LNAI 5559

Anne Håkansson Ngoc Thanh Nguyen
Ronald L. Hartung Robert J. Howlett
Lakhmi C. Jain (Eds.)

Agent and Multi-Agent Systems: Technologies and Applications

Third KES International Symposium, KES-AMSTA 2009
Uppsala, Sweden, June 2009
Proceedings

Springer

Lecture Notes in Artificial Intelligence 5559

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

Subseries of Lecture Notes in Computer Science

Anne Håkansson Ngoc Thanh Nguyen
Ronald L. Hartung Robert J. Howlett
Lakhmi C. Jain (Eds.)

Agent and Multi-Agent Systems: Technologies and Applications

Third KES International Symposium, KES-AMSTA 2009
Uppsala, Sweden, June 3-5, 2009
Proceedings

Springer

Volume Editors

Anne Håkansson
Stockholm University, 164 40 Kista, Sweden
E-mail: anne.hakansson@dis.uu.se

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

Ronald L. Hartung
Franklin University, 43215, Columbus, OH, USA
E-mail: Hartung@franklin.edu

Robert J. Howlett
University of Brighton, Brighton, BN2 4GJ, UK
E-mail: r.j.howlett@bton.ac.uk

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

Library of Congress Control Number: Applied for

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-01664-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-01664-6 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12674282 06/3180 5 4 3 2 1 0

Preface

This volume contains the proceedings of the Third KES Symposium on Agent and Multi-agent Systems – Technologies and Applications (KES-AMSTA 2009)—held at Uppsala University in Sweden during June 3-5, 2009. The symposium was organized by Uppsala University, KES International and its Focus Group on Agent and Multi-agent Systems. The KES-AMSTA Symposium series is a sub-series of the KES Conference series.

Following the successes of the First KES Symposium on Agent and Multi-agent Systems – Technologies and Applications (KES-AMSTA 2007), held in Wroclaw, Poland, from May 31 to 1 June 2007—and the Second KES Symposium on Agent and Multi-agent Systems – Technologies and Applications (KES-AMSTA 2008) held in Incheon, Korea, March 26-28, 2008—KES-AMSTA 2009 featured keynote talks, oral and poster presentations, and a number of workshops and invited sessions, closely aligned to the themes of the conference.

The aim of the symposium was to provide an international forum for scientific research into the technologies and applications of agent and multi-agent systems. Agent and multi-agent systems are an innovative type of modern software system and have long been recognized as a promising technology for constructing autonomous, complex and intelligent systems. A key development in the field of agent and multi-agent systems has been the specification of agent communication languages and formalization of ontologies. Agent communication languages are intended to provide standard declarative mechanisms for agents to communicate knowledge and make requests of each other, whereas ontologies are intended for conceptualization of the knowledge domain.

The symposium attracted a large number of scientists and practitioners who submitted their papers for 13 main tracks covering the methodology and applications of agent and multi-agent systems and eight special sessions on specific topics within the field. The papers were selected under the established KES quality principle, that we do not accept unsatisfactory papers simply to maintain or increase the size of the conference, but equally we do not unnecessarily or arbitrarily reject good papers. From the submissions to KES-AMSTA 2009 coming from more than 20 countries throughout the world, only 86 papers were selected, from a much larger number, for presentation and inclusion in the proceedings. All papers were rigorously reviewed by two referees who were experts in the field. Many of them were reviewed using the double-blind mode.

The Program Committee defined the following main tracks: Computational infrastructure and Agent Systems; Agent Systems with Implementation and Design; Agent Architectures; Negotiation Protocols; Social and Organizational Structures; Mobile Agents and Robots; Simulation Systems and Game Systems; Agent Systems and Ontologies; Privacy, Safety, and Security; Web Services and Semantic Web; Communication and Agent Learning Systems; E-commerce; Information Storage and Retrieval. In addition to the main tracks of the symposium there were the following eight special sessions: Workshop on Self-Organization in Multi-agent Systems, Management and eBusiness, Agent-Based Optimization, Mobile and Intelligent Agents for Networks and Services, Engineering Interaction Protocols, Agent-Based Simulation, Decision Making and Systems Optimization, Digital Economy, and Distributed Systems and Artificial Intelligence Applications.

We would like to thank the invited speakers – Rune Gustavsson (Sweden), Chengqi Zhang (Australia), and Caire Giovanni (Italy) – for their interesting and informative talks of world-class standard.

Special thanks go to the Program Co-chair for his great help in the organizational work and in managing the reviews and paper selection process. Thanks are due to the Program Committee and the Board of Reviewers, essential for reviewing the papers to ensure the high quality. We thank the members of the Local Organizing Committee, Publicity Chairs and Invited Sessions Chair. We acknowledge with gratitude the efforts of the KES Focus Group on Agent and Multi-agent Systems for coordinating the organization of the symposium. We extend cordial thanks to the KES Secretariat for the support with central administration and the registration process. Finally, we thank authors, presenters and delegates for their contribution to a successful event.

Thanks are also due to the many other experts who contributed to making the event a success.

We believe that KES-AMSTA 2009 significantly contributed to fulfillment of the KES mission of academic excellence, leading on to even greater successes in the future.

March 2009

Anne Håkansson
Ngoc Thanh Nguyen
Ronald Hartung
Robert J. Howlett
Lakhmi C. Jain

Organization

Honorary Chair

Vice-rector Lars Magnusson
Department of Economic History
Uppsala University, Sweden

General Chair

Anne Håkansson
Department of Computer and Systems Science
Stockholm University, Sweden

General Co-chair

Ngoc Thanh Nguyen
Institute of Informatics
Wroclaw University of Technology, Poland

Program Chair

Ronald L. Hartung
Department of Computer Science
Franklin University, USA

Executive Chair

Robert J. Howlett
Centre for SMART Systems
School of Environment and Technology
University of Brighton, UK

Invited Session Chair

Lakhmi C. Jain
School of Electrical and Information Engineering
University of South Australia, Australia

Program Co-chair

Dharmendra Sharma
Faculty of Information Sciences and Engineering
Canberra University, Australia

Local Organizing Chair

Torsten Palm
Department of Information Science/Computer Science
Uppsala University, Sweden

Publicity Co-chairs

Zeynep Kiziltan
Department of Electronics, Computer Sciences and Systems,
University of Bologna, Italy

Esmiralda Moradian
Department of Computer and Systems Sciences,
Stockholm University

Workshop and Invited Session Chairs

1. *Workshop on Self-Organization in Multi-agent Systems*
   Dariusz Krol
   Wroclaw University of Technology, Poland

2. *Management and eBusiness*
   Chulmo Koo
   Chosun University, South Korea,

   David Sorhammar
   Uppsala University, Sweden

3. *Invited Session on Agent-Based Optimization*
   Piotr Jedrzejowicz
   Gdynia Maritime University, Poland,

   Ireneusz Czarnowski
   Gdynia Maritime University, Poland

4. *Mobile and Intelligent Agents for Networks and Services*
   Gordan Jezic
   University of Zagreb, Croatia

5. *Engineering Interaction Protocols*
   Chihab Hanachi
   University of Toulouse, IRIT Laboratory, France

6. *Agent-Based Simulation, Decision Making and Systems Optimization*
   Hideki Katagiri
   Hiroshima University, Japan,

   Ichiro Nishizaki
   Hiroshima University, Japan,

   Kosuke Kato
   Hiroshima University, Japan

7. *Digital Economy*
   Yun-Heh (Jessica) Chen-Burger
   The University of Edinburgh, UK

8. *Distributed Systems and Artificial Intelligence Applications*
   Pilar Herrero
   Facultad de Informática - Universidad Politécnica de Madrid, Spain

Invited Speakers

1. Rune Gustavsson
   Blekinge Institute of Technology, Sweden
2. Chengqi Zhang
   University of Technology, Sydney, Australia
3. Giovanni Caire
   Research Labs of Telecom, Italy

International Program Committee

Abraham Ajith	Norwegian University of Science and Technology, Norway
Francesco Amigoni	Politecnico di Milano, Italy
Antonio Fernandez-Caballero	University of Castilla-La Mancha, Albacete, Spain
Maria Bielikova	Slovak University of Technology in Bratislava, Slovakia
Grazyna Brzykcy	Poznan University of Technology, Poland
Longbing Cao	University of Technology, Sydney, Australia
Frantisek Capkovic	Institute of Informatics, Slovak Academy of Sciences, Slovakia
Noelle Carbonell	Nancy Universite, France
Shyi-Ming Chen	National Taiwan University of Science and Technology, Taiwan
Jessica Chen-Burger	The University of Edinburgh, UK
Chihab Hanachi	IRIT Laboratory, University of Toulouse, France
Jayan Kurian Chirayath	Nanyang Technological University, Singapore
Kim Chonggun	Yeungnam University, Korea
Irek Czarnowski	Gdynia Maritime University, Poland
Paul Davidsson	Blekinge Institute of Technology, Sweden
Anne Håkansson	Uppsala University, Sweden
Ronald L. Hartung	Franklin University, USA
Sorin Hintea	Technical University of Cluj-Napoca, Romania

Adam Jatowt	Kyoto University, Japan
Piotr Jedrzejowicz	Gdynia Maritime University, Poland
Gordan Jezic	University of Zagreb, Croatia
Jason J. Jung	Yeungnam University, Korea
Krzysztof Juszczyszyn	Wroclaw University of Technology, Poland
Oleg Karsaev	St. Petersburg Institute for Informatics and Automation, Russia
Hideki Katagiri	Hiroshima University Kagamiyama, Japan
Radoslaw Katarzyniak	Wroclaw University of Technology, Poland
Przemyslaw Kazienko	Wroclaw Unniversity of Technology, Poland
Yanghee Kim	Utah State University, USA
In-Young Ko	Information and Communications University, Korea
Grzegorz Kolaczek	Wroclaw University of Technology, Poland
Dariusz Krol	Wroclaw University of Technology, Poland
Daniel Kudenko	University of York Heslington, UK
Kazuhiro Kuwabara	Ritsumeikan University, Japan
Zne-Jung Lee	Huafan University, Taiwan
Eun-Ser Lee	University of Soonsil, Korea
Huey-Ming Lee	Chinese Culture University, Taiwan
Jaeho Lee	The University of Seoul, Korea
Fuhua Lin	Athabasca University, Canada
Jie Lu	University of Technology, Sydney, Australia
NgWee Keong	Nanyang Technological University, Singapore
Orgun Mehmet	Macquarie University, Sydney
Jeng-Shyang Pan	National Kaohsiung University of Applied Science, Taiwan
Bhanu Prasad	Florida A&M University, USA
Zbigniew Ras	University of North Carolina-Charlotte, USA
Nancy Reed	University of Hawaii, Hawaii
Rajesh Reghunadhan	Bharathiar University, Coimbatore
Alessandro Ricci	University di Bologna, Italy
Bastin Tony Roy Savarimuthu	University of Otago, Dunedin, New Zealand
Janusz Sobecki	Wroclaw University of Technology, Poland
Yasufumi Takama	Tokyo Metropolitan University, Japan
Wojciech Thomas	Politechnika Wroclawska, Poland
Arkady Zaslavsky	Luleå University of Technology, Sweden
Daniel Zeng	Chinese Academy of Sciences and University of Arizona, USA
Justin Zhan	Carnegie Mellon University, USA
Wen-Ran Zhang	Georgia Southern University, USA

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

Ngoc Thanh Nguyen — Wroclaw University of Technology, Poland

KES International Conference Series Chairs

Robert J. Howlett — University of Brighton, UK
Lakhmi C. Jain — University of South Australia, Australia

Table of Contents

Social and Organizational Structures of Agents

Negotiation Protocols

Mobile Agents and Robots

Agent Design and Implementation

E-Commerce

Simulation Systems and Game Systems

Agent Systems and Ontologies

Agents for Network Systems

Communication and Agent Learning System

Web Services and Semantic Web

Self-Organization in Multi-Agent Systems

Management and eBusiness

Mobile and Intelligent Agents for Networks and Services

Engineering Interaction Protocols

Agent-Based Simulation, Decision Making and Systems Optimization

Digital Economy

Agent-Based Optimization (ABO2009)

Distributed Systems and Artificial Intelligence Applications

Dependency in Cooperative Boolean Games

Luigi Sauro[1], Leendert van der Torre[2], and Serena Villata[3]

[1] Dipartimento di Scienze Fisiche, University of Naples, Italy
sauro@na.infn.it
[2] CSC, University of Luxembourg
leendert@vandertorre.com
[3] Dipartimento di Informatica, University of Turin, Italy
villata@di.unito.it

Abstract. Cooperative boolean games are coalitional games with both goals and costs associated to actions, and dependence networks for boolean games are a kind of social networks representing how the actions of other agents have an influence on the achievement of an agent's goal. In this paper, we introduce two new types of dependence networks, called the *abstract dependence network* and the *refined dependence network*. Moreover, we show that the notion of stability is complete with respect to the solution concept of the *core* in the case of cooperative boolean games with costly actions. We present a reduction, called Δ-*reduction*, to pass from a cooperative boolean game G to game G' without loosing solutions.

1 Introduction

Dunne *et al.* [5] recently introduced a kind of coalitional games called cooperative boolean games, and they characterize the complexity of the solution concepts of core and stable sets. Moreover, Bonzon [3,2] defines stable sets and uses dependence networks to simplify the computation of the pure Nash equilibrium, showing that the notion of stability is complete with respect to the pure Nash equilibrium for actions without costs. In this paper, we propose a new step to make the computation of the core easier by means of the dependence networks associated with a cooperative boolean game. In particular, we study the following two questions:

1. Is the notion of stability complete also with respect of the solution concept of the core in the case of cooperative boolean games with costly actions?
2. How to pass from a cooperative boolean game G to game G' without loosing solutions?

We define two kinds of dependence networks and we show how to use them to calculate the core. Abstract dependence networks (ADNs) represent a structure based on the model presented by Bonzon et al. [3] without labels on the edges and AND-arcs, and refined dependence networks (RDNs) with AND-arcs labeled by the boolean variables composing the goals. We use the dependence network to decompose games, such as separating a game into two smaller independent games (with disconnected interdependencies) or isolating an agent requiring the execution of a costly action useless for all the other agents. Moreover, we present a number of abstractions that allow to reduce

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 1–10, 2009.

the search space by means of a set of criteria principally based on graphs visit algorithms that are computationally tractable. Moreover, we underline a number of hidden properties in the notion of core showing how, in certain cases, this notion is too strict and, thus, it can lead to counterintuitive results.

The reminder of the paper is as follows. Section 2 provides an introduction to the framework of cooperative boolean games and their solution concept, the core. Section 3 defines the dependency relations in the cooperative boolean games, defining the two types of dependence networks. Conclusions end the paper.

2 Cooperative Boolean Games

A Cooperative Boolean Game (CBG) [5] consists of a set of agents $1, \ldots, n$, a set of goals the agents desire to accomplish, the variables the agents control, and a cost function. Goals are represented by a propositional formula γ_i over some set of Boolean variables Φ.

Definition 1 ([5]). *A cooperative boolean game G is a $(2n+3)$-tuple $G = \langle A, \Phi, cost, \gamma_1, ..., \gamma_n, \Phi_1, ..., \Phi_n \rangle$ where $A = \{1, ..., n\}$ is a set of agents, $\Phi = \{p, q, ...\}$ is a finite set of boolean variables, $cost$ is a cost function defined in $\Phi \rightarrow \Re_+$, $\gamma_1, ..., \gamma_n$ are the boolean formulas over Φ representing the goals of the agents and $\Phi_1, ..., \Phi_n$ is a partition of Φ over n, with the intended interpretation that Φ_i is the set of Boolean variables under the control of agent i.*

The simple action theory works as follows, inspired by discrete event systems. For each variable $p \in Phi$, there is one agent who can set its truth value to true or false. Setting a variable $p \in \Phi$ to be $\top$ is *performing the action* p, and setting $p \in \Phi$ to $\bot$ is *doing nothing*; only in the former case, the costs of the action are taken into account. A set of variables $\xi \subseteq \Phi$ (or a strategy) stands for a valuation, in the sense that the value of the variables belonging to ξ is true and the value of the other ones is false, and we write $\xi \models \phi$ means that ϕ is true under the valuation ξ under the standard propositional semantics. $cost_i(\xi)$ denotes the cost to agent i of valuation $\xi \subseteq \Phi$, that is, $cost_i(\xi) = \sum_{v \in (\xi \cap \Phi_i)} cost(v)$. Agents minimize its costs only when they have achieved their goal, or cannot achieve their goal. In other words, if the only way an agent can achieve its goal is by making all its variables true, then the agent prefers to do this rather than not achieve its goal. However, if there are different ways to achieve its goal, then the agent prefers to minimize costs. This is represented by a utility function that is always positive if the valuation ξ satisfies the goal of the agent i and otherwise it is always negative. If μ represents the total cost of all variables, the utility for agent i of a valuation ξ, $u_i(\xi)$ is defined as follows.

$$u_i(\xi) = \begin{cases} 1 + \mu - cost_i(\xi) \text{ if } \xi \models \gamma_i \\ \qquad -cost_i(\xi) \;\; otherwise \end{cases}$$

The meaning of the fact that coalition D blocks ξ_1 through ξ_2 is that ξ_2 could do better than ξ_1 only flipping the value of some of the variables under the control of D. Here a a coalition $D \subseteq A$ is a set of agents without represented relationship, Φ_D denotes the set of variables under the control of some member of D, $\Phi_D = \bigcup_{i \in D} \Phi_i$, and $\xi_1 = \xi_2$ mod D means that valuation ξ_2 is the same as a valuation ξ_1 except at most in the value of variables controlled by D, $\xi_1 \succeq_i \xi_2$ iff $u_i(\xi_1) \geq u_i(\xi_2)$.

Definition 2 (Blocked Valuation [5]). *A valuation ξ_1 is blocked by a coalition $D \subseteq A$ through a valuation ξ_2 if and only if:(1) ξ_2 is a feasible objection by coalition D which means that $\xi_2 = \xi_1$ mod D; (2) D strictly prefers ξ_2 over ξ_1: $\forall i \in D : \xi_2 \succ_i \xi_1$.*

The core is a fundamental concept in coalitional game theory. A valuation is in the core if and only if no coalition has an incentive to defect.

Definition 3 (Core). *Given a CBG G, $\xi \in core(G)$ iff it is not blocked by any coalition.*

The following example reproduces the classical schema of the Prisoner Dilemma, and illustrates that Definition 3 is a strengthening of the well-known Nash equilibrium in non-cooperative game theory [6], usually called Strong Nash Equilibrium (SNE). This form of solution satisfies at the same time some requirements of both the cooperative and non-cooperative game theory. From the solution criteria developed in cooperative game theory, it *borrows* efficiency, a strategy cannot be a solution if the agents or a part of them can obtain better results. From non-cooperative game theory, it assumes that agents are suspicious and that agreements cannot be enforced, that is at any time agents may betray agreements.

Example 1. Let G be a CBG involving two agents. $\Phi_1 = \{a\}$ and $\Phi_2 = \{b\}$, $\gamma_1 = \gamma_2 = a \vee b$, $cost(a) = cost(b) = 1$. The CBG can be represented by the following payoff matrix:

a/b	1	0
1	$(1,1)$	$(-1,2)$
0	$(2,-1)$	$(0,0)$

$(0,0)$ is the only Nash equilibrium in this game, that is not efficient as the two agents collaborating can obtain better results, $(1,1)$, and hence it is not a SNE. On the other hand, as $(1,1)$ is not a Nash equilibrium, it is not a SNE too. Indeed, if $(1,1)$ is proposed as an agreement each agent would unilaterally betray it obtaining better results.

Note that whether SNE corresponds to the notion of core depends on how a strategic game is translated into a cooperative game (see [7]). For example, if we use for the strategic game in Example 1 the minimax representation of a corresponding cooperative game - as introduced by von Neumann and Morgensten - then the strategy $\{1,1\}$ is in the core. This kind of representation is called *offensive* because it is assumed that for a certain coalition D, the agents in $A \setminus D$ *works* in order to minimize the outcomes of D. On the contrary if we adopt a *defensive* representation where the primary aim of the agents is to maximize their own utility, the SNE represents the core of the corresponding cooperative game. The following example illustrates a number of cases in which, in contrast to Example 1, the core is not empty.

Example 2. Consider a game where we have four agents $A = \{1,2,3,4\}$ who want to go in holidays to the seaside or to the mountains. We represent with the boolean variable a to go to the seaside and with b to go to the mountains. Agent i going to the seaside is represented by setting a_i to true whereas the holiday to the mountains is represented by b_i. For each agent i, $cost(a_i) = 2$ and $cost(b_i) = 1$. Agent 1 is in love with agent 2 and he wants to go everywhere with her, thus its goal is represented by $\gamma_1 = (a_1 \wedge a_2) \vee (b_1 \wedge b_2)$. Agent 2 is in love with agent 1 but she cannot tolerate the change of temperature of the mountains thus her goal is $\gamma_2 = a_1 \wedge a_2$. Agent 3 is in love with 2 and, as he is jealous of agent 1, he would like to stay with 2 without the presence of 1 $\gamma_3 = (a_2 \wedge a_3 \wedge \neg a_1) \vee (b_2 \wedge b_3 \wedge \neg b_1)$. Agent 4 is in love with agent 3, who does not like him, but she is not able to swim thus its goal is $\gamma_4 = b_3 \wedge b_4$. Let us say that agent i is satisfied given a valuation ξ if $\xi \models \gamma_i$, i.e., if i's goal is satisfied. It can be verified that $\{a_1, a_2\}$ is in the core.

3 Dependency Relations in CBG

Dependence networks have been developed by Conte and Sichman [10] as a kind of social network representing how each agent depends on other agents to achieve the goals he cannot achieve himself. The notion of agent dependence used to define dependence networks is related to the concept of social power, introduced by Castelfranchi [4]. Sauro [9,1] shows how to use dependence networks to discriminate among different potential coalitions during the coalition formation process. The authors develop a criterion of admissibility called *do-ut-des property* describing a condition of reciprocity. Moreover, they define another criterion, called the indecomposable do-ut-des property, establishing which coalitions cannot be formed under the assumption that agents are self-interested. These two criteria have only a qualitative connotation and thus, they cannot be directly applied to the solutions developed in game theory. Moreover, goals are not structured and they do not represent explicitly costs of the actions.

The first attempts of use of the dependence networks to represent and simplify the computation of the solution concepts for boolean games are given by Bonzon [2] and Bonzon et al. [3]. Representing these dependencies on a graph, they show how to compute pure-strategy Nash equilibria without enumerating all combinations of strategies. This work does not consider costly actions and dependence networks are simple graph without labeled edges.

In this section we present two types of dependence networks defined starting from a CBG. These networks, on the one hand, explicitly represent the inter-dependencies among agents according to their goals, on the other hand, they abstract from the quantitative aspects of a game associated to the cost function. Abstract Dependence Networks describe only which agents can play a role in the satisfaction of an agent's goal, therefore they abstract from how they can contribute and in particular if they have to execute a costly action. As this information may help in the study of the core and reduces the search space, we define another type of dependence network, the Refined Dependence Networks, which, as the name suggests, refine Abstract Dependence Networks by representing how agents contribute to the satisfaction of a goal and whether this involves a positive cost (without quantifying it). Starting from the Refined Dependence Networks, we define a method called $\Delta - reduction$ to reduce the admissible strategies and we prove the completeness of the $\Delta - reduction$ with respect to the computation of the core.

3.1 Abstract Dependence Networks

As in Bonzon et al. [3], in order to correctly establish the dependencies among agents we need to define which variables are relevant for the satisfaction of the agents' goal. A variable p is said irrelevant for a formula ϕ in case there exists an equivalent formula ϕ' where p does not occur. With $RV_G(i)$ we represent the set of all variables $p \in \Phi$ that are relevant for γ_i, whereas $RA_G(i)$ is the set of agents $j \in A$ such that j controls at least one relevant variable of i. Using the notion of relevant agents, we define a dependence network where nodes represent agents and an edge from i to j represents the dependence of i from j ($j \in RA_G(i)$).

Definition 4 (Abstract Dependence Network)
Given the CBG $G = \langle A, \Phi, cost, \gamma_1, ..., \gamma_n, \Phi_1, ..., \Phi_n \rangle$, the abstract *dependence network of G is the directed graph ADN(G) $= \langle N, R \rangle$ such that the set of nodes N corresponds to the agents in G, $N = A$, and $(i, j) \in R$ iff $j \in RA_G(i)$.*

As in [3] we say that a set of agents in *ADN*(G) is stable in case it is *closed* under the relation R. $R(C)$ is the set of players from which C may need some action in order to be satisfied.

Definition 5 (Stable set). *Given a directed graph $\langle N, R \rangle$, $C \subseteq N$ is stable iff $R(C) \subseteq C$, i.e. for all $i \in C$, for all j such that $(i, j) \in R$ $j \in C$.*

Note that the notion of stable set is not related to the strategic criterion of the same name originally introduced by von Neumann and Morgenstern [8].

Definition 6 (ADN Projection)
Let $G = \langle A, \Phi, cost, \gamma_1, ..., \gamma_n, \Phi_1, ..., \Phi_n \rangle$ be a CBG, ADN(G) $= \langle N, R \rangle$ the corresponding abstract dependence graph and $C = \{i_1, \ldots, i_m\} \subseteq N$ a stable set, the projection of G on C is defined by $G_C = \langle C, \Phi_C, cost_C, \gamma_{i_1}, ..., \gamma_{i_m}, \Phi_{i_1}, ..., \Phi_{i_m} \rangle$, where $cost_C : \Phi_C \rightarrow \Re^+$ is the restriction of cost on Φ_C.

As shown in Bonzon et al. [3] the projection of a CBG on a stable set is itself a CBG.

Proposition 1. *Given a CBG G, if C is a stable set, G_C is a cooperative boolean game.*

In Bonzon et al. [3] the authors show that by restricting a boolean game G to the projection G_C of a stable set C, if a strategy profile ξ_C in G_C is not a Nash equilibrium, then all of its extensions in G are not a Nash equilibrium. Here we extend this result to the case of the core in CBG with costly actions.

Proposition 2. *Given a stable set C, if ξ is in core(G), then ξ_C is in core(G_C), where ξ_C is the projection of ξ on the variables controlled by C, $\xi_C = \xi \cap \Phi_C$.*

Proof. Let $\bar{\xi}$ a generic valuation in G and $\bar{\xi}_C$ the projection of $\bar{\xi}$ on the variables controlled by C. Clearly, for all $i \in C$, $cost_i(\bar{\xi}) = cost_i(\bar{\xi}_C)$. Furthermore, as C is stable, for all $i \in C$, $RV_G(i) = RV_{G_C}(i)$ and hence $\bar{\xi} \models \gamma_i$ iff $\bar{\xi}_C \models \gamma_i$. This entails that for two generic valuations $\bar{\xi}$ and $\hat{\xi}$ and for all $i \in C$, $u_i(\bar{\xi}) \leq u_i(\hat{\xi})$ iff $u_i(\bar{\xi}_C) \leq u_i(\hat{\xi}_C)$.

Assume that there exists a valuation ξ'_C that blocks ξ_C, i.e. there exists a $C' \subseteq C$ such that $\xi'_C = \xi_C \; mod \; C'$ and for all $i \in C'$ $\xi'_C \succ_i \xi_C$. Now let $\xi_{-C} = \xi \cap \Phi_{A \setminus C}$ and $\xi' = \xi'_C \cup \xi_{-C}$. Clearly, $\xi' = \xi \; mod \; C'$ and, since for all $i \in C$ $u_i(\xi) \leq u_i(\xi')$ iff $u_i(\xi_C) \leq u_i(\xi'_C)$, ξ is blocked by C' through ξ'.

Finally, we define *consistency* such that given two coalitions D_1 and D_2, we say that two relative strategies ξ_{D_1} and ξ_{D_2} are *consistent* if and only if for each agent $i \in D_1 \cap D_2$, $\Phi_i \cap \xi_{D_1} = \Phi_i \cap \xi_{D_2}$. The following two propositions hold - the proof is straightforward and it is left to the reader.

Proposition 3. *Let ξ be a strategy blocked by a coalition D through ξ' and C be a stable set such that $C' = C \cap D \neq \emptyset$, then ξ is blocked by C' through ξ'_{C}.*

Proposition 4. *Given the stable sets $C_1, \ldots, C_n$ and relative strategies $\xi_{C_1}, \ldots, \xi_{C_n}$, if*

1. *for all $1 \leq i \leq n$, $\xi_i \in core(G_{C_i})$;*
2. *$\xi_{C_1}, \ldots, \xi_{C_n}$ are consistent;*
3. *$A = \bigcup_{i=1}^{n} C_i$,*

then $\bigcup_{i=1}^{n} \xi_i \in core(G)$.

Propositions 2 and 4 provide a way to decompose the problem of determining the core of a CBG G into the subgames G_{C_i}, where $C_1, \ldots, C_n$ are stable sets that involve all the agents in G. Then, once each $core(G_{C_i})$ is determined, it remains to gather any union ξ of consistent strategies $\xi_{C_i} \in core(G_{C_i})$, with $1 \leq i \leq n$. Due to Proposition 4 $\xi \in core(G)$, whereas, Proposition 2 ensures that in this way we find all the elements in $core(G)$, indeed if $\xi' \in core(G)$, then $\xi'_{C_i} \in core(G_{C_i})$, with $1 \leq i \leq n$ and hence ξ' is the union of consistent solutions in each G_{C_i}.

Example 3 (Continued). The abstract dependence network of the game in Example 2 is $ADN = \langle N, E \rangle$, where $N = \{1, 2, 3, 4\}$ and $R = \{(1,2), (2,1), (3,1), (3,2), (4,3)\}$. This ADN is depicted in Figure 1-(a) where the edges represent the dependence of the first agent on a second one for a boolean variable composing its goal and the circle represents the stable set C. Let us consider agents 1 and 2 (see Figure 1-(a)), following Definition 5 they represent a stable set (all the edges that go out from agent 1 enter in agent 2 and converse), so we can consider first the projection of the game on them (Definition 6). We can represent valuations as $uvyz \in \{0,1\}^4$, where u represents the value of a_1, v that of a_2, y the value of b_1 and z that of b_2. Following Definition 6, it can be found that 1100 is the only strategy in the core. Due to Propositions 2, also in the complete game all the strategies containing either b_1 or b_2 and not containing one of a_1 and a_2 are blocked.

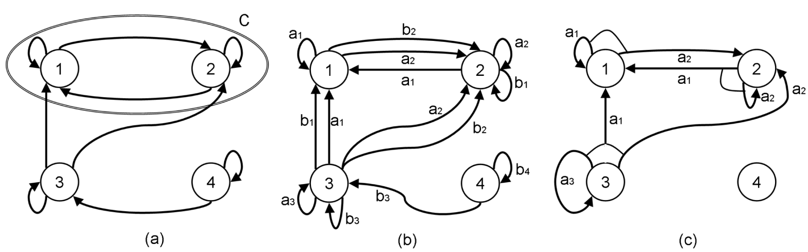


Fig. 1. a)- ADN of Example 3 with the stable set C, b)- RDN of Example 6, c)- RDN of Example 6 after the Δ-reduction

3.2 Refined Dependence Networks

As seen before, by using stable sets Abstract Dependence Networks can be safely used to split the original problem in subproblems without loosing solutions. However, Abstract Dependence Networks may hide some useful information that can also be used to prune some strategies that cannot belong to the core - and hence to reduce the search space. For this reason we define another notion of dependence network at a lower level of abstraction and we call it Refined Dependence Network (RDN). These networks may seem actually equivalent to the boolean game itself, except for the cost of variables. These costs, however, are not a minor point since two boolean games that result in the same RDN can have different solutions.

A Refined Dependence Network represents how the goals can be satisfied by means of AND-arcs among the agents whose single edges are labeled with literals. Furthermore, costly actions are *marked* in a set Δ.

Definition 7 (Refined Dependence Network)
A Refined Dependence Network is an AND-graph $\langle N, \Phi, \Delta, E, \Phi_1, ..., \Phi_n \rangle$ where N is the set of nodes, Φ is the set of boolean variables, $\Delta \subseteq \Phi$ is the subset of costly variables[1]*, $E \subseteq N \times 2^{(N \times Litt(\Phi))}$ where $Litt(\Phi) = \Phi \cup \{\neg p | p \in \Phi\}$ and $\Phi_i \subseteq \Phi$ where n is the cardinality of N.*

In the following, given a literal l, we denote by $| \ l \ |$ the corresponding boolean variable, that is if $l \in \Phi$, then $| \ l \ |= l$ whereas if $l = \neg p$, $| \ l \ |= p$. Furthermore, to simplify the formalism, we represent an AND-arc $(i, \{(j_1, l_1), \ldots, (j_m, l_m)\})$ as the set of triples $\{(i, j_1, l_1), \ldots, (i, j_m, l_m)\}$. Of course, a set as $\{(1, 3, p), (3, 4, q)\}$ has no meaning in our context.

As already done for Abstract Dependence Networks, we use Refined Dependence Networks to reveal the structure of interdependencies among agents. First, we assume that the goals of the agents do not contain irrelevant variables and are given in disjunctive normal form, i.e. $\gamma_i = \gamma_{i_1} \vee \cdots \vee \gamma_{i_m}$ where each γ_{i_j} is a conjunction of literals. Note that the deletion of these irrelevant variables is coNP-complete. We expect that a goal does not contain irrelevant variables, alternatively it is possible to consider the whole set of variables occurring in a formula, avoiding in this way this computational cost. To simplify again the formalism we describe respectively γ_i as a set of γ_{i_j} and each γ_{i_j} as a set of literals - the empty set has the usual meaning respectively of $\bot$ referred to γ_i and $\top$ referred to the γ_{i_j}. Roughly, each AND-arc e outgoing the agent i corresponds to a $\gamma_{i_j} \in \gamma_i$, where each single edge that composes e is labeled with a literal occurring in γ_{i_j} and reaches the agent that controls the corresponding variable. The set Γ consists of the actions that have a strictly positive cost.

Definition 8 (From CBGs to RDNs)
Given the CBG $G = \langle A, \Phi, cost, \gamma_1, ..., \gamma_n, \Phi_1, ..., \Phi_n \rangle$, the corresponding Refined Dependence Network RDN$(G) = \langle N, \Phi, \Delta, E, \Phi_1, ..., \Phi_n \rangle$ is such that $N = A$, $\Delta = \{p \in \phi \mid cost(p) > 0\}$ and $\{(i, j_1, l_1), \ldots, (i, j_m, l_m)\} \in E$ iff $\{l_1, \ldots, l_m\} \in \gamma_i$ and for all $1 \leq h \leq m$, $| \ l_h \ | \in \Phi_{j_h}$.

[1] The set of variables with an associated cost.

Example 4. Let G be a cooperative boolean game defined by $A = \{1, 2, 3, 4\}$, $\Phi = \{a, b, c, d, e\}$, $cost(a) = cost(b) = cost(c) = cost(d) = cost(e) = 1$, $\gamma_1 = a$, $\gamma_2 = c \wedge e$, $\gamma_3 = b \wedge c$, $\gamma_4 = d$, $\Phi_1 = \{b, e\}$, $\Phi_2 = \{d\}$, $\Phi_3 = \{a\}$, $\Phi_4 = \{c\}$. The associated refined dependence network $RDN_G = \langle A, \Phi, E, \Phi_1, ..., \Phi_n\rangle$, where E is composed by the following dependencies: $\{(1, 3, a)\}$, $\{(3, 1, b), (3, 4, c)\}$, $\{(2, 1, e), (2, 4, c)\}$, $\{(4, 2, d)\}$.

Given a Refined Dependence Network $RDN(G) = \langle N, \Phi, \Delta, E\rangle$, we mean with $R_E \subseteq N \times N$ the binary relation such that $(i, j) \in R_E$ just in the case there exists an AND-arc $e \in E$ that starts from i and reaches j, i.e. for some literal l, $(i, j, l) \in e$. It is easy to see that $ADN(G) = \langle N, R_E\rangle$ and hence $RDN(G)$ describes G at a lower level of abstraction with respect to $ADN(G)$.

We want to use Redefined Dependence Networks to impose some constraints to the set $core(G)$. To this scope some preliminary results are needed. A boolean variable $a \in \Phi_i$ is said to be *unfavourable* if and only if $a \in \Delta$, i.e. $cost(a) > 0$, and for each $\{l_1, \ldots, l_m\} \in \gamma_i$, $a \notin \{l_1, \ldots, l_m\}$. In the following we denote by $[i]^-$ the set of unfavourable variables of the agent i.

Proposition 5. *Given a cooperative boolean game G and an agent $i \in A$, for each $a \in [i]^-$, $\xi \in core(G)$ implies $a \notin \xi$.*

Proof. $a \in [i]^-$ means that $cost(a) > 0$ and a does not occur (positive) in γ_i. Assume that $\xi \models \gamma_i$, then, for some $\{l_1, \ldots, l_m\} \in \gamma_i$, $\xi \models \{l_1, \ldots, l_m\}$. Since $a \notin \{l_1, \ldots, l_m\}$, this means that also $\xi \setminus \{a\} \models \{l_1, \ldots, l_m\}$.

Now it remains to show that if for each ξ, $\xi \models \gamma_i$ implies $\xi \setminus \{a\} \models \gamma_i$, then $\xi \in core(G)$ implies $a \notin \xi$. Assume that $a \in \xi$, clearly $\xi \setminus \{a\} = \xi \ mod\ \{i\}$ as $a \in \Phi_i$. Furthermore, as $c_i(\xi \setminus \{a\}) < cost_i(\xi)$ and by hypothesis $\xi \models \gamma_i$ implies $\xi \setminus \{a\} \models \gamma_i$, then $\xi \prec_i \xi \setminus \{a\}$. But this means that ξ is blocked by i through $\xi \setminus \{a\}$.

Note that according to Proposition 5 it can be easily seen in Example 4 that, as all the variables are unfavourable, the core can contain only the empty strategy. We also prove that a goal depending on an unfavourable variable a can be reduced into one that do not depend on a without affecting the possible solutions. More precisely, we define the notion of reduction as follows.

Definition 9. *Given a cooperative boolean game G and an unfavourable variable $a \in [i]^-$, we say that the cooperative boolean game G' is a Δ-reduction of G just in the case it is obtained from G applying the following steps:*

1. *remove a from Φ_i;*
2. *remove from each γ_i any conjunction of type $\{l_1, \ldots, a, \ldots, l_n\}$;*
3. *replace in each γ_i any conjunction of type $\{l_1, \ldots, \neg a, \ldots, l_n\}$ with $\{l_1, \ldots, l_n\}$.*

Proposition 6. *Let G' be a Δ-reduction of a CBG G, $core(G) \subseteq core(G')$.*

Proof. For each valuation ξ that does not contain the unfavourable variable a, it clearly holds for each agent i that $u_i(\xi)$ in G is equal to $u_i(\xi)$ in G'. Therefore, if in G', ξ'_1 is blocked by C through ξ'_2, then the same holds in G and hence $core(G) \subseteq core(G')$.

Note that the converse does not hold. Consider Example 5:

Example 5. Let G be a cooperative boolean game composed by 3 agents and such that $\Phi_1 = \{a\}$, $\Phi_2 = \{b, c\}$ and $\Phi_3 = \{d\}$, $\gamma_1 = b$, $\gamma_2 = a \vee (c \wedge d)$ and $\gamma_3 = c \wedge b$, $cost(a) = cost(b) = 1$ and $cost(c) = cost(d) = 2$. The boolean variable a is an unfavourable variable then the Δ-reduced game G' is such that $\Phi_1 = \emptyset$, $\Phi_2 = \{b, c\}$ and $\Phi_3 = \{d\}$, $\gamma_1 = b$, $\gamma_2 = c \wedge d$ and $\gamma_3 = c \wedge b$. The function *cost* is the same as in G. It is easy to see that $\{c, d\} \in \mathit{core}(G')$ whereas in G it is blocked by $\{1, 2\}$ through $\{a, b\}$.

Note that the previous results do not use *quantitative* values of the cost function but only the fact that an action has a strictly positive value, therefore they reside in the level of abstraction of RDNs. We can now define a procedure on *RDN*(G) which uses unfavourable variables and Δ-reductions to reduce the search space in finding the core.

Definition 10 (Reduction rule). *Let RDN*(G) *be a Refined Dependence Network and let* ω *denoting a boolean formula initially set to* $\top$, *the reduction rule RDN*(G) *is given by applying exhaustively the following rule:*

Condition: *for some* $a \in \Phi_i \cap \Delta$, *there does not exist an AND-arc* e *outgoing from* i *such that* $(i, i, a) \in e$ *(i.e.* a *is unfavourable).*
Action: *remove any and-arc* e' *such that* $(j, i, a) \in e'$, a *from* Φ_i *and update* ω *with* $\omega \wedge \neg a$.

Due to Propositions 5 and 6, the boolean formula ω we obtain from definition 10 constraints the strategies that can be in the core of G, that is $\xi \in \mathit{core}(G)$ implies $\xi \models \omega$.

Example 6 (Continue). Let us consider again the cooperative boolean game of Example 2. In the corresponding RDN all the actions involve some positive cost, thus $\Delta = \Phi$. The AND-arcs are shown in Figure 1-(b) (without connections for simplicity of the figure) and all the edges are labeled with the boolean variable they represent. By exhaustively applying the rule in the Definition 10, b_2 satisfies the given condition, therefore we remove the AND-arc $\{(1, 2, b_2), (1, 1, b_1)\}$ and the AND-arc $\{(3, 3, b_3), (3, 2, b_2), (3, 1, \neg b_1)\}$. After these deletions, also b_1 and b_3 satisfy the condition in Definition 10, and hence also the AND-arc $\{(4, 3, b_3), (4, 4, b_4)\}$ has to be removed. Finally, also b_4 satisfies the condition, therefore as output ω is equal to $\neg b_1 \wedge \neg b_2 \wedge \neg b_3 \wedge \neg b_4 \wedge \neg a_4$, therefore the only strategies that can be in the core are the subsets of $\{a_1, a_2\}$. The RDN of Example 6 after the application of the Δ-reduction is depicted in Figure 1-(c).

4 Conclusion

In this paper we present a new approach to cooperative boolean games [5] based on dependence networks [10]. Differently from Bonzon et al. [3], we use dependence networks to reduce the search space thanks to the application of graphs' visit algorithms and to argue on the notion of core, showing a number of hidden properties of this solution concept. Moreover, we define two different kinds of dependence networks, abstract and refined dependence networks in which, differently from [3], we introduce costly actions, labeled edges and ADN-arcs. Finally, we present the Δ-reduction that allows to reduce the search space to find the strategies in the core without loosing solutions.

Concerning future work, we can address our methodology and results to the other solution concepts, for example instead of Strong Nash equilibrium we could represent the core with a less restrictive notion of stability such as the Coalitional-proof Nash equilibrium. In particular, it has been shown that if a game has only one Nash Equilibrium, then it is also a Coalitional-proof Nash equilibrium. Moreover, we are working on an algorithm FIND_CORE to search the strategies in the core using the $\Delta - reduction$ without loosing solutions.

Finally, we aim at evaluate the implemented algorithm in concrete domains - such as the Grid - which can be represented as exchange networks [9]. Several game theoretical approaches used to model such domains require numerical information about private utilities or the degrees of collaboration, however such information is generally not available. Conversely, Cooperative Boolean Games can be set up on the base of known information: the goals that the agents request when they join an exchange market and the costs of the resources they offer.

References

1. Boella, G., Sauro, L., van der Torre, L.: Strengthening admissible coalitions. In: ECAI, pp. 195–199 (2006)
2. Bonzon, E.: Modelisation des interactions entre agents retionnels: les jeux booleens. PhD thesis, Universite Paul Sabatier, Toulouse (2007)
3. Bonzon, E., Lagasquie-Schiex, M.-C., Lang, J.: Dependencies between players in boolean games. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS, vol. 4724, pp. 743–754. Springer, Heidelberg (2007)
4. Castelfranchi, C.: The micro-macro constitution of power. Protosociology 18, 208–269 (2003)
5. Dunne, P.E., van der Hoek, W., Kraus, S., Wooldridge, M.: Cooperative boolean games. In: AAMAS, pp. 1015–1022 (2008)
6. Keiding, H., Peleg, B.: Representation of effectivity functions in coalition proof nash equilibrium: A complete characterization. Technical report, University of Copenhagen (1999)
7. Myerson, R.B.: Game Theory. Harvard University Press (1997)
8. Neumann, J., Morgenstern, O.: Theory of games and economic behaviour, NJ (1944)
9. Sauro, L.: Formalizing admissibility criteria in coalition formation among goal directed agents. PhD thesis, University of Turin (2005)
10. Sichman, J.S., Conte, R.: Multi-agent dependence by dependence graphs. In: AAMAS, pp. 483–490 (2002)

Multi-Agent Artificial Immune Systems (MAAIS) for Intrusion Detection: Abstraction from Danger Theory

Chung-Ming Ou[1] and C.R. Ou[2]

[1] Department of Information Management, Kainan University, #1 Kainan Rd.
Luchu, Taoyuan 338, Taiwan
cou077@mail.knu.edu.tw
[2] Department of Electrical Engineering, Hsiuping Institute of Technology
Taichung, 412, Taiwan
crou@mail.hit.edu.tw

Abstract. Danger theory-inspired Multi-agent artificial immune system (MAAIS) is applied to intrusion detection systems (IDS). Antigens are profiles of system calls while corresponding behaviors are regarded as signals. The intelligence behind such system is based on the danger theory while dentricit cells agents (DC agent) are emulated for innate immune subsystem and artificial T-cell agents (TC agent) are for adaptive immune subsystem. This IDS is based on the dual detections of DC agent for signals and TC agent for antigen, where each agent coordinates with other to calculate danger value (DV). According to DV, immune response for malicious behaviors is activated by either computer host or Security Operating Center (SOC).

1 Introduction

Recently the Internet has become a powerful mechanism for propagating malicious codes. Internet worms spread through computer networks by searching, attacking and infecting remote computers automatically. The very first worm is the Morris worm in 1988; while Code Red, Nimda and other Internet worms have caused tremendous loss for the computer industry every since. Recently, OWASP issued the Top 10 vulnerability list for Web applications [14]; intrusions have become increasing threats for information assets. In order to defend against future network attacks, one efficient way is to understand various properties of virus, which include the impact of patching, awareness of other human countermeasures and the impact of network traffic, even the ways how these malicious codes reside in a certain hosts, etc.

One important aspect is the following: a computer security system should protect a host or network of hosts from unauthorized intruders, which is analogous in functionality to the immune system protecting the body from invasions by foreign pathogens. For example, anti-virus software has recently adopted some features analogous to the innate immune system, which can detect malicious patterns. However, most commercial products do not yet have the adaptive immune system's ability to address novel threats.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 11–19, 2009.

Artificial Immune Systems (AIS), based on Human Immune Systems (HIS), have been applied to anomaly detections [2][7][9][11][12][13]. AISs have been developed according to negative selection algorithm and clonal selection algorithm which are based on the classical self-nonself theory; nonselfs are entities which are not part of human organisms [1]. This so-called self-nonself classification theory had been challenged while failing to explain several immunological phenomena. Some alternative theories have been proposed, for example, the danger theory (DT). DT postulates that the human immune systems respond to the presence of molecules known as danger signals, which are released as results of unplanned cell deaths.

The major purpose of this paper is to facilitate multi-agent mechanisms with AIS based on danger theory to improve intrusion detection systems (IDS). Agents are entities that have the abilities of consciousness, solving problems and communications. Agents equipped with dentritic cell functionality (DC agent) can "detect" danger signal issued by computer hosts being attacked or suspiciously being attacks. While other agents such as T-cell agent (TC agent), antigen agent (Ag agent) and responding agents (RP agent), communicate one another to improve the efficiency of IDSs. Computer threats generally come from the Internet, which are very similar to those of pathogens to our bodies. The central challenge with computer security is how to discern malicious activities from benign ones. Our multi-agent based model majorly consists of the cooperation of DC agents in the innate immune system and TC agents in the adaptive immune system. This dual detective mechanism, where DC agent detects the behavioral information (i.e. signal) caused by an antigen and TC agent detects system call (i.e. antigen), can decrease false positive rate.

The arrangement of this paper is as follows. In section 2, preliminary knowledge such as AIS, danger theory, multi-agents system and intrusion detections are introduced. In section 3, MAAIS-based IDS model inspired by DT is discussed.

2 Preliminary Knowledge

2.1 Human Immune System (HIS)

HIS can be categorized as innate and adaptive immune system. The innate immune system is characterized by three roles, namely, host defense in the early stages of infection, induction of the adaptive immune response and determination of the type of adaptive response through antigen presenting cells (APCs) [12]. On the other hand, the main characteristics of the adaptive immune system are recognitions of pathogens.

The dentritic cell (DC) is a vital link between the innate and adaptive immune system and provides the initial detections of pathogenic invaders. DC is an APC which captures antigen protein from the surrounding area and processes it by ingesting and digesting the antigen. DCs are also part of innate immune system; once activated, they migrate to the lymphoid tissues where they interact with T-cells and B-cells to initiate the adaptive immune response. Moreover, adaptive immune response is "orchestrated" by DCs.

DCs are the first defense line for HISs which will arrive at the locations where antigens intrude and then swallow the latter to the pieces. These pieces will be attached to APCs and presented to the T-cells. DCs can be regarded as the commanders for HISs. DCs can combine the danger and safe signal information to decide if the tissue environment is in distress or is functioning normally. DT states that the immune system will only respond when damage is indicated and is actively suppressed otherwise.

2.2 Danger Theory (DT)

Matzinger [10] proposed the DT, which has become more popular among immunologists in recent years for the development of peripheral tolerance (tolerance to agents outside of the host). DT proposes that APCs, (in particular, DCs), have danger signal receptors (DSR) which recognize signals sent out by distressed or damaged cells. These signals inform the immune systems to initiate immune responses. APCs are activated via the danger signals. These activated APCs will be able to provide the necessary signals to the T-cells (more precisely, T-helper cells) which control the adaptive immune response.

Danger signals are generated by ordinary cells of the body that have been injured due to attacks by pathogens. These signals are detected by DCs. There are three modes of operation: immature, semi-mature and mature. In the DC's immature state, it collects antigens along with safe and danger signals from its local environment. DC is able to integrate these signals to decide whether the environment is safe or dangerous. If it is safe, DC becomes semi-mature. Upon presenting antigens to T-cells, DC will cause T-cells tolerance. If it is dangerous, DC becomes mature and causes the T-cells to become reactive on antigen-presentations.

2.3 Multi-Agent Based Models (MABM)

Agent is an entity that has the ability of consciousness, solving problem, self-learning and adapting to the environment. To have the agents learn, we may utilize AIS: the immune system response attributes of specificity, diversity, memory and self/non-self recognition are needed. It may be used to optimize the agent’s responses to intruders. Functionalities of the biological immune system (e.g., content addressable memory, adaptation, etc.) are identified for use in intelligent agents.

According to Forrest et al. [4], MABM is an appropriate method for studying immunology. As computers have been more powerful and less expensive, the MABM becomes a practical method for studying complex systems such as the immune system.

2.4 Architecture of Multi-Agents AIS (MAAIS)

MAAIS was proposed by Fu et al. [12]. The architecture is based on CARDINAL [13] which is basically not an agent-based system (Fig.1). This MAAIS, which consists of two components, namely agents and server processes, provides agent-based anomaly detection functionality. Agents monitor their corresponding hosts; while server will evaluate for selecting suitable strategies of immune responses, see Fig. 2. However, [12] did not designate roles of agents specifically. For further improvement, agents may be composed of diverse APCs and various types of artificial T-cells.

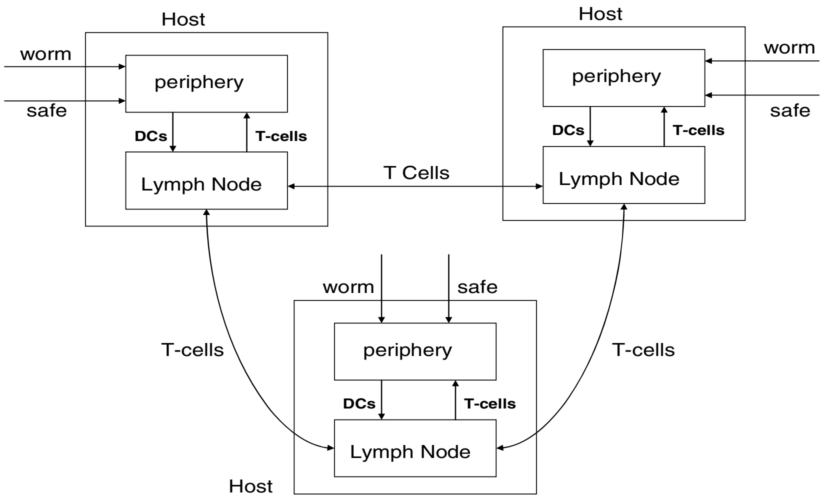


Fig. 1. Architecture of CARDINAL

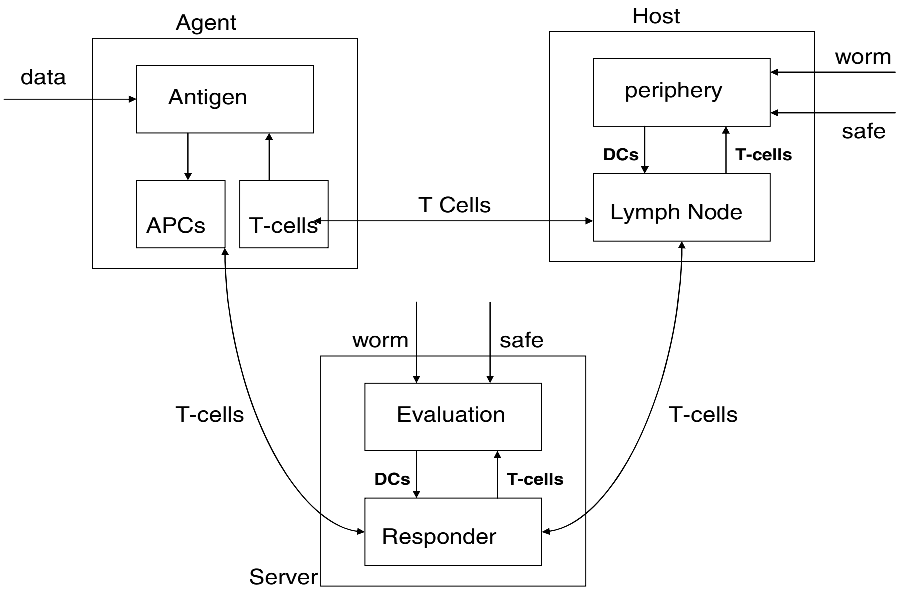


Fig. 2. Architecture of MAAIS

3 Intrusion Detection Mechanism Based on MAAIS

According to [11] [12] [13], we propose improved IDS based on MAAIA. Different from traditional self-nonself paradigm for immune systems, our IDS will first detect

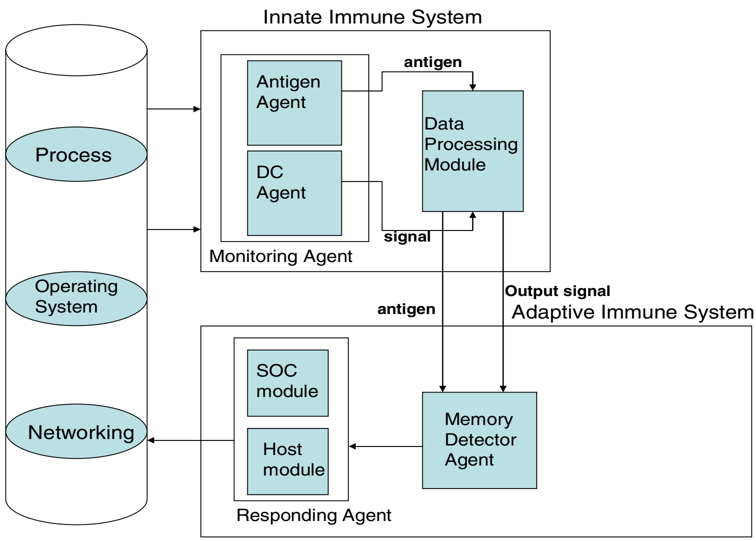


Fig. 3. Components of the MAAIS-based IDS model

danger signals emitted by computer hosts. These danger signals are based on some security threat profile, which defines by system calls generated by running processes. According to [11], threat profile may be composed of excessive CPU, memory load at the host, bandwidth saturation, high connection number of the host, etc. In this multi-agent system, several agents are generated which can communicate each other to emulate functionalities extracted from DT-inspired AIS, see Fig.3.

3.1 Agents in AIS

3.1.1 Antigen Agent (Ag Agent)

Antigens are profiles of input data such as IP packet, which includes IP address, port number, protocol type, and network connection, etc. Ag agent simply parses input data into format of antigens then sends them to the DC agents.

3.1.2 Dentricit Cell Agent (DC Agent)

In order to determine whether a malicious behavior is taking place, IDS needs to analyze the input data, i.e., antigens. DC agents are complex compared to other agents; they are installed at computer hosts. When an Ag agent issues a picked signal, DC agent will calculates the danger value (DV) and analyze the signal corresponding to this antigen.

3.1.3 T-Cell Agent (TC Agent)

When DV of an antigen exceeds some activated threshold, say T_2, it indicates that a high level danger occurs. The Security Operating Center (SOC) will respond to this antigen. TC agents are installed at the SOC; therefore it has all these resources to compare the output category with the original category of each Ag agent, to calculate the overall true positive or accuracy of the virus detections.

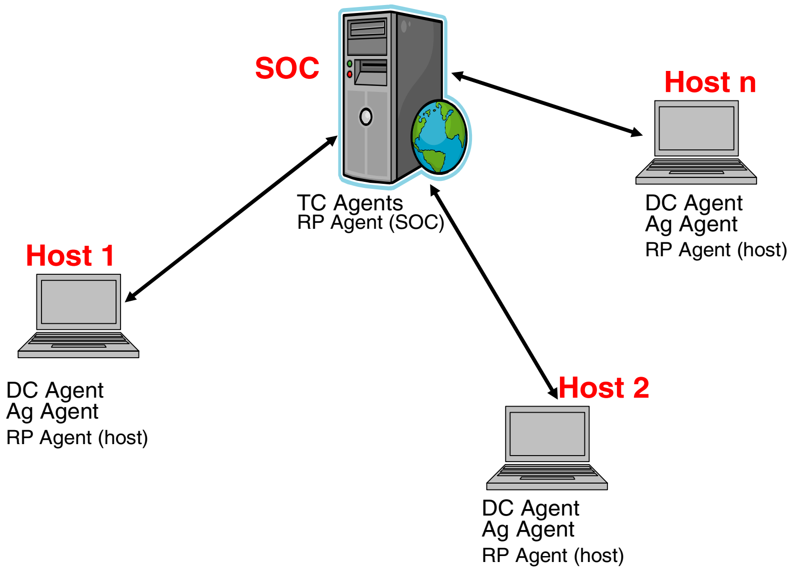


Fig. 4. Architecture of MAAIS-based IDS

TC agents are stimulated by the signals from DCs when the DV exceeds the secondary threshold T_1. Antigen presented by this DC will be "tacked" by this TC agent. According to the design of [12], DC agent will generate a mobile agent to carry this antigen to SOC, if the DV exceeds the threshold T_2. SOC will take stronger action in this case.

3.1.4 Responding Agent (RP Agent)

Ag agents, DC agents and TC agents are coordinating one another to perform immune responses. After DV exceeds threshold value T_2, TC agents will inform RP agent, which is installed at some SOC. It represents that an infected host is detected; RP agents will activate some control measure to such malicious action. Two measures are considered [11]:

1. Reporting to the SOC or security manager, for example, patches downloaded, activate relevant anti-virus software on this infected host and removes virus.
2. Disruption of intrusion, discards a suspicious packets, kill the related process, cut-off infected sub-network. These can prevent large-scale spreading of computer viruses, in particular internet worms, which have high spreading rate by their natures.

3.2 Agents and SOC Processes

3.2.1 Immune Response by DC Agents

The strategy of DC agent immune response is to reduce the false positive rate of corresponding IDS. DC agents first estimate DVs according to the input antigens and classify the threats according to DVs.

The degree of danger in this MAAIS is determined by three factors, namely, attack severity (S), certainty (C) and the length of attack time (T) [12]. There are different aspects of estimating S, C and T. From IDS viewpoints, these factors are functions of CPU usage, memory load, bandwidth saturation and connection numbers of the host. S, C, and T are normalized, namely, $S, C, T \in [0,1]$.

DVs are computed through these factors with their respective weights, namely,

$$DV = \frac{W_S * S + W_C * C + W_T * T}{W_S + W_C + W_T} \tag{1}$$

$DV \in [0,1]$. DC agent is responding according to two DV threshold values, $T_1, T_2 \in [0,1]$. If $T_1 \leq DV \leq T_2$, then DC agent will take the immune response. If $DV > T_2$, then SOC will take a stronger action based on its assessment, also see the following subsection. On the other hand, DC agent will not respond if $DV < T_1$.

3.2.2 Immune Response by Responding Agent

Once DV exceeds T_2, RP agent installed in the SOC is activated and makes a comprehensive evaluation for the received "danger" signal. This evaluation is a crucial factor to mitigate the threat of the whole network. If the evaluation is not good enough, the false positive responding action will produce the damage equal to the one caused by the attack itself. This observation, coincidently agreed with the paradigm of danger theory, suggests that the comprehensive evaluation should be depending not only on the danger signal, but also the number of DC agents emitting danger signals.

Let CE represent the comprehensive evaluation, AVE is the average value of DV exceeding T_2; n is the number of DC agents emitting danger signals, N is the total number of DC agents. The calculations of CE and AVE are as follows; Table 1 is the suggesting weightings W_A, W_n.

$$CE = \frac{W_A * AVE + W_n * (n/N)}{W_A + W_n} \tag{2}$$

$$AVE = \frac{1}{n}\sum_{i=1}^{n} DV_i \tag{3}$$

$CE, AVE \in [0,1]$. There will be a critical task for SOC to define its security measures according to CE values. This issue is out of our scope here.

Table 1. Weightings for CE

W_A	W_n
2	1
1	1
1	2

3.2.3 Agent Communications

One advantage of multi-agent based IDS is the communications and coordination between DC agents from corresponding hosts. The basic idea is as follows. TELL and ASK-based communication permit agents share their internal information, enhancing their performances to respond to intruders [12]. On the other hand, useful knowledge obtained by DC agent should be stored in a database. These databases could also be shared by each DC agent to improve their detection efficiencies.

4 Future Works

The evaluations of three factors S, C, and T are pragmatic issues. On the other hand, threshold values T_1 and T_2 have to be defined by SOCs according to security profiles. For example, the characteristics of general DDOS (distributed denial of services) attacks may help define these threshold values. Due to the paragraph limitation, simulations and evaluations of varied MAAIS will be preceded in the future. Also the responding scenarios of SOCs according to CE values are worth exploring in the future.

5 Conclusions

We propose a MAAIS-based intrusion detection system. The intelligence behind such system is based on the danger theory of human immune systems. In particular computations of danger values with two different thresholds will reduce the false positive rate of danger signals issued by computer hosts. Three agents, namely, Ag agent, DC agent and TC agents are coordinated to exchange information of intrusion detections.

References

1. Hofmeyr, S., Forrest, S.: Immunity by Design. In: Proc. Of the Genetic and Evolutionary Computation Conference (GECCO), pp. 1289–1296 (1999)
2. Aickelin, U., Bentley, P., Cayzer, S., Kim, J.: Danger Theory: The Link between AIS and IDS. In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 147–155. Springer, Heidelberg (2003)
3. Boukerche, A., Machado, R., Juca, K., Sobral, J., Motare, M.: An Agent based and Biological Inspired Real-time Intrusion Detection and Security Model for Computer Network Operations. Computer Communications 20, 2649–2660 (2007)
4. Forrest, S., Beauchemin, C.: Computer Immunology. Computer Communications 20, 2649–2660 (2007)
5. Greensmith, J., Feyereisl, J., Aickelin, U.: The DCA: SOMe Comparison. Evolutionary Intelligence 1(2), 85–112 (2008)
6. Nedjah, N., Abraham, A.: Computational Intelligence in Information Assurance and Security. Springer, Heidelberg (2007)
7. Liu, S., Li, T., Wang, D., Zhao, K., Gong, X., Hu, X., Xu, C., Liang, G.: Immune Multi-agent Active Defense Model for Network Intrusion. In: Wang, T.-D., Li, X.-D., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G.-L., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 101–111. Springer, Heidelberg (2006)

8. Yeom, K.-W., Park, J.-H.: An Artificial Immune System Model for Multi Agents based Resource Discovery in Distributed Environments. In: ICICIC 2006 (2006)
9. Gu, F., Aickelin, U., Greensmith, J.: An Agent-based Classification Model
10. Matzinger, P.: Tolarance, Danger and the Extended Family. Annual Review in Immunology 12, 991–1045 (1994)
11. Zhang, J., Liang, Y.: Integrating Innate and Adaptive Immunity for Worm Detection. In: Second International Workshop on Knowledge Discovery and Data Mining (WKDD 2009), pp. 693–696 (2009)
12. Fu, H., Yuan, X., Wang, N.: Multi-agents Artificial Immune System (MAAIS) Inspired by Danger Theory for Anomaly Detection. In: 2007 International Conference on Computational Intelligence and Security Workshops, pp. 570–573 (2007)
13. Kim, J., Wilson, W., Aickelin, U., McLeod, J.: Cooperative Automated Worm Response and Detection ImmuNe Algorithm (CARDINAL) Inspired by T-Cell Immunity and Tolerance. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 168–181. Springer, Heidelberg (2005)
14. The Ten Most Critical Web Application Security Vulnerabilities, 2007 update, 2002-2007 OWASP Foundation (2007)

Ants-Like Agents: A Model and Analysis Based on Natural Ants Behavior

Mohamed Hamada

University of Aizu
Aizu Wakamatsu, Japan
Hamada@u-aizu.ac.jp

Abstract. Ants are social insects that work in groups to collectively achieve certain goals that cannot be achieved by a single ant. One of the most interesting ants' behaviors is the highly optimized path that ants follow, in their foraging, between the source of food and the colony's nest. Researchers are inspired by such optimized behavior in several applications. In this paper we introduce an integrated environment for ants-like agents based on such ants' behavior. Our model can simulate and test the behavior of such agents under various conditions and environment changes.

1 Introduction

Ant colonies attracted scientists because of their ability to collectively achieve complex tasks through self-organization processes based on simple rules. It is believed that such collectively technique is the main reason for the survival of such small insects through millions of years on earth despite the drastic changes in environment and living conditions through history. Imitating this behavior, computer scientists and engineers are interested in building ants-like agents (e.g. ants-robots, see [11]) that has limited sensing and computational capabilities, but are simple to design, easy to program, and cheap to build. This makes it feasible to deploy groups of such agents, in places inaccessible otherwise, and take advantage of the resulting fault tolerance, parallelism, and collectively achievement of a certain goal. One of the most interesting ants' behaviors is the highly optimized path that ants follow, in their foraging, between the source of food and the colony's nest. The ants' decision is controlled by imitating and following of trails of a chemical substance, called *pheromone*. When there is a choice among several alternative paths, ants choose a path in a probabilistic way, based on the pheromone concentration over the possible paths. This mechanism allows the selection of the shortest path among several ones [7]. Hence, the pheromone concentration on those paths increase more rapidly and they attract more ants. This process of indirect communication relies on a positive feedback mechanism and depends on the environment characteristics, e.g. colony size [10], food type [13], number of food sources [12] and the nature of the environment on which the ants are moving [4]. Such optimized behavior inspired several applications, for example: traveling salesman problem [5], the quadratic assignment problem [6], the job shop scheduling problem [2], the graph coloring problem [3], the vehicle routing problem [1], and network routing algorithm

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 20–29, 2009.

with *digital pheromone* which used by British Telecommunications PLC in London to solve routing problem and to find the shortest path [9].

In this paper we introduce an integrated environment for ants-like agents based on such ants' behavior. Our system can simulate the behavior of such agents under various conditions and environment changes. In our model the ants are moving in random on an environment that contains a randomly distributed source of food. Ants move with some simple rules and can change direction according to environmental information. We test this model with various situations and conditions of the environment to study how the system works with the pheromone information. In the aim of realizing the practical implication of our model, a web-based computer simulation is given. This simulation enables the study and test for the model under different circumstances of the environmental conditions.

The paper is organized as follows. Section two contains a description of our model. In Section three we introduce the design of a web-based simulation of our model. Section four contains an analysis and study of our model and simulation. Finally we conclude our work and discuss further research in Section five.

2 The Model

Our model can be described as follows. At a time *t*, the environment has one colony, A_t ants, and F_t food. All are randomly distributed in the environment. The ants' objectives are to collect food to colony and to spread pheromone, with different levels, to attract other ants' attention. The environment has changeable conditions: ants follow the birth/death property according to its life cycle, more food may be created if necessary, and the pheromone amount can be change accordingly. The flowchart in Figure 1 (a) describes the birth/death property of ants, and in Figure 1 (b) describes the food creation process.

After the colony is initialized randomly at some position in the environment, the process of creating new ants starts. At a time *t*, if the number of ants A_t is less than the maximum amount of ants A_{max} then, if there is enough amount of food, a new ant is created at some random position in the environment, see the flowchart in Figure 1 (a).

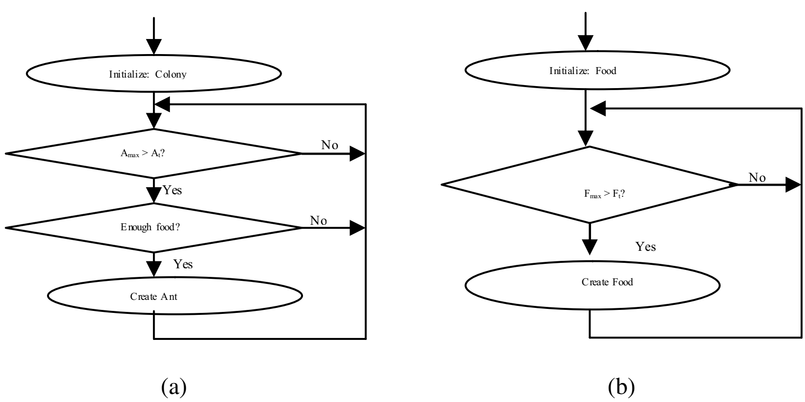


Fig. 1. (a) Ants birth/death process (b) Food creation process

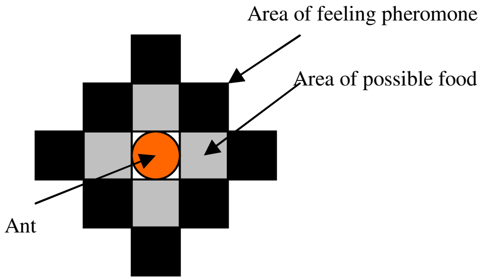


Fig. 2. Ant, food and pheromone positions

New food can also be created in the environment. After initializing the food, at a certain time *t*, if the amount of food F_t is less than the maximum amount of food F_{max} then, a new food is created at some random position in the environment, see the flow-chart in Figure 1(b).

An ant ant_i has a life cycle L_i, when $L_i = 0$, ant_i dies. An ant at a position *p* can find food in some of the surrounding four positions and can fell and affected by the pheromone in the surrounding twelve positions as shown in Figure 2.

Pheromone is the most important simple way of ants' communication. Pheromone is spread by ants and fades away rapidly with time. Ants always spread week pheromone and if they find food, a stronger pheromone is spread to attract other ants' attention. At a time *t*, the amount of pheromone at position *p*=(*x,y*) will be denoted by $Pheromone^t_{x,y}$, when no confusion, we simply use *Pheromone(p)*. We define $Pheromone^t_{x,y}$ recursively as follows:

$$Pheromone^t_{x,y} = Pheromone^{t-1}_{x,y} * \frac{D_{own\%}}{100} + \sum_{d \in Direction} Pheromone(d) * \frac{D_{neighbor\%}}{100}$$

Here $D_{own\%}$ and $D_{neighbor\%}$ denote the pheromone decrease percent in the position *p*=(*x,y*) and the neighboring area respectively. This decrease percentage is natural due to external factors in the environment (in nature, for example wind). The *Direction* set is the set of surrounding areas from the four directions around the position *p*, i.e. *Direction*= {*North, East, South, West*}.

Pheromone at a certain position can affect the amount of pheromone in its neighboring positions. Positions are initialized with a small amount of pheromone (may be zero). If the pheromone amount in a position is greater than zero, this position spreads pheromone to the neighbor positions in the environment. Then the amount of pheromone is decreased in that position. Finally, a new pheromone is calculated based on the decreased amount of pheromone in the position and the amount of pheromone from the neighboring positions in the environment and from other ants. The flowchart of this process is shown in Figure 3.

In our model, such pheromone information is used to help ants to decide how to select the next step. At a time *t*, the pheromone information I_t, in a rectangular area X*Y of the environment, is calculated using the formula:

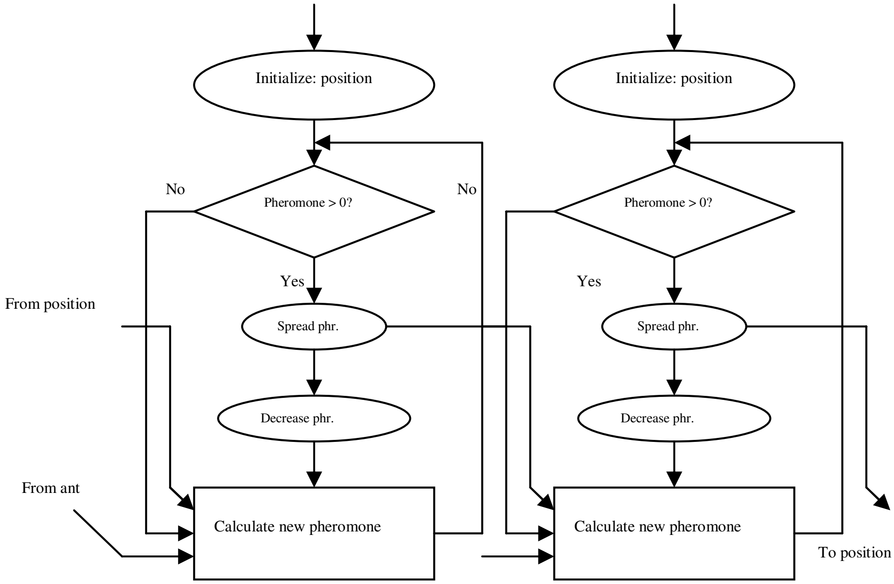


Fig. 3. How pheromone in one position affects another

$$I_t = \frac{\sum_{i,j=0}^{i<X,\ j<Y} Pheromone_t(i, j)}{X * Y}$$

At a time t, the amount of food carried to the colony by an ant_i is denoted by $F(ant_i)$. In our model we assume some dependency among ants. An initial dependency percentage (*Dep%*) can be initialized in the model. If *Dep%* is initialized to zero, then no dependency among components. The dependency *Dep(p)* at some position *p* is calculated by the percentage of the pheromone at *p*, *Pheromone(p)*, multiplied by the initial dependency percentage *Dep%*, i.e.

$$Dep(p) = \frac{Pheromone(p)}{100} * Dep\%$$

A unique behavior in our model is the property of ant's trait *homing instinct*. When ants' life is near its end, ants want to go home colony or rest some time. Also if ants have many food, they want to take back home quickly. At any moment, the ant_i desires to go back home to the colony can be measured by the probability *P(h)* as follows:

$$P(h) = \frac{F(ant_i) * w + F(ant_i) * D(ant_i, colony)}{u}$$

Here $D(ant_i, colony)$ is the distance between the ant_i and the colony, and *w* and *u* are some arbitrary values. When testing our model in our simulation with different values of *w* and *u*, we found that our ants-like agents work as near as real ants when the value of *w*=20 and the value of *u*=8.

In our model, ants move on a random walk. There is a set of five basic actions; *Actions*={*move one step forward, turn right, turn left, rest in same place, pick up food*}. Every action is initialized with a base probability, $P_{base}(i)$ for all i in the set of *Actions*. At a time t, if ant_i is at position (x,y), the probability of action of ant_i is calculated as follows.

- P(move) = P_{base}(move) + Dep(front position) + P(h)
- P(turnR) = P_{base} (turnR) + Dep(right position) + Dep(back position) + P(h)
- P(turnL) = P_{base} (turnL) + Dep(left position) + Dep(back position) + P(h)
- P(pick) = P_{base} (pick) + F(x,y)* Pheromone(x,y)
- P(Rest) = P_{base} (Rest) + ([(m- L_i)*(1+(F(ant_i))/n)

Here $F(x,y)$ and $Pheromone(x,y)$, represent the amount of food and the amount of pheromone at the position (x,y), i.e. the ant_i 's position, respectively. While m and n represent some arbitrary values. Experimentally, we found that our ants-like agents works as near as real ants when the value of m=120 and the value of n=20.

The ants' actions in our model are described in the flowchart of Figure 4.

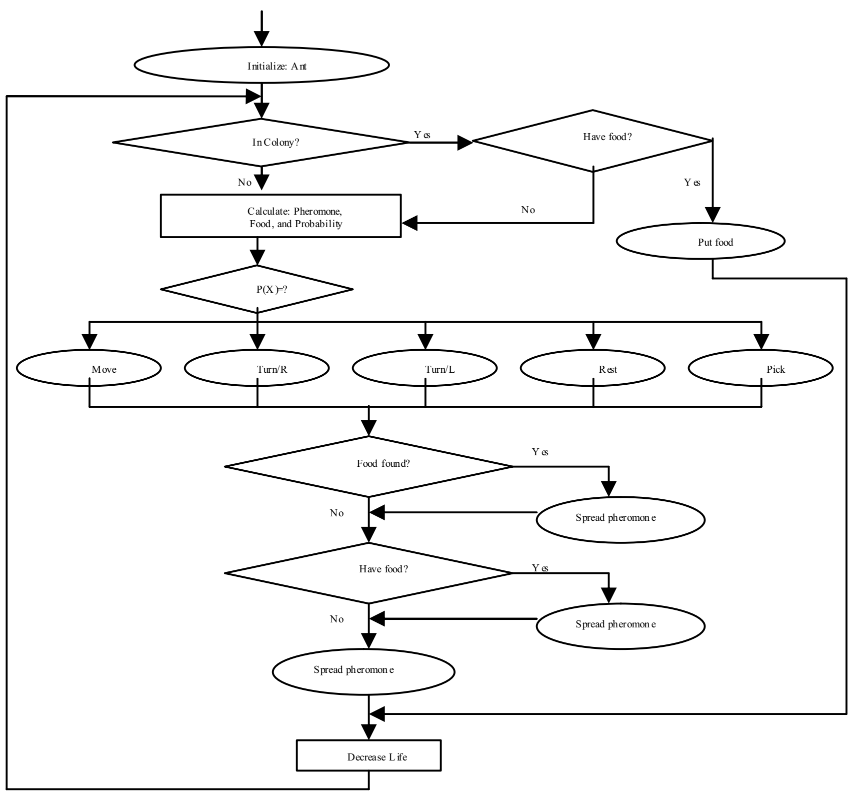


Fig. 4. The possible actions of ants in our model

Ants are randomly created in the environment. If the ant is in colony, then if it holds food, release that food, decrease the life cycle of the ant and check the ant's new situation. If not in colony or in colony but holds no food, then calculate the current pheromone, check for food, and calculate the action probability. According to the value of the action probability, the ant can: move one step forward, turn right, turn left, reset in its position, or pick food. If food is picked, then spread some amount of pheromone. If not, then if the ant already holds food, then spread some amount of pheromone. Otherwise spread normal (small) amount of pheromone. Next decrease the ant's life cycle and repeat the same process till the end of the ant's life. This process is shown in the flowchart in Figure 4.

3 Simulation

In this section we introduce a computer simulation of our ants' model described in section 2. The purpose of this simulation is to visualize the model elements such as ants, food and pheromone. It also enables us to test the model under different environmental conditions. In this simulation, we use *n* x *n* square lattice with absolute boundary conditions to represent the environment. All components such as colony, ants, and food are distributed randomly on the lattice cells. The simulation is designed as a web-based Java applet to enable its use over the internet. An overview of the main menu in our simulator is shown in Figure 5.

Environment is the most important component in our simulation. The environment cellular automaton is used to spread pheromone. Each cell holds an integer value between 0 and 255 to represent the amount of pheromone in it. The simulator allows a

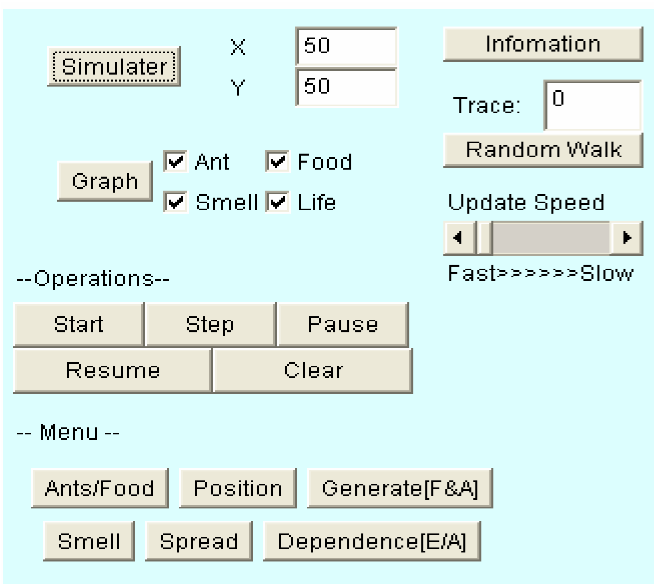
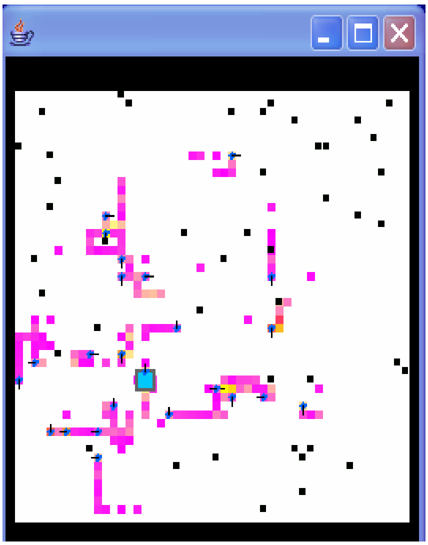
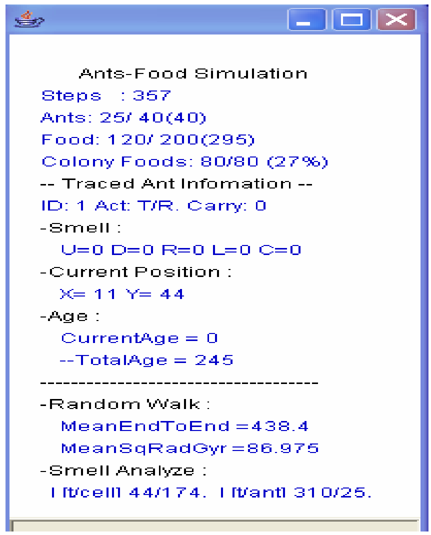


Fig. 5. The main menu of our simulator

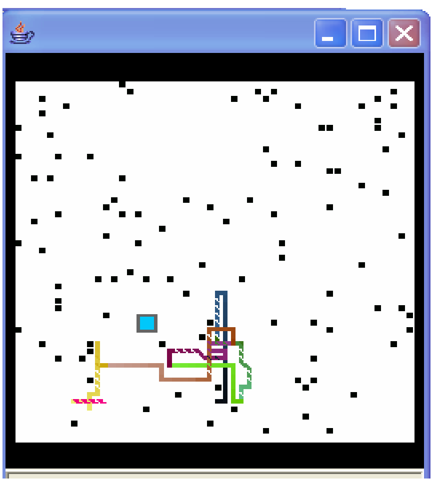
set of pheromone preconditions. Preconditions are preserved by the environment and are used to check the behavior of ants in different situations. Five preconditions are provided; normal, line, frame, two box and squares. These preconditions represent the waves of pheromone in the environment, for example framed precondition causes the pheromone waves to be shaped as a frame. All preconditions keep the present value of pheromone. Normal condition has no preset value.



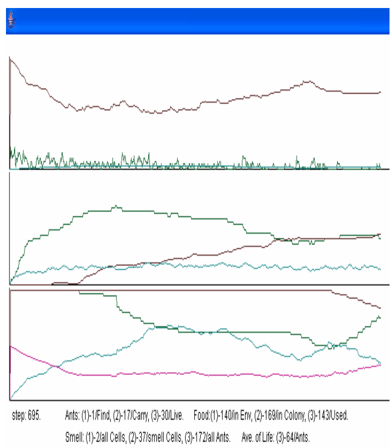
(a) The environment components



(b) Various information



(c) The random walk



(d) The graphical representations

Fig. 6. The simulation components

The cellular automaton, that represents the environment, is determined by the values of *X* and *Y*. The simulator has three major parts: visualization components, operations, and menu. The visualization components show the simulator; which visualize and simulate the model components (colony, ants, food, and pheromone) and the graph which graphically shows the relationship between the model components. We can trace a specific ant by initializing its ID number (from 1 to A_{max}). Then we can follow up the random walk of that ant by using the random walk button. The information part shows much information about the simulation and the traced ant. For example we can show the mean square end to end distance of all the ants' random walks and the mean square radius of gyration. We can also see the amount of pheromone as a function of time, and many more information. The second part of the simulator is the operations. The operations that controls the simulation are simple, it is to start running the simulator as a whole or as step by step, or to clear the initialization so we can try another run with new values. The menu part of the simulator has many options to control the simulation process. We can set the initial values of ants and food, the position of the colony, and we can control the creation of food and ant through changing the parameters (as defined in the previous section). The pheromone can be controlled and initialized. The spread conditions can be modified. Finally the dependence values can also be controlled. Figure 6 (a, b, c, and d) shows all the visual components of the simulator.

4 Analysis

To analyze our model we test it with two different conditions: when food is randomly distributed in a limited area of the environment and when food is randomly distributed in the entire environment. Within each condition five cases are considered for the pheromone dependency conditions. Then we compare the results to see how the model behaves. In this experiment we consider the cellular automaton environment with 50x50 cells, 40 ants, and 200 foods. We compare the values of average pheromone amount $I_{t/a}$, food in colony C_t, current food amount F_t, and number of ants A_t, in a time period *t* up to 1000. The five pheromone dependency conditions are:

No dependency i.e. the value of initial dependency *Dep%* = 0%

a. Dependency is *Dep%* = 30%, pheromone stay longer, and ants attracted to pheromone
b. Dependency is *Dep%* = 30%, pheromone stay shorter, and ants attracted to pheromone
c. Dependency is *Dep%* = 30%, pheromone stay longer, and ants disperse from pheromone
d. Dependency is *Dep%* = 30%, pheromone stay shorter, and ants disperse from pheromone

Cases b and c are testing the closing ants' behavior: ants spread strong pheromone when find food. Cases d and e are testing the avoiding ants' behavior: ants spread strong pheromone when they bring food to colony. The ants' action is pheromone dependent, so it is interesting to see how such pheromone conditions can affect the

Table 1. Experimental results for condition 1

	$I_{t/a}$	C_t	F_t	A_t
a	0	122	43	15
b	321	141	21	18
c	124	127	41	16
d	320	147	28	12
e	193	129	40	12

Table 2. Experimental results for condition 2

	$I_{t/a}$	C_t	F_t	A_t
a	0	130	45	7
b	104	101	56	9
c	134	93	70	7
d	90	103	90	3
e	74	107	77	10

ants' behavior. The following tables summarize the experiment results. Table 1 summarizes the results of the five cases with respect to condition one, i.e. food is randomly distributed in a limited area of the environment. Table 2 summarizes the results for the second condition, i.e. when food is randomly distributed in the entire area of the environment.

Since the mission of ants-like agents in our model is to collect food in colony, the results show that agents work better with condition one than condition two since the average collected food with condition one is 133.2, and the average collected food with condition two is 106.8. This result coincides with our intuition, since food in limited area can be collected faster than food in a large area. Within condition one, we found that case d is better since more food is collected. This shows that sometime too much pheromone can cause ants' confusion. Within condition two case "a" shows a better result since more food is collected.

5 Conclusion

In this work we introduced a model for ants-like agents based on ants' behavior. We also introduced a visual simulation of the model as a web-based java applet. Then we tested and analyzed our model through the web-based simulation. We figure out the conditions in which our model can behave like real ants in nature. However our model is more general, it can cover many more situations. This makes it a suitable model for ants-like agents (such as ants-robots). Like real ants, ants-like agents in our model rely on the pheromone as a mean method of communication and interaction with the environment. This kind of model may not be suitable in some situations in real life, where agents should be able to learn from the environment and deal with changeable environmental conditions more intelligently. For this purpose we plan to introduce intelligent ants: ants that can learn from its environment and change its behavior accordingly. We can utilize one of the neural networks techniques (such as back-propagation) or genetic algorithms for that purpose. Our system is temporarily available at [8].

References

[1] Bullnheimer, B., et al.: An Improved Ant Algorithm for Vehicle Routing Problem (1997)
[2] Clorni, A., et al.: Ant System for Job-Shop Scheduling. JORBEL-Belgian Journal of Operations Research. Statistics and Computer Science 34, 39–53 (1994)

[3] Costa, D., Hertz, A.: Ants Can Color Graphs. Journal of Operation Research Soc. 48, 295–305 (1997)
[4] Detrain, et al.: The Influence of the Physical Environment on the Self-organized Foraging Patterns of Ants. Nturwissenschaften 88(4), 171–174 (2001)
[5] Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Trans. Evol. Comp. 1, 53–66 (1997)
[6] Gambardella, L.M., et al.: Ant Colonies for the Quadratic Assignment Problem. Journal of Operation Research Soc. 50(2), 167–176 (1999)
[7] Goss, et al.: Self Organized Shortcuts in the Argentine Ant. Nturwissenschaften 67, 579–581 (1989)
[8] Hamada, M., Nakamura, Y.: Ants Behavior Simulation (2008), http://www.u-aizu.ac.jp/~hamada/ants/ants.html
[9] N. Hoida Information Technorogy(IT) Using Ants Behavior (in Japanese), COMPUTER WORLD 2001-06-18, No.2001062601, NetLaboratory (2008), http://venus.netlaboratory.com/salon/hoida/2001/20010626.html
[10] Mailleux, et al.: How Does Colony Groth Influence Communication in Ants? Ins. Soc. 50, 24–31 (2003)
[11] MIT Computer Science and Artificial Intelligence Laboratory, The Ants: A Community of Microrobots (2008), http://www.ai.mit.edu/projects/ants/
[12] Nicolis, et al.: Optimality of Collective Choices: a Stochastic Approach. Bull. Math. Biol. 65, 795–808 (2003)
[13] Portha, et al.: How Food Type and Brood Influence Foraging Decisions of Lasius Niger Scouts. Anim. Behav. (2005) (in press)

Simulated Annealing for Multi-agent Coalition Formation

Helena Keinänen

Helsinki University of Technology, Finland
helena.keinanen@tkk.fi

Abstract. We present a simulated annealing algorithm for coalition formation represented as characteristic function games. We provide a detailed analysis of various neighbourhoods for the algorithm, and of their effects to the algorithm's performance. Our practical experiments and comparisons with other methods demonstrate that simulated annealing provides a useful tool to tackle the combinatorics involved in multi-agent coalition formation.

1 Introduction

In multi-agent systems there are often run-time limits for finding solutions to problems, and after such strict deadlines the solutions may become useless. Coalition formation has recently received much attention in multi-agent settings. The applicability of the coalition formation formalisms to multi-agent settings with hard real-time limits requires that the used algorithms are up to their tasks.

In this paper, we address the question of how to achieve high quality solutions to the coalition structure generation problem (see e.g. [4] for a survey of the problem), especially when the search execution times to find solutions are very limited. Motivated by the observations in [12,13] that genetic algorithms provide a useful tool for searching the maximal sum of the values of coalitions, we show that a stochastic local search method of simulated annealing (SA) [1,7] provides a very competitive approach to the problem. This paper extends the preliminary results presented in [5].

One of the main research issues in multi-agent systems is the combinatorial problem of coalition structure generation, where the aim is to coordinate agents into disjoint coalitions. Finding the most beneficial way of partitioning a set of agents is a computationally hard task. It is shown in [11] that finding a coalition structure which yields a given value is an NP-complete problem. Initial algorithms for coalition structure generation are presented in [8,11], and these algorithms are extended and improved in [2,3].

More recent state-of-the-art algorithms directed at optimal coalition structure generation are reported in [9,10]. In multi-agent settings without hard real time limits, these algorithms are quite efficient to solve many real life problem instances. However, in a number of other settings with short execution times and strict deadlines there is still a need for alternative approaches.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 30–39, 2009.

In this paper, we devise novel SA algorithm that can find within extremely short runtimes *sufficiently* good coalition structures. An advantage of the SA algorithm is the much lower space complexity $O(v + n)$ than for the previous algorithms (here v is the size of the representation of the input characteristic function and n the number of agents). We demonstrate experimentally that a specific neighbourhood function with certain properties is crucial for the efficiency of the SA algorithm when applied to coalition formation in characteristic function games. According to the extensive experimental results, the SA algorithm often finds better values or even the optimal coalition structures well before the algorithms in [2,3,8,11].

This paper is organised as follows. In the next section we introduce coalition formation in characteristic function games. In Section 3 we present the SA algorithm for coalition formation. In Section 4 we present the experimental results. We conclude in Section 5.

2 Coalition Formation in Characteristic Function Games

We study coalition formation in characteristic function games (CFGs) [4,8,11] and we use the following standard notation. Let $A = \{a_1, a_2, \ldots, a_n\}$ be the set of agents. A coalition is any non-empty subset S of A, i.e. $S \subseteq A$ such that $S \neq \emptyset$. In CFGs a characteristic function $v : 2^A \rightarrow \mathbb{R}$ assigns a real value to each coalition S which is called the worth of S. Notice that the characteristic function can easily be represented compactly such that the size of its representation is only polynomial in the number n of agents.

A coalition structure CS is a partition of A into mutually disjoint coalitions in which, for all S, $S' \subseteq A$, we have $S \cap S' = \emptyset$, $S \neq S'$ and $\bigcup_{S \in CS} = A$. The value of a coalition structure $V(CS)$ is called social welfare, and it is defined as the sum $V(CS) = \sum_{S \in CS} v(S)$ of the worths of the coalitions included in the coalition structure. Given a set of agents $A = \{a_1, a_2, \ldots, a_n\}$ together with a characteristic function $v : 2^A \rightarrow \mathbb{R}$, our aim is to find a coalition structure CS which maximizes the social welfare among the agents in A.

Sandholm et al. [11] show the number of coalition structures is $O(n^n)$ and $\omega(n^{n/2})$ in an n-agent CFG. Consequently, for general CFGs with a large number of agents exhaustive enumeration of all coalition structures is out of the question in order to find the maximal social welfare. Furthermore, the number of all coalition structures is too large to allow exhaustive search to find a coalition structure with even satisfyingly valuable social welfare. The rest of the paper is devoted to devise an efficient simulated annealing approach for tackling the problem.

3 Simulated Annealing for Coalition Formation

The SA algorithm [1,7] is a widely used stochastic local search method. SA has been applied successfully in various domains and has proved to be an effective optimization technique in practice. It turns out that the SA ideas are well suited to maximize social welfare in CFGs. Algorithm 1 shows our SA algorithm for optimizing the social welfare of a CFG.

Algorithm 1

```
CoalitionFormationSA(c_max, t_init, alpha):
external V()
c = 0;
t = t_init;
CS = random initial coalition structure;
CS_best = CS;
while c < c_max do
  CS' = random neighbour of CS in Neighbour(CS);
  if V(CS') > V(CS) then
    CS = CS';
    if V(CS)>V(CS_best) then CS_best=CS;
  else
    with probability e^((V(CS')-V(CS))/t)
      CS = CS';
  c = c+1;
  t = alpha*t;
end while
return CS_best;
```

As its input, the algorithm requires a characteristic function $v : 2^N \rightarrow \mathrm{I\!R}$ for an n-agent CFG, from which a social welfare can be calculated using a routine $V(CS)$ to sum the worths of all coalitions in a coalition structure CS. Additional inputs to the algorithm are iteration limit c_{max}, initial temperature t_{init}, and the cooling ratio α. The variable c is to keep track of the number of iterations such that the algorithm terminates when c_{max} iterations have been executed. As the algorithm proceeds, the variable CS_{best} records the coalition structure with the highest social welfare among the ones seen.

At each iteration a random neighbour solution CS' of coalition structure CS is picked according to a specific neighbourhood $Neighbour()$. The search proceeds with an adjacent coalition structure CS' of the original coalition structure CS, if CS' yields a better social welfare than CS (i.e. $V(CS') > V(CS)$ holds). Otherwise, the search is continued with CS' with probability $e^{(V(CS')-V(CS))/t}$. The temperature t decreases after each iteration according to an annealing schedule $t = \alpha t$ where $0 < \alpha < 1$.

Generally, the performance of SA algorithms is very sensitive to parameter adjustments as well as to neighbourhood function selection. Given a set of agents $A = \{a_1, a_2, \ldots, a_n\}$ together with a characteristic function $v : 2^A \rightarrow \mathrm{I\!R}$, let M denote the set of all coalition structures that can be formed. The neighbourhood is a function $Neighbour : M \rightarrow 2^M \setminus \emptyset$ which maps coalition structures to the sets of their neighbour coalition structures.

As shown in Sect. 2, in the CFG setting the underlying search spaces may become of astronomical sizes. Therefore, the choice of an appropriate neighbourhood is very essential for the efficiency of Algorithm 1. A desirable property of a neighbourhood is to provide the algorithm with a search landscape where a social welfare maximizing coalition structure can always be found with only a relatively small number of search steps.

Through a comprehensive study of structural aspects of various neighbourhoods for coalition formation in CFGs, we found out that the following two neighbourhoods are particularly appropriate for Algorithm 1.

- **split/merge** neighbourhood, in which $CS' \in \mathit{Neighbour}(CS)$ if and only if CS' can be obtained from CS by either (i) splitting one coalition in CS into two disjoint coalitions in CS', or (ii) merging two distinct coalitions of CS into a single coalition in CS'.
- **shift** neighbourhood, in which $CS' \in \mathit{Neighbour}(CS)$ if and only if CS' can be obtained from CS by shifting exactly one agent from a coalition to another coalition.

The following Lemmata concern the structural aspects of split/merge and shift neighbourhoods. The Lemmata are useful in establishing the desirable property stated in Theorem 1. Namely, Algorithm 1 may always find a social welfare maximizing coalition structure through a small number of search steps, no matter where in the search space it conducts the search.

Lemma 1. *Let A be the set of agents and $n = |A|$ for some $n > 1$. Let M be the set of all coalition structures generated from A, and let $CS, CS' \in M$ be two different coalition structures. Then, CS can be transformed into CS' by doing at most $n - 1$ splits or merges of coalitions.*

Proof. Let i be the number of coalitions in CS and let j be the number of coalitions in CS'. Notice that from every coalition structure with $k > 1$ coalitions one can obtain another coalition structure with $k - 1$ coalitions by doing one merge of some coalitions. Similarly, from every coalition structure with $k < n$ coalitions one can obtain another coalition structure with $k + 1$ coalitions by doing one split of some coalition. It follows that there exist two different ways to obtain CS' from CS as follows:

1. Merge coalitions of CS exactly $i - 1$ times such that a grand coalition (coalition structure with only one coalition) is obtained, and then split coalitions $j - i$ times such that CS' is obtained.
2. Split coalitions of CS exactly $n - i$ times such that a coalition structure with n coalitions is obtained, and then merge coalition structures $n - j$ times such that CS' is obtained.

Thus, CS' can always be obtained from CS by doing either $(i - 1) + (j - 1)$ or $(n - i) + (n - j)$ splits or merges. Assume for a contradiction that the theorem does not hold. Then, both $i - 1 + j - 1 > n - 1$ and $n - i + n - j > n - 1$ must hold. It follows that both $i + j > n + 1$ and $-i - j > -n - 1$ hold, which is obviously a contradiction because $-i - j > -n - 1 \equiv i + j < n + 1$ We thus conclude that the lemma holds. □

Lemma 2. *Let A be the set of agents and $n = |A|$, for some $n > 1$. Let M be the set of all coalition structures generated from A and let $CS, CS' \in M$ be two different coalition structures. Then, CS' can be obtained from CS by shifting at most $n-1$ agents' coalitions.*

Proof. We transform CS' into CS through $n-1$ shifts. Let $S(k)$ denote the coalition of CS with the agent a_k $(1 \leq k \leq n)$, and let $S'(k)$ denote the coalition of CS' with the agent a_k. Let i be the number of coalitions in CS and j the number of coalitions in CS'. We order the coalitions of $CS = (S_1, S_2, \ldots, S_i)$ and $CS' = (S'_1, S'_2, \ldots, S'_j)$ such that the $S_1 = S(1)$ and $S'_1 = S'(1)$; otherwise, the order is irrelevant. We construct CS from CS' by shifting $n-1$ agent's coalitions in CS' in the following way. The agent a_1 is kept in $S'(1)$. For each $2 \leq l \leq n$, shift the agent a_l from the coalition $S'(l)$ to another coalition S'_m, where m is the index of the coalition $S(l)$ in the ordered structure CS. A new coalition will be formed in CS' whenever $m > j$. Obviously, after $n-1$ shifts we have transformed CS' into a coalition structure with i coalitions, and $S_1 = S'_1, S_2 = S'_2, \ldots, S_i = S'_i$ holds. □

For Algorithm 1, a search position is any coalition structure $CS \in M$, and a search step from CS is a pair of coalition structures $(CS, CS') \in M \times M$ such that $CS' \in Neighbour(CS)$. A search trajectory of the algorithm is a finite sequence $(CS_0, CS_1, \ldots, CS_k)$ of search positions such that, for all $0 \leq i \leq k-1$ the pair (CS_i, CS_{i+1}) is a search step.

The following Theorem is an immediate consequence of Lemmata 1 and 2.

Theorem 1. *Let CS be an arbitrary search position of Algorithm 1 with an n-agent CFG. If split/merge or shift neighbourhood is used for the algorithm, then there exists a search trajectory with the following properties: (i) CS is the initial search position of the trajectory, (ii) the trajectory consists of at most $n-1$ search steps, (iii) a coalition structure $CS* = argmax_{CS' \in M} V(CS')$ with the optimal social welfare appearing along the trajectory.*

4 Experimental Results

In order to evaluate the SA algorithm we have implemented it in the C programming language [6], and tested the behaviour on CFG problems taken from the literature. As our benchmarks we use coalition structure generation problems from [2,8,11,12]. To compare our SA algorithm with some other approaches, we have also implemented in C the algorithms presented in [2,8,11]. Unfortunately, an objective experimental comparison to the approaches in [9,10] was not possible because the implementations of these algorithms are not publicly available. In the following subsections we present experimental results on the behaviours of these algorithms for the benchmarks, considering in particular solution quality, robustness, and runtime performances.

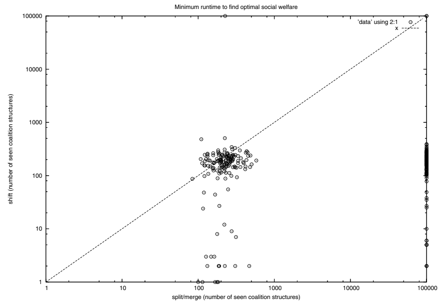


Fig. 1. Neighbourhod relation comparison

4.1 Neighbourhood Relation Comparisons

Figure 1 shows a robustness comparison of split/merge and shift neighbourhoods for the SA algorithm on 300 randomly generated 10-agent CFG problem instances. For each problem instance, a (possibly incomplete) characteristic function v was generated to assign random coalition values between $[0, 1]$. The precision used for each real valued number is 6 digits. For the problem instances, an exhaustive search was first used to find the social welfare maximizing coalition structures, and then the SA algorithm was used to find optimal solutions in the following way. For both neighbourhoods, we executed 11 runs on every problem instance with approximately optimal parameters $\alpha = 0.99$ and $t_{init} = 1$. The runtime limit for each run was set to 100000 coalition structures. We plot the minimum execution times of 11 runs to find an optimal social welfare.

As we can see in Fig. 1 the shift neighbourhood is much more robust than the split/merge. SA with shift neighbourhood is able to find the optimum solution in 298 of the 300 instances. In contrast, SA with the split/merge times out in 136 instances without finding an optimum solution. Moreover, it can be clearly seen in Fig. 1 that SA with the shift neighbourhood is mostly able to find the optimum values with substantially fewer search steps than SA with the split/merge. We tried to experimentally find parameters α and t_{init} to yield better performance for the split/merge neighbourhood, but irrespective of the parameter variation the behaviour of the shift neighbourhood was superior. To compare the solution qualities of SA with the two different neighbourhoods, we investigate the behaviours on 100 randomly generated 20-agent CFG problem instances, again random coalition values in $[0, 1]$. Figure 2 shows the correlation between the solution quality of SA with the split/merge neighbourhood and SA with the shift neighbourhood; (a) illustrates the minimum social welfares found, (b) the median social welfares and (c) the maximum social welfares. Each minimum, median and maximum was measured from 11 runs per neighbourhood. The runtime limit was set to $2^{20-1} = 524288$ coalition structures, and again we use the approximately optimal annealing schedule where $\alpha = 0.99$ and $t_{init} = 1$. Based on Fig. 2 it is easy to see that the split/merge neighbourhood is not very competitive when compared to shift neighbourhood. For instance, SA with

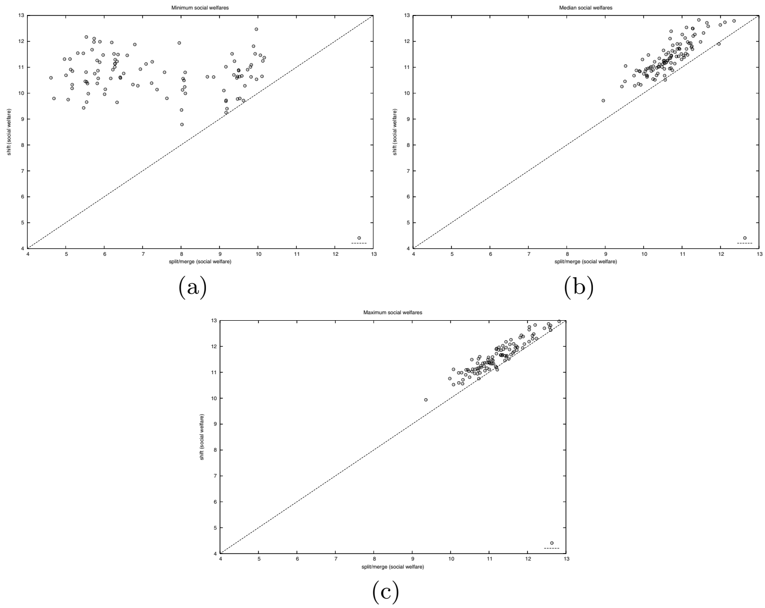


Fig. 2. A comparison of neighbourhood relations

the split/merge neighbourhood finds lower median social welfares than SA with the shift neighbourhood in 98 of the 100 problem instances. The differences in the performance are not large with the maximum social welfares (c), but we notice the performance difference is statistically significant with the minimum social welfares (a). To sum up, these results clearly show that SA with the shift neighbourhood outperforms SA with the split/merge neighbourhood.

4.2 Comparisons with Other Algorithms

Algorithm 1 relies essentially on the search heuristic which is based on the search steps controlled by the temperature and the cooling ration. As a comparison, it is of interest to investigate the behaviour of a rather simpler algorithm obtained by stripping away the SA search heuristics, and laying the bare algorithm to conduct purely randomized search steps in the search landscape. This is obtained by imposing 1 to be the probability to accept worsening moves, by removing the line "with probability $e^{((V(CS')-V(CS))/t)}$" from Algorithm 1. We call the resulting algorithm *Random search* for coalition structure generation. Iteratively, this algorithm simply performs random local search steps on the landscape induced by the selected neighbourhood relation.

To evaluate the SA and Random search algorithms we have also implemented the anytime algorithm from [8,11] outlined in Algorithm 2, whereby $CS(k)$ denotes the set of all coalition structures with exactly k coalitions.

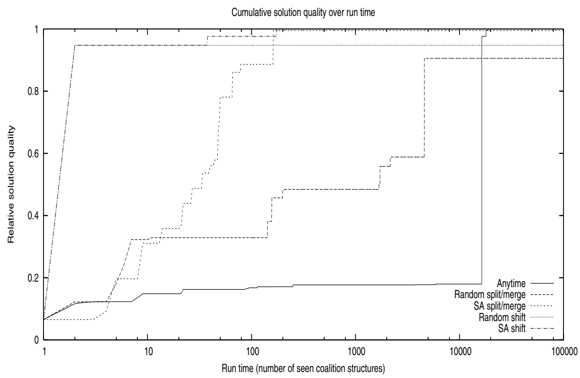


Fig. 3. A comparison of SA, Random search and Anytime algorithms

Algorithm 2

```
1. Search the coalition structure in CS(1).
2. Search the coalition structures in CS(2).
3. As long as there is time left, or until
   all coalition structures has been
   seen, continue the search by
   searching all coalition structures in
   CS(n), CS(n-1), CS(n-2), ...
4. Return the coalition structure that has the
   highest welfare among those seen so far.
```

Given a n-agent CFG the anytime algorithm begins with the grand coalition $CS(1)$, and then continues its search with the coalition structures residing in the sets $CS(2), CS(n), CS(n-1), \ldots$, until a given time limit has been exceeded or the whole search space has been traversed. For a more detailed description of the anytime algorithm see [8,11].

To investigate the behaviour of all the algorithms, we compared the performances of SA, Random search and the anytime algorithm on a set of randomly generated 10-agent CFGs with coalition's values picked randomly from a uniform distribution $[0, 1]$. Figure 3 shows the cumulative solution qualities over runtime (measured as seen coalition structures), on a representative problem instance. In these tests, for the SA the initial temperature is $t_{init} = 1$ and α is fixed to 0.8. Both SA and Random search are run only once. As the figure shows, the SA algorithm finds good solutions very quickly. The SA with the shift neighbourhood finds the optimum within short runtime, and also SA with the split/merge neighbourhood climbs very close to the optimum. Random search with both neighbourhoods manages to find quickly relatively good solutions. However, as SA with the split/merge, Random search does not find any maximal social welfare. The anytime algorithm searches for a long time without finding good solutions, but then finally sees a coalition structure with maximal social welfare.

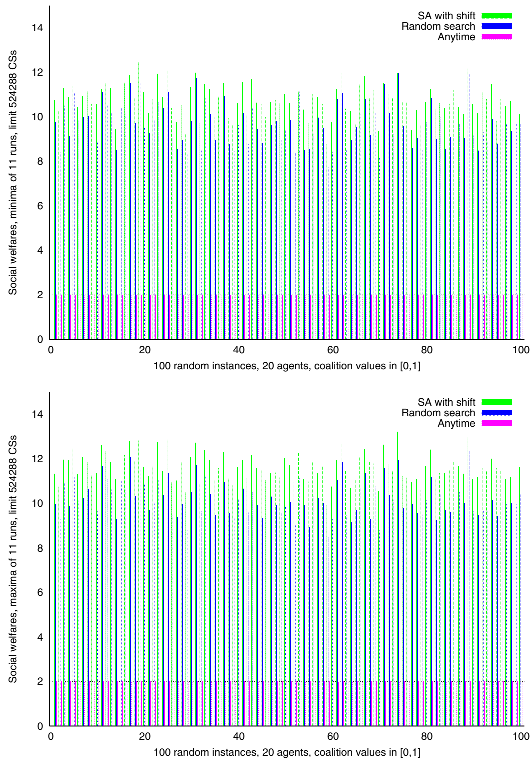


Fig. 4. A comparison between SA, Random search and Anytime algorithms

Notably, for a n-agent CFG, up to a runtime limit of 2^{n-1} search steps (measured as seen coalition structures), Algorithm 2 coincides with the more recent anytime algorithm presented in [2]. From an analytical point of view, we recall that [2,8,11] show both of these anytime algorithms guarantee the limit of 2^{n-1} search steps is sufficient to find a coalition structure whose social welfare is within the bound $n = \min\{k\}$ from the optimum, whereby $k \geq V(CS*)/V(CS*_N)$ with $CS* = argmax_{CS \in M} V(CS)$, $CS*_N = argmax_{CS \in N} V(CS)$, and N any subset $N \subseteq M$ such that $2^{n-1} = |N|$. It is thus particularly important to observe in practice whether our SA algorithm can find better social welfares than the anytime algorithms [2,8,11] when the runtimes are limited to 2^{n-1} search steps.

Consequently, to systematically compare the SA and Random search algorithms with the anytime algorithm, we conducted further experiments on 100 random 20-agent CFGs with coalition's values in $[0, 1]$. For each problem instance, we collected the minimum and maximum social welfares measured from 11 runs per algorithm. Figure 4 shows the results achieved in these tests. The runtime limit for all algorithms was set to $2^{19} = 524288$ coalition structures. We used both SA and the Random search with the shift neighbourhood, and the SA parameters were the approximately optimal $\alpha = 0.99$ and $t_{init} = 1$.

The results are consistent with the results of the previous experiments. For all problem instances the SA algorithm substantially outperforms random search and the anytime algorithms in [2,8,11]. Notice that, given the run time limit $2^{20-1} = 524288$ the anytime algorithms [2,8,11] traverse only coalition structures with two or less coalitions. Consequently, these anytime algorithms cannot find here better social welfares than 2.

5 Conclusion

Coalition structure generation is a computationally hard, combinatorial problem in multi-agent coalition formation. In this paper, we define a SA algorithm for coalition structure generation. We provide a detailed study to reveal properties of neighbourhoods suitable for the optimization of social welfares with the SA. The experiments demonstrate that, with an appropriate neighbourhood, the SA algorithm finds good or even optimal solutions in extremely short runtimes.

References

1. Černý, V.: Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. J. of Opt. Theory and Applications 45, 41–51 (1985)
2. Dang, V.D., Jennings, N.R.: Generating coalition structures with finite bound from the optimal guarantees. In: Proc. 3rd Int. Conf. on Autonomous Agents and Multi-Agent Systems, pp. 546–571 (2004)
3. Dang, V.D., Jennings, N.R.: Coalition structure generation in task based-settings. In: Proc. 17th ECAI, pp. 210–214 (2006)
4. Kahan, J.P., Rapoport, A.: Theories of Coalition Formation. Lawrence Erlbaum Associates Publishers, Hillsdale (1984)
5. Keinänen, H., Keinänen, M.: Simulated annealing for coalition formation. In: Proc. 18th ECAI, pp. 857–858 (2008)
6. Kernighan, B.W., Ritchie, D.M.: The C Programming Language. Prentice-Hall, Englewood Cliffs (1988)
7. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
8. Larson, K., Sandholm, T.: Anytime coalition structure generation: An average case study. J. Expt. Theor. Artif. Intell. 12, 23–42 (2000)
9. Rahwan, T., Ramshurn, S.D., Giovannucci, A., Dang, V.D., Jennings, N.R.: Anytime optimal coalition structure generation. In: Proc. AAAI 2007, pp. 1184–1190 (2007)
10. Rahwan, T., Jennings, N.R.: Coalition structure generation: dynamic programming meets anytime optimization. In: Proc. AAAI 2008, pp. 156–161 (2008)
11. Sandholm, T., Larson, K., Andersson, M., Shehory, O., Tohmé, F.: Coalition structure generation with worst case guarantees. Artificial Intelligence 111, 209–239 (1999)
12. Sen, S., Dutta, P.S.: Searching for optimal coalition structures. In: Proc. 4th Int. Conf. on Multi-Agent Systems, pp. 286–292 (2000)
13. Yang, J., Luo, Z.: Coalition formation mechanism in multi-agent systems based on genetic algorithms. Applied Soft Computing 7, 561–568 (2007)

Structural Changes in an Email-Based Social Network

Krzysztof Juszczyszyn and Katarzyna Musiał

Wroclaw University of Technology,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
{krzysztof.juszczyszyn,katarzyna.musial}@pwr.wroc.pl

Abstract. Different ways of detecting structural changes in email-based social networks are presented in the paper. A social network chosen for experiments was created on the basis of the Wroclaw University of Technology email server logs covering the period of 20 months. Structural parameters like degree centrality and prestige, clustering coefficients as well as betweenness and closeness centrality were computed for each of the consecutive months and their changes were analyzed. Our aim was to make an insight into dynamics of Internet-based social networks based on email service. It was found that the major changes in the structure of the network concern its local topology. Global indices like betweenness and closeness centrality remain relatively stable which also concerns the distribution of the local parameters such as degree centrality and prestige. However, the network size and local topology changes significantly which may be detected with motif analysis and visible changes in node clustering coefficients.

1 Introduction

The variety and many possible areas where social networks (*SN*) can be applied cause that they have become subject of many researches. The concept of *SN* is utilized to describe the relationships between friends, co–workers, members of the particular society, relatives in the family, etc. The main idea of a social network is simple (Fig. 1). It is the set of actors i.e. group of people or organizations, who are the nodes of the network, and ties that link the nodes [1][10][18] by one or more relations [9]. Social network indicates the ways in which actors are related. The tie between actors can be maintained according to either one or several relations [9]. Several examples of social networks can be enumerated: a community of scientists in the given discipline who collaborate and prepare common scientific papers, a corporate partnership networks a set of business leaders who cooperate with one other, friendship network of students, company director networks, etc.

The social networks of internet users differ substantially from the regular ones. Although social networks in the Internet have already been studied in many contexts and many definitions were created, they are not consistent. Also, different researchers name these networks differently. In consequence, these networks are called: computer-supported social networks (CSSN) [19], online social networks [9], web-based social networks, web communities, or virtual communities [1]. Nowadays, based on the kind of service people use, many examples of the social networks in the Internet

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 40–49, 2009.

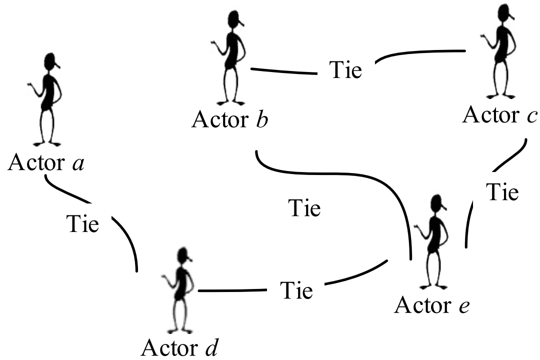


Fig. 1. A regular social network

can be enumerated. To the most commonly known belong: a set of people who date using an online dating system [5], a group of people who are linked to one another by hyperlinks on their homepages [1], customers who buy similar stuffs in the same e-commerce [21], the company staff that communicates with one another via email [22], people who share information by utilizing shared bookmarking systems [14] such as *del.icio.us.*

In this paper we analyse the structural measures (introduced in sec.2) for large social network based on email communication. The set of measures that were taken into consideration are mainly the centrality indices and clustering coefficients. Then their dynamics in the period of 20 months of operation of the university email network are investigated (sec.3). The analysis revealed that some of the parameters are more useful in order to detect structural changes while other are not sensitive to temporal changes occurring in social network.

2 Centrality and Other Structural Measures in Social Networks

The prevailing approaches to social network analysis have provided many measures to determine the characteristics of a member within the network like degree centrality, degree prestige, reachability, connectivity [6], [10], [18]. All of them indicate the importance of a member in the network. There are some particular approaches to evaluate one of the above mentioned measures – centrality: closeness centrality, betweeness centrality, and degree centrality and prestige [7].

Degree centrality $DC(x)$ of member x takes into account the number of outdegree of member x, [17]. It is expressed by the normalized number of neighbors that are adjacent (by means of their outcoming links) to the given person, as follows:

$$DC(x) = \frac{o(x)}{m-1}, \tag{1}$$

where: $o(x)$ – the number of the first level neighbors that are adjacent to x and there is a relation directed from x to them;

m – the total number of members in the social network.

The degree prestige is based on the indegree number so it takes into account the number of members that are adjacent to a particular member of the community [18]. In other words, more prominent people are those who received more nominations from members of the community [2]. The degree prestige $DP(x)$ of member x can be described with the following formula:

$$DP(x) = \frac{i(x)}{m-1}, \tag{2}$$

where $i(x)$ is the number of members from the first level neighborhood that are adjacent to x and have directed relation to it.

The closeness centrality pinpoints how close a member is to all the others within the social network [3]. Its main idea is that the member takes the central position if they can quickly contact other members in the network. A similar idea was studied for hypertext systems [4]. The closeness centrality $CC(x)$ of member x tightly depends on the geodesic distance, i.e. the shortest paths from member x to all other people in the social network [16] and is calculated as follows:

$$CC(x) = \frac{m-1}{\sum_{i=1}^{m} d(x,y)}, \tag{3}$$

where $d(x, y)$ is the length of the shortest path from member x to y.

Betweeness centrality of member x pinpoints to what extent x is between other members. It can be calculated only for undirected relationships by dividing the number of shortest geodesic distances (paths) from y to z by the number of shortest geodesic distances from y to z that pass through member x. This calculation is repeated for all pairs of members y and z, excluding x. Betweeness centrality of member x is the sum of all the outcomes, [8].

On the other hand, the local network connectivity may be investigated by means of *clustering coefficients*. The two coefficients considering the density of connections in the 1– and 2–neighborhood of given node are defined by Eq. 4 and 5.

$$CC1 = \frac{2|E(G1(n))|}{\deg(n)(\deg(n)-1)} \tag{4}$$

$$CC2 = \frac{|E(G1(n))|}{|E(G2(n))|} \tag{5}$$

where:

deg(n) – denotes degree of node *n*,
|*E(G1(n))*| – number of lines among nodes in 1–neighborhood of node *n*,
|*E(G2(n))*| – number of lines among nodes in 1 and 2–neighborhood of node *v*.

We also assume that for a node *n* with *deg(n)* ≤1 all clustering coefficients are 0.

A method that allows a deeper insight into the local network connection patterns is *motif analysis.* Network motifs are small (usually 3 to 7 nodes in size) subgraphs which occur in the given network far more (or less) often then in the equivalent random networks. The statistical significance of motif M is defined by its Z–score Z_M:

$$Z_M = \frac{n_M - \langle n_M^{rand} \rangle}{\sigma_M^{rand}} \tag{6}$$

where n_M is the frequency of motif M in the given network, $\langle n_M^{rand} \rangle$ and σ_M^{rand} are the mean and standard deviation of M's occurrences in the set of random networks, respectively [11]. The actual profile of the network is expressed by the set of Z–scores of the motifs. Their concentration values for all triads form so–called *Triad Significance Profile* of the network (*TSP*) [15].

3 Dynamics of the Structural Measures in Social Networks

3.1 Extraction of the Social Network from Email Communication

The experiments were carried out on the logs from the Wrocław University of Technology (WUT) mail server, which contain only the emails incoming to the staff members as well as organizational units registered at the university [5]. First, the data cleansing process was executed. The bad email addresses were removed from the analysis and the duplicates were unified. WUT social network consists of nodes and relations between these nodes. The email addresses are the nodes of this network and the relationship between two nodes exists if and only if there is any email communication between them. In order to track the temporal changes in the considered network the information from the logs was extracted for every month in the period of February 2006 – September 2007 separately. This allowed us to create 20 networks reflecting the structure of organizational communication between the WUT employees. The size of the networks varied from 3 257 to 4465 nodes, reflecting fact that various numbers of employees were active in different months.

3.2 Changes in the Network Size

Research has revealed that the changes in network size are periodic – this effect is connected with the general business profile of the WUT employees' activities. Obviously – August (months nr 7 – 3257 nodes and 19 – 2905 nodes on Fig.2), as a peak of summer holiday season, may be associated with minimal communication activity of the personnel. This is reflected in small decrease in the number of active email users (network nodes) and rapid drop in the number of network edges – 7941 edges in August 2006 and 7555 edges in August 2007. Similar effect is typical also for July and September, when the holidays respectively start and end.

In the following sections we will investigate how these changes are reflected in the centrality and clustering measures, as well as in the local topology patterns.

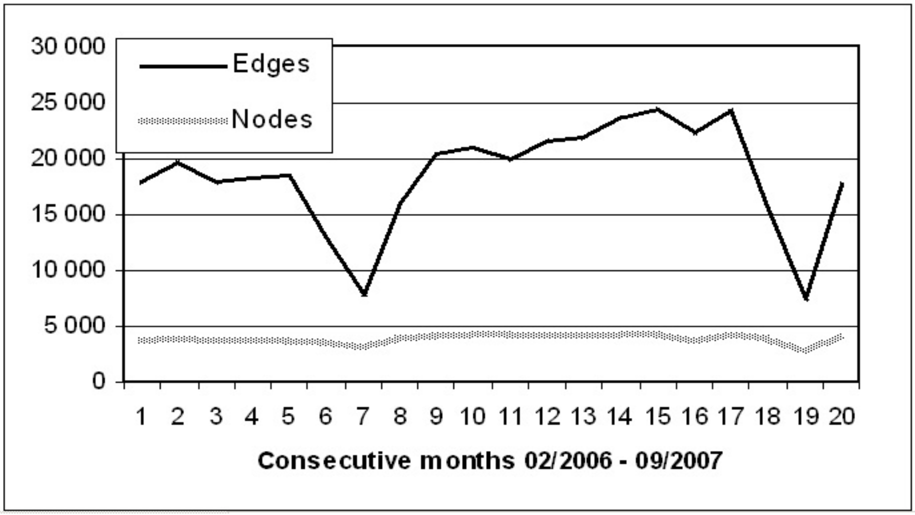


Fig. 2. Temporal changes in the size of WUT email network

3.3 Centrality Measures

While investigating the closeness and betweenness centrality it can be noticed that they are relatively stable during the analysed period of 20 months. The average value for network betweenness centrality was 0.0975 with very small variance of 0.00075. Average network closeness centrality was 0.2719, its variance being 0.00065. This means that these parameters practically do not change despite the visible changes in the size and local structure of the network.

We have also found that the shape of distribution of node closeness centrality as well as the degree prestige and degree centality of the nodes is stable, and their fluctuations are not significant. The variance of the typical distribution of the above parameters are shown below (Figs. 3 and 4):

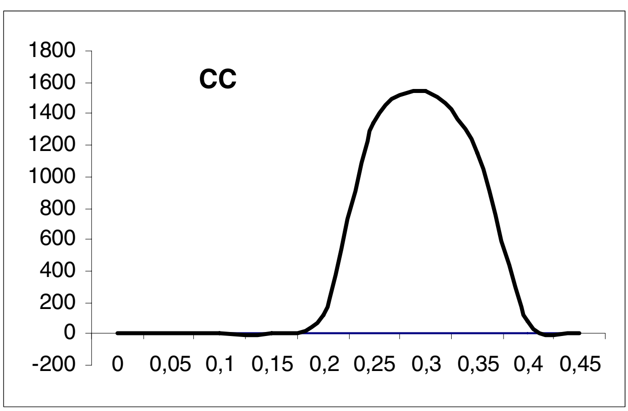


Fig. 3. Typical distribution of CC values for network nodes

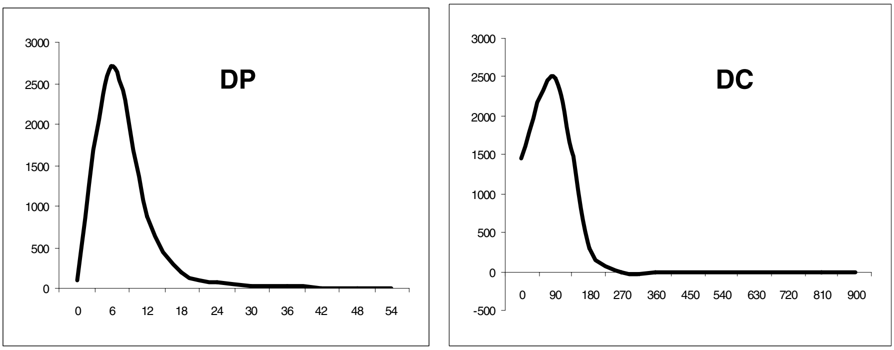


Fig. 4. Typical distribution of DP and DC values for network nodes

Note that there exist nodes with very high – reaching 1000 – DC (broadcasting hubs, typical for email networks) while the typical values of DP are not so huge. However, of course, the average values for DC and DP in the network are equal (5.668 in our case) because the number of incoming and outcoming links is the same.

This is an interesting result showing that, despite changes in network size the average distance between actors (closeness) is almost constant. The same concerns the role of actors which intermediate in the communication (network betweenness). Because the closeness centrality is based on the distance between the actors in the in the network. However, the local topology of the network undergoes visible changes.

3.4 Clustering Coefficients

Along with the size of the network (measured as the number of nodes and edges), two more structural measures were computed clustering coefficients *CC1* and *CC2*. Fig.5 presents the changes in the values of *CC1*. We may note that two moments of the most significant changes in the network structure (months no 7 and 19) are visible in the values of *CC1*.

The temporal characteristic of *CC2* (which takes into account the density of the 2–neighbourhood of given node) is shown on the Fig.6. We can see that the two minima clearly denote August – the peak of summer holidays, associated with rapidly fading communication between the employees of WUT. Moreover, the next two smallest values (for February’2006 and February’2007) point to the two–week winter holidays (also: the end of winter semester on the university).

The above results have shown that there are visible temporal changes in the local structure of network connections. It was also confirmed by means of network motif analysis.

3.5 Motif Analysis

The research (further discussion on network motifs in email-based social networks is presented in [13]) was conducted for 3-node network motifs. There are 13 different motifs that consist of three nodes (Fig. 7). Their IDs=1,2,…,13 are used in the further descriptions interchangeably with the corresponding abbreviations M1, M2,…, M13.

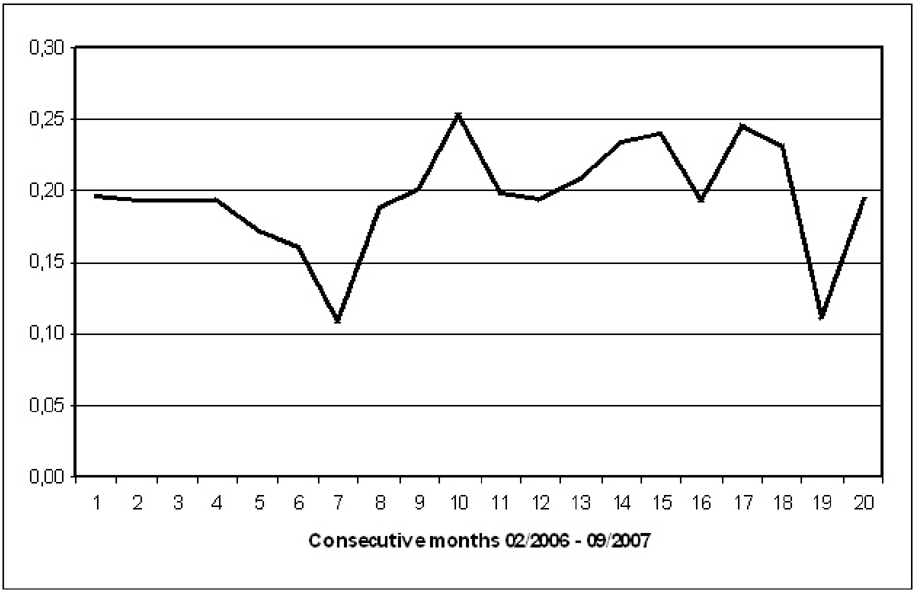


Fig. 5. Average network clustering coefficient *CC1*

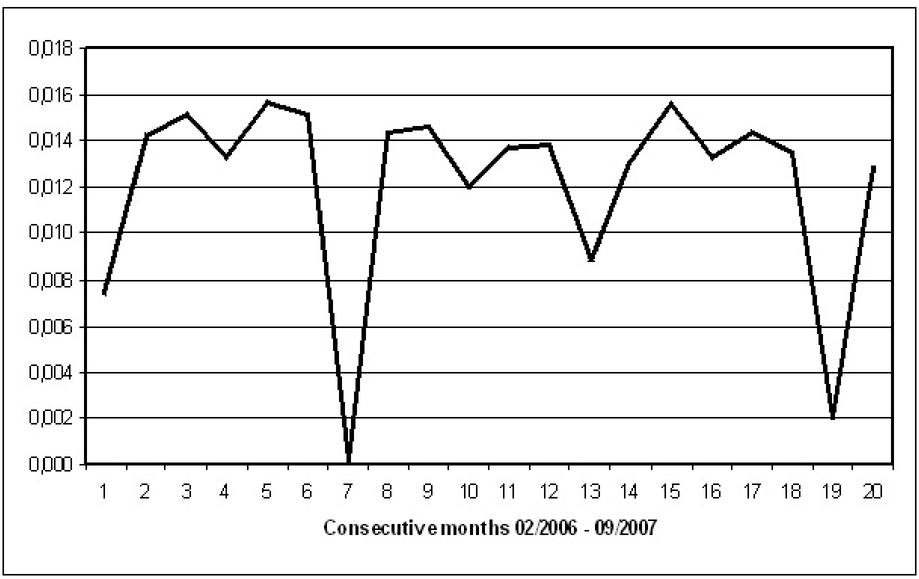


Fig. 6. Average network clustering coefficient *CC2*

The investigation confirmed the typical property of social networks – the small–world phenomenon. Loosely connected motifs with only 2 edges, like M2, M3, M4, M7, M10 occur less frequently compared to the random networks (which property

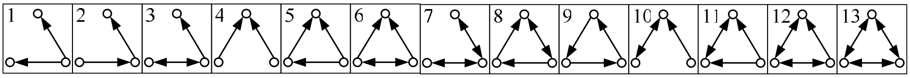


Fig. 7. Directed triads and their IDs

corresponds to their negative Z-score). Triads with edges between any of the three nodes have typically positive (and large) Z-scores [13]. The only exception is M1 which is met relatively often. This reflects specific property of email–based social networks: there are relatively many broadcasting nodes who spread messages which are never answered. On Fig.8. the changes in the Z-score of M1 and M13 are presented in detail [20]. It can be noticed that maximum values (which correspond to strong above-statistical occurrence of subgraphs M1 and M13) were detected for March 2006 (month no. 2) and February 2006 (month no. 13) in the case of M1 - August 2006 (no.6) and 2007 (no.19) for M13 respectively.

Holiday months (July-September) are characterized by relatively big (clearly bigger then these of M1) Z-scores of M13. For the M13 – a fully connected triad – it means that bi-directional communication in dense cliques is sustained regardless of decrease in the number of both unidirectional and bi-directional graph edges.

On the other hand, unidirectional communication characterized by the over-statistical presence of M1 (bigger or equal to M13 Z-scores) is typical for normal

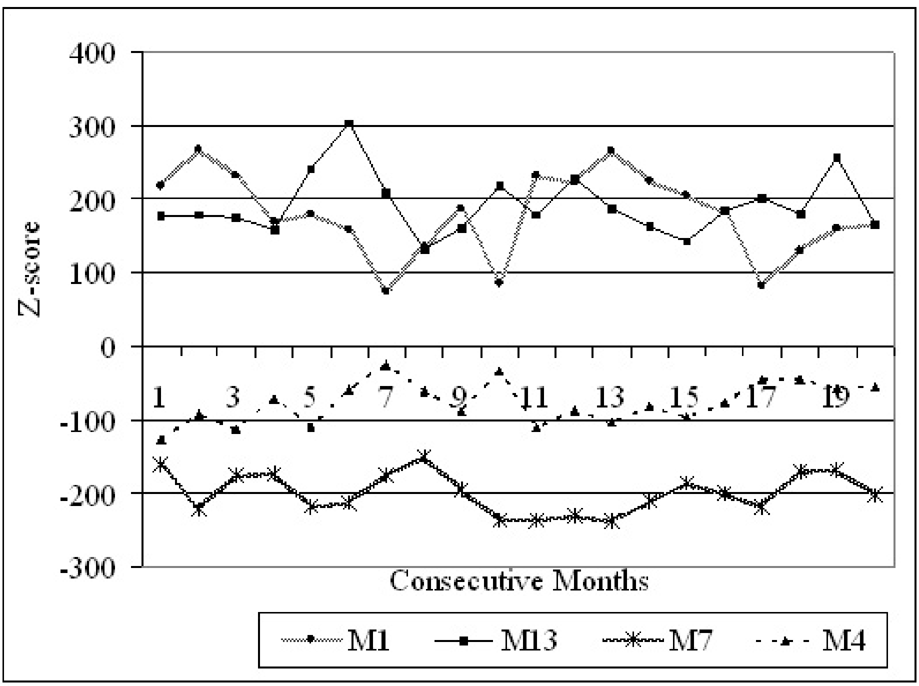


Fig. 8. The changes in Z-score of chosen motifs [12]

business operation during both university semesters. Moreover, from the results of experiments described in [13] and concerning the same dataset, we know that the Z–score of M13 exceeding that of M1 is typical for subnetworks of strong links (network edges formed only by the intense email communication). It can be noticed that these links do not extinct during low activity periods, prevailing over weak, unidirectional ties. Another conclusion concerns M7, a motif containing a node bidirectionally communicating with the two neighbours from different cliques. The changes in the occurrence of M7 (which was shown to have the minimal representation in both classic and email-based social networks, with decisively the smallest Z–score [15][13]), however significant, cannot be associated with the abovementioned temporal changes – the links connecting groups of users remain unchanged regardless of the level of activity.

4 Conclusions

We have analysed temporal changes of the parametres charcterizing structural properties of the large email-based social network. The degree centralities, clustering coefficients and centrality measures were computed for 20 month-long periods. During this time, due to business profile of the university (winter breaks, holidays, gradual network development) the size of the network undergone significant changes measured in the number of nodes and edges.

We concluded that global parameters (closeness and betweenness centrality) remained practically unchanged while the local topology of the investigated network (measured by clustering coefficients and motif analysis) was changing which process is correlated with the dynamics of the network size.

The most interesting result of our experiments is the global network stability expressed by the values of centrality measures and the node distribution of centrality values. We have also identified the measures (like CC2 clustering coefficient and several network motifs) which are especially suited for tracking and the interpretation of the changes in local network topology.

Another important conclusion is that the topology changes of social networks based on Internet communication services may be tracked and investigated mostly on the basis of local information from the immediate vicinity of chosen network nodes. This may also suggest the use of software agents for network structure monitoring – in this case the agents do not have the complete knowledge about network topology and may draw justified conclusions on the basis of partial (local) information.

The next stages of research will address temporal changes in the roles of certain network nodes (esp. hubs and clique connectors) and the possible ways of predicting the network evolution.

Acknowledgements

This work was supported by The Polish Ministry of Science and Higher Education, grant no. N N516 264935.

References

1. Adamic, L.A., Adar, E.: Friends and Neighbors on the Web. Social Networks 25(3), 211–230 (2003)
2. Alexander, C.N.: A method for processing sociometric data. Sociometry 26, 268–269 (1963)
3. Bavelas, A.: Communication patterns in task – oriented groups. Journal of the Acoustical Society of America 22, 271–282 (1950)
4. Botafogo, R.A., Rivlin, E., Shneiderman, B.: Structural analysis of hypertexts: identifying hierarchies and useful metrics. ACM Transaction on Information Systems 10(2), 142–180 (1992)
5. Boyd, D.M.: Friendster and Publicly Articulated Social Networking. In: CHI 2004, pp. 1279–1282. ACM Press, New York (2004)
6. Carrington, P.J., Scott, J., Wasserman, S.: Models and Methods in Social Network Analysis. Cambridge University Press, New York (2005)
7. Freeman, L.C.: Centrality in social networks: Conceptual clarification. Social Networks 1(3), 215–239 (1979)
8. Freeman, L.C., Roeder, D., Mulholland, R.R.: Centrality in Social Networks: II. Experimental Results. Social Networks 2(2), 119–141 (1980)
9. Garton, L., Haythorntwaite, C., Wellman, B.: Studying Online Social Networks. Journal of Computer-Mediated Communication 3(1) (1997)
10. Hanneman, R., Riddle, M.: Introduction to social network methods, `http://faculty.ucr.edu/~hanneman/nettext/` (01.04.2006)
11. Itzkovitz, S., Milo, R., Kashtan, N., Ziv, G., Alon, U.: Subgraphs in random networks. Physical Review E 68, 026127 (2003)
12. Juszczyszyn, K., Kazienko, P., Musiał, K., Gabrys, B.: Temporal changes in connection patterns od an email-based social network. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 9–12. IEEE Computer Society, Los Alamitos (2008)
13. Juszczyszyn, K., Musiał, K., Kazienko, P.: Local Topology of Social Network Based on Motif Analysis. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS(LNAI), vol. 5178, pp. 97–105. Springer, Heidelberg (2008)
14. Millen, D., Feinberg, J., Kerr, B.: Social bookmarking in the enterprise. Queue 3(9) (2005)
15. Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M., Alon, U.: Superfamilies of evolved and designed networks. Science 303(5663), 1538–1542 (2004)
16. Sabidussi, G.: The centrality index of a graph. Psychmetrica 31(4), 581–603 (1966)
17. Shaw, M.E.: Group structure and the behavior of individuals in small groups. Journal of Psychology 38, 139–149 (1954)
18. Wasserman, S., Faust, K.: Social network analysis: Methods and applications. Cambridge University Press, New York (1994)
19. Wellman, B., Salaff, J., Dimitrova, D., Garton, L., Gulia, M., Haythornthwaite, C.: Computer Networks as Social Networks: Collaborative Work, Telework, and Virtual Community. Annual Review Sociology 22, 213–238 (1996)
20. Wernicke, S.: Efficient detection of network motifs. IEEE/ACM Transactions on Computational Biology and Bioinformatics 3(4), 347–359 (2006)
21. Yang, W.S., Dia, J.B., Cheng, H.C., Lin, H.T.: Mining Social Networks for Targeted Advertising. In: 39th Hawaii International International Conference on Systems Science (HICSS-39 2006). IEEE Computer Society, Los Alamitos (2006)
22. Zhu, W., Chen, C., Allen, R.B.: Visualizing an enterprise social network from email. In: 6th ACM/IEEE-CS joint Conference on Digital Libraries. ACM Press, New York (2006)

Concept of a Multi-Agent System for Assisting in Real Estate Appraisals

Tadeusz Lasota[1], Zbigniew Telec[2], Bogdan Trawiński[3], and Krzysztof Trawiński[4]

[1] Wrocław University of Environmental and Life Sciences, Department of Spatial Management
Ul. Norwida 25/27, 50-375 Wroclaw, Poland
[2,3] Wrocław University of Technology, Institute of Informatics,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
[4] European Centre for Soft Computing, Edificio Científico-Tecnológico, 3ª Planta,
C. Gonzalo Gutiérrez Quirós S/N, 33600 Mieres, Asturias, Spain

Abstract. The general architecture of a multi-agent system for real estate appraisal (MAREA) is presented in the paper. The appraisal data warehouse is filled with data drawn from source cadastral databases. Data driven appraisal models are created using different machine learning algorithms. Architecture of the MAREA system in the JADE platform was also proposed. Several experiments aimed to assess the usefulness of different machine learning algorithms for the system were conducted using the KEEL tool.

1 Introduction

The notion of an agent is rather loosely defined, it usually means a software component which is able to act like a human in order to accomplish tasks on behalf of its user. Its properties were gathered by Wooldridge and Jennings [21] and the basic ones are autonomy, social ability, reactivity, and pro-activeness , human-like ones are knowledge, belief, intention, obligation, and emotion, and others are mobility, veracity, benevolence, and rationality. A typology of software agents was proposed by Nwana [15]. She identified seven types of agents: collaborative agents, interface agents, mobile agents, information/internet agents, reactive agents, hybrid agents, smart agents, and the applications which combine agents from two or more of these categories she named heterogeneous agent systems.

Machine learning in multi-agent systems (MAS) has brought substantial interest of researchers, because while designing agents it is not possible to predict all situations that agent may encounter. Therefore, the agents must be able to learn from the environment in order to be autonomous. This area evolved from single-agent learning, through multiple single-agent learning to social multi-agent learning [11]. In single-agent learning, an agent is taking whole decisions, it is responsible for all the given tasks. It models itself, the environment, and their interactions. One of the learning techniques such like explanation-based learning, Q-learning, inductive logic programming might be applied. The environment in MAS is dynamic and more unpredictable, thus application of the ML is more complex. There are two approaches distinguished for that. First one so called multiple single-agent learning uses the

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 50–59, 2009.

single-agent learning technique for all agents. In the latter one agents are aware of existence of the other ones and incorporate knowledge of partners in the learning process. There are several types of learning in the MAS environment. Inductive learning also called supervised learning is a technique where knowledge is gained based on the information examples provided by the expert, whereas reinforcement learning is based on data gained from the environment and agent learns during the runtime. The last approach, unsupervised learning is used seldom. Different approaches to machine learning in multi-agent systems are presented in the numerous overview articles [2], [8], [16], [19].

Information systems proposed in [9], [10] to assist with real estate appraisals have a distributed nature The appraiser accesses the system through the internet and chooses an appropriate section of a city or a district and input the values of the attributes of the property being evaluated into an appropriate appraisal model. The appraisal models are based mostly on sales comparison method, therefore they are generated and learned using actual data extracted from a cadastral system, cadastral digital map and a registry of real estate sales/purchase transactions. The final result calculated using a chosen model is sent to the appraiser as a suggested value of the property. Agent-based approach seems to be suitable to resolve the problem of developing such system. The purpose of this paper is to outline the concept of a multi-agent system assisting property valuation and to assess the usefulness of different machine learning algorithms to be employed in the system.

2 Multi-Agent System for Real Estate Appraisal

The general architecture of the proposed multi-agent system for real estate appraisals (MAREA) is presented in Figure 1. Source databases are heterogeneous and distributed, they comprise a cadastral database, cadastral digital map and registry of real estate sales/purchase transactions. The source databases are continuously updated by an information centre located at a self-government office. Data agents retrieve data necessary to build appraisal models, cleanse and integrate them and save into an appraisal data warehouse. Learning agents create data driven appraisal models from scratch or update existing ones basing on the analysis of data included in the warehouse. Interface, query and result agents serve the appraisers.

Interface Agents. Interface agents interact with appraisers, i.e. system users, gather parameters of properties being appraised and present suggested prices of properties exposed to valuation.

Query Agents. Query agents are generated at each demand of a user. They apply to available models assisting with real estate appraisals. These models are used to generate the suggested prices of properties being appraised.

Result Agents. Result agents process the results provided by the models assisting with valuation and prepare suggested prices to be presented to the users.

Data Agents. Data agents process data gathered in source databases, they extract, cleanse and integrate data preparing them to be useful to learn real estate appraisal models. They create and maintain a certain kind of appraisal data warehouse.

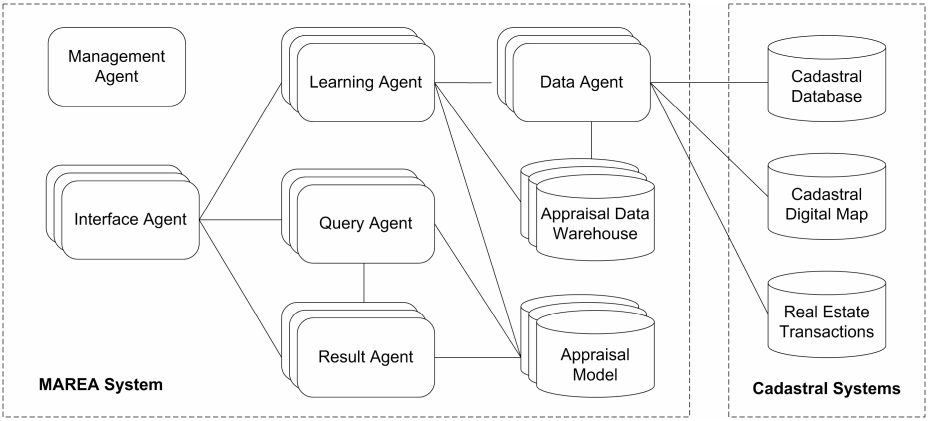


Fig. 1. General architecture of the MAREA system

Learning Agents. Learning agents perform overall and supplementary learning, creating in result different data driven models for real estate appraisal. They are informed by Data Agents that a new portion of data was taken from the source databases and then extracted, cleansed, integrated and introduced into the appraisal data warehouse. After having saved new data they cluster them and analyze to decide whether existing models should be only modified or completely re-generated from scratch.

The JADE (Java Agent Development Framework) platform is convenient to build the MAREA system [3]. The JADE platform developed by Telecom Italia, is an agent platform completely implemented in Java and based on the FIPA specifications [7]. Thanks to the support of many programming libraries and system components like transmitting protocols, addressing services, agents searching or communication procedures the platform allows to create multi-agent systems. The JADE platform provides also a starting platform on which created agents can be executed. The platform has graphic user interface thanks to which it is possible to monitor and manage multi-agent system and the whole platform.

An agent platform consists of one or more agent containers in which agents are activated. Platforms are abstract creations containing one or more physical machines (computers). Container, however, is connected with specific computer, and one computer can have more than one containers activated. This containers do not have to be part of the same platform. One container is distinguished and called main container, system agents are activated in it. The JADE environment's main container contains at least two system agents: AMS (Agent Management System) and DF (Directory Facilitator). First one is responsible for system management, to be more specific it is responsible for the management of a specific agent platform and its collaboration with other platforms. In turn the second one is responsible for keeping the register of agents, it is responsible for registering new agents and services as well as searching for agents responsible for specific services etc.

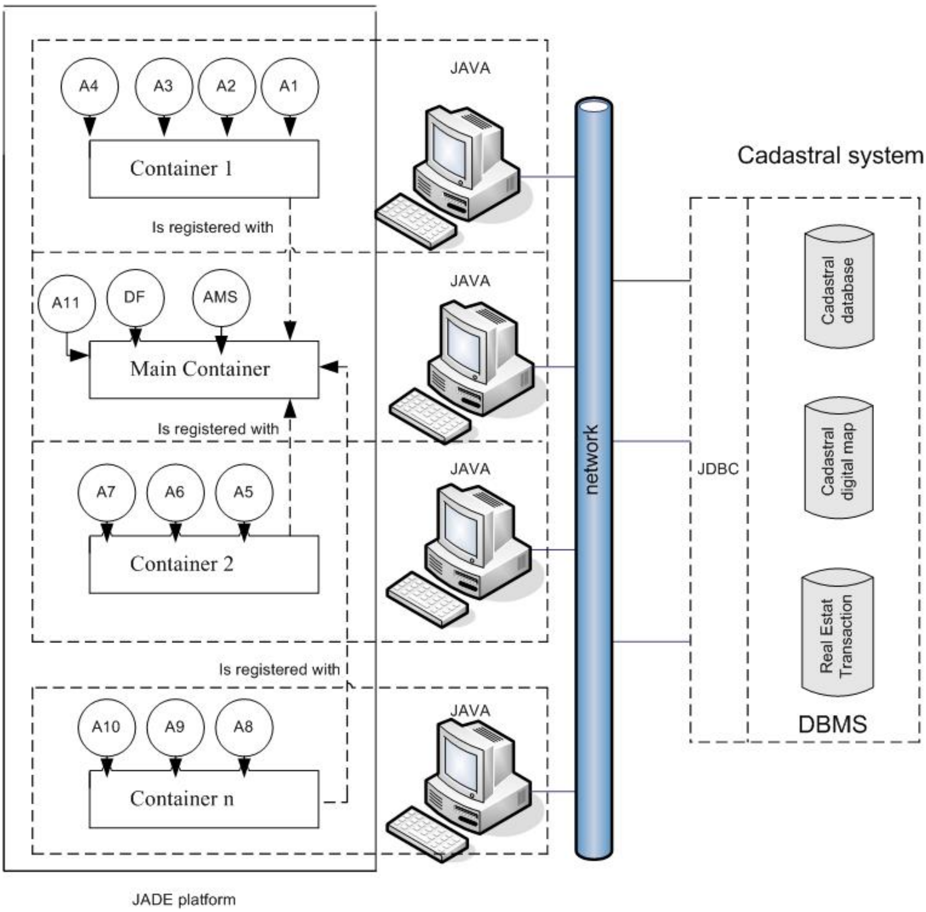


Fig. 2. Architecture of the MAREA system in the JADE platform

The JADE platform supplies mechanisms ready for communication between agents and their collaboration and mobility. JADE's advantage is also possibility to create distributed systems working on many computers at once. Architecture of the MAREA system in the JADE platform is depicted in Figure 2. A MAREA system construction is brought to proper implementation of programming agents in the JADE environment. Each agent is attributed with different role during creation of MAREA system on the JADE platform:

A1 – Interface Agents
A2 – Query Agents
A3 – Result Agents
A4 – Learning Agents (using Linear regression algorithm)
A5 – Learning Agents (using Decision trees for regression)
A6 – Learning Agents (using Support vector machines for regression)

A7 – Learning Agents (using Multilayer perceptron for modeling)
A8 – Learning Agents (using Genetic fuzzy rule learning)
A9 – Learning Agents (using Wang-Mendel algorithm)
A10 – Learning Agents (using Genetic tuning of the Wang-Mendel model)
A11 – Data Agents

The MAREA system is also based on the FIPA standards and therefore allows to activate the system on many computers and to execute distributed calculations. This solution is beneficial for Learning Agents, which use time consuming algorithms. In the MAREA system, if one of the computers is heavily loaded, agents may be copied into other machines within the borders of the JADE platform, so that the agents can perform their functions efficiently.

3 Machine Learning Algorithms Used in Experiments

The second goal of our study was to assess the usefulness of different machine learning algorithms to be employed in a multi-agent system assisting property valuation. For this purpose the KEEL tool was used. The algorithms we applied to create data driven models are listed in Table 1. We divided them into two groups for three methods: deterministic ones, which yield the same results using given sets of training and testing data, and computational intelligence ones, which are based on random techniques and each time produce different output. The deterministic algorithms comprised linear regression methods (LRM), decision trees for regression (DTR) and support vector machines (SVM), in turn the computational intelligence ones contained artificial neural networks incorporating conjugate gradient multilayer perceptron (ANN), genetic fuzzy rule learning system (FRL), and the Wang-Mendel algorithm for fuzzy rule learning tuned by means of evolutionary post-processing algorithms (WMP).

The KEEL is a non-commercial Java software tool designed to assess different algorithms for regression, classification, clustering, pattern mining problems [1]. The KEEL contains several dozen of algorithms for data pre-processing, designing and conducting the experiments, data post-processing and evaluating and visualizing the results obtained, which have been bound into one flexible and user friendly system. The great advantage of the KEEL is the possibility to create different experimental

Table 1. KEEL algorithms used in study

Code	KEEL name	Description
LRM	Regr-LinearLMS	Linear regression models
DTR	Regr-M5	Decision trees for regression
SVM	Regr-NU_SVR	Support vector machines for regression
ANN	Regr-MLPerceptronConj-Grad	Multilayer perceptron for modeling
FLR	Regr-COR_GA	Genetic fuzzy rule learning
WMP	Regr-Fuzzy-WM + Post-G-G-Tuning-FRBSs	Wang-Mendel Algorithm with Global Genetic Tuning of the Fuzzy Partition

sets of data automatically and to perform cross validation of learned models within the same process, what substantially decreases time needed to prepare and accomplish experiments and allows to avoid or diminish the threat of model overfitting. Previous investigations by the authors [12], [13] proved the appropriateness of the KEEL to build and evaluate data driven models for real estate appraisal. Following KEEL algorithms for building, learning and tuning fuzzy models were applied to carry out the experiments.

Regr-LinearLMS (LRM). Linear regression method is a standard statistical approach to build a linear model predicting a value of the variable while knowing the values of the other variables. It uses least mean squares in order to adjust the parameters of the linear model/function [18].

Regr-M5 (DTR). The M5 model tree is a system solving regression problems. It is based on decision tree approach which can learn efficiently being capable to solve tasks with high dimensionality. As it is in decision trees, it builds tree based model, however instead of having values at their nodes it contains multivariate linear regression models at each node. The main advantage of M5 approach over traditional regression trees is that model trees are much smaller than regression trees [17].

Regr-NU_SVR (SVM). The SVM (Support Vector Machine) model uses the sequential minimal optimization training algorithm and treats a given problem in terms of solving a quadratic optimization problem. The NU-SVR, called also v-SVM, for regression problems is an extension of the traditional SVM and it aims to build a loss function. [6].

MLPerceptronConj-Grad (ANN). Proposed by Moller [14] conjugate gradient based algorithm is an approach for supervised learning of the neural nets avoiding a time consuming line-search. Algorithm is performed on networks consisting of multiple layers, usually interconnected in a feed-forward way, where each neuron on layer has directed connections to the neurons of the subsequent layer.

Regr-COR_GA (FLR). The cooperative rules methodology is an approach to generation simple, accurate linguistic fuzzy rules [4]. It is an improvement for the ad hoc data-driven methods. Instead of choosing the consequents with highest performance in current subspaces this methodology investigates other consequents, which will effectively cooperate in knowledge base, to be chosen.

Regr-Fuzzy-WM + Post-G-G-Tuning-FRBSs (WMP). Wang-Mendel algorithm was used. The aim of approach is to provide a common fuzzy rule base from fuzzy rules which are generated from the data pairs (examples) and the linguistic fuzzy rules [20]. Rules are determined by choosing rule with the highest degree from each group established according to their antecedents. The model generated was then tuned using the *Post-G-G-Tuning-FRBSs* post-processing method. Dealing with linguistic variables the method adapts the data base of a Mamdani Fuzzy Rule-Based System (FRBS) with use of genetic algorithm [5].

4 Results of Experiments

Actual data used to generate and learn appraisal models came from the cadastral system and the registry of real estate transactions referring to residential premises sold in one of big Polish cities at market prices within eight years form 1999 to 2006. Four attributes were pointed out as price drivers, namely usable area of premises, floor of the building on which premises are located, year of construction, number of storeys in the building. The set of 5007 transactional records could not be directly used to our experiments, because the prices of premises were changing substantially in the course of time. Therefore each data set was furthermore split into subsets covering individual years, and we might assume that within one year the prices of premises with similar attributes were roughly comparable.

All the experiments were run for all eight one-year subsets of data separately using 10-fold cross validation (10cv) and 5x2-fold cross validation (5x2cv) as well as with no validation (NO-cv). The latter was to reflect the periodic re-learning process of agents. In order to obtain comparable results, the normalization of data was performed using the min-max approach. As fitness function the mean square error (MSE) implemented in the KEEL was applied.

The comparison of the performance of computational intelligence models for testing sets and with 10cv, 5x2cv and with NO-cv is depicted in Figures 3, 5, 7 respectively, and analogous comparison for deterministic models is illustrated in Figures 4, 6, 8. No result for NO-cv applied to DTR algorithm was obtained, because due to an internal bug in the KEEL system application, for this experiment run no model was generated.

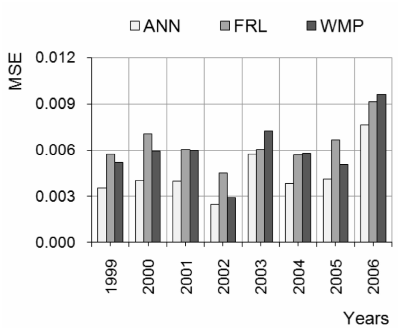


Fig. 3. Performance of soft computing models for 10cv and test sets

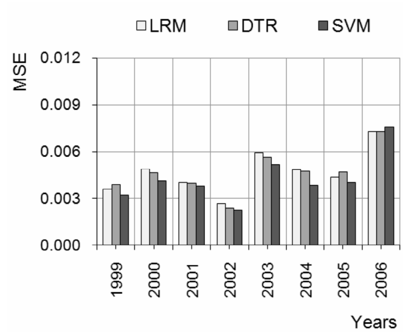


Fig. 4. Performance of deterministic models for 10cv and test sets

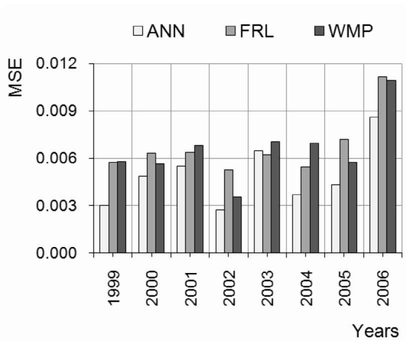


Fig. 5. Performance of soft computing models for 5x2cv and test sets

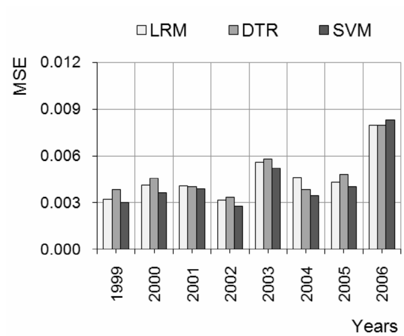


Fig. 6. Performance of deterministic models for 5x2cv and test sets

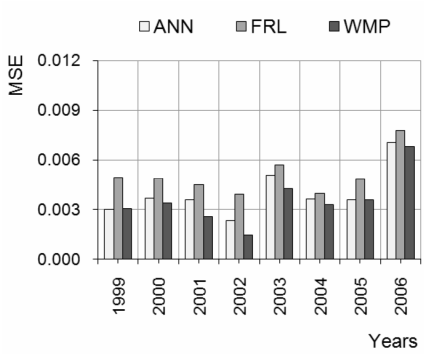


Fig. 7. Performance of soft computing models for NO-cv and test sets

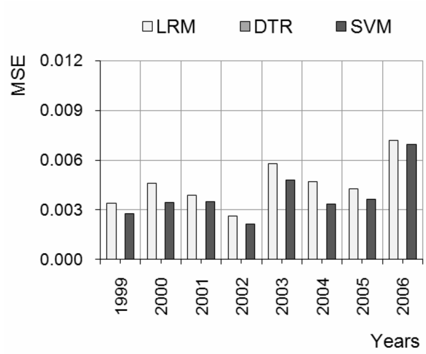


Fig. 8. Performance of deterministic models for NO-cv and test sets

Analyzing Figures 3-8 it can be concluded that the differences between individual algorithms are not enough big to exclude any algorithm and there are not substantial differences between results obtained using different validation methods.

In order to assess the usefulness of individual algorithms to a multi-agent system two additional measures were employed: execution time, and correlation between prices predicted by individual models and actual ones.

Execution times of the algorithms for all eight one-year subsets are presented in Figure 9. They ranged from less than one minute in the case of deterministic methods, through several minutes for ANN and FRL, to some hours for WMP algorithm. The longest time of above 15 hours was observed in the case of WMP algorithm with 10-cv. However periodical re-learning of the model will be made with no validation, and the maximum learning time which is no longer than two hours may be acceptable for this purpose.

Correlation coefficients between actual and predicted prices for all 8 data sets (testing sets) in 10cv and 5x2cv and for all folds of training sets in NO-cv are shown in Figure 10.

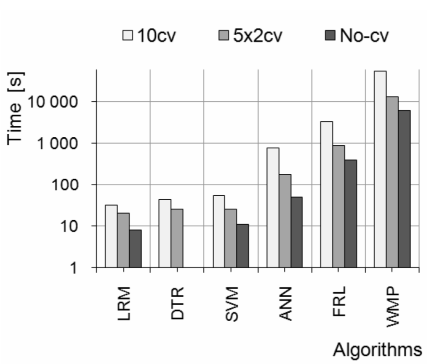


Fig. 9. Execution time of algorithms for all 8 data sets and for 10cv, 5x2cv, and No-cv (logarithmic scale was used)

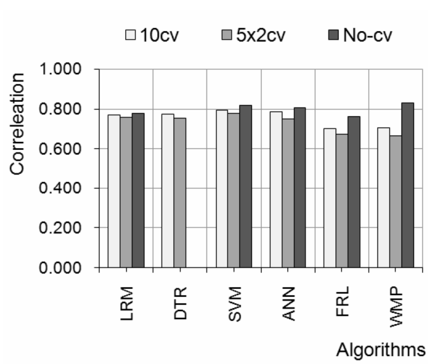


Fig. 10. Correlation coefficients between actual and predicted prices for all 8 data sets and for 10cv, 5x2cv, and No-cv

Values of correlation coefficients ranged from 0.664 to 0.831 what can be considered a fairly good fit.

5 Conclusions and Future Work

The concept of a multi-agent system for real estate appraisals was outlined in the paper. The architecture of the system comprises programming agents which serve appraisers' queries using the appraisal models. The models are frequently updated or re-generated from scratch by learning agents. The source databases comprising a cadastral database, cadastral digital map and registry of real estate sales/purchase transactions are continuously updated by an information centre located at a self-government office. Data agents retrieve data necessary to build appraisal models, cleanse and integrate them and save into an appraisal data warehouse. The core of the system are appraisal models which are created and updated by means of machine learning algorithms. Therefore a major part of the study was devoted to evaluate six deterministic as well as computational intelligence algorithms with regard to their appropriateness to the multi-agent systems. The experiments conducted using actual data and the KEEL tool revealed that all algorithms could be employed. So that the result agent is necessary in order to process the output produced by individual algorithms and to present it adequately to the appraisers. It is planned to implement the system using the JADE platform and to investigate its performance.

References

1. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J.C., Herrera, F.: KEEL: A software tool to assess evolutionary algorithms for data mining problems. Soft Computing 13(3), 307–318 (2009)
2. Alonso, E., d ' Inverno, M., Kudenko, D., Luck, M., Noble, J.: Learning in multi-agent systems. Knowledge Engineering Review 16(3), 277–284 (2001)
3. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE. A White Paper. EXP. 3(3), 6–19 (2003)
4. Casillas, J., Cordón, O., Herrera, F.: COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. IEEE Trans. Syst. Man Cybern. Part B: Cybernetics 32(4), 526–537 (2002)
5. Cordón, O., Herrera, F.: A Three-Stage Evolutionary Process for Learning Descriptive and Approximate Fuzzy Logic Controller Knowledge Bases from Examples. Int. J. Approximate Reasoning 17(4), 369–407 (1997)
6. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using the second order information for training SVM. Journal of Machine Learning Research 6, 1889–1918 (2005)
7. FIPA Agent Management Specification (Standard), Foundation for Intelligent Physical Agents, Geneva (2004)
8. Mannor, S., Shamma, J.S.: Multi-agent learning for engineers. Artificial Intelligence 171(7), 417–422 (2007)

9. Król, D., Lasota, T., Nalepa, W., Trawiński, B.: Fuzzy system model to assist with real estate appraisals. In: Okuno, H.G., Ali, M. (eds.) IEA/AIE 2007. LNCS(LNAI), vol. 4570, pp. 260–269. Springer, Heidelberg (2007)
10. Król, D., Lasota, T., Trawiński, B., Trawiński, K.: Investigation of evolutionary optimization methods of TSK Fuzzy Model for Real Estate Appraisal. International Journal of Hybrid Intelligent Systems 5(3), 111–128 (2008)
11. Kudenko, D., Kazakov, D., Alonso, E.: Machine Learning for Agents and Multi-Agent Systems. In: Plekhanova, V. (ed.) Intelligent Agent Software Engineering. Idea Group Publishing (2002)
12. Lasota, T., Mazurkiewicz, J., Trawiński, B., Trawiński, K.: Investigation of Fuzzy Models for the Valuation of Residential Premises using the KEEL Tool. In: Proceedings of the Eighth International Conference on Hybrid Intelligent Systems (HIS 2008), pp. 258–263. IEEE, Los Alamitos (2008)
13. Lasota, T., Trawiński, B., Trawiński, K.: An attempt to use the KEEL tool to evaluate fuzzy models for real estate appraisal. In: Zgrzywa, A., Choroś, K., Siemiński, A. (eds.) New Trends in Multimedia and Network Information Systems, pp. 125–139. IOS Press, Amsterdam (2008)
14. Moller, F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6, 525–533 (1990)
15. Nwana, H.S.: Software Agents: An Overview. Knowledge Engineering Review 11(3), 205–244 (1996)
16. Panait, L., Luke, S.: Cooperative Multi-Agent Learning: The State of the Art. Autonomous Agents and Multi-Agent Systems 11(3), 387–434 (2005)
17. Quinlan, J.R.: Learning with Continuous Classes. In: 5th Australian Joint Conference on Artificial Intelligence (AI 1992), Singapore, pp. 343–348 (1992)
18. Rustagi, J.S.: Optimization Techniques in Statistics. Academic Press, London (1994)
19. Stone, P., Veloso, M.: Multiagent systems: A survey from a machine learning perspective. Autonomous Robotics 8(3), 345–383 (2000)
20. Wang, L.X., Mendel, J.M.: Generating Fuzzy Rules by Learning from Examples. IEEE Transactions on Systems, Man and Cybernetics 22(6), 1414–1427 (1992)
21. Wooldridge, M.J., Jennings, N.R.: Intelligent agents: Theory and practice. Knowledge Engineering Review 10(2), 115–152 (1995)

Learning as Meaning Negotiation: A Model Based on English Auction

Elisa Burato and Matteo Cristani

Dipartimento di Informatica, Università di Verona, Cà Vignal 2,
strada Le Grazie 15, I-37134 Verona
elisa.burato@univr.it, matteo.cristani@univr.it

Abstract. In this paper we focus upon a specific process of knowledge sharing, the *meaning negotiation* one, in presence of a number of negotiating subjects greater than two. Current approaches of the literature are based upon argumentation techniques, persuasion dialogue and game theory and all of them are specific of the negotiation subject and typically limited to two agents. In this paper we present a formalisation of the problem based upon a specific game, the English Auction, that results independent of the number of involved agents and of the negotiation subject. In our framework, each agent's viewpoint is denoted by a logical theory. We show that the problem as posed in the framework is decidable, under a very general assumption and that the computational complexity of the algorithm is polynomial in the number of agents and in the number of preferred representations of the agents.

1 Introduction

Meaning Negotiation (henceforth MN) is the process that takes place when at least two agents build a shared knowledge starting from a set of disagreeing and incompatible points of view. This can be viewed as one of the major processes in learning activities: when a subject a teaches something to a collection of subjects B, she shares the knowledge she has with each of the subjects in B. However, the subjects in B have their own beliefs, so the knowledge sharing process does not merely consist in passing on a knowledge corpus, since a has to persuade the subjects in B that she is right. Naively speaking, *a justifies* the passage of knowledge to the subjects in B, that means that she negotiates the *meanings of the terms* she uses in order to be understood by the subjects in B on one hand, and the *knowledge* she has, which is delivered by means of the negotiated terms on the other.

In the recent past, Knowledge Representation scholars have considered with interest the problem of how to formalise MN and have tried several approaches. However, a large class of cases do not fall into any of these approaches. In fact, the situation in which the number of agents is at least three and one of these agents is nominated a referee is not treated in the current literature. This would not be great loss if the above mentioned condition were not frequent, but, conversely, in many application areas we can find examples in which this occurs, and particularly in learning processes in which we can identify the referee with the teacher.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 60–69, 2009.

If the learning process is asymmetrical, like for instance in front-to-class traditional model of knowledge delivery, no MN process takes place. Differently, in higher education, in cooperative learning models, and in scientific discussions, mediation is needed much more than just delivery for the participants to the process.

What is common of these applications is that the referee attempts at obtaining the shared knowledge as the most restricted acceptable definition of the terms whose meaning is negotiated. The involved agents, in fact, behave as in an auction. Every agent, not in the role of the referee, tries to resist on her ideal position (the initial offer), but she is committed as well to winning the competition, and therefore she is able to accept a weaker point of view, possibly not her one, that is still acceptable. The referee conversely searches for the best *price*, namely the most generally accepted point of view.

As often argued in many studies on the foundations of artificial intelligence, the meaning of a concept has an axiomatic nature. In other words, the definition of a set of terms can be viewed as a logical theory[1]. Here, the negotiated meaning is not the outcome of the negotiation of common terms of a shared language, but, instead, the set of the axioms representing the agents' most specific viewpoint about the contended words, derived from the agents' initial definitions. How an agent derives her most specific theory is subject to her preferences amongst her definitions. Since a preference relation is commonly represented as a partial order, the subjective hierarchy relation we employ here supports a preferential ordering of the "features" to be negotiated. In order to provide a general model of agents who negotiate theories, that constitute the framework within which we provide a formalisation of MN, we need some further notions with respect to the basic architecture of English Auction. We introduce these notions in the technical part of the paper. Indeed, our framework can be fruitfully employed for representing a debate about *facts of the world*, or abstraction about the world itself, not only for the negotiation of terms. Therefore we can report a discussion in which not only the meaning of a set of terms is considered as subject of discussion but also the descriptions of facts occurring in a world observed by all the *contenders*. In a negotiation process there are two possible outcomes: *agreement* when the process ends with a theory shared by the agents, *disagreement* otherwise. A multiple parties negotiation process may also conclude with a theory to which a list of agreeing agents is attached; this situation arises when not all the agents agree with a theory. As proposed in Game Theory literature[2], we assume that a negotiating agent is characterised by a behavioral nature representing her predisposition to negotiate. There are two natural predispositions: an agent can be *flexible* or she can be *stubborn* towards her axioms' set. The behavioral nature of an agent is determined by axioms that are *fundamental* and *unquestionable* for her. Depending on her behavioural nature, an agent can negotiate her theory by *weakening* her beliefs to accept the opponent's viewpoint, or by *rejecting* a

[1] Note that the two processes of Negotiation of Meaning and Negotiation of Knowledge, by this analysis, fall into the *same formal modelisation*: the negotiation of logical theories.

[2] The general notions introduced henceforth are taken from Kambe's paper [5].

world representation of the counterpart agent. An agent is flexible, when she can accept to weaken her current theory, whilst she is stubborn when she will refuse any generalisation of her theory henceforth. Agents become stubborn when they have tried to negotiate all the axioms they do not consider fundamental and unquestionable. Figure 1 shows a scenario in which four fictitious agents, Trisha, Harry, Craig and Oliver discuss on the effectiveness of game in education. This is an extension of the example of Ravenscroft and McAlister [8], Table 1 p.82.

In this paper we omit all the proofs of the claimed results, for the sake of space. The rest of the paper is organised as follows: next section summarises

(1) Trisha I think... a lot of learning games suck and that's why they're not used
(2) Harry Why do you say that?...
(3) Trisha Let me explain... a large number of games on the market have not caught up with the mainstream video industry with regard to narrative sophistication etc. When they are presentated to an already media savvy kid...they don't play them
(4) Harry I agree because...I guess as million selling computer games don't even make any money till the third edition, it must be impossible for educational developers to keep up
(5) Trisha I agree because...the educational industry is outgunned by the "blow'em up" video game manufacturer, rather depressing...
(6) Craig I disagree because...I imagined educational game being played only in educational establishment and shoot em up being played at home. Therefore, the two should not cross.
(7) Trisha Why do you say that?...if an educational games was seen as fun, wouldn't it stand on equal footing with the shoot em ups?
(8) Harry Let me explain...the funding for comp games is massive compared to educational games, the make more money now than the film industry, but its not an immediate turn around. So for an educational game to compete is nigh on impossible.
(9) Oliver Do we all agree?...that educational games are poorly designed and people are not really interested in them?
(10) Harry I disagree because...although most are, you can probably learn more about ancient history from playing something like rise of nations than you would remember from a book
(11) Trisha Good point.
(12) Oliver Let me explain...the reason why the diffusion of educational games is poor lies on the lack of advertisement of them. It could be useful to produce signed gadgets and clothes for children.
(13) Oliver I agree because... signed gadgets and clothes may become a new fashion for children.
(14) Trisha Good point.
(15) Harry Good point.

Fig. 1. A critical discussion as reasoning game demonstrating constructive conflict, explanation and partial agreement (originally [8] Table 1 p. 82). The sentence "good point" is here considered to be a deliberative conclusion of the discussion by the contenders.

some MN application areas and the current investigations on the MN problem; in third section we present the formalisation of the MN problem based upon English Auction and exhibit the formal properties of the model; fourth section shows the algorithmic approach followed in our formalisation and proves some properties of the proposed algorithms; finally we take some conclusions and sketch further work.

2 Meaning Negotiation Application Areas and Related Work

In this section we make a survey of the main contexts in which the MN process may take place: e-learning, e-Commerce and legal reasoning.

The online learning processes involve at least two agents, a teacher and a student: the latter has to learn something about the world or she has to ground her knowledge by teacher's justification. The dialog between a teacher and a student can be viewed as a litigation because each of them proposes to the opponent her viewpoint and supports it by known axioms or theorems.

In e-Commerce, by MN buyers and sellers can share knowledge and agree on properties of the exchanged objects, as argued, for instance, in Walton and Krabbe's work [11] by means of persuasion dialogue: the seller tries to persuade the buyer about the rightness of the contending object by argumentation techniques as showed in the work of Parsons et al. [6].

Finally, the core of negotiation in legal reasoning is the support to discussions about the truth of assertions made by agents. In a trial each counsel gets her description of facts, built by abduction from the matter in discussion (F) and from her beliefs; the prosecutor advances her theory T_p about the fact F and the same thing does the defendant's advocate by T_d. Then $T_p \models F$ and $T_d \models F$. Also the jury abduces a theory that is a mediation of the prosecutor's and the defendant advocate's theories: it is the result of a MN process. The researches in e-learning by dialogue games aim at implementing automatic tools to support on-line learners. Ravenscroft and Matheson [7] describe the properties of existing tools and discuss about their superficiality (they are incoherent and with no agreed closure), their ambiguity (they lack in shared meanings). Authors approach the e-learning problem in a *investigation by design (IBD)* method that includes the examination of the dialogue features like participant goals, participant roles, dialogue tactics/moves and rules governing the dialog. The two IBD models authors present in the papers are: facilitating and elicit-inform dialogue game; the models differ in the way the teacher informs the student on the rightness of her theory.

On the other hand, the MN problem has been largely investigated in the Artificial Intelligence literature. In general the main approach is to define a negotiation protocol to guide the process. The frameworks in Computational Game Theory literature start from a different point: they use a game to model the negotiation process and the rules of the game govern the agents' discussion. For a rather complete classification of argumentation technics see Walton's work [10]. A well known formal model of such techniques, that is also employed for general formalisation, is the Toulmin model [9].

Burato and Cristani [3] present a formalisation to the problem of Contract Clause Negotiation (CCN) based upon the Bargaining game and involving two players. The authors assume that both agents have an initial set of clauses represented by axioms and that they make in turn a proposal, i.e. a new clauses' set, until an agreement is reached. Benameur et al. [1] discuss the negotiation process as an auction game. They make a survey of the principal auction's type depending on the nature of the negotiation subject and the number of involved "buyers" and "sellers". Bichler [2] presents how a negotiation process can be designed as an auction through a classification scheme based on microeconomics theory.

3 A General Framework for MN with Multiple Parties

In this section we formalise the MN problem as an English Auction game. The framework we present models the MN process by assuming that there are at least three *agents* involved and that one of them is nominated the *auctioneer* (referee) namely the agent who controls and decides the process development. The existence of a referee is relevant but not a constraint to our model because, generally the first bidding agent is the one that regulates the game for every type of auction. The negotiation is performed by a dialogue among the agents involved: bids and accepting or rejecting proposals are send as messages to the agents. A very general definition of *dialogue* may be "an exchange of messages between two or more parties" [4] so the definitions of *agent* and *message* are necessary to formalise what is a negotiation dialogue.

In the rest of this paper we assume that a message exchanged during a dialogue is an assertion, a well-formed formula of classical propositional logic (CPL) or first-order logic (FOL) (we denote by FOL the first-order logic without identity operator) that are here considered commonly known. We define negotiating agent below.

Definition 1. *A* negotiating agent*, indicated by Ag, is defined as a triple* $(\mathcal{L}, Ax, Stub)$ *where $\mathcal{L}$ is the expression language of the agent, Ax is the axioms' set establishing her point of view about the world and $Stub \subset Ax$ represents the axioms that she never concedes.*

By Definition 1, agents are characterised by a behavioral nature interpreting their negotiation power. We assume that agents have at least one axiom and that they have at least one element in the stubborness set.

Given $Ag = (\mathcal{L}, Ax, Stub)$ and the set of the inference rules Δ, we say that Th is the set of theorems built by Ax applying a finite number of inference rules in Δ. $Th = \{\phi | (Ax) \vdash^* \phi\}$ is the theory about world of the agent Ag and Ag is identified by it. Negotiating agents can be in four relations depending on how their theories relate to each other. Agents' theories relate by:

- if $Th_1 \models \bigwedge_{a \in Th_2} a$ and $Th_2 \models \bigwedge_{a \in Th_1} a$ and $\models Th_1 \cup Th_2$ then theories are *equivalent* denoted with $Th_1 \sim Th_2$;
- if $Th_1 \models \bigwedge_{a \in Th_2} a$ and $Th_2 \not\models \bigwedge_{a \in Th_1} a$ and $\models Th_1 \cup Th_2$ then Th_1 is limited respect to Th_2; we denote this with $Th_1 \prec Th_2$;

- if $Th_1 \not\models \bigwedge_{a \in Th_2} a$ and $Th_2 \not\models \bigwedge_{a \in Th_1} a$ and $\models Th_1 \cup Th_2$ then theories are consistent but not comparable. In this case we say that the theories are *compatible* and denote it with $Th_1 \bowtie Th_2$;
- if $\not\models Th_1 \cup Th_2$ the theories are inconsistent; we denote this with $Th_1 \dashv\vdash Th_2$.

The MN subject is to find, if it exists, a theory to which every negotiating agent is committed: Th represents the shared knowledge among a set of agents A iff all of them agree with it, i.e. $\forall i \in A : Th_i \sim Th$.

We now introduce the definition of *theory hierarchy* of an agent; it is a key role concept to our formalisation because a negotiating agent follows it to make her bids during the auction. When an agent is involved in a MN process, she has to think how she can negotiate her theory. She does it by building a graph generated by *weakening* axioms of her theory. We refer to this graph as *Subjective Hierarchy*[3]. By S^* we denote the *deductive closure* of S, namely the set of all theorems derivable by a set of formulas S.

Definition 2. *Given Th, an agent theory, based on a set of axioms Ax, the* subjective hierarchy *of Ag is a graph $TheoryTree = (V, E)$ where:*

i) $Th_0 = Ax^*$ *is the head;*
ii) $V \subseteq \{Th_i\}$ *where for every Th_i, $Th_i \succ Th_0$;*
iii) $E \subseteq \{(Th_i, Th_j)\}$*, where both Th_i and Th_j are in V and $Th_i \prec Th_j$;*
iv) *there's only one leaf, $\emptyset$, representing a theory of the stubbornness axioms, then $Th_\perp = (Stub)^*$*
v) *The relation E is antitransitive.*

The last property of antitransitivity is settled because the represented relation is an order, therefore transitive. The transitive closure of the graph is the preference relation. This settling makes the model most compact. A straightforward consequence of the definition of subjective hierarchy and of the definition of the relation $\prec$ is the following proposition.

Proposition 1. *A subjective hierarchy $TheoryTree = (V, E)$ is a directed acyclic graph.*

The idea governing the formalisation we present here is to find the least generalisation of the theory that the auctioneer proposed that is *good* for all agents or for an acceptable subset of them (decided by the referee and assumed as a MN parameter). When to consider *good* a theory is the core issue of our formalisation. We state that a negotiating agent decides that T is good if her theory S is either a (weak) generalisation of T or the way around. Therefore the two theories are: $T \sim S$ or $T \prec S$ or $T \succ S$. T is not a good theory for an agent when $T \bowtie S$ or $T \dashv\vdash S$.

---
[3] We limit ourselves to those hierarchies that are one-rooted and one-leafed. It can be proved that the results obtained here can be extended to the general case. However this choice substantially simplifies the proofs, so we decided to follow this line for the sake of clarity.

We formalise the MN by viewing it as a sequence of *beats* and every beat is started by the auctioneer and participated by all the other agents in the game. Because the MN process works to build shared knowledge among a set of negotiating agents, we assume that the referee allows the *degree of sharing* she wants to obtain in front of the auction. The degree establishes the minimum number of agreeing agents the auctioneer admits to consider the MN positive: if n is the number of negotiating agents, the degree of sharing α is such that $0 < \alpha \leq n$. The case in which $\alpha = 1$ is clearly degenerated being possible an outcome in which the auctioneer is the only agreeing agent.

A *beat* in the negotiation auction is made by the following steps.

1. The auctioneer proposes a theory Th. Each player adopts the theory in their subjective hierarchy's root.
2. Every agent keeps Th proposed by the auctioneer and computes the relation among her current theory, T, and Th to controls if Th is good.
3. If Th is not good, each agent searches the least general theory in her subjective hierarchy, starting by T node, that is good with respect to Th. The way the agent visits her hierarchy depends on the relation she tests between Th and her current theory.
4. If Th is good or if a good new theory Th_{new} is found in the subjective hierarchy, the agent responds with Th or Th_{new} respectively.
5. The auctioneer receives all the agents' messages.

At the beginning of a beat every agent has adopted a representation theory belonging to her subjective hierarchy defined in Definition 2.

At the end of each beat the referee controls the state of the MN process. She is able to state if the process is positively or negatively ending or if it has to continue so a new beat starts with referee's new proposal.

The general phases the referee has to pass in the MN process by English Auction are:

Proposal. The auctioneer makes a proposal and receives the negotiating agents' proposals.

Elaboration. The auctioneer has to perform a new proposal because the agreement has not been reached. She does so by visiting in breadth-width her subjective hierarchy.

Test. The auctioneer tests if any ending condition is reached with respect to the last proposals received by the negotiating agents.

A negotiating agent makes a bids by following her subjective hierarchy: she visits the graph "horizontally", namely by visiting those nodes that are at the same depth of the current theory with respect to the root, and share the parent node, if she tests the compatibility relation between the theory proposed by the auctioneer Th and her current theory S, or she visits the graph in depth if she tests S is a restriction of Th so that she finds a theory T such that $T \succeq Th \succ S$.

A MN process is considered *positive* when the auctioneer asserts that a commonly accepted theory is found (*total* agreement) or that an acceptable number

α of agreeing agents exist on that theory (*partial* agreement), as *negative* if there is not a common viewpoint shared by all the negotiating agents or by an acceptable number of them.

In the example of Figure 1 the agents close the dialog in a totally positive way because all of them agree on what Oliver said in step 13. Conversely, suppose the dialog ends at step 11, only Trisha and Harry agree on a viewpoint, so the negotiation ends in a partially positive way for them and negatively for the others.

When at the end of a beat the auctioneer controls the MN state she checks the existence of a winning theory among those proposed by the agents and checks if there are other proposals she may perform. The referee is the only one able to decide and state how the MN ends because no other agent knows what the others proposed.

Because of the finiteness of the involved variables, the number of beats in a MN-English Auction game is finite. Consequently, if establishing that a bid is good is a decidable problem then also the MN process based on English Auction is decidable.

4 An Algorithm for Meaning Negotiation Based on English Auction

We provide an algorithm that takes as input $n + 1$ subjective hierarchies where $n \geq 2$ is the number of negotiating agents not nominated referee. At the beginning of the MN process the auctioneer Ag_a, makes a proposal and waits until all the negotiating agents perform their own ones. The bids advanced by all the agents are logical formulae representing their viewpoints. Then if T is the theory an agent wants to bid, she sends $\varphi(T) = \bigwedge_{\psi \in T} \psi$ in the auction. In Figure 2 Algorithm `MNEngAuction` is shown. We henceforth refer to this algorithm as Algorithm MNEA.

If there are only two agents involved in the auction, a referee and a negotiated agent, the MN process automated by `MNEngAuction` computes in the same way as `MeaningNegotiation` algorithm presented in Burato and Cristani [3] page 77.

We now provide some theorems about the correctness (Theorem 1) and the completeness (Theorem 2) of the algorithm (called henceforth MNEA) presented above. Assuming some restrictions to the representation of the agents viewpoints we finally show that the algorithm is polynomially decidable (Theorem 3).

First of all we claim soundness and completeness of the Algorithm.

Theorem 1. *If T is the theory that Algorithm MNEA outputs then T is the least general theory with respect to the one advanced by the auctioneer and which is shared by at least α agents where α is the degree of sharing assumed by the referee.*

Theorem 2. *Let T be the theory which is the least generalisation among those the negotiating agents may admit. MNEA with the the $n+1$ subjective hierarchies $\{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_n, \mathcal{H}_a\}$ in input, computes the MN problem and outputs T.*

ALGORITHM `MNEngAuction`
INPUT: n subj. hier.; a auctioneer; α the degree of sharing;
OUTPUT: A negotiated theory, if possible.

Step 1 Ag_a proposes her current theory Th_a to other agents;
Step 2 For every agent Ag_i that assumes Th_i for $i \in [1, n]$:
Test the relation $\mathfrak{R}$ that relates Th_a and Th_i
Step 2.1 case $\mathfrak{R} = \sim$: Ag_i bids Th_i;
Step 2.2 case $\mathfrak{R} = \prec$: Ag_i bids Th_i;
Step 2.3 case $\mathfrak{R} = \succ$: Ag_i visits in depth $\mathcal{H}_i(Th_i)$; let S be the next node visited, goto Step 2.1 replacing Th_i with S;
Step 2.4 case $\mathfrak{R} = \bowtie$: Ag_i visits in breadth $\mathcal{H}_i(Th_i)$; let S be the next node visited, goto Step 2.1 replacing Th_i with S;
Step 2.5 case $\mathfrak{R} = \dashv\vdash$: Ag_i bids Th_i;
Step 3 Ag_a receives all the agents bids $bids =< b_1, b_2, \ldots, b_n >$;
Step 4 Ag_a finds the least general theory b_i with respect to Th_a;
Step 5 if no bids generalise Th_a:
Step 5.1 if Th_a is not the Ag_a's stubbornness theory, Ag_a visits in breadth $\mathcal{H}_a(Th_a)$; let S be the next node visited and go to Step 1 replacing Th_a with S;
Step 5.2 else goto Step 11;
Step 6 else let b_{win} be the least generalisation of Th_a among those in *bids*.
Step 7 Ag_a counts the number of agents whose bids generalised b_{win}. Let m that number;
Step 8 if $\alpha \leq m$ then goto Step 10;
Step 9 else goto Step 5.1;
Step 10 Outcome: the negotiated theory b_{win}; stop.
Step 11 Outcome: agents cannot agree; stop.

Fig. 2. The algorithm for MN between $n + 1$ agents by English Auction

The complexity of the algorithm automating the MN problem depends on the representation of the agents' points of views. Henceforth we assume that establishing if a theory is good for an agent is a decidable problem.

Theorem 3. *If n is the number of the negotiating agents and h the maximum number of nodes among those of the subjective hierarchies of the negotiating agents, then MNEA computes the MN process in $\mathbf{O}(h^2 \times n \times \mathbf{C})$ where $\mathbf{C}$ is the computational cost of the relationship test among the theories advanced by agents during the MN.*

If we assume that the agent knowledge bases are CPL or FOL formulae without identity, we can prove that Algorithm MNEA computes in polynomial time with respect to negotiating agents, number of theories involved, and number of axioms in the theories. This is claimed in the following theorem.

Theorem 4. *Let n be the number of the negotiating agents, h the maximum number of nodes among those of the subjective hierarchies of the negotiating agents. If m is the maximum number of the axioms in the agents knowledge* *bases and l the maximum number of symbols occurring in an axiom, then MNEA computes the MN process in* $\mathbf{O}(h^2 \times n \times m \times 2^l)$.

A nondeterministic version of the algorithm, based upon linear resolution can be easily built so that we can inherit the computational properties of this approach. Based on this approach we can prove that MN with English Auction is an NP problem[4].

5 Conclusions

There are different ways in which we can take this research further. In particular we aim at establishing the computational lower bound of the formulated problem, starting from the conjecture that the problem is NP-hard. We will moreover investigate the substitution of a referee during the negotiation, introducing a category of non-monotonic negotiation framework, that is well exemplified in real-world examples. A third direction consists in comparing this approach to the elicit-inform dialog of the educational dialog game [7].

References

1. Benameur, H., Chaib-draa, B., Kropf, P.: Multi-item auctions for automatic negotiation. Information and Software Technology 44(5), 291–301 (2002)
2. Bichler, M.: A roadmap to auction-based negotiation protocols for electronic commerce. In: HICSS 2000: Proceedings of the 33rd Hawaii International Conference on System Sciences, Washington, DC, USA, vol. 6, p. 6018. IEEE Computer Society Press, Los Alamitos (2000)
3. Burato, E., Cristani, M.: Contract clause negotiation by game theory. In: ICAIL 2007: Proceedings of the 11th International Conference on Artificial Intelligence and Law, pp. 71–80. ACM, New York (2007)
4. Hamblin, C.L.: Mathematical models of dialogue. Theoria 11(2), 130–155 (1971)
5. Kambe, S.: Bargaining with imperfect commitment. Games and Economic Behavior 28(2), 217–237 (1995)
6. Parsons, S., Sierra, C., Jennings, N.: Agents that reason and negotiate by arguing. J. Logic and Computation 8(3), 261–292 (1998)
7. Ravenscroft, A., Matheson, M.P.: Developing and evaluating dialogue games for collaborative e-learning. Journal of Computer Assisted Learning 18(1), 93–101 (2002)
8. Ravenscroft, A., McAlister, S.: Designing interaction as a dialogue game: Linking socil and conceptual dimensions of the learning process. In: Interactions in Online Education, pp. 73–88. Routledge (2006)
9. Toulmin, S.: The Uses of Argument. Cambridge University Press, Cambridge (2003)
10. Walton, D.N.: Deceptive arguments containing persuasive language and persuasive definitions. Argumentation 19(2), 159–186 (2005)
11. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue. State University of New York Press (1995)

[4] We omit this part of the investigation for the sake of space.

An Approximate Model for Bidders in Sequential Automated Auctions*

Erol Gelenbe and Kumaara Velan

Intelligent Systems and Networks Group
Electrical and Electronic Engineering Department
Imperial College
London SW7 2BT, UK
{e.gelenbe,kumaara.velan}@imperial.ac.uk

Abstract. In this paper, we propose a probabilistic model to study the interaction of bidder and seller agents in sequential automated auctions. We consider a designated "special bidder" (SB) who arrives at an auction and observes the ongoing activities among a number of bidders and the seller jointly, as a stochastic system that is parameterised by the rate of the bidding and selling events. The auction is modelled as a random process with discretised state-space, where the state-space represents the recently attained price for the good. We distinguish the statistical properties of the SB from that of the others, and isolate the system states that denote the desirable outcomes for the SB. In this manner, we define the measures that are of interest to the SB: its winning probability, the average time it takes to purchase an item, and its expected savings with respect to the maximum payable. For tractability, we introduce an approximately equivalent model that yields convenient and closed form expressions of these measures with minimal loss in the accuracy. We examine the effects of the SB's time to bid, and study how decisions may be taken to balance the trade-offs between its interests.

1 Introduction

Computerised auctions are increasingly prevalent in the Internet-based economy, where a significant part of transactions are conducted by automated buyers and sellers acting on behalf of their human counterparts. Since physical restrictions do not apply in this setting, the buyers can move freely between sites, and hence, have access to a large choice of identical or substitutable goods that may be of interest. In many instances, the buyers will be aware of possible sales of similar items in future [3], probably by a different seller at a different place. This knowledge of future availability of similar items adds flexibility to the buyers

* This research was undertaken as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Networks) project and is jointly funded by a BAE Systems and EPSRC (Engineering and Physical Research Council) strategic partnership (EP/C548051/1).

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 70–79, 2009.

decisions; one may choose to restrain oneself from bidding aggressively and pull back under intense competition, in hope of securing a better deal later.

We can also imagine the good on sale being non-exhaustible. For instance, in allocation problems [4] where the resource on offer is not actually sold, but rented out for some period, and is returned to the seller for reuse. This concept may find application in computer networks where the bandwidth is rented to potential users, or in disaster management applications where the resource may be an ambulance, fire engine, or help personnel, that is dispatched to the most needy. Again, the buyers will have to balance their need to consume the resource immediately against their willingness to pay, possibly a higher cost, for the impatience.

Recent work [1] has proposed a model to study the outcomes of ascending auctions over repeated cycles, and [2] has extended it to a network of interconnected auctions. In this paper, we generalise the approach in [1], to analyse systems where bidders with distinguishable statistical properties engage in an auction. Suppose a "special bidder" (SB) arrives and observes ongoing activities of the other bidders and the seller combined, as a stochastic system that is parameterised by the rate of the occuring events. The SB's problem is then to adjust its decisions with respect to the rest of the agents so as to maximise its benefits. To this end, we propose a mathematical model to characterise the performance measures that are of interest to the SB, and study its trade-offs.

In the following parts, we first outline the model to be studied and detail the mathematical formalisation in Sect. 2. Then, we introduce an approximately equivalent model that yields tractability, and provide analytical solutions for this case. In Sect. 3, we define the performance measures that are of interest to the SB and the seller, where we show that the approximate solutions allow us to derive these measures in neat closed forms that facilitate straightforward computation. Numerical examples are given in Sect. 4, where we illustrate the model's predictions and evaluate the accuracy of the approximation approach. Finally, we draw conclusions in Sect. 5, and suggest future work.

2 An Ascending Auction with Discretised Increments

We consider an ascending auction, where the possible valuations of successive bids are increasing by one unit price with respect to the previous offer. The analysis follows directly if the increments take on any size, so long as they are discrete and fixed in advance; but this case is not discussed here. Suppose there are $n + 1$ participating bidders, one of whom is the SB. Each of the "other" bidders raises the price at some rate λ, while the SB raises the price at rate β; these parameters model the average of the random delay before the price is incremented. Here we let the rest of the bidders collectively share a common statistical behaviour that is distinct from the SB, but it will be easy to extend the model to distinguish behaviours within smaller groups among the other bidders.

We assume that, during the auction, the bidders will not raise the price by any more than what is minimally necessary to stay as the winning bidder; in fact this is the standard approach used by the eBay's proxy bidding agent [5]. The good on sale will have some valuation $v > 0$ that represents the maximum price anyone is willing to pay, and all bidding activities stop once this maximum is achieved. To start with, we let v take a known and fixed valuation, and subsequently, show how the results generalise when it is replaced by a random variable V.

Upon receiving a bid offer, the seller may deliberate on accepting it while waiting for a better deal. This duration is modelled as a random variable with average δ^{-1}. If, when an offer is being thus considered, a higher bid is made, the seller rejects the old bid to consider the new offer. The rejected bidder will rejoin the rest of the bidders, and may attempt bidding again. On the other hand if, at some point during the auction, the waiting time lapses without interruption from incoming bids, then the auction concludes with a sale to the owner of the highest bid and enters a random "rest" period with an average r^{-1}. This is simply the overhead duration until the next auction can start. For instance, in the case where the item is rented out, instead of sold, this rest time will include the rent duration before it is returned for another round of auction.

Following that, the auction restarts and repeats itself for an infinite number of times, each auction being independent of the previous. All the discussed random variables are modelled via exponential distributions with parameters of their respective rates, and they maintain their statistical properties between auctions.

2.1 The Mathematical Model

Let $\{ t_i : i = 1, 2, \dots \}$, be the sequence of instants when the i–th auction starts, and assume $t_1 = 0$. For any of the auctions, using a fixed v, we model the system as a continuous time Markov process $\{ X(t) : t \geq 0 \}$, which takes valuation from the state-space $\{ 0, O(l), R(l), A(O, l), A(R, l) \}$ where $1 \leq l \leq v$. The valuation $X(t_i + t) = 0$ denotes the case when the auction has restarted for the i–th round and is yet to receive any bids after some time $t \geq 0$. Then $t_{i+1} = \inf\{ t : t > t_i \text{ and } X(t_{i+1}) = 0 \}$ defines the restarting instants. Using these reference points, we can describe all the state valuations.

1. The state $X(t_i + t) = R(l)$, if at time $t_i + t$ for $t > 0$ during the i–th auction $(t_i + t < t_{i+1})$, the price has attained valuation l for $1 \leq l \leq v$, where the l–th bid was placed by the SB (disregarding all previous $l - 1$ bids).
2. The state $X(t_i + t) = A(R, l)$ corresponds to the case at time $t_i + t$ for $t > 0$ during the i–th auction $(t_i + t < t_{i+1})$, the seller has accepted the bid placed by the SB at price l for some $1 \leq l \leq v$.
3. The state $X(t_i + t) = O(l)$, if at time $t_i + t$ for $t > 0$ during the i–th auction $(t_i + t < t_{i+1})$, the price has attained valuation l for $1 \leq l \leq v$, where the l–th bid was not placed by the SB.
4. The state $X(t_i + t) = A(O, l)$, if at time $t_i + t$ for $t > 0$ during the i–th auction $(t_i + t < t_{i+1})$, the seller has accepted the highest bid at price l for some $1 \leq l \leq v$, which was not owned by the SB.

When the auction is active, any bidder, with the exception of the winning bidder, may place a bid at its respective rate. Thus, at any price $1 \leq l \leq v - 1$, only n

bidders will be active, while at price 0 all $n+1$ will be involved. No bidders will raise the price beyond valuation v.

The transition from any state $U(l)$ to the corresponding state $A(U,l)$, where $U=O,R$ and $1 \le l \le v$, marks the event of a sale at price l, and occurs at rate δ; while the transition from any state $A(U,l)$, where $U=O,R$ and $1 \le l \le v$, to state 0 denotes the event of an auction restart, and occurs at rate r.

Let $P(\cdot)$ represent the stationary probability distribution for the Markov chain with state-space $\{\,0, O(l), R(l), A(O,l), A(R,l) : 1 \le l \le v\,\}$. Then these probabilities will satisfy the following system balance equations:

$$\begin{aligned}
&P(O(1))((n-1)\lambda+\beta+\delta) = n\lambda P(0), \\
&P(R(1))(n\lambda+\delta) = \beta P(0), \\
&P(O(l))((n-1)\lambda+\beta+\delta) = (n-1)\lambda P(O(l-1)) \\
&\qquad\qquad\qquad\qquad\qquad + n\lambda P(R(l-1)), \quad 2 \le l \le v-1, \\
&P(R(l))(n\lambda+\delta) = \beta P(O(l-1)), \quad 2 \le l \le v-1, \\
&P(O(v))\delta = (n-1)\lambda P(O(v-1)) + n\lambda P(R(v-1)), \\
&P(R(v))\delta = \beta P(O(v-1)), \\
&P(A(O,l))r = \delta P(O(l)), \quad 1 \le l \le v, \\
&P(A(R,l))r = \delta P(R(l)), \quad 1 \le l \le v, \\
&P(0)(n\lambda+\beta) = r \sum_{U=O,R} \sum_{l=1}^{v} P(A(U,l)), \\
&1 = P(0) + \sum_{U=O,R} \sum_{l=1}^{v} \big[P(U(l)) + P(A(U,l))\big].
\end{aligned} \tag{1}$$

Solving this model will result in closed form, but lengthy and unwieldy expressions of the probabilities of interest. Instead, in pursuit of an analytically tractable model, we will introduce an approximation that will simplify the problem.

2.2 Approximate Model for a Large Number of Bidders

When the number of bidders (other than the SB) is very large, i.e. $n >> 1$, the model is considerably simplified. Starting from (1) we can write:

$$\begin{aligned}
&a = \frac{n\lambda}{(n-1)\lambda+\beta+\delta} \approx \frac{n\lambda}{n\lambda+\beta+\delta}\,, \quad b = \frac{\beta}{n\lambda+\delta}\,, \quad \text{and} \\
&P(l) = P(O(l)) + P(R(l)),
\end{aligned}$$

so that, the stationary probabilities related to the other bidders become

$$\begin{aligned}
&P(O(1)) = aP(0), \\
&P(O(l)) = aP(l-1), \quad 2 \le l \le v-1, \\
&P(O(v))\delta = n\lambda P(v-1), \\
&P(A(O,l))r = \delta P(O(l)), \quad 1 \le l \le v,
\end{aligned} \tag{2}$$

and the stationary probabilities related to the SB become

$$
\begin{aligned}
&P(R(1)) = bP(0), \\
&P(R(l)) = bP(O(l-1)), \quad 2 \le l \le v-1, \\
&P(R(v))\delta = \beta P(O(v-1)), \\
&P(A(R,l))r = \delta P(R(l)), \quad 1 \le l \le v.
\end{aligned} \tag{3}
$$

Writing down the probability for $P(0)$, and from the law of total probability, we have

$$
P(0)(n\lambda + \beta) = r \sum_{U=O,R} \sum_{l=1}^{v} P(A(U,l)), \tag{4}
$$

$$
1 = P(0) + \sum_{U=0,R} \sum_{l=1}^{v} \big[P(U(l)) + P(A(U,l))\big].
$$

To proceed, we first define the sequences

$$
H(l) = \begin{cases} a, & l = 1; \\ a\left[H(l-1) + G(l-1)\right], & 2 \le l \le v-1; \\ \dfrac{n\lambda}{\delta}\left[H(l-1) + G(l-1)\right], & l = v, \end{cases}
$$

and

$$
G(l) = \begin{cases} b, & l = 1; \\ bH(l-1), & 2 \le l \le v-1; \\ \dfrac{\beta}{\delta} H(l-1), & l = v, \end{cases}
$$

so that the stationary probabilities can be written as

$$
\begin{aligned}
P(R(l)) &= G(l)P(0), \\
P(O(l)) &= H(l)P(0), \\
P(A(R,l)) &= \frac{\delta}{r} G(l)P(0), \\
P(A(O,l)) &= \frac{\delta}{r} H(l)P(0).
\end{aligned} \tag{5}
$$

Then, after some algebraic manipulation, we have the probability that the auction has started and is waiting for bids:

$$
P(0) = \frac{\delta r}{(\beta + n\lambda)(\delta + r) + \delta r}\,. \tag{6}
$$

Now, using a simple substitution of $G(l)$ by its valuation defined as a function of $H(l-1)$, we obtain the roots for the recurrent sequence $H(l)$:

$$
R_{1,2} = \frac{1}{2}\left[\, a \pm \sqrt{a^2 + 4ab}\,\right].
$$

As a consequence, we can write $H(l)$ in a closed form solution:

$$H(l) = \frac{R_1^{l+1} - R_2^{l+1}}{R_1 - R_2}, \quad 1 \le l \le v-1, \tag{7}$$

and at the boundary $l = v$, the solution involves a different set of coefficients:

$$H(v) = \left[\frac{R_1^v - R_2^v + b(R_1^{v-1} - R_2^{v-1})}{R_1 - R_2} \right] \frac{n\lambda}{\delta}. \tag{8}$$

Since $G(l)$ is defined as a function of $H(l-1)$, its solutions are easily derived from the above:

$$\begin{aligned} G(1) &= b, \\ G(l) &= \left[\frac{R_1^l - R_2^l}{R_1 - R_2} \right] b, \quad 2 \le l \le v-1, \\ G(v) &= \left[\frac{R_1^v - R_2^v + b(R_1^{v-1} - R_2^{v-1})}{R_1 - R_2} \right] \frac{n\lambda}{\delta}. \end{aligned} \tag{9}$$

We know that by the truth of the inequalities $a, b > 0$, $a^2 + 4ab > 0$, and $R_1 - R_2 = \sqrt{a^2 + 4ab} > 0$, the given solutions exist and take real valuations. In addition, when the maximum valuation v is only known in terms of some general probability distribution $p(v) = \text{Prob}[V = v]$, the solutions are directly computed as expectations with respect to the random variable.

3 Performance Measures of Interest

In this section, we define and obtain the performance measures of interest, both for the SB and the seller. First we will derive the average duration of an auction, τ. The probability $P(0)$ can be interpreted as the ratio of the average time the process spends in state 0 to the average time between two consecutive reentries to that state. The first is simply the average time the auction waits for the first bid after having restarted, i.e. $[\beta + n\lambda]^{-1}$, and the latter is τ. Thus, using (6) we get

$$\begin{aligned} P(0) &= \frac{\text{Average time spent in state 0}}{\tau}, \\ \tau &= \frac{\delta r + (\delta + r)(\beta + n\lambda)}{\delta r(\beta + n\lambda)}. \end{aligned} \tag{10}$$

In any auction, the *probability that the SB wins the item, as opposed to any one of the other bidders*, represented by π, is the conditional probability that the auction ends in a state $A(R, l)$ for some $1 \le l \le v$, given that a sale is made. The latter is the probability of any event from the set $\{A(U,l) : U = O, R \text{ and } 1 \leq l \leq v\}$ occurring. Thus

$$\pi = \frac{\sum_{l=1}^{v} P(A(R,l))}{\sum_{l=1}^{v}[P(A(R,l)) + P(A(O,l))]} = \left[\sum_{l=1}^{v} P(A(R,l))\right] \cdot \left[\frac{\delta r + (n\lambda + \beta)(\delta + r)}{\delta(n\lambda + \beta)}\right]. \tag{11}$$

Suppose we are interested in the *average time that the SB waits until it succeeds in purchasing an item*, which we denote by $\psi(v)$. This is readily derived from the above measures. Note that, due to the auctions being independent and identical processes, the winning probabilities are identical across the auctions. Thus this measure is the average of multiples of τ, probabilistically weighted by the number of auction rounds it takes to win:

$$\psi(v) = \sum_{k=1}^{\infty} \pi(1-\pi)^{k-1} k\tau = \frac{1}{r\sum_{l=1}^{v} P(A(R,l))}. \tag{12}$$

The final measure of interest for the SB is the *average savings it generates with respect to the maximum that it is willing to pay, given that it wins the item*, or

$$\phi(v) = \frac{\sum_{l=1}^{v}(v-l)P(A(R,l))}{\sum_{l=1}^{v} P(A(R,l))}, \tag{13}$$

which we will refer to as the SB's "expected payoff".

On the other hand, the seller's expected income I from an auction is computed as the average of the closing price l weighted by the probability of a sale at that price, regardless of the identity of the winner:

$$I = \frac{\sum_{l=1}^{v} l[P(A(R,l)) + P(A(O,l))]}{\sum_{l=1}^{v}[P(A(R,l)) + P(A(O,l))]}. \tag{14}$$

Over many such auctions, the seller's interest may be to maximise its *income per unit time* $\iota = \frac{I}{\tau}$.

3.1 Approximate Solutions

Using the solutions from (7), (8) and (9), we can readily derive the probability that the SB wins the good:

$$\sum_{l=1}^{v} P(A(R,l)) = P(0)\left[\frac{\delta b}{r} + \frac{\delta b}{r(R_1 - R_2)}\left[\frac{R_1^v - R_1^2}{R_1 - 1} - \frac{R_2^v - R_2^2}{R_2 - 1}\right] + \frac{\beta}{r}\left[\frac{R_1^v - R_2^v}{R_1 - R_2}\right]\right]. \tag{15}$$

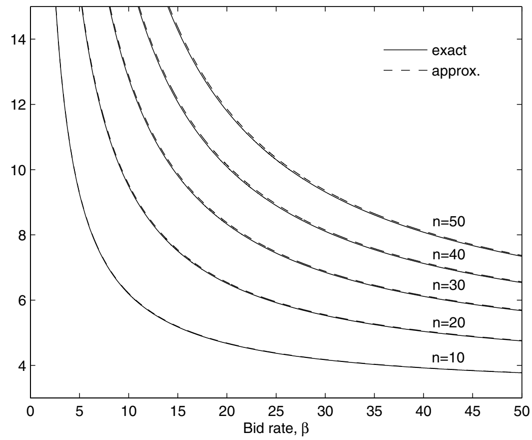


Fig. 1. The SB's expected time to win against bidding rate β, comparing exact solution with approximation result for various number of bidders. Other parameters are $\delta = 0.5$, $r = 1$, $\lambda = 1$, and $V \sim U(80, 100)$.

Similarly, the expected price paid by the SB when it makes the purchase is

$$\sum_{l=1}^{v} lP(A(R,l)) = P(0)\Bigg[\frac{\delta b}{r} + \frac{v\beta}{r}\Big[\frac{R_1^v - R_2^v}{R_1 - R_2}\Big] + \frac{\delta b}{r}\frac{1}{R_1 - R_2}\Big[\frac{(v-1)R_1^{v+1} - vR_1^v - R_1^3 + 2R_1^2}{(R_1-1)^2} - \frac{(v-1)R_2^{v+1} - vR_2^v - R_2^3 + 2R_2^2}{(R_2-1)^2}\Big]\Bigg]. \quad (16)$$

The ability to express these two quantities in a neat and closed form is an important gain and justifies our approximation approach; this is not possible using the exact model (1). We can now express the quantities $\psi(v)$ and $\phi(v)$ in a lengthy but convenient form that can be directly computed using formulas (12) and (13), respectively.

4 Numerical Examples

We will present the model's predictions and assess the accuracy of the approximation approach via some numerical examples. We let all parameters of the system be fixed, where their valuations capture the stochastic nature of the activities, and examine the impact of the SB's bidding rate β on the quantities $\psi(v)$ and $\phi(v)$.

In the following, we compare the results that are obtained by solving the original model (1) numerically and exactly, with that evaluated from the approximate analytical solutions (15) and (16). Figure 1 confirms, as intuition suggests, that

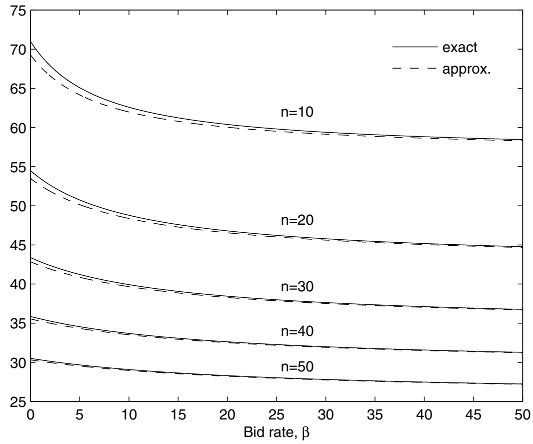


Fig. 2. The SB's expected payoff against bidding rate β, comparing exact solution with approximation result for various number of bidders. Other parameters are $\delta = 0.5$, $r = 1$, $\lambda = 1$, and $V \sim U(80, 100)$.

by bidding at a high rate (the $x - axis$), the SB can expect to win the item in a shorter time (the $y - axis$). Also, the slope $d\psi/d\beta < 0$ for all β, and it increases with β. The increase is more rapid for smaller valuations of n, and accentuates the effect of diminishing returns for the SB.

The quantity $\phi(v)$ is shown in Fig. 2. The intensity of the competition that the SB faces, represented by different valuations of n, determines the levels of payoffs that the SB may expect. Within these levels, however, increasing the SB's rate of bidding adversely affects its expected payoff given that it wins. In other words, when n is fixed, $\phi(v)$ drops with β; this drop is more significant for smaller valuations of n.

Notably, in all the results, the approximations are close to the exact solutions. In all cases, the expected time to win is overestimated, while the expected payoff is underestimated, by the approximate solutions. This can be intuitively explained. In reality, the cumulative bidding rate of the other bidders is conditional upon the event that the highest bidder is the SB, in which case it takes a multiplicative factor of n; or the event that the highest bidder is one of themselves, in which case it takes a multiplicative factor of $n-1$, because the winning bidder refrains from further participation. By approximating, we have "lumped" these events together, taking the cumulative rate as a factor of n. Therefore we have modelled a process more competitive than the actual, and, consequently, the results will be less favourable to the SB: the expected payoff is lower, and the expected time to win is higher than the actual case.

Interestingly, there is an observable trade-off here. Increasing the bid rate will result in the SB buying the item quickly at the expense of some payoff. If the SB is pressed for time and needs to purchase the item quickly, this may be a good option. On the other hand, for the "bargain hunters", buying the item at the best possible price may be a priority, and the time spent, perhaps, is of less importance. Thus, clearly, the SB will have to balance its willingness to forgo some payoff against the prospect of winning the item sooner.

5 Conclusions

In this paper, we propose a discrete state-space continuous time probability model to study auctions that proceed with unit increments, where the state-space represents the attained price. We distinguish the SB from the others, and isolate the states corresponding to the desirable outcomes for the SB, so that we can compute the measures of interest for the SB. These measures are the probability that the SB wins rather than one of the other bidders, the average time the SB waits to be able to purchase an item, and its expected savings with respect to the maximum payable. For tractability, we introduce an approximation that allows us to conveniently express these measures in closed form, with minimal loss in the accuracy.

The model is useful when the SB is aware of the availability of similar or substitutable goods for sale in the future. For instance, in using auctions for allocating reusable goods for a period of consumption, after which it is returned to the seller for reallocation. The good could be some resource such as network bandwidth. In such applications, the model can provide the aforementioned measures of interest for the SB, and quantitatively relate these to the speed at which the SB raises the price.

Some interesting extensions of this model will be pursued in future work. In particular, we will characterise the time constraints faced by the bidders, and examine optimal decisions under this formulation. Also, the possibility and the benefits of bidders moving between different auction sites can be considered.

References

1. Gelenbe, E.: Analysis of automated auctions. In: Levi, A., Savaş, E., Yenigün, H., Balcısoy, S., Saygın, Y. (eds.) ISCIS 2006. LNCS, vol. 4263, pp. 1–12. Springer, Heidelberg (2006)
2. Gelenbe, E.: Analysis of single and networked auctions. ACM Trans. Internet Technology 9(2) (March 2009)
3. Zeithammer, R.: Forward-looking bidding in online auctions. J. Marketing Res. 43(3), 462–476 (2006)
4. Gagliano, R.A., Fraser, M.D., Schaefer, M.E.: Auction allocation of computing resources. Commun. ACM 38(6), 88–102 (1995)
5. eBay, http://ebay.co.uk
6. Gelenbe, E., Mitrani, I.: Analysis and Synthesis of Computer Systems. Academic Press, New York (1980)
7. Guo, X.: An optimal strategy for sellers in an online auction. ACM Trans. Internet Technology 2(1), 1–13 (2002)
8. Rothkopf, M.H., Harstad, R.M.: Modeling competitive bidding: A critical essay. Manage. Sci. 40(3), 364–384 (1994)

Agent-Based Modeling of a Mobile Robot to Detect and Follow Humans

José M. Gascueña and Antonio Fernández-Caballero

Universidad de Castilla-La Mancha, Escuela de Ingenieros Industriales de Albacete &
Instituto de Investigación en Informática de Albacete, 02071-Albacete, Spain
{jmanuel,caballer}@dsi.uclm.es

Abstract. This paper introduces a multi-agent system (MAS) approach using the detailed process provided by Prometheus methodology for the design of a moving robot application for the detection and following of humans. Our conjecture is that complex autonomous robotic systems have to be fully modeled in their initial design stages by means of agent-based technology. The application has been completely modeled with the Prometheus Design Tool (PDT), which offers full support to Prometheus methodology.

Keywords: Multi-agent systems, Agent-based software engineering, Mobile robots, Surveillance.

1 Introduction

At present, there are many applications that benefit from the use of mobile robots that incorporate the capability of following persons. Some examples are carrying objects that people working in hospitals, airports, museums, or domestic environments need; or detecting and following intruders. In literature, several studies that use information picked up by devices mounted on a robot to track humans can be found. It is usual to use a camera to detect faces or color blobs, or to follow a contour [13]. Other researchers implement the following task by using information provided by a laser [7]. A hybrid approach is considered in [10], [16], where visual information provided by a camera and information gotten with a laser are used jointly. In addition to the vision sensor, a voice recognition sensor is mounted on the mobile robot in [8] to follow humans in an outdoor environment. Another option involves placing a tracking device on humans [2]. For example two LEDs that are detected by a camera mounted on a robot; or an ultrasonic transponder that allows a robot ultrasonic sensor to distinguish between persons and obstacles.

The works cited previously do not use a methodology that allows requirements capture and design before carrying out the application implementation. Our proposal is to introduce Software Engineering techniques, as these produce a very well documented application from requirements up to implementation [14], [6]. Moreover, using a methodology allows sharing the same terminology, annotation, models, and development processes [1].

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 80–89, 2009.

Like humans, robots need a certain level of autonomy, reactivity, pro-activity and social ability to perform their tasks. These characteristics are often cited as a rationale for adopting agent technology [17]; so an agent-oriented methodology will be useful for modeling these kinds of systems. In the last few years a great number of agent-oriented methodologies have been proposed, but only some of them have been applied to develop robotic applications. As far as we know, the only agent-oriented methodologies used to analyze and design a robotic system are Cassiopeia [3], MaSE [5], PASSI [4], and the methodology proposed in [9] that uses concepts from GAIA, Mas-CommonKADS and MaSe methodologies. INGENIAS has been tested in an advanced surveillance system composed of different types of sensors [12].

This paper presents how the detailed process provided by Prometheus methodology has been used to design a robotics application, namely the detection and following of a person, using a multi-agent system (MAS) approach. We have chosen this methodology because it provides a collection of guidelines helping to determine the elements (for instance, agents and interactions) that form the MAS. These guidelines are also helpful to the experts in MAS development. They will be able to transmit their experience to other users through explaining why and how they have obtained the different elements of the agent-based application. In addition, Prometheus is also useful because it explicitly considers agent perceptions and actions as modeling elements. In robotics, percepts are environment data collected by several robot sensors (temperature, light, distance, etc) and actions represent the control carried out by the robot actuators (motors, LEDs, and so on). Lastly, the use of plans also seems a good fit for developing robotic systems.

2 Overview of the Prometheus Methodology

Prometheus [11] is defined as proper detailed process to specify, implement and test/debug agent-oriented software systems. It offers a set of detailed guidelines that includes examples and heuristics, which provide a better understanding of what is required in each step of the development. This process incorporates three phases. The system specification phase identifies the basic goals and functionalities of the system, develops the use case scenarios that illustrate its functioning, and specifies which are the inputs (percepts) and outputs (actions). It obtains the analysis overview diagram, scenarios diagram, goal overview diagram, and system roles diagram. The architectural design phase uses the outputs produced in the previous phase to determine the agent types that exist in the system and how they interact. It obtains the data coupling diagram, agent-role diagram, agent acquaintance diagram, and system overview diagram. The detailed design phase focuses on developing the internal structure of each agent and how each agent will perform its tasks within the global system. It obtains agent overview and capability overview diagrams. Finally, Prometheus details how the entities obtained during the design are transformed into the concepts used in a specific agent-oriented programming language (JACK). The design process for Prometheus methodology is supported by Prometheus Design Tool (PDT) [15].

3 System Specification

The process to detect and follow moving objects using the robot is described next. The robot is moving randomly around the environment while the images collected are shown to the guard (state *wandering*). After some elapsed time (*Timer_P*) the robot stops in order to analyze the images captured in that instant (state *detecting*). After that, if movement has been detected, (1) information about the detected blob is obtained, and, (2) the guard is warned to decide if the robot should follow the blob or not. The process to follow persons is started (state *following*) if he chooses to follow it (*Follow_P*). When the robot is wandering, the guard may perceive that something is moving in the environment, according to the images displayed on his interface. In that case, the guard orders (*Detect_P*) that the images are analyzed to check if there is or not movement. If the image analysis does not detect movement, then the robot goes on moving randomly. In order to achieve tracking an object correctly (state *following*) the images are captured, displayed, and analyzed continuously in order to obtain blob information. The object is followed until the tracking phase finishes. This condition can be satisfied by three different reasons: (1) the guard has decided not to continue to follow the target (*Follow stop_P*), (2) the target is out of the field of vision, or, (3) it is impossible to follow it because some physical inaccessibility is encountered in the environment (for example, the target takes a staircase). After that, the robot wanders again.

Usually, the System Specification phase begins with the analysis overview diagram, which shows the interactions between the system and the environment (see Fig. 1). An actor is an external entity – human or software/hardware – that interacts with the system. At this level, firstly, an actor for each device mounted on the robot (sonar, bumpers, camera, and wheels) has been identified; there is also a *Guard_A* actor to represent a human that interacts with the system, and a *Timer_A* actor which submits time percepts (*Timer_P*) to the system. There are two scenarios (*Motion detection scenario* and *Object following scenario*) that correspond to the main requirements of the system, and another scenario (*Start system scenario*) to represent the robot components initialization process. Secondly, the information that comes into the system from the environment has been identified (percepts). It corresponds to impacts detected by the bumper device (*Collision_P*), images captured by the camera (*Image_P*), distance to obstacles/targets perceived by the sonar (*Distance_P*), and orders issued by the guard to control the change of the system state (*Detect_P*, *Follow_P*, *Follow stop_P*). On the other hand, everything produced on the actors by the system is also identified (actions). It corresponds to the camera movements carried out based on the tilt, pan and zoom parameters provided (*Set camera focus_a*), commands to control wheel motion (*Set direction_a*, *Stop,_a*, *Move_a*), and an action *Show images_a* to show the images captured. *Show results_a* also highlights with a square the image regions where movement has been detected.

Scenarios are specified in more detail in a scenario diagram. A scenario is a sequence of structured steps – labeled as action (A), percept (P), goal (G), or other scenario (S) – that represents a possible execution way of the system. As an example, *Object following scenario* begins with the order given by the guard in order to follow the blob detected (step 1, P). Then, images are captured (step 2, G) and analyzed

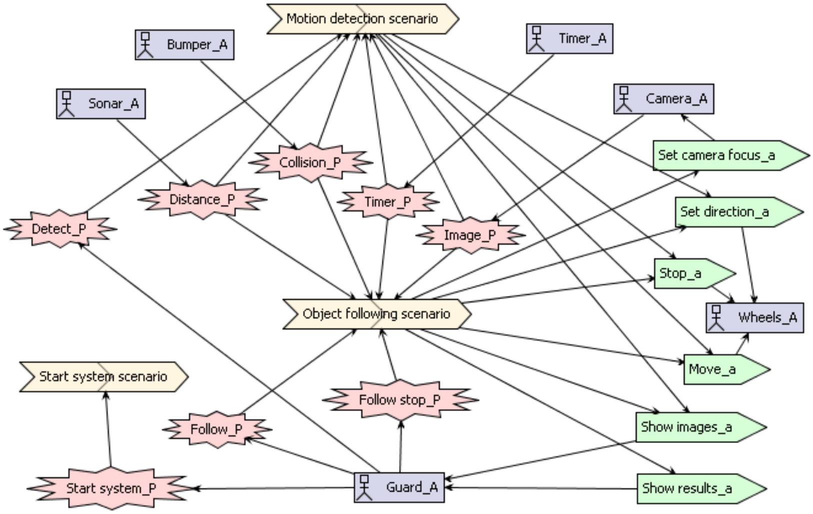


Fig. 1. Analysis Overview Diagram

(step 3, G), information about the blob to be followed is gotten (step 4, G), and analysis results are displayed (step 5, A). Based on this information, the robot is oriented to follow the object (step 6, A) and moves towards it (step 7, A). The scenario returns to step 2 until the guard orders to stop object following (step 8, G). The tracking phase is finished when the goal *Finish following* is achieved (step 9, G).

In our MAS approach, several agents communicate and coordinate to pursue the *Environment surveillance* common goal. A goal is associated for every scenario in order to represent the goal that the scenario is intended to achieve. So, in goal overview diagram there are three goals (*Object following*, *Movement detection*, and *Start system*) related to the scenarios identified (*Object following scenario, Motion detection scenario* and *Start system scenario*, respectively). And they contribute to satisfy the common parent goal *Environment surveillance*. Likewise other goals, they are also decomposed into several sub-goals to denote how to achieve each parent goal. Detection and following processes use information provided by the sonar (to avoid obstacles), the bumper (to control collisions), the guard, the information captured by the camera, and perform commands on the wheels (to move robot). So, there are common sub-goals to accomplish *Movement detection* and *Object following* goals.

The roles are identified by clustering goals and linking perceptions and actions (see Fig. 2). *Start System_R* role handles the guard's request to start the robot devices. *Control Collision_R* role is responsible for achieving *Control Collision* goal, for which it needs inputs detected by the physical bumper device. *Observe environment sonar_R* uses the sonar to perceive distances to obstacles in order to avoid them. *Management guard order_R* aims to meet the guard's orders to control the system operation, which has already been started. These orders correspond to perceptions that allow to start/stop the tracking phase (*Follow_P, Stop follow_P*), and to analyze the images (*Detect_P*). *Wander_R* objective is to control the robot "wandering" process. It consists in randomly moving the robot around the environment, avoiding obstacles

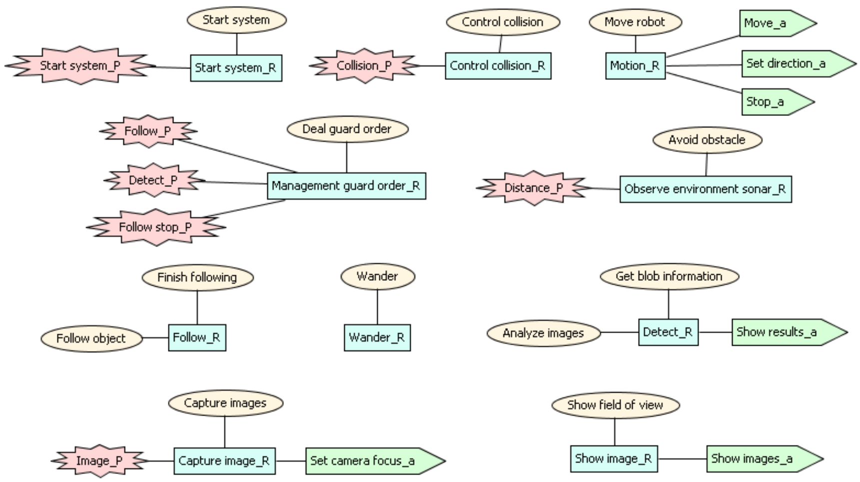


Fig. 2. System Roles Diagram

and controlling situations when a collision has been detected. *Follow_R* is responsible for controlling the robot's movement when the system is following an object. *Follow_R*/*Wander_R* roles do not include perceptions from the environment or actions on the environment, but as it will be described later on, it uses information obtained from physical sensors different from the camera, and therefore they need to "communicate" with the roles responsible for achieving *Follow object*/*Wander* sub-goals (*Avoid obstacle*, *Move robot*, *Control collision*). *Detect_R* is responsible for the goals of analyzing images captured by the camera, getting information from the detected moving blob, and performing an action to display results to the guard. *Capture Image_R* perceives images from the environment (*Image_P* percept), and moves the camera to set the camera focus (*Set_focus_a* action) to capture images in an optimum way (*Capture image* goal). *Show Image_R* is responsible for displaying the camera field of view to the guard. To satisfy this goal, *Show Image_a* action is executed when no movement is detected. *Motion_R* uses wheels to move the robot around the area (*Move robot* goal). This is controlled by actions that allow to stop, move and set the motion direction of the robot (*Stop_a*, *Move_a*; *Set motion direction_a*).

It has been shown in previous descriptions that there are entities, such as goals, which appear in several diagrams. This means that updating some diagram may lead to the need of updating another diagram when taking an iterative approach.

4 Architectural Design

One task carried out in this phase is to decide the agent types (as collections of roles). This is drawn in the agent-role grouping diagram. In our case we have grouped (1) *Start System_R* and *Management guard order_R* roles into *Central* agent, (2) *Wander_R* and *Follow_R* roles into *Motion Manager* agent, and, (3) *Show image_R* and *Detect_R* roles into *Image Manager* agent. Finally, *Control Collision_R*, *Observe environment sonar_R*, *Motion_R*, *Capture Image_R* roles are related with *Bumper*, *Sonar*, *Wheels*, and

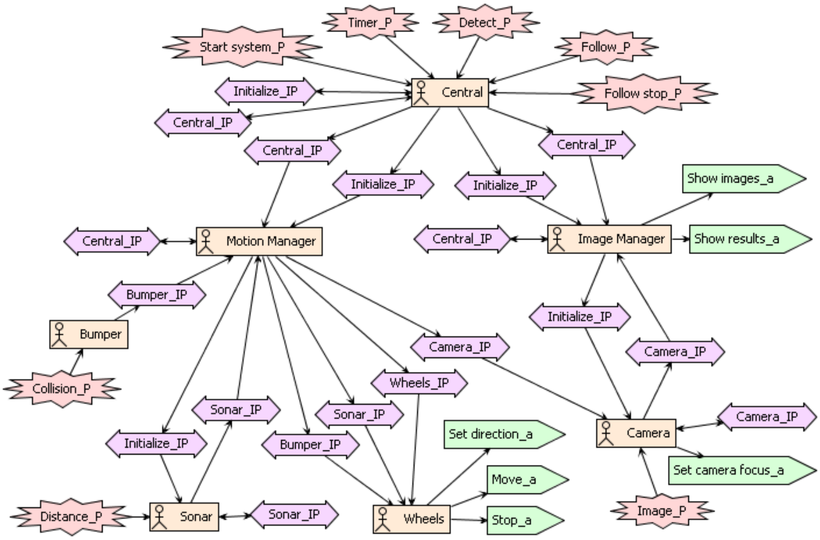


Fig. 3. System Overview Diagram

Camera agents, respectively. An agent is responsible for the functionalities – roles – related. Once roles have been grouped into agents, information about percepts and actions related to roles, depicted in system roles diagram, it is automatically propagated and linked with the agents in the system overview diagram (see Fig 3).

Once the agents have been identified, the next task is to define agent conversations (interaction protocols - IP) in order to describe what should happen to realize the specified goals and scenarios. Fig. 3 shows the system overview diagram for our system design. *Initialize_IP* means that there are communications between *Central*, *Motion Manager*, *Image Manager*, *Sonar* and *Camera* agents when the system is started for activating the sonar and setting the camera initial parameters. *Bumper_IP* specifies interactions between agents (*Bumper*, *Motion Manager* and *Wheels*), and between agents and environment through *Bumper_A* and *Wheels_A* actors, which occurs when the robot collides with something – the robot should stop and establish a new direction, denoted by actions, in order to continue moving. When the *Bumper* agent perceives that there has been a collision, there is a communication with the *Motion Manager* agent through *Collision_M* message. Then, the *Motion Manager* agent sends messages to the *Wheels* agent to execute the actions mentioned. Collisions occur because the sonar has not been able to detect an obstacle on time. *Sonar_IP* includes messages exchanged between *Sonar*, *Motion Manager* and *Wheels* agents as a result of using information provided by the physical device sonar (it measures the distance from an obstacle to the robot). In this protocol, the *Wheels* agent also executes actions to stop the robot and to orient it towards a new direction when the sonar device detects an obstacle. *Wheels_IP* represents the possible messages sent from the *Motion Manager* agent to the *Wheels* agent in order to execute an action on the robot's wheels. *Central_IP* contains messages sent from the *Central* agent to manager agents (*Motion Manager* and *Image Manager*) to monitor the robot's state (*wandering*,

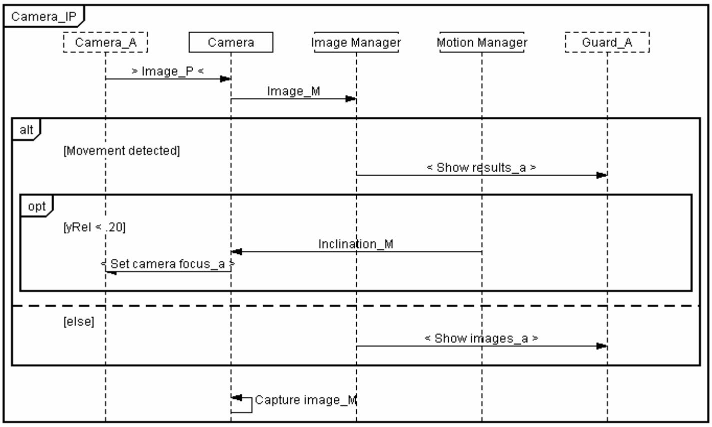


Fig. 4. Camara Protocol Diagram

following, *detecting*) according to the orders provided by the guard (*Detect_P*, *Follow_P*, *Follow stop_P* percepts) or end of a time slice (*Timer_P* percept). *Wander_M* message is sent to the *Motion Manager*, and it includes 'start_wander', 'continue_wander' or 'stop_wander' value to control the wandering state. The same idea is used with *Follow_M* and *Detect_M* messages sent to the *Motion Manager* and the *Image Manager*, respectively.

Finally, Fig. 4 details the *Camera_IP* interaction protocol internal structure, where interactions involve three agents and two actors (identified by the dotted squares in the diagram). As we can notice, the interaction with the environment is carried out by actors (percepts originated by an actor and going to an agent, whereas actions go from an agent to an actor). Firstly, *Camera_A* actor sends *Image_P* percept, which contains the captured frame to *Camera* agent. This agent sends the information perceived to *Image Manager* agent through *Image_M* message in order to determine if there is motion or not (these options are represented by using an alternative box). If the *Image Manager* evaluates that there is no motion, then it shows the image on the graphical guard interface using *Show images_a* action. Otherwise, it shows an image with a frame on the detected moving blob using *Show results_a* action, and an optional box (opt) will be executed if [yRel <.20] is satisfied. yRel is calculated by *Motion Manager*, only when some object is being followed. This optional box means that the *Motion Manager* agent sends an *Inclination_M* message to the *Camera* agent. Next, the *Camera* agent executes *Set focus_a* action using information about new camera focus contained in the message received. *Camera* agent is continuously receiving images captured by the *A_Camera* actor. This is modeled with *Camera* agent sending to itself an idle *Capture image_M* message, so a new image is captured.

The agent acquaintance diagram contains communication links between agents. It is automatically generated from information messages included in the interaction protocols. In short, there is a hierarchical communication between agents. *Central* sends messages to *Motion Manager* and *Image Manager* depending on the robot's state. The *Motion Manager* sends messages to *Camera* and *Wheels* agents in order to move robot mobile components. Moreover, it receives messages from the *Bumper* and

Sonar agents with the information they have collected. *Image Manager* receives messages from *Camera* agent with images perceived in order to detect if there is motion or just to show them.

5 Detailed Design

In this phase, the internal details of each agent are specified in a way that is consistent with its related roles and the interface that has been specified with both the environment (percepts and actions) and other agents (messages). This section only shows the Motion Manager agent internal structure (see Fig. 5) as an example. This agent is responsible for handling the movement of the robot's mobile components (camera and wheels). It pursues Wander and Follow object goals related to the roles associated. In order to satisfy these goals it is necessary to achieve Avoid obstacle, Move robot, Control collision sub-goals, which are pursued by Sonar, Wheels, and Bumper agents, respectively. Thus, the Motion Manager agent has a communication with these agents. Start sonar_p plan is triggered (this is denoted with a dashed arrow) by Control sonar_M message sent by Central agent. It sends Activate sonar_M message to Sonar agent in order to start perceiving distance measures. After that, it sends an Analyze sonar_M message to itself, which triggers the Control sonar_p plan. Once the sonar has been activated, Store sonar percept_p plan updates continuously Buffer sonar_D data with information received within the Distance_M message sent by Sonar agent. Control sonar_p plan sends Stop_M message to Wheels agent in order to stop the robot when the obstacle detected by the sonar device is in the robot advance direction. After that, New direction_M message is sent (this contains new robot's direction and velocity) to Wheels agent according to the reading made on the data represented with a cylindrical shape. Wandering_D and Following_D are Boolean data that contain whether the robot is in state wandering and following, respectively. Blob_D data is used to calculate the new direction that the robot should take when it is following an object. Finally, it sends itself an Analyze sonar_M message in order to continuously execute the process that controls the sonar information. Control collision_p plan is triggered by Collision_M message, which is sent by the Bumper agent when a collision has been perceived. To ensure robot's progress, this plan uses an algorithm similar to the one used in Control sonar_p plan.

Moreover, a capability has been created for each role related to this agent. The wandering process is executed in *Wandering_p* plan included within *Wandering_c* capability. It consists in setting a new random direction in a regular time slice. *Wandering_p* is triggered by *Wander_M* message sent by *Central* agent (the message contains 'start wander' or 'stop wander') or *Motion Manager* to continue the wandering process (the message contains 'continue wander'). The messages and data which appear in Fig. 5 related to *Follow_C* capability are propagated automatically towards the capability overview diagram for *Follow_C* depicted in Fig. 5b. *Follow_p* plan is included within *Follow_c* capability. *Follow_p* plan determines the procedure used by the robot to move through the environment when it is following an object. It can be triggered for three different reasons: (1) *Central* agent sends a *Follow_M* message that contains 'start follow' to begin the following process, (2) *Central* sends *Follow_M* with information 'stop follow', which leads to send *Stop_M* and to finish the

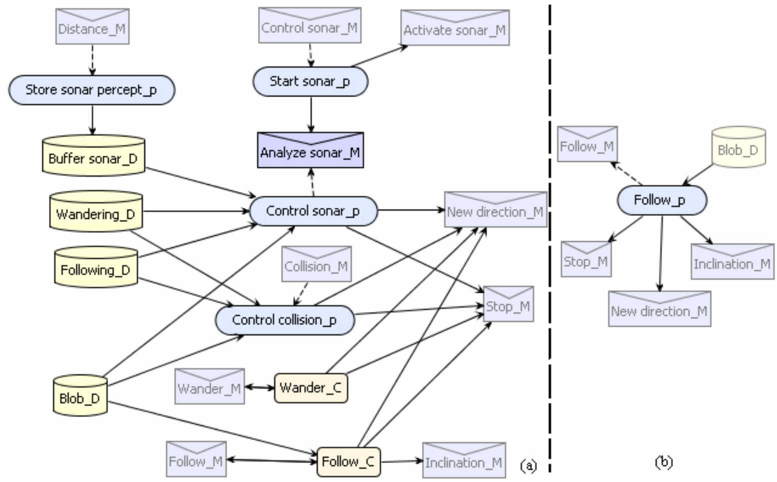


Fig. 5. (a) Agent Overview Diagram for *Motion Manager*, (b) Capability Overview Diagram for *Follow_C*

following process, and (3) *Motion Manager* agent sends itself a *Follow_M* message that contains 'continue follow'. Cases one and two use *Blobs_D* data (a) to determine the robot's direction and velocity, which are sent to the *Wheels* agent through *New direction_M* message, and (b) to calculate the camera focus to continue detecting the blob followed, and to send this information within *Inclination_M* message to *Camera* agent. Each plan descriptor includes a procedure field where it is specified in an informal way what the agent will execute.

6 Conclusions

A MAS approach using the detailed process provided by Prometheus methodology for the design of a robotic application for the detection and following of humans has been introduced in this paper. Traceability between the entities (concepts) identified along the three phases of the Prometheus methodology has allowed progress in the robotic application design. That is to say, the concepts identified in one phase are helpful in order to identify new concepts that appear in other models of the same phase or another later phase. It has been shown that Prometheus methodology can be used to model the behavior of a single robot that incorporates several sensors. PDT allows automatic code generation from the design, and it can be imported by JACK Development Environment. In the near future, we will try to show the suitability to use it into multi-robot systems.

Acknowledgments

This work is supported in part by the Spanish Ministerio de Ciencia e Innovación TIN2007-67586-C02 grant, and the Junta de Comunidades de Castilla-La Mancha PBI06-0099 grant.

References

1. Bordini, R.H., Dastani, M., Winikoff, M.: Current issues in multi-agent systems development. In: O'Hare, G.M.P., Ricci, A., O'Grady, M.J., Dikenelli, O. (eds.) ESAW 2006. LNCS(LNAI), vol. 4457, pp. 38–61. Springer, Heidelberg (2007)
2. Chan, H.K., Ye, W., Lam, T.L., Ou, Y., Xu, Y.: Sensor System for a Human-Following Robot. In: International Conference on Automation, Control and Applications, pp. 350–355. Novosibirsk, Russia (2005)
3. Collinot, A., Drogoul, A., Benhamou, P.: Agent Oriented Design of a Soccer Robot Team. In: 2nd International Conference on Multi-Agent Systems (ICMAS 1996), pp. 41–47 (1996)
4. Cossentino, M., Sabatucci, L., Chella, A.: A possible approach to the development of robotic multi-agent systems. In: IEEE/WIC Conference on Intelligent Agent Technology (IAT 2003), pp. 13–17 (2003)
5. DeLoach, S., Matson, E., Li, Y.: Applying Agent Oriented Software Engineering to Cooperative Robotics. In: 15th International Florida Artificial Intelligence Research Society Conference, pp. 391–396 (2002)
6. Gascueña, J.M., Fernández-Caballero, A.: Towards an integrative methodology for developing multi-agent systems. In: 1st International Conference on Agents and Artificial Intelligence, ICAART 2009 (2009)
7. Gockley, R., Forlizzi, J., Simmons, R.: Natural Person Following Behavior for Social Robots. In: ACM/IEEE International Conference on Human-Robot Interaction (HRI 2007), pp. 17–24 (2007)
8. Inamura, T., Shibata, T., Matsumoto, Y., Inaba, M., Inoue, H.: Finding and following a human based on on-line visual feature determination through discourse. In: International Conference on Intelligent Robots and Systems (IROS), pp. 348–353 (1998)
9. Jiménez Builes, J.A., Vallejo Valencia, M., Ochoa Gómez, J.F.: Methodology for the Analysis and Design of Multi-Agent Robotic Systems: MAD-Smart. Revista Avances en Sistemas e Informática 4(2), 61–70 (2007)
10. Kobilarov, M., Hyams, J., Batavia, P., Sukhatme, G.S.: People tracking and following with mobile robot using an omnidirectional camera and a laser. In: IEEE International Conference on Robotics and Automation, pp. 557–562 (2006)
11. Padgham, L., Winikoff, M.: Developing intelligent agents systems: A practical guide. John Wiley and Sons, Chichester (2004)
12. Pavón, J., Gómez-Sanz, J.J., Fernández-Caballero, A., Valencia-Jiménez, J.J.: Development of intelligent multi-sensor surveillance systems with agents. Robotics and Autonomous Systems 55(12), 892–903 (2007)
13. Schlegel, C., Illmann, J., Jaberg, K., Schuster, M., Wörz, R.: Vision based person tracking with a mobile robot. In: 9th British Machine Vision Conference (BMVC), pp. 418–427 (1998)
14. Sokolova, M.V., Fernández-Caballero, A.: Facilitating MAS complete life cycle through the Protégé-Prometheus approach. In: Nguyen, N.T., Jo, G.S., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2008. LNCS, vol. 4953, pp. 63–72. Springer, Heidelberg (2008)
15. Thangarajah, J., Padgham, L., Winikoff, M.: Prometheus Design Tool. In: 4th International Conference on Autonomous Agents and Multi-Agent Systems, pp. 127–128 (2005)
16. Zivkovic, Z., Kröse, B.: People Detection Using Multiple Sensors on a Mobile Robot. In: Unifying Perspectives in Computational and Robot Vision. Lecture Notes in Electrical Engineering, pp. 25–39 (2008)
17. Wooldridge, M., Jennings, N.R.: Intelligent agents: Theory and practice. The Knowledge Engineering Review 10(2), 115–152 (1995)

Component Based Approach for Composing Adaptive Mobile Agents

Kutila Gunasekera[1], Arkady Zaslavsky[1], Shonali Krishnaswamy[1], and Seng Wai Loke[2]

[1] Faculty of Information Technology, Monash University, Australia
{kutila.gunasekera,arkady.zaslavsky, shonali.krishnaswamy}@infotech.monash.edu.au
[2] Department of Computer Science & Computer Engineering, La Trobe University, Australia
S.Loke@latrobe.edu.au

Abstract. Mobile software agents are increasingly identified as suitable for developing pervasive applications. This paper presents a novel approach to compose adaptive mobile agents. The proposed agents are context-driven, adaptive, component-based and are able to exchange their capabilities with peer agents. The approach requires a flexible, lightweight and standardized component model. We describe the proposed agent system, issues related to selection of a component model, implementation and future research issues.

Keywords: Mobile agents, Compositional adaptation, OSGi.

1 Introduction

Mobile software agents are seen as an attractive approach to develop software for pervasive environments due to features such as their migratory nature, flexibility, scalability and ability to reduce complexity by delegation [1]. However, it is extremely difficult, if not impossible, to design and program an agent system to cater for all possible situations it may encounter in a complex and dynamic environment [2, 3]. Therefore agent systems, despite their adaptive nature, have limitations and need to be replaced or redesigned when faced with unexpected situations. Hence, it is important to be able to model and build agent systems that can learn and adapt dynamically to environmental changes. Much research effort in recent times has been exerted on solving issues related to the modelling and building of such systems [2].

Our current research is on compositional adaptation of mobile software agents. A software application adapts compositionally when its internal structure is changed by acquiring new components [4]. Such algorithmic change allows an application to gain completely new behaviours. Our aim is to create a highly flexible software agent which supports adaptation at runtime and is suitable for carrying out vastly different tasks during its lifetime. We propose a novel approach to runtime compositional adaptation of mobile software agents that is suited for pervasive environments. Our solution leads to software agents that are context-driven, can adapt by acquiring new software components at runtime, and can execute on dynamic heterogeneous environments. In our approach, an agent's useful functionality is provided by reusable

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 90–99, 2009.

software components. A component model that is flexible, lightweight and provides for easy component development is essential for the success of the approach. This paper focuses on the selection and implementation of such a component model.

The rest of this paper is structured as follows: Section 2 describes the background and several related works in compositionally adaptive agents. In section 3, an overview of the proposed solution is presented. Section 4 describes the component model, its requirements and several available options. In section 5 we go through some implementation details and compare two component model implementations before concluding in section 6.

2 Background

Software agents are inherently adaptive compared to traditional software systems. However, an agent's adaptivity is still limited to future changes that can be foreseen at design time. Hence, many agent systems developed to date are static and able to deal with only a limited set of "expected" situations. Therefore, much research effort in recent years has been spent on solving issues related to modelling and building of adaptive agent systems [2]. We identify the following approaches in which agent systems can be made adaptive.

- At multi-agent systems (MAS) level, interactions between agents, multi-agent learning and injecting new agents into a system lead to adaptation.
- At individual agent level, learning (by a single agent) and compositional adaptation of an agent can lead to adaptation.

These methods are not mutually exclusive and are mostly used together. Adaptive MAS [2, 3] and agent learning [5] have been the focus of a considerable body of research over time.

Compositional adaptation of agents occurs when agents change their internal structures by acquiring new software components. This algorithmic change allows agents to gain completely new behaviours. Research on compositional adaptation in software has increased as a result of pressure from fields such as pervasive and autonomic computing [4] and is the focus of our research. Next we briefly describe several previous works on compositionally adaptive agents.

The Generic Adaptive Mobile Agent architecture (GAMA) [6] takes a component-based approach to developing adaptive mobile agents. A GAMA [6] agent is composed of multiple components, and is capable of adapting itself to suit new environments after migration. GAMA [6] agents have a limited scope of adaptation because selection is limited to the set of components available on its new location. The agents are however self-adaptive and their adaptation is triggered by contextual input. The purpose of adaptation is to be able to execute in face of environmental changes. Adaptation occurs only after an agent arrives at a new location, which indicates that agents are not responsive to environmental changes which occur while they are stationary (e.g. variations in CPU load, battery power etc.).

The Dynamic agent infrastructure [7] aims to provide agents with dynamic modifiability of behaviours and is a complete agent architecture built on top of the Java language. Individual Dynamic agents [7] are general purpose carriers of programmes.

An agent is composed of a fixed carrier part that provides housekeeping services and a dynamic part which contains useful capabilities. An agent can dynamically load new programs when it is presented with a task that requires capabilities beyond what it has at present. New components are located either from information available in request messages or by querying a resource-broker. Then a component is downloaded from a URL and added to a local object store.

Dynamically Configurable Software (DynamiCS) [8] provides an architecture which allows dynamic inclusion of negotiating capabilities into mobile agents. It specifically limits itself to adapting agents to support different negotiation schemes.

The open source JADE agent platform [9] provides support for runtime adaptation of agent functionality. An agent's useful functionality is represented as programmer developed *Behaviours* and adaptation support is provided through the `jade.core.behaviours.LoaderBehaviour` class and agent messaging. JADE provides the basics required for agent applications to support runtime adaptation of functionality. However, the API only provides limited control on the adaptation process.

Above systems provide support for runtime compositional adaptation of agents. However, they vary in their support for pervasive computing environments and the scope of adaptation available to agents. Many allow adaptation only in response to tasks assigned or limit adaptation to specific situations. Thus, there is the need for an integrated approach to compositionally adaptive mobile agents that specifically targets pervasive environments.

3 VERSAG System Overview

VERSAG is a novel approach to runtime compositional adaptation of mobile software agents suited for pervasive environments [10]. It leads to software agents that are context-driven, adapt by acquiring new software components at runtime, and execute on dynamic heterogeneous environments. The framework allows agents of different architectures to be embedded within it and an agent's useful functionality is provided by reusable software components.

The salient features of the proposed solution are: the ability of agents to acquire new behaviours from peer agents without depending on designated component providers, and an agent's ability to adapt based on contextual input. Self-adaptiveness is important in pervasive applications to manage the growing complexity of applications and to be able to execute in different environments. Agents are carriers of software components; thereby making agents reusable and easily extensible. Since it does not require any changes to the underlying agent platform it is possible to use VERSAG agents on already deployed agent networks. In VERSAG, these reusable software components are termed as Capabilities.

A VERSAG agent's high-level task is to execute an itinerary assigned to it. An itinerary specifies a list of places the agent has to traverse and actions to execute at each location where an action is the unit of work an agent has to carry out at a particular location. To carry out an action, the agent may need multiple capabilities. The agent decides when and from where the necessary capabilities are acquired. It may

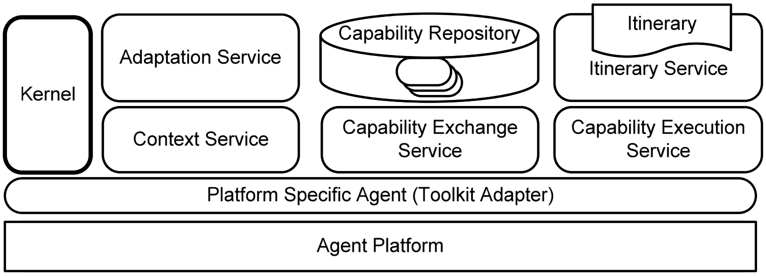


Fig. 1. Structure of a reference VERSAG agent

load necessary capabilities in advance, or load them at a later location based on criteria such as capability availability, number of locations a particular capability is required at, network cost and resource constraints at locations. An agent also has the ability to pass on its capabilities to other agents when requested, making all VERSAG agents potential capability providers.

Figure 1 illustrates the structure of a reference VERSAG agent. The *kernel* controls the agent and passes control to other modules when required. The *capability repository* is where the agent stores its application specific capabilities. The *itinerary service* holds the agent's itinerary and provides methods to interpret itinerary commands. The *capability execution service* provides the means to load, run and stop capabilities that are available in the repository. The *adaptation service* contains the logic which allows an agent to adapt. The adaptation process would involve removing and acquiring capabilities, making changes to running capabilities, selecting suitable capabilities from multiple available ones etc. The *context service* is a related capability which influences the adaptation of an agent. It is expected to consume external context services and also to maintain internal (agent) context information and makes the agent context-aware. The *capability exchange service* fulfils the dual roles of a capability requestor and provider. In its provider role it would listen for capability requests from peer agents and respond as appropriate. The requestor role gives an agent the ability to request capabilities from peers. In most situations this service would run as a low priority process secondary to the agent's main tasks. The *platform specific agent* is the framework's point of contact with the underlying agent platform. Through it, basic services provided by the platform (e.g. mobility, communications) are made available to the upper layers of the agent. Base agent implementations for different agent platforms (e.g. JADE, Voyager) would allow cross-platform migration of agents. Capabilities themselves are agent-platform agnostic and can execute wherever a suitable execution environment is made available. In addition to these, an agent's repository would contain other components which allow it to fulfil application-specific needs including alternate mobility and communication services.

4 Capability Model

A capability represents a central concept in VERSAG and is a software component which can be attached to and detached from a software agent to provide the agent with a particular behaviour.

A capability was formally defined in [10] as a tuple *<id, F, credentials, Env>* where, *id* is a unique identifier, *F* represents the set of functions that the capability contains, *credentials* represent meta-data such as origin, version, security certificates, algorithms and units used and optimizations available in the *capability* and *Env* represents a set of environments that the *capability* can execute in.

Capability exchange among peer agents gives VERSAG several advantages over systems where runtime component loading depends on a designated provider. Designated providers need to have high-performance and be highly available to satisfactorily serve requests from a large number of agents, and failure of a provider or broker could cripple the system. For example, in an environment where a group of agents form an ad-hoc network and are out of reach of a designated component provider, an agent which needs to acquire a component, is unable to do so even if a neighbouring agent has the component because the provider is unavailable. Peer capability exchange becomes highly valuable in such situations. Also, acquiring capabilities from a nearby peer instead of a distant provider reduces network traffic by allowing local exchanges instead of global ones [11].

As previously mentioned, an agent acquires capabilities based on the tasks assigned to it. Thus, the agent should be able to look for capabilities and acquire them from its peers. Also, it should be able to manage the capability life cycle (start, stop, restart or replace with another capability etc.). For this, capabilities have to be described in a manner that allows agents to search for and reason about them (i.e. match tasks to capabilities and compare between alternatives and evaluate the "fitness" of a capability to a task). It should also be possible to ascertain that capabilities acquired are safe. Computationally heavy tasks could have capabilities optimized for different hardware platforms. Capabilities could similarly describe themselves as memory/CPU efficient for use in resource limited environments. In previous research on compositionally adaptive agents, component (i.e. capability) description and matching has not been given prominence and emphasis was not given to defining components in a manner suited for reuse in a wider scope. However, there has been research on component description in other areas of agent research as well as in fields such as web services and service-oriented systems (e.g. WSDL, ontology) which VERSAG agents can make use of.

Thus, we see that capabilities need to be built in accordance to a standard model which allows agents to look for, use, store, discard and exchange them. We first identify several requirements VERSAG has of a capability model. The agent should be able to install, uninstall, start and stop capabilities without adversely affecting the functioning of the agent or other capabilities executing on the agent. Capabilities should also be able to communicate with each other and access services provided by the agent. Since application-specific capabilities would be developed by third parties, a well-defined contract according to which components can be developed is essential. There should also be minimum limitations set on the components being developed. The flexibility to have capabilities of different sizes and features, and being lightweight to suit resource constrained environments is also important. As they would be updated with time, the ability to transparently upgrade a capability would be further enhanced if multiple versions of the same capability could coexist on an agent. This would also be useful to manage conflicting dependency requirements (e.g. Capabilities X and Y respectively need versions 0.9 and 1.1 of capability Z). Being reusable components, if capabilities could be built to adhere to widely used standards, they could be used in a wider environment and components available in the wider environment could be used in VERSAG.

Previous works on compositionally adaptive agents have preferred to develop their own component models and mechanisms. While a custom-developed component model would be the simplest first approach, to meet some of the advanced requirements previously described would consume considerable time and effort. It is also unlikely to meet the requirement of being reusable in the wider environment. Therefore we first examine several existing component models which could be utilized.

JADE [9] allows dynamic loading of functionality. An agent that runs the `LoaderBehaviour` is capable of receiving new *Behaviours* over incoming messages. The required functionality has to be developed as a JADE *Behaviour*. At the recipient agent, the *Behaviour* is added to the existing pool of *Behaviours* and scheduled with the others. This approach results in capabilities being JADE specific and only allows limited control over the loading process. For example, it is not possible for an agent to explicitly unload the *Behaviour* classes when it desires.

µCode [12] is a Java toolkit for code mobility. It does not provide most of the features we require, however it does provide a flexible and lightweight approach to code mobility, a grouping of classes, and allows multiple versions of the same class to co-exist within a single Java virtual machine. Hence, it could be used as the basis for building a capability model. Since µCode does not place additional constraints on the type of classes that can be made mobile, it is possible to apply on top of it a standard for the capabilities. Other features have to be custom developed.

OSGi [13] (Open Services Gateway Initiative), known as the “dynamic module system for Java”, provides a framework where resources (named OSGi bundles) can be installed and executed. An OSGi container provides mechanisms to start, stop, install and uninstall bundles without restarting the container. Bundles need to be developed in accordance with the provided APIs and packaged as Java archives (JAR). OSGi is a popular standard with industry backing and supports platforms ranging from mobile phones and embedded devices to vehicles and high-end computers. It provides life cycle management of components, a well-defined contract for component development, supports co-existence of multiple versions of components and is an industry standard with a large population of available components. While larger than µCode, an OSGi container is still lightweight since it can be deployed on mobile phones and embedded devices. OSGi however does not provide mechanisms for component exchange and description as required by VERSAG.

Above approaches are not able to meet all the requirements specified. Many others are too heavy for small devices and were not described. Thus, a practical approach would be to utilize one of the above as basis and build support for the additional requirements around it. The JADE [9] approach, while simple, would make VERSAG agents JADE specific. µCode [12] provides some desirable features, but is not a standard and is not actively maintained. OSGi [13] is a popular standard with a large number of components already available and provides useful bundle life cycle management features. Thus, it was decided to build the VERSAG capability model around the OSGi standard.

We previously mentioned that a VERSAG agent executes *actions* at various locations according to an itinerary assigned to it. An *action* specifies a list of one or more capabilities, mandatory and optional *credentials* of each capability and an order in which they should be executed. *Credentials* are used to help select suitable capabilities. Described next is an algorithm that illustrates how a VERSAG agent acquires capabilities to execute an *action*. We assume that the agent acquires all needed capabilities just prior to execution and that execution fails if suitable capabilities could not be found. Factors such as contextual changes, and non-functional application requirements, which would contribute to deciding on an optimal algorithm, have not been considered.

Step 1: An agent requires multiple capabilities to carry out its action at a particular location. These capabilities should be selected according to a specification consisting of mandatory and optional credentials.

Step 2: For each capability spec, look in the agent's repository for capabilities which meet the mandatory credentials. A reference to each matching capability is recorded against the spec and if at least one match is found the spec is flagged as found.

```
1   Let CS = {cs_1, ..., cs_n} be the set of capability specifications
2   for each cs_i
            find local capabilities matching mandatory credentials
            assign discovered capabilities to cs_i
            if discovered capability count > 0
                    cs_i.found = true
            end if
    end for
3   for each cs_i where cs_i.found == false
            find remote capabilities matching mandatory credentials
            assign discovered capabilities to cs_i
            if discovered capability count > 0
                    cs_i.found = true
            end if
    end for
4   for each cs_i
            if cs_i.found == false
                    return fail
            end if
    end for
5   for each cs_i
            sort capabilities in decreasing order of match using
    optional credentials
    end for
6   for each cs_i
            while capability not loaded and cs_i has more capabilities
                    load next best matching capability
            end while
            if no capability loaded
                    return fail
            end if
    end for
7   execute contents of CS
```

Fig. 2. Simplified algorithm illustrating how a VERSAG agent searches for capabilities in the local repository and from peer agents to execute an *action*

Step 3: For each un-flagged capability spec (i.e. a matching capability was not found locally), search remotely for capabilities meeting the mandatory credentials. Discovered matches are recorded against the spec and the spec flagged as found.

Step 4: If there are any capability specs for which a match was not found, terminate as failure.

Step 5: For each capability spec, sort the matching capabilities in decreasing order of fitness, according to the optional credentials.

Step 6: For each spec, load the best matching capability. If loading fails, try the next matching capability until a capability is loaded or the list of matching capabilities is exhausted. If a matching capability could not be loaded, terminate as failure.

Step 7: Execute the capabilities in the required order.

5 Implementation

We first describe the current implementation of VERSAG using OSGi [13] as the basis of the capability model. Figure 3 illustrates the structure of an agent. Comparing this structure with the reference structure of figure 1 we note that the base agent is now JADE specific and the kernel is implemented as a JADE *Behaviour* [9]. The repository, itinerary execution service and capability execution service are implemented as key constituents of the agent. However, the context service and adaptation service have been implemented as *capabilities,* with the major consequence that context-awareness and self-adaptability become optional features of the agent. This could be useful in situations such as a LAN environment where adaptivity and context-awareness may not be needed. Also it is possible for an agent to switch to different implementations of these services. For example, an agent may use different context sensing mechanisms in different environments.

The capability repository is a service provided by the agent which needs to be accessed from within capabilities (i.e. OSGi bundles). Hence, the agent exports the repository as an OSGi service. The Executor service API provides a `registerService()` method for this purpose. Capabilities can then request a reference to the repository by specifying the class name it is registered under.

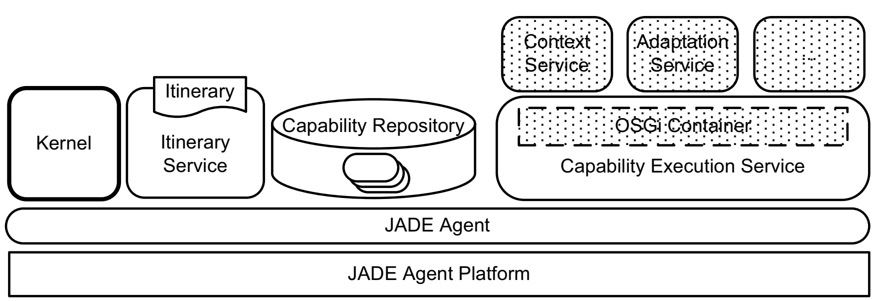


Fig. 3. Structure of JADE/OSGi based VERSAG agent

Table 1. Comparison of custom-developed and OSGi-based capability models

	Custom Capability Model		**OSGi Capability Model**	
	Capability 1	*Capability 2*	*Capability 1*	*Capability 2*
Source statements	46	43	86	83
Size (KB)	1.75	1.73	4.56	4.55
Avg execution time (ms)	34.5	287.5	35.9	286

```
public class ExecutorService {
  ...
  public void registerService(Object serviceRef, String
  serviceClass);
  public STATUS execute(Command cmd, AgentIF agent);
}
```

The capability execution service utilizes an OSGi container to run capabilities. Functionality represented by a capability is implemented as an OSGi bundle. An OSGi container is, however, not developed with mobility in mind and uses file based configurations which cannot be used on a migrating agent. OSGi caching also had to be disabled since agents should not leave behind cached information on remote agencies. Therefore, it was necessary to make configuration changes to the OSGi container to ensure it adheres to these VERSAG specific requirements. Our implementation also pioneers deploying an OSGi container inside a mobile agent.

Table 1 compares two capabilities developed according to a custom-developed capability model and an OSGi-based capability model and tested with VERSAG agents. For both capabilities, we see that in terms of source statements, the OSGi model has more statements. While in the given examples the increase is in excess of 50%, the number of additional statements is fixed and unlikely to increase with the size of the capability. They are due to OSGi requirements and due to the OSGi based model being multi-threaded. The class sizes follow the trend of source statements. The increase in size is unlikely to have an adverse effect since OSGi itself targets mobile devices. Also the results show that OSGi does not result in any extra cost in terms of execution time.

6 Conclusions

In this paper we presented a novel approach to compositionally adaptive mobile software agents. The solution leads to agents that are context-driven and adapt by exchanging software components between agents. The solution is suitable for dynamic heterogeneous environments, eliminates the dependence on global component servers and reduces bandwidth consumption by agents.

The reference structure of an agent was presented and compared with an implemented agent. We also identified the requirements of VERSAG's capability model. Several available options were discussed, from which the OSGi architecture was selected as suitable. A simple comparison with a custom-developed capability model illustrated that there is no capability execution overhead due to an OSGi-based capability model. It also allows reuse of VERSAG capabilities across a wide range of OSGi based environments.

We are currently working on runtime efficiency evaluation of VERSAG agents using an information retrieval example. Our future work includes research into capability description mechanisms for the capability model and supporting context-awareness of mobile agents on heterogeneous environments through the use of different context sensing approaches.

References

1. Zaslavsky, A.: Mobile agents: can they assist with context awareness? In: IEEE International Conference on Mobile Data Management, pp. 304–305. IEEE Press, Los Alamitos (2004)
2. Marín, C.A., Mehandjiev, N.: A Classification Framework of Adaptation in Multi-Agent Systems. In: Klusch, M., Rovatsos, M., Pay, T.R. (eds.) CIA 2006. LNCS, vol. 4149, pp. 198–212. Springer, Heidelberg (2006)
3. Alonso, E., Kudenko, D., Kazakov, D.: Adaptive Agents and Multi-Agent Systems: Adaptation and Multi-Agent Learning. In: Alonso, E., Kudenko, D., Kazakov, D. (eds.) AAMAS 2000 and AAMAS 2002. LNCS, vol. 2636. Springer, Heidelberg (2003)
4. McKinley, P.K., Sadjadi, S.M., Kasten, E.P., Cheng, B.H.C.: Composing Adaptive Software. Computer 37, 56–64 (2004)
5. Weiß, G.: Adaptation and Learning in Multi-Agent Systems: Some Remarks and a Bibliography. In: Weiß, G., Sen, S. (eds.) IJCAI-WS 1995. LNCS(LNAI), vol. 1042, pp. 1–21. Springer, Heidelberg (1996)
6. Amara-Hachmi, N., Fallah-Seghrouchni, A.E.: Towards a Generic Architecture for Self-Adaptive Mobile Agents. In: European Workshop on Adaptive Agents and Multi-Agent Systems, Paris (2005)
7. Chen, Q., Chundi, P., Dayal, U., Hsu, M.: Dynamic Agents. International Journal of Cooperative Information Systems 8, 195–223 (1999)
8. Tu, M.T., Griffel, F., Merz, M., Lamersdorf, W.: A Plug-in Architecture Providing Dynamic Negotiation Capabilities for Mobile Agents. In: Rothermel, K., Hohl, F. (eds.) MA 1998. LNCS, vol. 1477, pp. 222–236. Springer, Heidelberg (1998)
9. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing Multi-agent Systems with JADE. John Wiley and Sons, Chichester (2007)
10. Gunasekera, K., Zaslavsky, A., Krishnaswamy, S., Loke, S.W.: VERSAG: Context-Aware Adaptive Mobile Agents for the Semantic Web. In: 32nd Annual IEEE International Computer Software and Applications Conference, pp. 521–522. IEEE, Los Alamitos (2008)
11. Gunasekera, K., Zaslavsky, A., Krishnaswamy, S., Loke, S.W.: Context Driven Compositional Adaptation of Mobile Agents. In: International Workshop on Data Management in Context-Aware Computing (DMCAC 2008), Beijing (2008)
12. Picco, G.P.: μCODE: A lightweight and flexible mobile code toolkit. In: Rothermel, K., Hohl, F. (eds.) MA 1998. LNCS, vol. 1477, p. 160. Springer, Heidelberg (1998)
13. About the OSGi Service Platform: Technical Whitepaper. OSGi Alliance, 19 (2007)

A Ratio-Based Update Scheme for Mobile Agent Location Management*

Ting Yuan Yeh and Tzong I. Wang

Laboratory of Intelligent Network Application-LINA
Department of Engineering Science,
National Cheng Kung University
Tainan, Taiwan, 701, R.O.C.
wti535@mail.ncku.edu.tw,
aska@linuxeagle.es.ncku.edu.tw

Abstract. A location management scheme is critical for a mobile agent system to trace mobile agents and support on-the-go communications between peer agents. For the time being, most mobile agent location management schemes consist of both a tracking phase and a message delivery phase, and dutifully create or incur plenty of update messages and interaction messages in order to guarantee the effectiveness of the schemes. Excessive transmission of messages not only downgrades the overall performance of mobile agent location management scheme, but also causes network overhead. This paper proposes a ratio-based update scheme that can minimize the overall transmission costs of produced messages by intelligently adjusting the ratio between update messages and forwarding pointers. Experiments are conducted to evaluate the proposed scheme, which reveal that this scheme is general, adaptable, and possessed of higher granularity in the network latency. Moreover, the results also indicate that this scheme is better than previous schemes in optimizing the overall performance.

Keywords: mobile agent, location management, tracking phase, message delivery phase, network latency, update messages, forwarding pointers.

1 Introduction

Mobility and autonomy make mobile agent especially advantageous in bandwidth savings, limited latency, higher degree of robustness, and mobile agent therefore becomes a promising technology in distributed system [1]. Mobility, however, makes messages difficult to transmit to a mobile agent accurately. Many significant services in a mobile agent system, nevertheless, have to be accomplished through interacting the messages exchanged between mobile agents, such as mobile agent communication [2], fault tolerance, and mobile agent control [3][4]. For transmitting interaction messages to a mobile agent accurately, a location management scheme is forced to operate in two phases; one is the tracking phase, which keeps track of location information of mobile

* This research was supported by National Science Council, Taiwan, under the contract NSC-95-2221-E-006-158-MY3.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 100–109, 2009.

agents continuously, and the other is the message delivery phase, in which the interaction messages are forwarded to a mobile agent along the location information.

In the home-based architecture, the home host serves as a Location Management System (LMS) for mobile agents. Although it suffers from the bottleneck problems as a centralized server, the feature of simplicity remains it extremely popular in the current researches. Update message and forwarding pointer are two most widely adopted methods for tracking phase. An update message is sent by a mobile agent to refresh its location information that is maintained in a LMS. Forwarding pointer is a location reference left by a mobile agent at current location to indicate its next destination before it starts moving. Each host in a mobile agent system maintains a location table for storing location information of mobile agents created in the host, and forwarding pointers of other mobile agents which visited this host. The maintained location information must be able to connect consecutively for providing interactive messages a channel leading to the next immediate mobile agent. A channel starts at a responsible LMS for a mobile agent, may include several intermediate hosts, and ends at the current location of the mobile agent. Each mobile agent creates plenty of messages (update messages and interactive messages) as executing assigned tasks on a network. Transmitting these messages not only affects the overall performance of the location management scheme, but also increases the load of the network.

Figure 1 depicts the tracking model of a LMS. Current studies make use of update messages and forwarding pointers alternately in the tracking phase for minimizing the overall transmission cost of messages. In other words, a mobile agent can decide to leave location information by using either of the two tracking methods when it arrives at a new host. In a home based LMS architecture, update messages of mobile agents are sent back to home host. As shown in figure 1, there are zero or more forwarding pointers between two successive updates. The more frequent uses of update messages there are, the fewer forwarding pointers yields, and vice versa.

When a mobile agent updates its location information at H_{i+k}, and its channel becomes direct from LMS to the current host H_{i+k}. The length of a channel will get longer and longer with the accumulating of forwarding pointers, due to mobile agent's migration, until the next location update is commanded.

The interaction between entities, such as users, applications and mobile agents, is referred to as message delivery phase, as illustrated in Figure 2. An entity, attempting to interact with a mobile agent, has to send an interaction (enquiry) message to the LMS first. As a LMS receives a message, it forwards the message through the correct

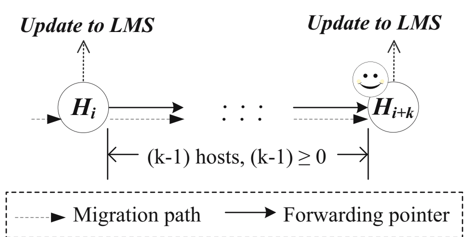


Fig. 1. The Tracking Model

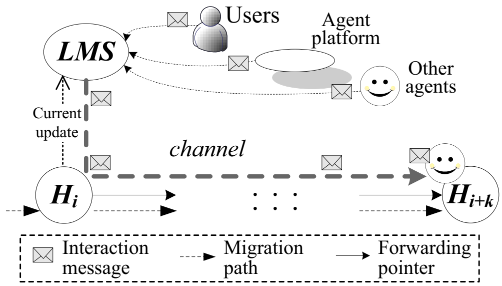


Fig. 2. The Message Delivery Model

channel stored in the location table to the targeted mobile agent. Hosts on a channel are supposed to repeat the forwarding process until an interaction message arrives at the current location of mobile agent.

Increasing the frequency of updates of a mobile agent decreases the transmission cost of interaction messages, while the overall transmission cost could be increased due to the growth of the number of update messages. Obviously, there exists a contradiction between the transmission cost of update messages and interaction messages. This study discover the updating location information of a mobile agent, as the ratio between the costs of transmitting update message and interactive message reaches a threshold, can optimize the overall transmission cost of produced messages more effectively. Therefore, this paper proposes a ratio based scheme which is established on this update criterion and enacts a number of simulation experiments to evaluate and compare the scheme with other well-known schemes, and the results exhibit that the overall transmission cost of messages in our scheme is lower than those previous, and that this scheme is more general and adaptive to various network latencies.

The rest of this paper is organized as follows: Section 2 introduces related works and analyzes the key issue of designing a mobile agent location management scheme. Section 3 presents the proposed intelligent update algorithm. Evaluation results are indicated in Section 4. Conclusions are given in Section 5.

2 Related Works

Both the earlier mobile agent location management schemes, the Home-proxy scheme (HP) and the Forwarding-proxy scheme (FP), are characterized to facilitated easy implementation, and, thus, adopt home-based LMS architecture. The HP scheme uses only update messages to track the location of a mobile agent; once the mobile agent changes its location, it sends an update message to refresh the location information maintained in the LMS. In this way, a channel always consists of the LMS and a mobile agent's current location only, which, thus, keeps the transmission cost of interaction messages extremely low at any moment. But the overall performance is not as good as expected, due to the excessive usage of update messages. On the other hand, the FP scheme, differing from the HP scheme in utilizing only forwarding pointers in tracking phase, which produces no messages, instead causes no additional performance overhead in tracking phase. However, in the FP scheme the length of a channel gets longer and longer with the increasing number of a mobile agent's movements and interaction messages must route through a long distance to reach a mobile agent, which consequently damages the overall performance seriously when mobile agents interact frequently. This is referred to as the lengthy channel problem in mobile agent location management. Mobile agent platforms such as MOA [5], Grasshopper [6], and Aglet [7] use HP scheme, while platforms such as Mole [8] and Voyager [9] adopt FP scheme.

For balancing the costs between the transmission of update messages and interaction messages to optimize the overall performance, a movement based scheme, called the dU scheme, has been proposed in [10], and a time-based scheme has been described in [11]. In the dU scheme, a mobile agent updates its location information after *d* migrations between hosts, while in the time-based scheme, after an interval of time τ has elapsed, a mobile agent updates its location information to its LMS. Both of the two schemes use different criteria to decide the adequate update timing, and their evaluation also proved the overall performance can be largely improved by this way. Both the dU scheme and the time-based scheme have been proved, at the right timing, to improve the performance of mobile agent location management schemes. Because the higher the transmission cost of update messages is, the lower the transmission cost of interaction messages will be, and vice versa, therefore, there should exist a specific ratio between the transmission costs of update messages and interaction messages as the best update timing mechanism.

3 The Ratio Based Update Scheme

Hop counts between each pair of hosts on a network can be regarded as network latency metric according to the requirement of an application. Different network latency metric represents different requirements of costs [13]; for example, hop counts reflect network resources consumed and RTTs indicates response times. This paper assumes that each host in a mobile agent system can detect the transmission costs between itself and other hosts in the system periodically by using network tools, such as *traceroute* or *ping*.

3.1 Ratio Based Update Scheme in Tracking Phase

A mobile agent updates its location information as the ratio between its current update cost and message delivery cost is below a threshold r. Threshold r is defined as follows as the message delivery cost is proportional to the length of a channel.

$$r = \frac{update\ cost}{length\ of\ channel}$$

where *length of channel* is regarded to as the delivery cost of interaction messages, which is proportional to the path length. Figure 3 illustrates how the relation of r is established. In this figure, mobile agent MA migrates from H_{i+k} to H_{i+k+1}. As MA is at H_{i+k}, the transmission cost of interaction messages is represented by the length of the current channel. After MA arrives at H_{i+k+1}, the transmission cost of interaction

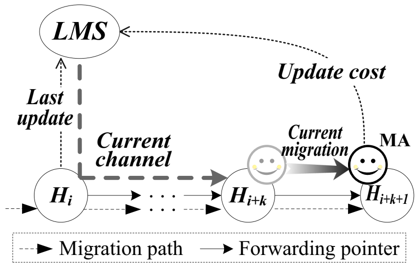


Fig. 3. The Relation of r

messages become the length of the current channel plus the length of the current migrations just made. Thus, the computation of r is to divide the update cost on current host (H_{i+k+1}) by the length of current channel plus current migration of MA.

As described, before the next update of location information, a channel gets longer and longer with the increasing number of mobile agent's migrations. In the definition, the value of r inverses to the length of a channel and, therefore, the value of r gets smaller and smaller as the mobile agent keeps migrating on the network.

For continuously calculating the value of r, a mobile agent has to carry the length of its current channel and update it during its life span. When a mobile agent arrives at a new host, it asks the host for the update cost and the cost of current migration for computing the value of r. If the value of r is smaller than a default threshold value, the mobile agent updates its location information. In addition, the mobile agent leaves a forwarding pointer which indicates to the next destination before it departs from the current host. In the update case, the channel of mobile agent would be renewed as LMS points to the current host, while in the forwarding pointer case, the current channel would be updated to a new channel which is obviously the current channel plus the current migration. The optimal threshold value of r will be obtained by simulation experiments described in later sections.

4 Simulation Experiments and Evaluations

The ratio-based scheme proposed is evaluated by both computing the transmission cost of total messages produced and conducting simulation experiments. To create a large-scale experiment platform with thousands of nodes on the Internet would be highly difficult and virtually impossible. The network topology of the simulation environment used in this paper is established by using the nem [12] that can create a network map resembling a real network. In the simulated network, distance between a pair of hosts on the network map is measured by using the hop counts and over 10 000 routers is created with the longest distance between hosts controlled in 22 hops. Experiments are conducted for numerous rounds for a mobile agent, and the mobile agent's each itinerary contains 100 randomly selected hosts. For the best assessment, each experiment also compares the performance of the ratio-based scheme with that of the two famous schemes, the HP and the dU.

4.1 Formulation and Computation of Network Overhead

Total cost of the update messages is denoted as C_T and formulated as follows:

$$C_T = N_U \times C_U \quad (1)$$

where N_U represents the total number of update messages and C_U represents the average cost of each update message, i.e. the total cost is the sum of the transmission cost of each update message.

The total cost C_M, of interaction messages is denoted as follows:

$$C_M = N_I \times L_C \quad (2)$$

where N_I represents the total number of interaction messages and L_C represents the average length of a channel.

The total transmission cost of produced messages, denoted as C_{total}, is then computed by summing C_T and C_M, and is formulated as follows:

$$C_{total} = C_T + C_M \quad (3)$$

Computing the total transmission cost of all the messages created for managing location information and communications of mobile agents is a main criterion in evaluating a location management scheme. The computation process is suitable not only for the ratio-based scheme, but also for other home based LMS architecture schemes. In the following section, the performance of the ratio-based scheme is evaluated according to the computation process.

4.2 Optimal Threshold

A simulation experiment is conducted to find out the best value of r for the ratio-based scheme. The interaction rate is defined as the average number of messages received per host during the course of an itinerary. For an itinerary with 100 hosts, if a mobile agent receives 10 messages throughout its itinerary, the interaction rate is 10% in average, while if receives 100 messages, the average interaction rate is 100%, over

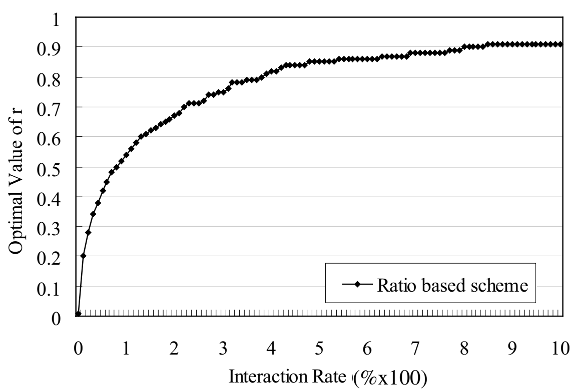


Fig. 4. The Best Value of r

100% when receiving more than one message. Figure 4 shows the experiment result of the best value of r under various interaction rates.

The result of this experiment shows that the best value of r varies under different interaction rates. As can be seen in Fig. 4, the value of r gets bigger and approaches 0.92 with the increase of interaction rate from 0% to 1000%. The result forms a basis of selecting a best value of r with different interaction rate. Moreover, when the value of r becomes very small, e.g. almost zero, the ratio-based scheme functions just the same as the FP scheme, which has no update cost, does. On the contrary, as the value of r approaches the other extreme value 0.92, the ratio-based scheme performs like the HP scheme.

4.3 Number of Updates

As aforementioned, the way for optimizing the overall performance evaluation is to optimally balance the trade-off between the costs of transmitting update messages and interaction messages. Thus, experiments are conducted to compare the performance of HP scheme, dU scheme, and ratio-based scheme in the aspects of transmission cost of update messages, interaction messages, and all produced messages respectively.

Fig. 5 displays the average number of updates of HP scheme, dU scheme, and ratio-based scheme with interaction rate stretching from 0% to 1000% using the same network topology and latency metric; the comparison of average number of update messages that a mobile agent produced is sufficient to reveal the average costs of updates in the schemes. As shown in figure 5, when the interaction rate is between 80% and 100% the dU scheme performs better than the ratio-based scheme, but not the rest interaction rates, which concludes that the ratio-based scheme can control and even reduce the cost of updates more effectively.

4.4 Length of Channel

When the interaction rate is high, the length of a channel has more impact than the number of updates on the overall performance. Interaction messages can seriously damage the performance of a location management scheme due to the lengthy channel

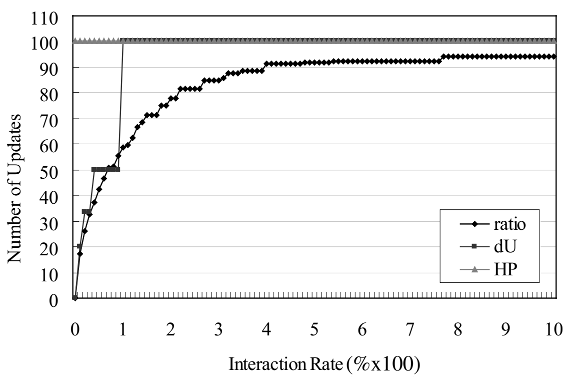


Fig. 5. Average Number of Updates

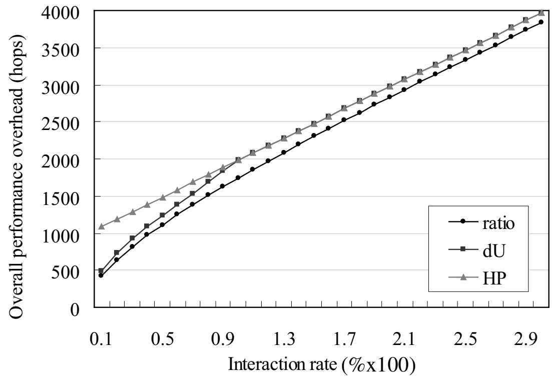


Fig. 6. Comparison of Overall Performance Overhead

problem, not to mention the enormous number of interaction messages if mobile agents interact frequently.

4.5 Total Transmission Cost

The experiment result of computing the total transmission cost of produced messages (C_{total}) for HP scheme, dU scheme, and ratio based scheme is shown in figure 6, which displays the average total costs with the increase of interaction rate.

As can be seen in figure 6, the HP scheme is worse than both the dU and the ratio-based schemes as interaction rate is low. The reason is that the update mechanism of HP scheme is monotonous and creates exceeding update messages in addition to the interaction messages. On the other hand, the dU and the ratio-based schemes update location information more flexible than the HP scheme and, therefore, they perform well in all interaction rates. The ratio-based scheme always performs better than the dU scheme irrelevant to interaction rates and this is because it controls the update timing more accurate than the dU scheme. This is the reason behind the superiority of ratio-based scheme to the other two schemes in figure 5.

4.6 Error Scope of r

In accordance with the variation of interaction rate to assign the best value of r is no easy, because the creation of interaction messages is difficult to predict. Instead of proposing a solution for this issue, this paper proposed the usage of an *Error Scope* of r within which the performance of the ratio-based scheme is guaranteed to be better than those of the HP and the dU schemes for a same interaction rate. An experiment is conducted to examine such an *error scope* of r corresponding to different interaction rates and Figure 7 shows the result of the experiment.

From the experiments, three curves are drawn in figure 8, where the upper curve represents the upper bound, the lower curve represents the lower bound, and the curve in between represents the best value of r. The curve of the best value of r is the same as the one shown in figure 4 except where the vertical scale set to 1 only. At any

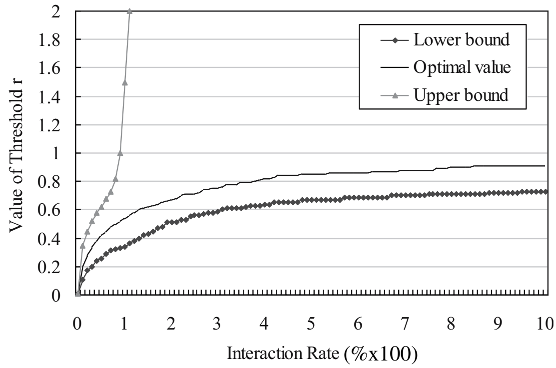


Fig. 7. The Error Scope of r

interaction rate, select a value of r between the upper curve and the lower curve would perform better than the other two schemes, e.g. at the interaction rate 100%, the ratio-based scheme could be better than HP scheme and dU scheme in the aspect of performance by setting the value of r between upper bound 1.5 and lower bound 0.34. For this example, the corresponding overall performances are 1958, 1952, and 1744 hops respectively when r is at upper bound, at lower bound, and at the best value 0.54, while the overall performance of the dU and the HP schemes are both 1981 hops, because dU is the same as HP when interaction rate is greater than 100%.

One point to note in figure 7 is that the line of upper bound grows rapidly. The reason for the sharp growth can be deducted as follows. When the interaction rate is fixed at 100%, the overall performance is almost invariant as r goes beyond the best value 0.54, and consequently r has no upper bound. Because the error scope of r extends to a wide range so that the ratio-based scheme can still achieve satisfactory performance without having to modify the value of r constantly with the variant of interaction rates. This proves that the proposed ratio-based scheme is more flexible than other peer schemes.

5 Conclusion

To optimize the overall performance of a home-based location management scheme, location information of mobile agents must be updated at the right time. As this study discovered that there exists a specific ratio between the location update cost and the message delivery cost, a ratio-based update scheme is proposed in this paper for optimizing the overall performance of mobile agent location management schemes. Simulation results confirm that the overall performance of this ratio-based scheme is better than the other two well-known schemes, the dU and the HP schemes.

Location management of a mobile agent is rather essential for a mobile agent system, in which many significant services are accomplished among interactions, for example, task communications and coordination, fault tolerance, control of mobile agents, and etc. The proposed ratio-based scheme, with its promising performance, can be embedded in mobile agent systems for these topics without any doubts.

References

1. Harrison, C.G., Chess, D.M., Kershenbaum, A.: Mobile Agents: Are they a good idea? IBM Research Report. T. J. Watson Research Center, NY (1995)
2. Cao, J., Feng, X., Lu, J., Sajal, K., Das, S.: Mailbox-based scheme for mobile agent communications. IEEE Computer 35(9), 54–60 (2002)
3. Baumann, J.: Control Algorithms for Mobile Agents. PhD thesis, University of Stuttgart, Germany (1999)
4. Wang, H., Zeng, G., Lin, S.: An efficient control mechanism of mobile agents. In: Proceedings of the International Conference on Computer Networks and Mobile Computing, October 2001, pp. 371–379 (2001)
5. Milojicic, D.S., LaForge, W., Chauhan, D.: Mobile Objects and Agents (MOA). Distributed System Eng. 5(4) (December 1998)
6. Bäumer, C., Magedanz, T.: Grasshopper: A Mobile Agent Platform for Active Telecommunication. In: Albayrak, Ş. (ed.) IATA 1999. LNCS, vol. 1699, pp. 19–32. Springer, Heidelberg (1999)
7. Lange, D., Oshinma, M.: Programming and Deploying Java Mobile Agents with Aglets. Addison Wesley, Reading (1998)
8. Baumann, J., Hohl, F., Strasser, M.: Mole-A Java Based Mobile Agent System. In: Proc. ECOOP 1996 Workshop Mobile Agent Object Systems (1996)
9. Glass, G.: ObjectSpace Voyager Core Package Technical Overview. Mobility: process, computers and agents. Addison-Wesley, Reading (1999)
10. Li, T.Y., Lam, K.Y.: An optimal location update and searching algorithm for tracking mobile agent. In: Proceedings of the 1st international joint conference on Autonomous agents and multiagent systems, pp. 639–646 (2002)
11. Songsiri, S.: A Naming Service Architecture and Optimal Periodical Update Scheme for a Multi Mobile Agent System. In: Proceedings of CIMCA-IAWTIC 2005, vol. 1, pp. 778–785 (2005)
12. Magoni, D.: nem: A software for network topology analysis and modeling. In: Proc. of the MASCOTS 2002, pp. 364–371. IEEE Computer Society, Los Alamitos (2002)
13. Obraczka, K., Silva, F.: Network Latency Metrics for Server Proximity. In: IEEE GLOBECOM, pp. 421–427 (2000)

jDALMAS: A Java/Prolog Framework for Deontic Action-Logic Multi-Agent Systems

Magnus Hjelmblom and Jan Odelstad

Department of Mathematics, Natural and Computer Sciences
University of Gävle, SE-801 76 Gävle, Sweden
mbm@hig.se, jod@hig.se

Abstract. A norm-regulated Deontic Action-Logic Multi-Agent System (DALMAS) is regulated by a normative system consisting of norms, which are expressed in an algebraic notation based on the Kanger-Lindahl theory of normative positions. A general-level Prolog implementation of the abstract DALMAS architecture makes it possible to implement specific systems in Prolog. This work describes jDALMAS, a Java library that may be used to create DALMAS applications in Java. A jDALMAS application has a client/server architecture, where a Prolog implementation of a specific system acts as a logic server. Together, the general-level Prolog implementation and the jDALMAS packages offer a framework for implementation of specific systems. Two examples of such implementations are presented here.

Keywords: deontic action-logic, normative positions, norm-regulated MAS, DALMAS, logic server.

1 Introduction

During the last decade, the study of norm-regulated multi-agent systems has emerged as an active field of study within AI. The notion of norm-regulated DALMAS (deontic action-logic based multi-agent system) was introduced by Odelstad and Boman in [11]. DALMAS is an abstract architecture for a global clock, global state and global dynamics multi-agent system. A norm-regulated DALMAS is regulated by a normative system which consists of norms expressed in an algebraic notation (cf. [8], [9]) based on the Kanger-Lindahl theory of normative positions [7].

In earlier work, a general-level Prolog implementation of the DALMAS architecture was developed. [5] The implementation consists of a Prolog module, *dnrDALMAS*, which may be used to implement specific DALMAS systems as logic servers. This paper presents *jDALMAS*, a general Java library that may be used to create Java applications that connect to dnrDALMAS servers. Together with the dnrDALMAS module, the jDALMAS library offers a framework for the implementation of specific DALMAS systems in Java, for example as Swing applications with graphical user interfaces.

Two such graphical applications have been developed to illustrate the function and use of the jDALMAS library. The Java code for jDALMAS and the two specific

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 110–119, 2009.

implementations is available in electronic form, together with the Prolog code for dnrDALMAS and the two specific logic servers.

The abstract architecture for DALMAS, and its theoretical foundations, is outlined in section 2.1 - 2.3. Two specific systems are described in section 2.3.1. Section 2.4 gives an overview of the dnrDALMAS implementation, and a description of jDALMAS and the two specific implementations is given in section 3.

2 Background

2.1 Deontic Action-Logic

The term *action-logic* will be used in this paper to denote a logical system which includes the binary action operator *Do*. The intended interpretation of *Do*(x, p) is "x sees to it that p".

Deontic logic may be used to create normative sentences, using norm-building operators such as *Shall* or *May*. The intended interpretation of *Shall* is "it shall be that" ("it shall be the case that"), and the intended interpretation of *May* is "it may be that". A conditional normative sentence consists of a combination of a descriptive sentence and a purely normative sentence. A typical conditional norm has the form $p \rightarrow$ *Shall* s. The interpretation of this sentence would be "if p, then it shall be that s".

Stig Kanger contributed to the deontic logic by combining it with action-logic. [6] For example, the intended interpretation of *Shall Do*(x, q) is "it shall be the case that x sees to it that q". The result of combining *Shall* or *May*, *Do* and $\neg$ is a powerful language for expressing normative sentences. A conditional norm may for example have the form $c(x,y) \rightarrow$ *Shall Do*($x,\neg d(y)$). According to the logic of *Shall* and *Do*, this means that "if $c(x,y)$, then it shall be the case that x sees to it that not $d(y)$". A more detailed presentation of deontic action-logic is given in [5], section 3.

The use of deontic logic within the design of multi-agent systems has been explored by many researchers. One example is the use of Constraint Handling Rules (CHR) to express deontic constraints within the area of agent communication. [2] Another example is IMPACT [3], an agent platform where deontic operators of permission, obligation and prohibition is the basis for the specification of what an agent is obliged to do, may do or cannot do. A third example is Sergot's Norman-G [12], a Prolog program based on the theory of normative positions.

2.2 Normative Positions

The Kanger-Lindahl theory of normative positions is based on Kanger's "deontic action-logic". The theory contains three systems of types of normative positions. The simplest of these systems is a system of seven one-agent types of normative positions, based on the logic of *Shall* and *Do*. [7]

$\mathbf{T}_i$ (where $1 \leq i \leq 7$) denotes the i:th type of *one-agent positions*. For example, $\mathbf{T}_2(\omega,d)$ denotes the deontic action-logic sentence *May Do*$(\omega,d) \wedge$ *May Pass*$(\omega, d) \wedge \neg$*May Do*$(\omega, \neg d)$, where *Pass*(ω, d) is an abbreviation of $\neg Do(\omega,d) \wedge \neg Do(\omega,\neg d)$. The complete list of one-agent types is presented in [5], section 4, together with a detailed discussion and references.

The types $\mathbf{T}_1 - \mathbf{T}_7$ may be used as operators on descriptive conditions to get deontic conditions. If, as an example, d is a unary condition, then $T_i d$ ($1 \leq i \leq 7$) is the binary condition such that

$$T_i d(y,x) \text{ iff } \mathbf{T}_i(x,d(y))$$

where $\mathbf{T}_i(x,d(y))$ is the i:th formula of one-agent normative positions.

2.3 Norm-Regulated DALMAS

In recent years, the study of norm-regulation of mult-agent systems has developed into a sub-discipline of AI called Normative Multi-Agent Systems, attracting the attention of specialists from different areas such as computer science, logic, sociology, and cognitive science. Central topics include the the use of norms as a mechanism in multi-agent systems and the use of multi-agent systems to study the concept and theories of norms and normative behaviour. An overview of this field of study is given by Boella et al. in [4], where ten research challenges for the NORMAS community are identified.

The abstract DALMAS architecture is one possible approach within this area. A deterministic DALMAS $\boldsymbol{D}$ is defined formally in [11], pp. 152f, as an ordered 9-tuple $\langle \Omega, A, \mathrm{S}, A_f, \Delta, \Pi, \Gamma, \tau, \gamma \rangle$. The arguments are specific sets, operators and functions which are models for the theory of DALMAS, defining the unique features of a specific DALMAS:

1. Ω is the *set of agents* in $\boldsymbol{D}$.
2. A is an *act set*. An element a in A is a function such that $a(\omega,s) = s^+$ means if agent ω performs act a in state s, then the resulting state will be s^+.
3. S is the *state space* of $\boldsymbol{D}$, that is, the the set of all states that may be reached when the agents perform feasible actions.
4. A_f is a function such that $A_f(\omega,s)$ is *the set of feasible acts* for agent ω in state s.
5. Δ is a deontic structure-operator, such that $\Delta(\omega,s)$ is ω's *deontic structure* on $A_f(\omega,s)$ in state s. In a *simple* DALMAS, the deontic structure for ω in state s is the set of permissible actions for ω in state s. An act is permissible if it is not prohibited by the DALMAS's normative system.
6. Π is a preference structure-operator such that $\Pi(\omega,s)$ is ω's *preference structure* on $A_f(\omega,s)$ in state s. The preference structure is an ordering of the acts in $A_f(\omega,s)$ according to the "utility" of the acts for the agent. In other words, Π determines which acts are the most preferable for the agent in the current state.
7. Γ is a function such that $\Gamma(\omega,s)$ is *the set of actions for ω to choose from* in state s, ordered according to the preference structure. In a *simple* DALMAS, the choice-set consists of the most preferred of the permissible actions.
8. τ is a turn-operator such that $\tau(\omega_1) = \omega_2$ means that ω_2 is to move after ω_1.
9. γ is a tie-breaking function, where $\gamma(\{a_1, \ldots, a_n\}) = a$ means that a is *the act to choose* out of a set of permissible and equally preferred actions.

When an agent is to move, it chooses an act out of a set of feasible acts. This leads to a new state, depending on the state of the system when the act is performed. The choice of act is determined by the combination of the DALMAS's preference structure and deontic structure.

The representation of norms is based on Lindahl-Odelstad's algebraic representation of normative systems, which in turn uses the Kanger-Lindahl theory of normative positions. For example, if c is a binary condition and d is a ternary condition, the norm $\langle M_1c, T_7d\rangle$ may be interpreted as $M_1c(\omega_1,\omega;\ \omega_m,s) \rightarrow T_7d(\omega_1,\omega_2,\omega;\ \omega_m,s)$ where ω_i is an agent, ω_m is the agent to move and s is the current state of the system. A deeper discussion is given in section 5 of [5].

2.3.1 Examples: Colour and Form and Waste-Collectors

Two simple systems, Colour & Form and Waste-collectors, are used in [5] and this paper as running examples to illustrate the ideas behind DALMAS. The Waste-collector system was originally used in [11] as an example, inspired by [14].

Colour & Form is a very simple multi-agent system consisting of only two agents called chroma and forma. States of the system are represented by the values of the colour ("black" or "white") and form ("circle" or "square") attributes for each agent.

The agents can choose between two acts: *change colour* and *change form*. The agents' behaviour is controlled by

1. a simple utility function which is defined such that agent chroma prefers to change colour, and agent forma prefers to change form; and
2. a normative system containing one norm that prohibits an agent to choose an act leading to a state where both agents have identical attribute values.

In other words, if $\omega_1 \neq \omega_2$, then the moving agent must not act so that all of ω_1*'s* attributes are identical with all of ω_2*'s* attributes. This norm may be expressed in logical form (with the universal quantifier ∀ omitted) in the following way:

$$Diff(\omega_1,\omega_2;\ s) \rightarrow \neg May\ Do(\omega_1, Eq(\omega_1,\omega_2;\ s))$$

Diff is defined such that $Diff(\omega_1,\omega_2;\ s)$ iff $\omega_1 \neq \omega_2$, and *Eq* is defined such that $Eq(\omega_1,\omega_2;\ s)$ iff agents ω_1 and ω_2 have identical attributes in state s. Using the appropriate T_i operator (see section 2.2), the norm may be expressed

$$M_1Diff(\omega_1,\omega_2,\omega;\ \omega_m,s) \rightarrow T_7Eq(\omega_1,\omega_2,\omega;\ \omega_m,s)$$

where the operator M_1 unifies agent ω_1 with the acting agent ω_m. The algebraic form of this norm is $\langle M_1Diff, T_7Eq\rangle$.

The **Waste-collector** system is a system of agents operating in an environment which consists of a grid of squares ordered in rows and columns. Each square is assigned coordinate in the form of an ordered pair of integers. Some squares contain an amount of waste, represented by a number. The agents can move one square at a time in four directions: up/north, down/south, left/west and right/east, and at the same time pick up waste from the square where the agent starts its move. The agents may also pass[1], that is, do nothing except pick up waste (if any) from the current square. To summarize, an agent may choose one of the following actions: go_{north}, go_{south}, go_{west}, go_{east}, or *pass*. An agent in the system tries to cooperate with other agents to collect as much waste as possible. Each waste-collector has a utility function, such that the utility of an act depends on the amount of waste in the squares surrounding the target

[1] Note that the *pass* act, meaning that an agent does nothing, should not be confused with the *Pass* operator, meaning that an agent is passive with regard to some state-condition *s*.

square. Also, there are some restrictions on how the waste-collector agents may act, especially on how they may move near other agents. These restrictions may be expressed as norms in a normative system. For simplicity, let us consider a normative system with a single norm stating that it is not permissible to move to a square that is already occupied by another agent. Omitting the ∀ quantifier, this norm may be expressed in logical form in the following way:

$$Diff(\omega_1,\omega_2; s) \rightarrow \neg May\ Do(\omega_1, Lap_9(\omega_1,\omega_2; s))$$

The intended meaning of $Lap_n(\omega_1,\omega_2)$ is that the protected spheres (that is, the square where the agent is presently located and the eight squares surrounding it) overlap with n squares. See figure 1 of [5] for an illustration. This may be expressed in algebraic form as $\langle M_1 Diff, T_7 Lap_9 \rangle$.

2.4 dnrDALMAS

Previously, a Prolog implementation of the abstract DALMAS architecture has been developed. [5] The implementation consists of a Prolog module, *dnrDALMAS*, which may be used to create standalone text-based applications or logic servers for specific DALMASes. The module is written in SICStus Prolog, version 4.

The dnrDALMAS module contains a set of predicates for creating and initialising specific DALMAS implementations and querying the state of the system. An implementation of a specific system defines a set of (user-defined) primary predicates written in Prolog. These primary predicates correspond to the primary functions characterizing the system. The heart of the implementation is the `prohibited/3` predicate, which determines if an act is permissible or not according to the norms in the normative system.

Using dnrDALMAS, two implementations of the example systems in section 2.3.1 have been created. These implementations function both as standalone applications with simple text-based user interfaces and as logic servers within the jDALMAS framework.

The implementation of dnrDALMAS and the two implementations of specific systems are thoroughly described and discussed in [5]. The details will not be repeated here. The code is publicly available via *sourceforge.net*.

3 The jDALMAS Framework

jDALMAS is a general Java library consisting of a set of Java classes that can be used to create Java applications that communicate with dnrDALMAS servers. The Java part of the application may then provide for a (graphical) user interface, while the dnrDALMAS server provides for the DALMAS logic. The code for jDALMAS is publicly available via *sourceforge.net*.

A jDALMAS application has a client/server architecture, where a Java client communicates with a dnrDALMAS server.

3.1 Server Side

The concept of a logic server is somewhat analogous to a database server, but a logic server provides for application logic rather than for data storage and persistence.

Basically, the server side of a jDALMAS application consists of a dnrDALMAS server, which is a logic server for a specific system built with the dnrDALMAS Prolog module. The `prologbeans` module [13] in SICStus Prolog is used to initialize and start a server listening on a certain (user-defined or OS-assigned) port, register general queries that are to be accepted by the server, and register a set of event listener callback predicates.

To facilitate the administration of a dnrDALMAS server, the jDALMAS framework provides for a simple graphical control panel for the server.

3.2 Client Side

A jDALMAS client is a Java Swing applet. It uses the PrologBeans Java library [13] in SICStus Prolog to communicate with the dnrDALMAS server. The user interface consists of a generic control panel (see figure 1) that lets the user configure the specific system. Through the panel, the user may:

- add agents to the agent set;
- add acts to the act set;
- register and/or update functors of primary and secondary Prolog predicates;
- add functors of ground-predicates to the ground set;
- add functors of consequence-predicates to the consequence set; and
- add norms to the norm set.

To create an implementation of a specific system, the user needs to perform the following steps:

- create Prolog definitions of the appropriate predicates, for example user-defined primary and/or secondary predicates, or state-conditions used in the grounds or consequences of norms;
- deploy the user-defined Prolog files on the server;
- create a "logic handler" that translates between the client's own representation of knowledge and the representation used by the dnrDALMAS server; and
- determine how the specific system's knowledge base shall be visualized on the client side, and write the necessary Java code.

When the client control panel is started, the user adds agents to the agent set and acts to the act set. The user also adds functors of state conditions to the ground set and consequence set, and adds norms to the norm set. Finally, the user registers functors of primary or secondary predicates, and adds knowledge to the knowledge base which is initially empty.

The client application should display the specific system's current knowledge base in an appropriate way, and let the user perform the desired actions on the system. For instance, the user may want to ask the system for its next situation, which is the resulting situation when the agent to move chooses and performs its most preferred act that is not prohibited by the norm-system.

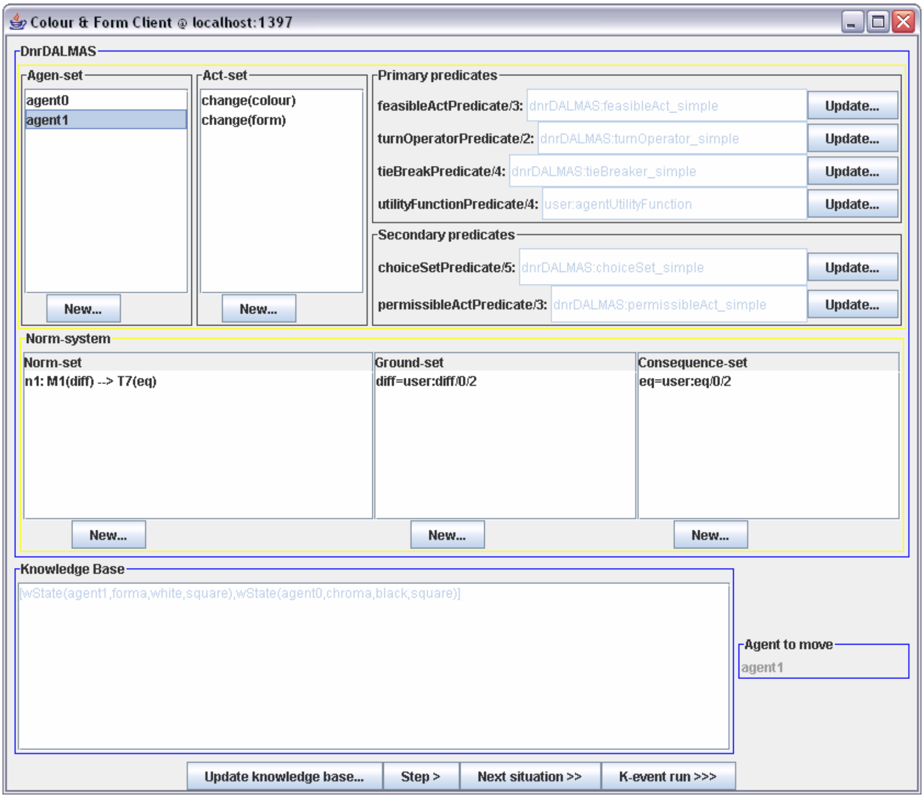


Fig. 1. Control panel for a specific jDALMAS client

3.2.1 Example 1: CFDALMAS

CFDALMAS is a graphical Java client for a Colour & Form system; see section 2.3.1. Aside from the client control panel shown in the previous section, the user interface consists of an agent frame for each agent. The agent frame (figure 2) shows a graphical representation of the agent's state.

3.2.2 Example 2: WasteDALMAS

WasteDALMAS is a graphical Java client for the Waste-collector system described in section 2.3.1. The user interface consists of two parts: a client control panel and state

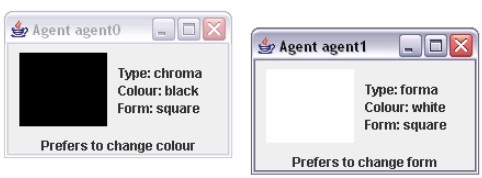


Fig. 2. Graphical representation of a state of the Colour & Form system

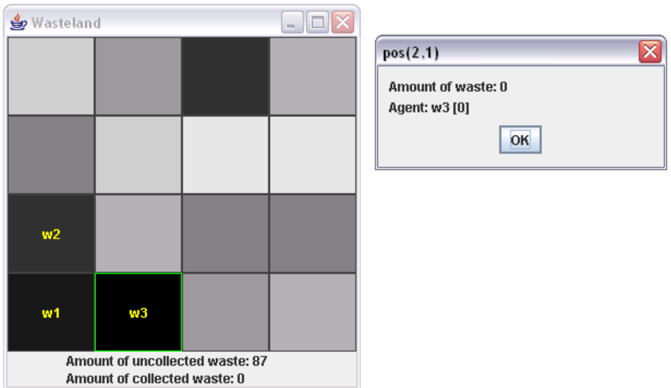


Fig. 3. Graphical representation of a state of the Waste-collector system

frame that shows a graphical representation of the state of the system. The client control panel is very similar to the one shown in section 3.2.

The state frame (see figure 3) shows the state of each square in the wasteland grid, including the position of agents and the amount of waste carried by each agent. The background colour of a square indicates how much waste the square contains; the darker the square, the less waste it contains. The square containing the agent which is next to move is marked by a green border. A dialog frame which contains more detailed information about a certain square is shown if the user clicks on the square.

4 Discussion

The jDALMAS framework is still work in progress. The current version works well, but some improvements and extensions can be made. An issue to deal with is how the implementations should behave in situations where the deontic structure is empty, that is, when all feasible acts for the acting agent are prohibited by the normative system. The current version of the framework leaves this issue to the user: if, in a given situation, there are no permissible acts, the run of the system fails. Another approach in this situation could be to let the moving agent be "confused", in the sense that it does nothing. However, this would mean that the "pass" (i.e. "do nothing") act will always be permissible, even if prohibited by the normative system. Another issue is to extend the framework to handle non-elementary norms, i.e. norms whose grounds and/or consequences are Boolean expressions.

5 Conclusion and Future Work

jDALMAS offers a framework for the creation of specific DALMAS applications, making it possible to implement a wide range of norm-regulated multi-agent systems. It also offers the possibility to design and experiment with different normative systems for a given agent system. For a system such as the Waste-collectors it would be possible to experiment with different combinations of normative systems and utility functions, to see which normative system gives the best overall performance or desired behaviour according to some evaluation function or the user's preferences. If combined with some learning mechanism, such as a genetic algorithm, the system itself could try to find an optimal combination for a given class of problems. [11], pp. 164f.

Other applications of DALMAS can be found in many areas. Somewhat related are the fields of education, edutainment and entertainment. A norm-regulated DALMAS could for example be part of a game environment which lets the user act as a "legislator", creating and playing with different "laws" for the game world.

Another idea is to use the framework for experiments involving humans. Is it, for example, possible for human observers to conclude which normative system is used by a particular application, just by examining the application's behaviour? This could be an interesting tool within many different disciplines.

It could also be rewarding to explore the use of DALMAS within decision support. Possible applications within this area could be to use norm-regulated systems as an analytical tool for the creation and/or evaluation of decision-theoretical expert systems. In a given situation, such a system could for example facilitate decision making by eliminating those feasible acts that are prohibited by some normative system.

Ahonen-Jonnarth and Odelstad have discussed the area of forest cleaning as a possible area of application for norm-regulated multi-agent systems. A single cleaning agent may be regarded as a one-agent system where the agent's decision making is regulated by a normative system. [1], [10]

References

[1] Ahonen-Jonnarth, U., Odelstad, J.: Evaluation of Simulations with Conflicting Goals with Application of Cleaning of Young Forest Stands. In: Proceedings of ISC 2006 (Fourth Annual International Industrial Simulation Conference), Palermo, Italy, June 5-7 (2006)

[2] Alberti, et al.: Logic Based Semantics for an Agent Communication Language. DEIS Technical Report no. DEIS-LIA-03-001. LIA Series no. 62

[3] Arisha, K.A., Ozcan, F., Ross, R., Subrahamanian, V.S., Eiter, T., Kraus, S.: IMPACT: A platform for collaborating agents. IEEE Intelligent Systems 14(2), 64–72 (1999)

[4] Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. Computation and Mathematical Organizational Theory, special issue on normative multiagent systems 12(2-3), 71–79 (2006)

[5] Hjelmblom, M.: Deontic Action-Logic Multi-Agent Systems in Prolog. Institutionen för matematik, natur- och datavetenskap, Högskolan i Gävle (2008) ISSN 1403-8749;30, `http://hig.diva-portal.org/smash/record.jsf?pid=diva2:118137`

[6] Kanger, S.: New foundations for ethical theory, Part 1 (1957). In: Holmström-Hintikka, et al. (eds.) Collected Papers of Stig Kanger With Essays on His Life and Work, pp. 99–119. Kluwer, Dordrecht (2001)

[7] Lindahl, L.: Position and Change: A study in Law and Logic. Reidel, Dordrecht (1977)

[8] Lindahl, L., Odelstad, J.: An Algebraic Analysis of Normative Systems. Ratio Juris 13(3), 261–278 (2000)

[9] Lindahl, L., Odelstad, J.: Normative Positions within an Algebraic Approach to Normative Systems. Journal of Applied Logic 2, 63–91 (2004)

[10] Odelstad, J.: Agents, Norms and Forest Cleaning. In: Boella, G., van der Torre, L., Verhagen, H. (eds.) Normative Multi-Agent Systems. Dagstuhl Seminar Proceedings, 07122 (2007) ISSN 1862-4405, http://drops.dagstuhl.de/portals/index.php?semnr=07122

[11] Odelstad, J., Boman, M.: Algebras for Agent Norm-Regulation. Annals of Mathematics and Artificial Intelligence 42, 141–166 (2004)

[12] Sergot, M.J.: A computational theory of normative positions. ACM Trans. Comput. Logic (TOCL) 2, 581–622 (2001)

[13] SICStus Prolog User's Manual. The Intelligent Systems Laboratory, Swedish Institute of Computer Science, Kista (2005)

[14] Steels, L.: Cooperation between distributed agents through self organization. In: Decentralized, A.I., Demazeau, Y., Muller, J.P. (eds.) Proc. of Maamaw 1989, pp. 175–196. Elsevier Science Publisher, Amsterdam (1990)

Belief Revision in a Fact-Rule Agent's Belief Base

Hai H. Nguyen

School of Computer Science,
University of Nottingham, UK
hhn07u@cs.nott.ac.uk

Abstract. When there is much incoming information with different levels of reliability, the agent's belief base could become inconsistent, and therefore it needs a revision procedure. The algorithm which implements this revision should be rational and computationally efficient. Besides, because an agent's belief could be either a fact or a rule, it is essential to perform belief revision by removing not only facts but also rules. In this paper, the work presented in [1] and [2] is extended to implement contraction by rules as well as by facts in polynomial time.

Keywords: Belief revision, Rule contraction, Rule-based agents.

1 Introduction

To perform its operation, an agent needs information from many sources. The more information the agent has, the more rational and sound its decisions could be. However, not all of incoming information which the agent receives are reliable, because they might be from different channels such as distributed knowledge base, the agent's observation, derivations from agent's assumptions or agents' communication. Unreliable information mixing with reliable knowledge could easily lead to inconsistence in the agent's belief base. As a result, the agent needs to revise its beliefs to maintain the consistency, and this process, therefore, is called *belief revision*. There are three main operations in a belief revision process: expansion, contraction and revision. In this paper, the main concern is on the contraction operation, although a brief construction of revision is also given.

In the literature, two main approaches to belief revision have been introduced: AGM (Alchourrón, Gärdenfors and Makinson) style belief revision (also known as coherent style) [3] and reason-maintenance style belief revision [4]. AGM style belief revision focuses on the logical structure of the beliefs, i.e. the way one belief related to other accepted beliefs in the current state, and informational economy, which is the condition that changes to belief state should be as minimal as possible. This style of belief revision is characterised by AGM postulates [3]. In contrast, reason-maintenance, or foundational, style of belief revision based on tracking the justification of one's beliefs. If all justifications for a belief are removed then the belief itself is also removed from the belief state. This paper

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 120–130, 2009.

combines the two approaches: it uses justifications together with beliefs but does not remove unsupported beliefs (i.e. beliefs with no justification).

Despite of the fact that belief revision is considered one of the central problems of modern Artificial Intelligence, most current practical agent programming languages have not featured the automatic belief revision yet. The reason is the lack of efficient implementations of belief revision operations for resource-bounded agents. However, recent work by Alechina et al. [1] have been successful in integrating belief revision feature in ***Jason*** [5], an interpreter of a popular agent-oriented programming language AgentSpeak. This feature in ***Jason*** helps to release the burden of keeping the belief-base consistent from programmers.

However, the previous work of Alechina et al. in contraction by literals (also known as McAllester's contraction[1]) still treats facts and rules separately and considers that rules of an agents could not be changed, i.e. expanded or removed. This approach is, however, not ideal in some contexts. Furthermore, it limits the ability of an agent to exchange rules (or plans in some agent programming languages) among other agents, and hence also reduces the richness of information that agent should have. The aim of this paper is to implement a more general contraction, in which rules and facts are treated in the same way, based on the previous work reported in [1,2].

The paper is organised as follows. In section 2, a rule-based agent and its corresponding logic are described. In section 3, we argue the need for contraction by rules and facts together, present the algorithm of a general contraction by beliefs, analyse its complexity and show the methods of computing preferences and constructing revision based on contraction. In section 4, we give examples of contraction by beliefs based on the defined algorithm. Comparison to other approaches is presented in section 5. Finally, the paper is concluded by summarising the achievement and introducing directions of future work.

2 The Agent and Its Logic

In this section, we introduce a rule-based agent and its logic. The model of the agent is similar to [2] but here rules are treated as beliefs. The agent has a finite state which is also its current *belief-base*. Beliefs in the agent can be in the form of ground literals and rules. All beliefs are represented in predicate logic. Assume that there are a fixed set of predicate symbols P, a fixed set of variables $\mathcal{X}$ and a domain $\mathcal{D}$, which is a fixed set of constants. *Literals* are predicate symbols of n arguments or their negations followed by n variables or constants. The literal is called *ground literal* when all arguments of its predicate symbol are constant. For example, $\neg\texttt{HasA}(\texttt{x}, \texttt{PairOfWings})$ and $\texttt{IsA}(\texttt{Penguin}, \texttt{Bird})$ are both literals, given that `IsA` and `HasA` are two binary predicate symbols, `Penguin` and `Bird` are elements of the domain $\mathcal{D}$, and x belongs to $\mathcal{X}$. Note that the latter is a ground literal. In our agent, the belief-base contains a finite set of rules, denoted

[1] The algorithm of contraction by literals is similar to Boolean Constraint Propagation (BCP) algorithm introduced in [6] by McAllester.

by $\mathcal{R}$. Rules are universally-quantified positive Horn clauses with a non-empty body, which are of the form:

$$A_1, \ldots, A_n \rightarrow B$$

where $A_1, \ldots, A_n (n \geq 1), B$ are literals. Each of A_i and B are called a *premise* and the *consequence* respectively. All variables in rules, more precisely in all premises and the consequence, are universally quantified and they also do not contain function symbols.

The agent's logic is a weak logic W, which has only a single rule of inference, a *generalised modus ponens* (GMP):

$$\frac{\delta(A_1), \ldots, \delta(A_n), \quad A_1, \ldots, A_n \rightarrow B}{\delta(B)}$$

where δ is a function which substitutes all free variables in the rule by constants. The belief set is closed under consequence in W. Because of this inference rule, the belief base contains only rules and ground literals, assumed that originally, it has only ground literals and rules. Besides, the belief base is also finite if the agent's original state is finite.

The procedure of firing a rule in the agent is simple. It firstly creates an instance of the rule $\delta(A_1, \ldots, A_n \rightarrow B)$ by replacing all of the rule's free variables with constants. Then it uses the rule of inference GMP and ground literals corresponding to that instance, i.e. $\delta(A_1), \ldots, \delta(A_n)$, to derive a new ground literal $\delta(B)$. Finally, $\delta(B)$ is added to current belief base if this ground literal has not existed yet.

To perform belief revision and contraction in the agent, the agent's rules have to be fired to *quiescence*, which is the setting that the agent cannot obtain more new knowledge when firing all of its rules. This setting is similar to the concept of fixed point of a function in mathematics. In the quiescent setting, the belief base is closed under agent's rules and with respect to derivability in W. A belief base is said inconsistent when it is in the quiescent setting and there is at least one pair of a fact and its negation.

An example of the agent's belief base is as follows:[2]

R1 $\texttt{IsA}(\texttt{x}, \texttt{Bird}), \texttt{HasA}(\texttt{x}, \texttt{PairOfWings}) \rightarrow \texttt{Fly}(\texttt{x})$
R2 $\texttt{IsA}(\texttt{x}, \texttt{Bird}) \rightarrow \neg\texttt{Swim}(\texttt{x})$
R3 $\texttt{Fly}(\texttt{x}) \rightarrow \neg\texttt{Swim}(\texttt{x})$
F1 $\texttt{IsA}(\texttt{Penguin}, \texttt{Bird})$
F2 $\texttt{HasA}(\texttt{Penguin}, \texttt{PairOfWings})$
F3 $\texttt{IsA}(\texttt{Swan}, \texttt{Bird})$
F4 $\texttt{Swim}(\texttt{Dog})$.

Firstly, the agent fires its rule **R1** with facts **F1** and **F2** to derive a new fact:

F5 $\texttt{Fly}(\texttt{Penguin})$

[2] Rules and facts (ground literals) are denoted by R_i and F_i respectively.

Then it fires rule **R2** twice, the first time with fact **F1** and the second time with fact **F3** to derive two more facts:

F6 ¬`Swim`(`Penguin`)
F7 ¬`Swim`(`Swan`)

Finally, this agent fires rule **R3** with a derived fact, **F5**, to obtain a fact ¬`Swim`(`Penguin`). However, this fact is already in the current belief base (known as **F6**); therefore, the agent does not add the fact to its belief base. At this stage, the agent's belief base is still consistent because there is no pair of facts such that one is the negation of another.

3 Contraction by Beliefs

In this section, we examine the need of contraction by rules as an addition to the contraction defined in [2], which is only by literals. It is followed by an extension of the algorithm in order to perform a more general contraction, contraction by beliefs. Finally, a method to assign preferences to beliefs based on the quality of non-inferential justifications is provided.

3.1 Contraction by Literals[3] and Rules

In previous work reported in [2], contraction is done only by literals for various reasons. For example, rules are usually the representation of agents' plans in some agent-oriented programming languages or rules are considered as a part of agent program, which is unlikely to be modified. One could also argue that rules could be a domain knowledge, such as ontological rules; therefore they might not be changed or removed safely. On the other hand, facts could be likely from various sources with different levels of reliability. They could be assumptions of the agent, its observation, or the new information which the agent collects from other agents. As the result, it is only necessary to perform contraction by literals.

However, this is not ideal in the new context, where rules are considered as non-inferable beliefs. In this context, rules are no longer fixed for each agent, in particular, these rules, as other beliefs, could be exchanged among agents. This context could help to increase the richness of information because not only the rules but also the inferable facts are added into the agent's belief base although this could also lead to the increase of noise in the belief base. Moreover, in some ontology settings, even if rules are domain knowledge and assumed by the agent, it is not necessary that such rules are always correct, or in other words, they could always derive reliable facts. For example, in our example of belief base, rule **R1** could not be considered reliable if the agent observes the new fact that our `Penguin` cannot fly, or formally ¬`Fly`(`Penguin`). Note that the assumed

[3] More precisely, it is the contraction by facts (ground literals). However, for simplicity, in this section, we assume literals are ground literals, i.e. literals with no free variable.

rules might be less reliable than facts observed by the agent.[4] In general, if one contracts by a derived literal, it is unreasonable not to consider whether the rule, from which the new literal is inferred, is less reliable than other facts or not.

3.2 Contraction by Beliefs

In this section, a more general contraction, *contraction by a belief*, is defined as the removal of the belief itself and some other beliefs so that the contracted belief is no longer inferable from the agent's belief base. To construct a formal definition, it is necessary to have following notions:

Each belief B in the belief set is assigned a number, $p(B)$, to represent its *degree of preference*. One step further, a total preference order relation $\preceq$ on the set of beliefs is defined as follows. A total preference order relation $\preceq$ on the set of beliefs is a binary relation such that for every two beliefs we could decide which one is strictly more reliable than another. For two beliefs A and B, we denote $A \preceq B$ if $p(A) < p(B)$; if $p(A) = p(B)$, we can use alternative orders such as lexicographical order or established-time order to determine whether $A \preceq B$ or $B \preceq A$. Since $\preceq$ is total, every set of beliefs has the minimal element. Therefore, for any finite set of beliefs Γ, we could denote $w(\Gamma)$ as the weakest element of Γ, or in other words, the minimal element of Γ in regards to relation $\preceq$.

Based on $\preceq$ relation, let us denote a dependence relation in the set of beliefs, $\gg_K$, where K is the finite set of agent's beliefs. For two beliefs B_1 and B_2 in K, $B_1 \gg_K B_2$ (we say that B_2 depends on B_1 in K), if either:

1. $B_1 = B_2$; or,
2. if B_2 is a ground literal $\delta(B)$, $R = A_1, \ldots, A_n \to B \in K$, $\delta(A_1), \ldots, \delta(A_n) \in K$, and B_1 is the minimal element, with respect to $\preceq$, in the set of rule R and its instance's premises $\{R, \delta(A_1), \ldots, \delta(A_n)\}$, formally: $B_1 = w(R, \delta(A_1), \ldots, \delta(A_n))$; or,
3. for some $R = A_1, \ldots, A_n \to C \in K$, $\delta(A_1), \ldots, \delta(A_n) \in K$, $B_1 = w(R, \delta(A_1), \ldots, \delta(A_n))$, and $\delta(C) \gg_K B_2$.

Note that both B_1 and B_2 have to be in the belief set, and it is not true that if B_1 is a belief which is used to derived B_2 then $B_1 \gg_K B_2$. More precisely, B_1 is a least preferred belief involved in some step of a derivation of B_2.[5]

With $\vdash$ and Cn denoting derivability in logic W and closure under the consequence in W, the formal definition of a contraction by a belief is given as follows:

[4] In this example, we assume that each premise of the rule instance is more reliable than the rule itself (i.e. **F1** is assumed to be more reliable than rule **R1**).

[5] There are possibly more than one derivations of a belief. For example, if we have two rules $A, B \to D$, $C \to D$ and three facts A, B and C, we have two different derivations of D from the original belief base. Also note that a derivation might also contain several steps. An example is that if we have $A \to B$, $B, C \to D$ and, $\delta(B)$ is less preferred than $\delta(C)$ and the rule $B, C \to D$, then the derivation of $\delta(D)$ has two steps, corresponding to two used rules. In this example, if $\delta(A)$ is less preferred than the rule $A \to B$, we have $\delta(A) \gg_K \delta(D)$ and $\delta(B) \gg_K \delta(D)$.

Definition 1. *A contraction of K by a belief B, $K \dot{-} B$, is defined as*

$$K \dot{-} B \stackrel{df}{=} Cn(K \backslash \Gamma)$$

where $\Gamma \subseteq \{A : A \gg_K B\}$ and $K \backslash \Gamma \nvdash B$.

3.3 Implementation and Complexity of Contraction by Beliefs

This section presents the implementation of the contraction operator defined in Definition 1.[6] Moreover, the time complexity of this implementation is shown to be polynomial to the size of the agent's belief base.

Let us define a *justification* for a belief A is a pair of the belief and its *support list*, denoted by (A, s) where s is the support list of the justification. The support list of a justification contains all premises of the rule's instance used to derive that belief and the rule itself. In general, if the inferable belief $B(a)$ is derived by the rule $A_1(x), \ldots, A_n(x) \rightarrow B(x)$ and facts $A_1(a), \ldots, A_n(a)$, its justification is in the form of

$$(B(a), [A_1(a), \ldots, A_n(a), (A_1(x), \ldots, A_n(x) \rightarrow B(x))]).$$

For example, the belief base in previous example has belief **F6** $\neg$`Swim`(`Penguin`). This belief has two justifications because it was derived by two different rules. Two justifications of **F6** are $(\mathbf{F6}, [\mathbf{F1}, \mathbf{R2}])$ and $(\mathbf{F6}, [\mathbf{F5}, \mathbf{R3}])$.

Note that initially each belief in the agent's belief base has at least one justification. If the belief has only one justification and its justification has an empty support list, we call that belief *foundational belief*. It is also clear that foundational beliefs could not be derived and rules in our agent are always foundational beliefs. More formally, each rule R in our belief base has only one justification in the form of $(R, [])$.

Let us assume that the agent's belief base is represented by a directed graph of beliefs and justifications. For each belief B in K, there are incoming edges from its justifications and outgoing edges to justifications whose support list contains B.[7] For each support list s, there is a least preferred belief in s, denoted by $w(s)$, which is assumed accessible in constant time.

The algorithm implementing contraction by belief A has two loops. The first loop is to remove all justifications in A's outgoing edges. The upper bound of number of steps in this loop is $r(k + 2)$ where r is the number of rule instances[8] and k is the maximum number of premises in a rule. To remove a justification, the agent needs to remove all of its incoming edges, including edges from its premises

[6] The algorithm is almost the same as in [2], but the definitions of justification and support are changed to include rules. The complexity analysis of the algorithm is also slightly modified.

[7] It is also obvious that each justification could have $n (n \geq 0)$ incoming edges but exactly one outgoing edge. If there is no incoming edge, the justification is called non-inferential justification.

[8] Each rule instance is corresponding to a justification with an unempty support.

Algorithm 1. Contraction by belief A

```
for all j = (B, s) with an edge from A do
    remove j (and all edges to and from j) from the graph
end for
for all j = (A, s) with an edge to A do
    if s == [] then
        remove j (and the edge from j to A)
    else
        contract by w(s)
    end if
end for
remove A
```

and one edge from the rule, and its outgoing edge. In other words, at most $k+1$ incoming edges and one outgoing edge for each justification are removed in the first loop. In the second loop, for each justification in A's incoming edges, if its support list is empty, the justification is removed; otherwise, the contraction by the weakest element of its support list is recursively done. This loop will take maximum n steps where n is the number of beliefs. Finally, the belief is removed from agent's belief base. In general, the contraction algorithm has the time complexity of $O(kr + n)$, given that removing beliefs, justification and their edges are constant time operations. Algorithm 1 can be extended to adapt reason-maintenance style. Modification of the algorithm to implement this style of contraction is presented in [2].

3.4 Preferences and Revision

In this section, the term *quality* is used to imply the strength of each justification. Justifications with higher quality is stronger than one with lower quality. Let us assume that initially the agent's belief base is empty. Then any new belief added to the belief base has a justification with empty support, namely, *non-inferential justification*. This non-inferential justification is assigned a quality by the agent. The agent can use different ways to assess the quality of that non-inferential justification such as it might trust its assumptions more than information collected by communication to other agents, but less than its own observations. Alternatively, it could also prefer new beliefs to the old ones or prefer information from some agent to information from another agent. Within the scope of this paper, the methods agents use to assess the quality of information are not concerned.

The preference of a belief A, denoted by $p(A)$[9], is the highest quality of its justifications: $p(A) = \max\{qual(j_0), \ldots, qual(j_n)\}$, where $j_0, \ldots, j_n$ are justifications in A's incoming edges. The quality of an inferential justification is the degree of preference of the weakest belief in its support: $qual(j) = \min\{p(A) : A \in \text{ support of } j\}$. This is related to the idea of *partial entrenchment ranking* introduced in [7]. Based on these definition of preference and quality, we can

[9] Recall Section 3.2.

assign preferences to non-foundational beliefs and qualities to inferential justifications. An implementation of preferences computation of literals is given in [8]. Although this implementation is to compute preferences of literals, one could use it to compute preferences of general beliefs, given the new definition of belief and the fact that preference notion is now applicable to beliefs, not only literals.

The above preferred contraction could be applied to define the revision operation. This implementation could be found in [2], which also includes representation theorems for belief contraction and revision. The main idea is simple, to revise the belief base by a belief A, firstly the agent adds A to its belief base and fires its rules to quiescence. Then it lists all contradictory pairs of facts[10] and orders them by the degree of preference of each pair's weaker fact. Finally, it loops through this list and contracts by the weaker fact in each pair.

4 Examples

In this section, various examples of contractions by beliefs, including literals and rules, based on Algorithm 1 are introduced. Let us look back at the previous example of the agent's belief base, which contains

R1 2 $\mathtt{IsA(x, Bird), HasA(x, PairOfWings)} \rightarrow \mathtt{Fly(x)}$
R2 2 $\mathtt{IsA(x, Bird)} \rightarrow \neg\mathtt{Swim(x)}$
R3 2 $\mathtt{Fly(x)} \rightarrow \neg\mathtt{Swim(x)}$
F1 3 $\mathtt{IsA(Penguin, Bird)}$
F2 4 $\mathtt{HasA(Penguin, PairOfWings)}$
F3 3 $\mathtt{IsA(Swan, Bird)}$
F4 4 $\mathtt{Swim(Dog)}$
F5 2 $\mathtt{Fly(Penguin)}$
F6 2 $\neg\mathtt{Swim(Penguin)}$
F7 2 $\neg\mathtt{Swim(Swan)}$.[11]

The agent associates beliefs with preferences, which are the numbers after beliefs' label. For simplicity, let us classify beliefs reliability into four categories: 1 for information from unreliable communication, 2 for rule assumptions (i.e. rules which the agent assumes to be valid), 3 for domain knowledge and 4 for beliefs from observation.

Assume that the agent hears from an unreliable agent that `Dog` is a `Bird` and adds that fact into our belief base

F8 1 $\mathtt{IsA(Dog, Bird)}$

[10] Note that there is no contradiction of rules in a belief base. For example, if the belief base contains $A(x) \rightarrow B(x)$ and $A(x) \rightarrow \neg B(x)$, it is still consistent as long as there is no fact such as $A(a)$, which can be used to derive contradictory facts $B(a)$ and $\neg B(a)$.

[11] Remember that **F5**, **F6**, **F7** are derived facts.

and then the agent can use rule **R2** derive a new fact **F9** with justification (**F9**,[**F8**,**R2**],1)[12]

F9 1 ¬`Swim(Dog)`.

The belief base is no longer consistent, since there are contradictory facts (**F4** and **F9**). Therefore, it has to perform contraction by the less preferred fact, **F9**. This also leads to the contraction by less preferred beliefs among **F8** and **R2** so that **F9** is no longer inferable. Because **F8** is a foundational belief, it is removed without contracting by any other belief. Finally, **F9** is removed. At the moment, the belief base is consistent (actually, it goes back to the original state).

Now let our agent observe that `Penguin` can swim and adds the fact to its belief base

F10 4 `Swim(Penguin)`.

After firing its rules, the agent's belief base gains no new knowledge. However, it is now inconsistent because **F10** contradicts **F6**. Clearly, the agent has to contract by **F6**. Note that **F6** has two justifications, (**F6**,[**F1**,**R2**],2) and (**F6**,[**F5**,**R3**],2). With the former justification, because $p(\mathbf{F1}) > p(\mathbf{R2})$, the agent needs to contract by **R2** and consequently, removes **R2** from its belief base. With justification (**F6**,[**F5**,**R3**],2), because $p(\mathbf{F5}) = p(\mathbf{R3}) = 2$, the agent needs to choose which one to contract. Assume that the agent now prefers assumptions to derived beliefs, and recall that **F5** is a derived belief from **F1**, **F2** and **R1**, the agent performs contraction by **F5**. As a result, it has to contract either **F1**, **F2** or **R1**. Since **R1** is the least preferred belief among those, it is contracted and removed. The agent finally removes **F5** and **F6**. In total, contraction by **F6** leads to the removals of **R2**, **F5**, **R1** and **F6**. The belief base is still deductively closed and now consistent.

R3 2 `Fly(x)` → ¬`Swim(x)`
F1 3 `IsA(Penguin, Bird)`
F2 4 `HasA(Penguin, PairOfWings)`
F3 3 `IsA(Swan, Bird)`
F4 4 `Swim(Dog)`
F7 2 ¬`Swim(Swan)`
F10 4 `Swim(Penguin)`.

5 Comparison to Other Approaches

Due to the lack of space, only a brief comparison section is given. The most related approach to the paper is reported in [2]. As argued in section 3.1, [2] does not allow to remove rules in belief contraction. Based on this approach, the result of contraction by **F6** in the previous example will lead to the removals of only

[12] Each justification now has a quality number attached, which is the minimal preference of its support list.

F5, **F1** and **F6**. This result, however, could cause the loop of similar contracting processes when similar facts are frequently sent to the agent to trigger unreliable rules (e.g., **R1** and **R2**).

The other implemented belief revision system relevant to our approach is [9]. However, this approach does not keep track of justifications for a belief; instead, it only cares about the subset of beliefs from which that belief is derived. Besides, the implementation in [9] works under full first order logic and always has to check all beliefs in the belief base to compute a particular belief's rank.

6 Conclusion and Future Research

In this paper, a rule-based agent whose belief base contains rules and facts has been presented. Unlike in [2], the agent's rules are now considered as foundational beliefs and no longer fixed, and hence, could be expanded or contracted. Also, a more general contraction algorithm for the agent compared to [2], contraction by beliefs, is provided and shown to work in polynomial time so that it is applicable to practical agent programming languages. Before the conclusion, examples of contraction in different scenarios are also given.

The current work can be extended in different ways. For example, the agent's logic now is a weak logic, which has only a single rule of inference. If the definition of rules in the agent's belief base could be extended so that premises could also be rules then the agents would be able to create different rules of inference beside the GMP. Another direction of future research is to explore how foundational beliefs could be associated with their degrees of preference, which could help to complete the automatic belief revision process.

References

1. Alechina, N., Bordini, R.H., Hubner, J.F., Jago, M., Logan, B.: Belief revision for agentspeak agents. In: Proceedings of AAMAS 2006, Japan. IEEE Press, Los Alamitos (2006)
2. Alechina, N., Jago, M., Logan, B.: Preference-based belief revision for rule-based agents. Synthese 165(1), 159–177 (2008)
3. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet functions for contraction and revision. Journal of Symbolic Logic 50, 510–530 (1985)
4. Doyle, J.: Truth maintenance systems for problem solving. In: Proceedings of the Fifth International Joint Conference on Artificial Intelligence, IJCAI 1977, p. 247 (1977)
5. Moreira, A.F., Vieira, R., Bordini, R.H., Hübner, J.: Agent-oriented programming with underlying ontological reasoning. In: Baldoni, M., Endriss, U., Omicini, A., Torroni, P. (eds.) DALT 2005. LNCS, vol. 3904, pp. 155–170. Springer, Heidelberg (2006)
6. McAllester, D.A.: Truth maintenance. In: Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI 1990), pp. 1109–1116. AAAI Press, Menlo Park (1990)

7. Williams, M.A.: Iterated theory base change: A computational model. In: Proceedings of (IJCAI 1995), pp. 1541–1549. Morgan Kaufmann, San Francisco (1995)
8. Alechina, N., Jago, M., Logan, B.: Resource-bounded belief revision and contraction. In: Baldoni, M., Endriss, U., Omicini, A., Torroni, P. (eds.) DALT 2005. LNCS, vol. 3904, pp. 141–154. Springer, Heidelberg (2006)
9. Dixon, S., Wobcke, W.: The implementation of a first-order logic AGM belief revision system. In: Proceedings of 5th ICTAI, pp. 40–47 (1993)

Modeling Multi-Agent Systems as Labeled Transitions Systems: A Unifying Approach

Walid Chainbi

Sousse National School of Engineers/LI3
Avenue 18 Janvier, Bab Jdid - 4000 - Sousse - Tunisia
Walid.Chainbi@gmail.com

Abstract. In this paper, labeled transition systems are proposed as a unifying approach to deal with the semantics of a multi-agent system. The presented model represents a multi-agent system by a triplet consisting of a set of states, a set of actions and a set of all possible system executions. We argue that notions that appear with different and complex forms in the literature such as possible world semantics, Petri nets, and process calculus can be described in the same formal framework namely transition systems. Besides, the advantages related to the use of transition system based model are enumerated.

Keywords: Transition systems, multi-agent systems, behavioral semantics, modal logics.

1 Introduction

Agent-oriented techniques provide a new approach that aims at supporting the whole software development process. The goal of agent-oriented techniques is to handle all phases of this process with a single, uniform concept, namely that of agents. An agent is a computational entity that can be viewed as perceiving and acting upon its environment and that it is autonomous in that its behavior at least partially depends on its own experience [1]. A multi-agent system is a system designed and implemented as several interacting agents [2]. Examples of interaction include: cooperation (working together towards a common goal); coordination (organizing problem solving activity so that harmful interactions are avoided or beneficial interactions are exploited); and negotiation (coming to an agreement which is acceptable to all parties involved).

This paper is about the use of formal models to deal with multi-agent systems behaviors. We focus on the use of labeled transition systems, which have been initially introduced to define the semantics of parallelism [3], as a semantic model of a multi-agent system. Each labeled transitions system is a kind of a directed graph with vertices corresponding to states, and edges labeled with actions. Each state denotes the complete instantaneous state of the system. Consider, for instance, a system with two states defined by the value of a boolean-valued parameter p. Executing action a causes p to be true. Figure 1 represents graphically this system.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 131–140, 2009.

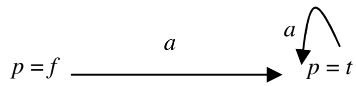


Fig. 1. An example of a transition system

Other models have been also used in the analysis of distributed computer systems including multi-agent systems. Among these models, we find possible world semantics [4], Petri nets [5], Milner's process calculus [6]. We argue in this paper that these models amount to specify a set of states, and a set of transitions between these states. The presented study shows also the use of labeled transition systems to deal with the truth conditions of formulae related to a multi-agent system's specification language.

This paper is organized as follows. Section 2 shows the use of labeled transitions systems to deal with the behavioral aspect of a multi-agent system. The presented model represents a multi-agent system by a triplet consisting of a set of states, a set of actions and a set of all possible system executions. Section 3 identifies an application of the presented model, consisting in a semantics basis for a specification language. In section 4, we discuss related work and we show that labeled transition systems may be envisioned as a unifying approach. We conclude in section 5 with a general discussion of the appropriateness of the presented study, and we give directions for future work. Our work is illustrated with the well known prey/predators problem in which four predators try to capture a prey on a bidimensional grid (i.e., getting closer to the prey from four sides: north, south, east, and west).

2 Labeled Transition Systems to Model Multi-Agent Systems

The proposed model represents an agent system by a triplet < **S**, **A**, **Ω** > consisting of:

- a set **S** of states where each element describes the complete instantaneous state of the system,
- a set **A** of actions, and
- a set **Ω** of all possible system executions which are finite or infinite sequences of the form

  $$s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \ldots$$

  where each s_i is a state and each a_i is an action.

First, we deal with the behavioral semantics of a single-agent system. Then, we extend this model in order to represent the behavioral semantics of a multi-agent system.

2.1 A Single-Agent System Model

Before specifying the possible state transitions caused by each action a of **A**, we define an agent's state. It indicates the aspects that may change over time.

Definition 2.1 [An agent state]
An agent state is a tuple <*B*, *G*, *R*> defined as follows: *B* is the set of beliefs of the agent, *G* are his goals, and *R* is his role.

The justification of such an intentional structure is out of the scope of this paper, but can be found in [7, 8]. Beliefs, goals, and roles denote respectively, the information, motivational, and deliberative states of an agent. We represent beliefs by ground atoms of first-order logic (i.e., atomic formulae containing no variables). An agent's goal is represented by a first-order formula. An agent's role is defined by an ordered set of actions which can be physical (i.e., interactions between agents and the spatial environment), communicative (i.e., emission or reception actions), private actions (i.e., internal functions of an agent), or decision action (i.e., can generate communicative, physical and private actions).

Definition 2.2 [The state transition due to a physical and a private action]
Let s = <B, G, R> be a state of an agent and a a physical or private action. The state transition due to a is described by the following axiom:

$$< B, G, R> \xrightarrow{a} <B', G', R'>$$

$B' = (B - Bels_a(s)) \cup Bels'_a(s)$ where $Bels_a(s)$ denotes the set of beliefs of the agent at a state s and on which the action a has an impact. This impact corresponds to the application of scalar functions that is, if $Bels_a(s) = \{R_1(...), ..., R_i(...), ..., R_k(...)\}$ where R_i is a relation symbol, then there are functions symbols $f_1, ..., f_i, ..., f_k$ such *that* $Bels'_a(s) = \{R_1(f_1(...)), ..., R_i(f_i(...)), ..., R_k(f_k(...))\}$[1].

- $G'=G$, and
- R' is the result of suppressing a from R i.e., $R=a.R'$ where "." is the operation of concatenation.

For example, in the prey/predators game, *move_up* is a physical action which a predator is able to do. The ground atom *Position (self,* v_1, v_2*)* is an agent's belief which indicates his position on the grid. Let $Bels_{move\_up}(s)$ = { *Position (self,* v_1, v_2*)* } and G a predator's goal (e.g., getting closer to a prey from the western side). Let the predator's role = <*move_up, move_right, move_right*> = R. Then, we have

$$< B, G, R> \xrightarrow{move\_up} <B', G', R'>$$

Where : $B' = (B - \{Position\ (self, v_1, v_2)\}) \cup \{Position(Inc\_Ord(self, v_1, v_2))\} = (B - \{Position\ (self, v_1, v_2)\}) \cup \{Position\ (self, v_1, v_2+1)\}$ such that *Inc_Ord* is a function which increments the second component of the agent coordinates on the grid (i.e., the second term of the relation symbol *Position*); $G'=G$; R' =<*move_right, move_right*>.

Definition 2.3 [The state transition due to a communicative action]
Let s = <B, G, R> be a state of an agent and a a communicative action. The state transition due the action a is described by the following axiom:

$$< B, G, R> \xrightarrow{a} <B', G', R'> \text{ such that:}$$

- If a is an emission action then $B' = B$ else (i.e., a is a reception action) $B' = (B - Bels_a(s)) \cup Bels'_a(s)$.
- $G'=G$, and
- $R=a.R'$.

[1] The application of a function symbol on terms of first-order logic results in possibly different terms -- see the example with the use of the function *Inc_Ord*.

Definition 2.4 [The state transition due to a decision action]
Let s = <*B, G, R*> be a state of an agent and τ a decision action. The state transition due to the execution of a decision action is described by the following axiom:

$$<B, G, R> \xrightarrow{\tau} <B', G, R'> \text{ such that:}$$

- $B' = (B - Bels_\tau(s)) \cup Bels'_\tau(s)$
- $R' = mod(R)$ where *mod* is an update function of the role resulting in the following operations :
  - The generation of an action (physical, private or communicative). In this case the action will be integrated in the role and *mod (R) = Insert (R, a)*. -- *Insert* is a function which integrates the action *a* in the role *R*
  - The suppression of an action (physical, private or communicative). In this case *mod (R) = Delete*(*R*, *a*). -- *Delete* is a function which deprives the role of one of its actions.
  - The repetition of the two operations mentioned above.

2.2 A Multi-Agent System Model

In this section, we deal with the behavioral semantics of a multi-agent system.

Definition 2.5 [A multi-agent system state]
A multi-agent system state is a tuple $<<B_1, G_1, R_1>, ..., <B_i, G_i, R_i>, ..., <B_n, G_n, R_n>, GG>$ where : $<B_i, G_i, R_i>$ is the state of the agent *i*, and *GG* is the global goal of the system such that **C** $(G_1, G_2, ..., G_n) \Rightarrow GG$ where **C** is a relation over the goals of the agents.

Note that **C** is closely related to the system to be modelled. For example, in the prey/predators problem, *GG* is the capture of the prey, G_i is getting closer to the prey from one side (north, south, east or west) and **C** is the conjunction of all the local goals.

Definition 2.6 [the state transition due to an action]
Let $<<B_1, G_1, R_1>, ..., <B_i, G_i, R_i>, ..., <B_n, G_n, R_n>, GG>$ be a state of our system, $A_i = <B_i, G_i, R_i>$ a state of an agent *i* (*i*: 1..n). The state transition due to an action is described by one of the following rules:

i. *a* is a physical or private action.

$$\frac{<B_i, G_i, a.R_i> \xrightarrow{a} <B'_i, G_i, R_i>}{<A_1, ..., A_i, ..., A_n, GG> \xrightarrow{a} <A_1, ..., <B'_i, G_i, R_i>, ..., A_n, GG>}$$

If an agent is ready to execute a physical or private action, then the whole system is ready to execute the same action.

ii. *a* is an emission action, and $\bar{a}$ is a reception action.

$$\frac{<B_i, G_i, a.R_i> \xrightarrow{a} <B_i, G_i, R_i>\ ; \quad <B_j, G_j, \bar{a}.R_j> \xrightarrow{\bar{a}} <B'_j, G_j, R_j>}{<A_1, ..., A_i, ..., A_n, GG> \xrightarrow{a\,\bar{a}} <A_1, ..., <B_i, G_i, R_i>, ..., <B'_j, G_j, R_j>, ..., A_n, GG>}$$

If an agent is ready to send a message a and another agent is ready to receive the same message, then the whole system is ready in this case to execute this interaction. We note it by $a\,\overline{a}$. One can see that in this rule, we model communication in a synchronous way[2].

iii. τ is a decision action.

$$\frac{<B_i, G_i, R_i> \xrightarrow{\ \tau\ } <B'_i, G_i, R'_i>}{<A_1, ..., A_i, ..., A_n, GG> \xrightarrow{\ \tau\ } <A_1, ..., <B'_i, G_i, R'_i>, ..., A_n, GG>}$$

iv. η is a negotiation action.

$$<A_1, ..., A_i, ..., A_n, GG> \xrightarrow{\ \eta\ } <A'_1, ..., A'_i, ..., A'_n, GG'>$$

where $\mathbf{C}\,(G'_1, G'_2, ..., G'_n) \Rightarrow GG'$.

A negotiation action is a finite sequence of communicative actions[3] initiated to update the goals of the agents. Indeed, under certain circumstances, it can be useful for an agent to modify his proper goal. We assume that the global goal can change if the agents perceive that it can't be achieved any longer.

3 Application

In this section, we propose a specification language (let it be $\mathcal{L}_\omega$) with a semantics based on the semantic model presented in the previous section. $\mathcal{L}_\omega$ is multi-modal and linear-time logic.

Definition 3.1 [$\mathcal{L}_\omega$ syntax]

i. The set of first-order logic formulae is included in $\mathcal{L}_\omega$;

ii. if $\varphi, \Psi \in \mathcal{L}_\omega$ then $\square\, \varphi \in \mathcal{L}_\omega$, $\mathbf{O}\, \varphi \in \mathcal{L}_\omega$, $\varphi\, \mu\, \Psi \in \mathcal{L}_\omega$, $\blacksquare\, \varphi \in \mathcal{L}_\omega$, $\bullet\, \varphi \in \mathcal{L}_\omega$, $\varphi\, \mathcal{S}\, \Psi \in \mathcal{L}_\omega$;

iii. if $\varphi, \Psi \in \mathcal{L}_\omega$ then $\neg\, \varphi \in \mathcal{L}_\omega$, $\varphi \vee \Psi \in \mathcal{L}_\omega$;

iv. if v is a variable, $\varphi \in \mathcal{L}_\omega$ then $\exists v\ \varphi \in \mathcal{L}_\omega$;

v. if x is an agent, and $\varphi \in \mathcal{L}_\omega$ then $\mathbf{B}\, x\, \varphi \in \mathcal{L}_\omega$, $\mathbf{G}\, x\, \varphi \in \mathcal{L}_\omega$;

vi. if x is an agent, a an action, $\varphi \in \mathcal{L}_\omega$, and r a role then $ENABLED\ a \in \mathcal{L}_\omega$, $OCCURRED\ a \in \mathcal{L}_\omega$, $AGENT\ x\ a \in \mathcal{L}_\omega$, and $HAS\_ROLE\ x\ r\ \varphi \in \mathcal{L}_\omega$.

Before dealing with the semantics of $\mathcal{L}_\omega$, we give some definitions.

Definition 3.2 [preliminary definitions]

i. Let $\mathbf{M}$ be the interpretation model of $\mathcal{L}_\omega$, $\mathbf{M}$ is a state transition sequence (i.e., $\mathbf{M} \in \Omega$)[4].

---

[2] Actually, agents can also communicate in an asynchronous way.

[3] The communicative actions used to negotiate are different from the usual communicative actions mentioned above. The latters cannot modify the individual goals whereas the formers are designed to update these goals.

[4] An agent system has been presented as by a triplet < S, A, Ω > consisting of a set S of states, a set A of actions, and a set Ω of all possible system executions (see § 2).

$$< \mathbf{M}, s> \models \Box\, \varphi \text{ iff } \forall\, s' \in ACC(s), < \mathbf{M}, s'> \models \varphi$$

$$< \mathbf{M}, s> \models \mathrm{O}\, \varphi \text{ iff } < \mathbf{M}, s' > \models \varphi \text{ with } s \angle s'$$

$$< \mathbf{M}, s> \models \varphi\, \mu\, \Psi \text{ iff } \exists s' \in ACC(s) \,/\, < \mathbf{M}, s' > \models \Psi \text{ and } \forall\, s'' \,(s \angle^* s'' \angle^* s'), < \mathbf{M}, s'' > \models \varphi$$

$$< \mathbf{M}, s> \models \blacksquare\, \varphi \text{ iff } \forall\, s' \,/\, s' \angle^* s, < \mathbf{M}, s' > \models \varphi$$

$$< \mathbf{M}, s> \models \bullet\, \varphi \text{ iff } < \mathbf{M}, s' > \models \varphi \text{ with } s' \angle s$$

$$< \mathbf{M}, s> \models \varphi\, \mathcal{S}\, \Psi \text{ iff } \exists\, s' \,/\, s' \angle^* s, < \mathbf{M}, s' > \models \Psi, \text{ and } \forall\, s'' \,(s' \angle^* s'' \angle^* s), < \mathbf{M}, s'' > \models \varphi$$

Fig. 2. Semantics of temporal formulae

ii. Let $ST(\mathbf{M})$ the set of states of **M**. We define the immediate successor of a state $s \in ST(\mathbf{M})$ by the relation $\angle$ such that : $\forall\, s, s' \in ST(\mathbf{M})$, $s \angle s'$ iff there is a state transition $s \xrightarrow{a} s'$ in **M**. We use $\angle^*$ to denote the reflexive and transitive closure of $\angle$.

iii. The set of accessible states from a given state s of **M** is defined as follows : $ACC(s) = \{e \in ST(\mathbf{M}) \,/\, s \angle^* e\}$.

The semantics of temporal formulae is given in a state sequence M, with respect to a current time point s of M (see figure 2).

With respect to an action, it may be interesting to reason about its occurrence. Let a be an action, (*ENABLED a*) means that the action a is ready to be executed. (*OCCURRED a*) means that the action a has just been executed. The semantics of the enabling as well as the occurrence of an action are given in a sequence **M** and at a current time point s of **M**.

$<\mathbf{M}, s> \models ENABLED\ a$ iff there is a state transition $s \xrightarrow{a} s'$ in **M**.

$<\mathbf{M}, s> \models\ OCCURRED\ a$ iff there is a state transition $s' \xrightarrow{a} s$ in **M**.

The fact that an action a is of an agent x, is denoted by the formula (*Agent x a*) which semantics is given in a state sequence **M** as follows:

$<\mathbf{M}> \models AGENT\ x\ a$ iff there is $s \xrightarrow{a} s'$ in **M**, and a is an action of x's role.

We use (*OCCURRED x a*) to denote that the agent x has just executed the action a.
$(OCCURRED\ x\ a) \equiv (AGENT\ x\ a) \wedge (OCCURRED\ \ a)$.

Each agent has a role denoting its organisational component. A role is modelled as mentioned above by a sequence of actions. The formula (*PERFORM x r*) means that agent x achieves his role r and its semantics is given by the following rule:

$< \mathbf{M}, s> \models PERFORM\ x\ r$ iff $\exists\, s' \,/\, s \angle^* s'$, and $\forall a$ in r, $<\mathbf{M}, s'> \models OCCURRED\ x\ a$

We use the formula (*HAS_ROLE x r* φ) to denote that an agent *x* has a role *r* to achieve φ. The semantics of this formula is given in **M** and at a current time point *s*.

$$< \mathbf{M}, s> \models HAS\_ROLE\ x\ r\ \varphi \quad \text{iff} \quad < \mathbf{M}, s> \models \Box\, (\blacklozenge (PERFORM\ x\ r) \Rightarrow \varphi)$$

We're also interested in knowing whether an agent would succeed to achieve φ. Let (*SUCCEEDS x* φ) be a formula denoting the success of an agent *x* to achieve φ. Formally: (*SUCCEEDS x* φ) ≡ ∃ *r* (*HAS_ROLE x r* φ)

Beliefs are modelled by a finite set of ground atoms associated to each state. Similarly, goals are represented by a finite set of first-order logic formulae associated to each state. A belief formula is B x φ and means that agent x has φ as a belief. A goal formula is G x φ and means that agent x has φ as a goal. The formal semantics of beliefs and goals are given in a state transition sequence M with respect to a current time point s of M.

$$<\mathbf{M}, s> \models (\mathbf{B}\ x\ \varphi) \quad \text{iff} \quad \varphi \in B_x \text{ (beliefs of agent } x)$$
$$<\mathbf{M}, s> \models (\mathbf{G}\ x\ \varphi) \quad \text{iff} \quad \varphi \in G_x \text{ (goals of agent } x)$$

According to the semantics of beliefs, a formula is said to be believed at a point time *s* by an agent if and only if it belongs to the set of his beliefs at this time point. Similarly, an agent is said having φ as a goal if and only if it belongs to the set of his goals at the time point *s*.

4 Towards a Unifying Approach

Transitions systems have been initially introduced to define the semantics of parallelism. Other models have been also proposed among which possible world semantics, Petri nets, Milner's process calculus.

Most previous work on multi-agent systems' theories uses possible world semantics (for example [9, 10]). The possible worlds model for logics of knowledge and beliefs was originally proposed by Hintikka [4], and is now most commonly formulated in a normal modal logic using the techniques developed by Kripke [11]. Each world represents one state of affairs considered possible, given what an agent knows. Hintikka coined the term epistemic alternatives to describe the worlds possible given one's beliefs. A Kripke structure is a tuple <*W*, π, *R*>, where *W* is a set of possible worlds (considered as *states*), $\pi : \phi \rightarrow 2^W$ is an interpretation that associates with each propositional constant[5] in ϕ a set of states in *W*, and *R* is a binary relation on *W*, that is, a set of pairs of elements of *W*. *R* is considered as a *state transition*. It is intended to capture the possibility relation: $(s, s') \in R$ if the state *s'* is possible given the information in the state *s* (see figure 3 where *p* is an atom and w_i is a possible world).

Petri nets are particular cases of transition systems where the states of a Petri net are its markings, and the state transitions are implemented with the simultaneous (or not) firing, according to the semantics given to the net, of net transitions.

[5] In this example, we wish to reason about worlds that can be described in terms of a nonempty finite set ϕ of propositional constants.

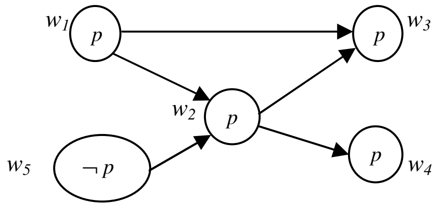


Fig. 3. The Kripke structure

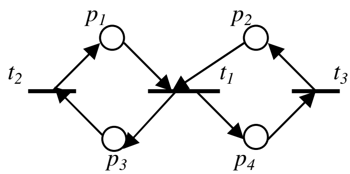


Fig. 4. An example of a Petri net

For example, let the Petri net of figure 4. It has three transitions t1, t2, t3 and four places p1, p2, p3, p4. If the initial marking is m0 = (1, 1, 0, 0) then the set of reachable markings is {m0, m_1, m_2, m_3} such that m_1 = (0, 0, 1, 1), m_2 = (1, 0, 0, 1), and m_3 = (0, 1, 1, 0). We reach a marking by firing either of the three transitions or by firing simultaneously transitions t_2 and t_3.

Accordingly, we obtain the following transition system.

$$m_0 \xrightarrow{t_1} m_1 \qquad m_1 \xrightarrow{t_3} m_3 \qquad m_3 \xrightarrow{t_2} m_0$$

$$m_1 \xrightarrow{t_2} m_2 \qquad m_2 \xrightarrow{t_3} m_0 \qquad m_1 \xrightarrow{t_2,\, t_3} m_0$$ [6]

Examples of the use of Petri nets in agent based studies include Chainbi and Cranefield works [12], [13]. Process calculus and other models amount to specify a set of states, and a set of transitions between those states. For example, in process algebras (which are inspired of Milner's calculus communicating systems -CCS-), transitions are defined by the calculus operational semantics which indicates how and under which conditions, a calculus term can be transformed into another term. For instance, consider a two elements buffer agent. When this buffer contains one element, either of two actions is possible: an *in* action accepting another element or an *out* action to dispense its stored element. The behavior of a two elements buffer can be defined in CCS as follows:

Buffer2 = in.Buffer2'
Buffer2' = in.Buffer2'' + out.Buffer2
Buffer2'' = out.Buffer2'

[6] This state transition will not be considered if we give the Petri net a semantics preventing the simultaneous firing of transitions.

Notice that each equation defines an agent state. In the *Buffer2'* state, either the *in* or the *out* action is possible. In the *Buffer2''* state, the buffer is full and only an *out* action is possible. Examples of the use of process algebras in agent community are Kinny and Sumpter et al. studies [14, 15].

Previously, Vasconcelos has used finite-state machines to describe the interactions among agents in the context of electronic institutions [16]. In his proposal, the author doesn't deal with the behavioral semantics of individual agents. He focused on the representation of global protocols that is all possible interactions among components of a multi-agent system from a global perspective.

Labeled transition systems have also been used by Gelfond et al. to describe the effects of actions [17]. Nevertheless, their work was not dedicated to agent systems. Indeed, the authors propose formal models of parts of the natural language that are used for talking about the effects of actions, central to their approach is the concept of transition systems. Besides, they focus their work on propositional transition systems (i.e., transitions systems where state variables are logical propositions).

5 Conclusion

In this paper, we have explored the use of labeled transitions systems as a unifying approach to model multi-agent systems. The use of such formalism has been basically justified by the following facts.

Firstly, it can be used as a behavioral model as well as a basis for the semantics of an underlying specification language to a multi-agent system.

Secondly, notions that appear with different and complex forms in the literature such as possible world semantics, Petri nets, process calculus can be described in the same formal framework namely transition systems. We have shown that these notions amount to specify a set of states, and a set of transitions between these states. Consequently, agent technology can take advantage of the results related to transition systems. For example, the comparison between two multi-agent systems becomes possible. That is what we envision to investigate in future work. Indeed, two systems are equivalent if they have the same semantics i.e., we can draw from the describing text (the syntax), the same information about their behavior (the semantics). Clearly, the equivalence should be defined formally, but we think that the literature about the equivalence of transition systems is very helpful at this level (for example [18]). Besides, it becomes also possible to check that a system implementation respect a specification if its semantics is expressed in the same formalism (namely transition systems) as another system whose properties are already known. This is possible because of the existence of an equivalence between the underlying transition systems of the considered systems.

References

1. Weiss, G.: Multiagent Systems A Modern approach To Distributed Artificial Intelligence. MIT Press, Cambridge (1999)
2. Jennings, N.R., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research Development. Autonomous Agents and Multi-Agent Systems 1, 7–38 (1998)

3. Arnold, A.: Systèmes de transitions finis et sémantique des processus communicants. Techniques et Sciences Informatiques Journal 9(3), 193–216 (1990)
4. Hintikka, J.: Knowledge and Belief. Cornell University Press, Ithaca (1962)
5. Murata, T.: Petri Nets: Properties, Analysis and Applications. Proc. IEEE 77(4) (April 1989)
6. Milner, R.: A Calculus of Communication Systems. LNCS, vol. 92. Springer, Heidelberg (1980)
7. Chainbi, W.: An Ontology Based Multi-Agent System Conceptual Model. Special Issue of the International Journal of Computer Applications in Technology 31(1/2), 35–44 (2008)
8. Chainbi, W.: Belief-Goal-Role agents versus BDI agents. In: 2nd Pacific Rim International conference on Multiagents (PRIMA 1999), Kyoto (1999)
9. Wooldridge, M.: This is MYWORLD: The Logic of An Agent-Oriented DAI Testbed. In: Wooldridge, M.J., Jennings, N.R. (eds.) ECAI 1994 and ATAL 1994. LNCS(LNAI), vol. 890. Springer, Heidelberg (1995)
10. Cohen, P.R., Levesque, H.J.: Intention is Choice with Commitment. Artificial Intelligence 42, 213–261 (1990)
11. Kripke, S.: Semantical analysis of modal logic. Zeitshcrift für Mathematische Logik und Grundlagen der Mathematik 9, 67–96 (1963)
12. Chainbi, W.: Multi-Agent Systems: A Petri Net with Objects Based Approach. In: Proc. of the 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Beijing, China, September 20-24 (2004)
13. Cranefield, S., Purvis, M.: Ontologies for Interaction Protocols. In: Proc. of the 2nd International Workshop on Ontologies in Agent Systems, AAMAS 2002, Bologna, Italy, July 15-19 (2002)
14. Kinny, D.: The Ψ calculus: An algebraic agent language. In: Meyer, J.-J.C., Tambe, M. (eds.) ATAL 2001. LNCS, vol. 2333, pp. 32–50. Springer, Heidelberg (2001)
15. Sumpter, D.J.T., Blanchard, G.B., Broomhead, D.S.: Ants and Agents: a Process Algebra Approach to Modeling Ant Colony Behavior. Bulletin of Mathematical Biology 63, 961–980 (2001)
16. Vasconcelos, W.W.: Logic Based Electronic Institutions. In: Leite, J., Omicini, A., Sterling, L., Torroni, P. (eds.) DALT 2003. LNCS(LNAI), vol. 2990, pp. 221–242. Springer, Heidelberg (2004)
17. Gelfond, M., Lifschitz, V.: Action Languages. Electronic Transactions on Artificial Intelligence 2, 193–210 (1998)
18. Dicky, A., Arnold, A.: Equivalencies and Preorders of Transition Systems. In: Borzyszkowski, A.M., Sokolowski, S. (eds.) MFCS 1993. LNCS, vol. 711, pp. 20–31. Springer, Heidelberg (1993)

A Multi-Agent Framework for Storage and Retrieval of Documents from Distributed XML Collections

Jakub Gieryluk, Radosław Katarzyniak, Przemysław Maciejczyk, Magdalena Sendal, and Grzegorz Skorupa

Wrocław University of Technology, Institute of Informatics,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{magdalena.sendal,jakub.gieryluk,
przemyslaw.maciejczyk}@student.pwr.wroc.pl,
{radoslaw.katarzyniak,grzegorz.skorupa}@pwr.wroc.pl

Abstract. A distributed and dynamic multi-agent system for integrated performing of gathering, processing, and retrieval of highly heterogeneous XML documents' collections is presented. Collections of XML documents are assumed to be distributed among nodes of a network. The management process consists of a few steps: documents in each node are clustered, clusters' representatives are computed and stored in metadata repository, queries are confronted with metadata repository, distributed retrieval is carried out over relevant clusters, retrieval results are integrated and final answer is presented to an end user. Major functions of such strategy were designed and implemented on JADE platform. Actual implementations of the proposed multi-agent system can vary, depending on applied models of XML documents' clustering, indexing and retrieval.

Keywords: multi-agent system, XML documents' collection, distributed database, distributed retrieval, meta-data repository.

1 Introduction

In many applications XML is used as a tool for the representation and exchange of semi-structured data. Many examples of effective XML applications can be found in such fields as e-commerce and e-business, collaborative authoring of large electronic documents, exchange of messages in SOAP, and management of large-scale documents' directories. Depending on the nature of application domain various strategies for XML documents' management and processing are needed, including the need for adequate approaches to managing distributed collections of documents and semantic mapping between distributed sources of knowledge. In many of them autonomy of documents' management plays an important role due to serious need for intensive human-independent support [6]. In a natural way multi-agent technology suits to achieve this target.

The aim of this work is to design and implement a prototype of a multi-agent system that is tailored to actual implementation of a certain predefined strategy for

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 141–150, 2009.

managing of XML documents. The set of management tasks assumed for this strategy consists of autonomous gathering, autonomous processing (clustering) and autonomous retrieving of XML documents. The XML documents processed in the system can be highly heterogeneous and stored in a distributed database, accessible to end users via network. Our prototype of a multi-agent system is supposed to fulfill these management functions that are mutual for all instantiations of considered strategy for XML documents' management.

It is further assumed that the distributed collection of XML documents is organized into clusters and each cluster is located in a particular node and consists of structurally similar documents. All cluster computed by dedicated agents are represented by their metadata structures. Such meta-representatives are stored and processed in a special metadata repository. The whole system is assumed to make it possible for each end user to represent his/her own information need as a pattern (query), confront this pattern with the content of metadata repository, locate clusters relevant to the pattern, distribute the pattern over the nodes with relevant clusters, carry out adequate search, integrate partial results into one answer, and present this answer to the end user.

The target of this work is to implement these functions of our multi-agent system that make it possible to handle the above defined management meta-actions regardless of possible models for processing of stored XML documents. For example, it is assumed in our work that the choice of particular models for the evaluation of document - pattern similarity and pattern - cluster similarity, as well as algorithms for the computation of cluster and clusters' representatives, is a further matter of each individual implementation and agents. The choice should depend on specific requirements resulting from the nature of a target domain. *The prototype of our system has been implemented and tested on JADE platform* [1,3,11].

The further organization of the text is as follows. In section 2 the architecture of multi-agent system is described; in particular, main classes of agents are defined and their major functions named. Section 3 contains a semi-formal model for the main components of our multi-agent system, the proposed knowledge base, and the flow of control realized within multi-agent population. In section 4 the implementation of a prototype is briefly reported.

2 System Architecture

The proposed system, as shown in Figure 1, consists of many document sources, meta-data repository and an access point for a user, from which he/she can perform searches and view search results.

The document sources are spread among different computers and consist of many XML documents containing various types of information. Each document source is managed by two agents: *meta-data creator* and *document search performer*. The *meta-data creator* agent is responsible for creation of meta-data representing groups of similar documents available in particular source. The *document search performer* agent performs a search in order to find relevant XML documents in given source.

The meta-data repository gathers information about existing sources and higher level information (meta-data) about types of documents available on these sources.

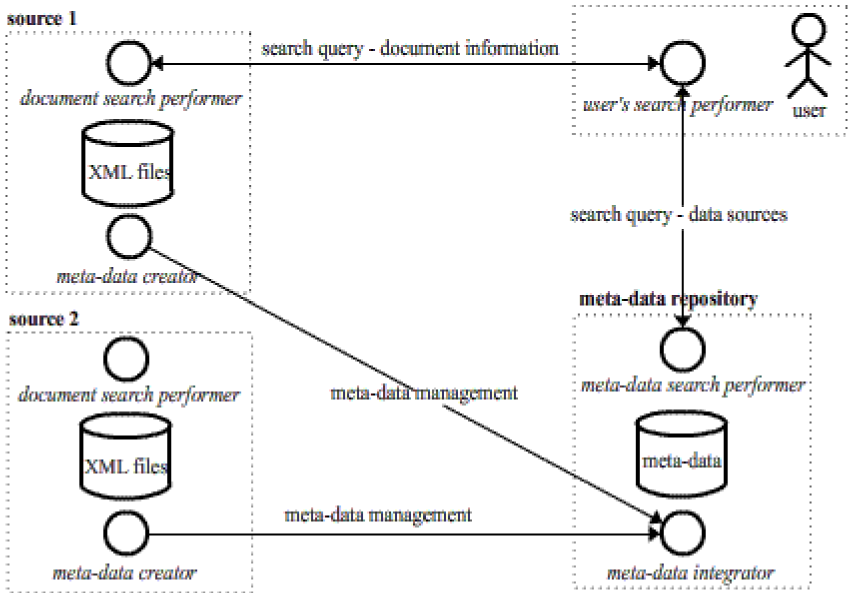


Fig. 1. System architecture and basic tasks. Agents are marked with circles.

Much like document source, the meta-data repository is managed by two agents: *meta-data integrator* and *meta-data search performer*. The *meta-data integrator* agent locates existing sources, gathers information created by meta-data agents in sources and integrates the higher level data in order to create sufficient information for *meta-data search performer*. The *meta-data search performer* agent performs an initial search among created meta-data about documents and sources in order to obtain a list of sources that can contain documents of user interest.

In the access point for a user there is one agent called *user's search performer*. The agent provides an interface where user can input search queries and view search result pages. The *user's search performer agent* searches for relevant documents in two steps. At first it sends query created by user to *meta-data search performer* in order to obtain a list of applicable sources. Then it sends the same query to *document search performer* agents located in found sources in order to gather information about relevant documents. In the end the agent integrates received results and displays them to the user as one list containing basic information about documents.

3 System Workflow

The behavior of the system can be described in two separate workflows: document searching related workflow and meta-data acquisition/update workflow.

User's, *meta-data* and *documents search performer* agents take part in the first workflow. The *user's search performer* is responsible for sending user's queries to *meta-data search performer* in meta-data repository and *document search performers* in sources. The *meta-data search performer* searches for sources containing documents that may be of user interest. Each *document search performer* searches for XML documents in its source. The workflow and agent tasks are further described in “Document Searching” paragraph.

The meta-data acquisition/update workflow is managed by *meta-data integrator* agent in meta-data repository and *meta-data creator* agents in sources. The *meta-data creator* agents are responsible for clustering documents containing similar contents. The *meta-data integrator* agent gathers clustering information from sources and integrates it in order to make it useful for *meta-data search performer*. The workflow is explained in detail in “Sources Meta-Data Creation” and “Meta-Data Integration in Meta-Data Repository” paragraphs.

3.1 Sources Meta-Data Creation

In each source *meta-data creator* agent prepares sufficient meta-data about documents available in its location. The prepared meta-data is then delivered to *meta-data integrator* agent in meta-data repository.

The meta-data creation process consists of two phases. The first phase is document clustering, that results in a partition of available documents’ set, based on some similarity measure [7,16]. In the second phase agent creates representative for each cluster. To illustrate the function we assume that the representative is an XML document containing nodes without any text information. A set of representatives for all clusters is the resulting meta-data repository.

Below major elements of this process are captured in a semi-formal system of symbols. Let us assume that *meta-data creator* agent in a node is given a set of XML documents:

$$D=\{\mathrm{d}_1, \mathrm{d}_2, \ldots, \mathrm{d}_\mathrm{n}\} \tag{1}$$

Using clustering algorithm this agent creates a partition D_1, D_2, …, D_m of D:

$$D = D_1 \cup D_2 \cup \ldots \cup D_m\,, \quad D_i \subset D\,, \quad D_i \cap D_j = \varnothing \text{ for } i \neq j\,,\ i,j = 1..n. \tag{2}$$

When this partition is ready, the agent has to create representatives r_i for each sets D_i. Such representatives can be XML trees built from labeled nodes. Let the set R be a meta-data collection extracted from documents stored in a particular node N_i, i=1…k:

$$R=\{\mathrm{r}_1, \mathrm{r}_2, \ldots, \mathrm{r}_\mathrm{m}\} \tag{3}$$

This collection is further sent to *meta-data integrator* agent. The clustering and creation of representatives can be realized in various ways, e.g. by applying the algorithm given in [2]. The choice depends on the domain and specific needs.

3.2 Meta-Data Integration in Meta-Data Repository

Meta-data integrator agent is responsible for discovery of new documents sources, and collecting of meta-data incoming from various nodes of the distributed XML collection. In the simplest case this agent can collect simple lists of cluster representatives and references to their location. A more advanced agent can be proposed in order to determine similar clusters of documents in different sources (nodes) and

Table 1. *meta-data creator* agent role description

meta-data creator
Placed in: documents' source
Responsibilities: Clustering available XML documents into similar groups; Creating representatives (meta-data) for each cluster.
Communicates with: *meta-data integrator* agent to pass generated meta-data
Required algorithms:
XML documents clustering **(DC)** – create clustering of available XML documents. Input: set *D* (see eq. 1); Output: partition of that set *D* (see eq. 2).
XML cluster representative creating **(CRC)** – the algorithm has to create new representatives after the clustering is done. Input: partition of set *D* (see eq. 2); Output: representatives (see set *R* in eq. 3).
XML documents clustering update **(DCU)** – algorithm run when a document is added or removed. The algorithm can change partition of set *D* and/or representatives from set *R* (see eq. 3).

merge them into generalized groups. The scope of this and similar implemented functionality depends strictly on the target domain's nature and needs.

The simplest case of *meta-data integrator* agent can be described as follows. The agent locates sources S_i located in nodes N_i, i=1…k, respectively:

$$S=\{S_1, S_2, \ldots, S_k\} \tag{4}$$

For each i=1..k, agent asks the node N_i for related set R_i defined in (3). In consequence, this agent becomes responsible for managing the following relation *SR*:

$$SR \subset S \times \overline{R}\ ,\ \overline{R} = R_1 \cup R_2 \cup \ldots \cup R_k\ ,\ SR = \{(S_i, r) : S_i \in S \wedge r \in R_i\}\ . \tag{5}$$

The set $\overline{R}$ contains all representatives and the relation *SR* describes what sources are described by what representatives. The sets *S*, $\overline{R}$ and the relation *SR* are used by *meta-data search performer* agent to find sources containing documents of user's interest.

Table 2. *meta-data integrator* agent role description

meta-data integrator
Placed in: meta-data repository
Responsibilities: Discovery of new document sources; Collecting meta-data from document sources; Integrating the collected meta-data (handling relation *SR* in eq. 5)
Communicates with: *meta-data creator* agents in sources.
Required algorithms:
Meta-data integration **(MDI)** – the algorithm can modify the received representatives; i.e. the algorithm can merge some representatives.
input: set $\overline{R}$ (see eq. 5); output: modified representatives set.
Meta-data update **(MDU)** – what should be done when set $\overline{R}$ changes. In particular when some representatives are added or removed. Input: new set $\overline{R}$ together with information about changed representatives; Output: changed representatives set

3.3 Document Searching

At the beginning user inputs his/her search query q using interface provided by *user's search performer*. It can be assumed that the query is an XML document. The system is supposed to find documents most similar to the supplied query.

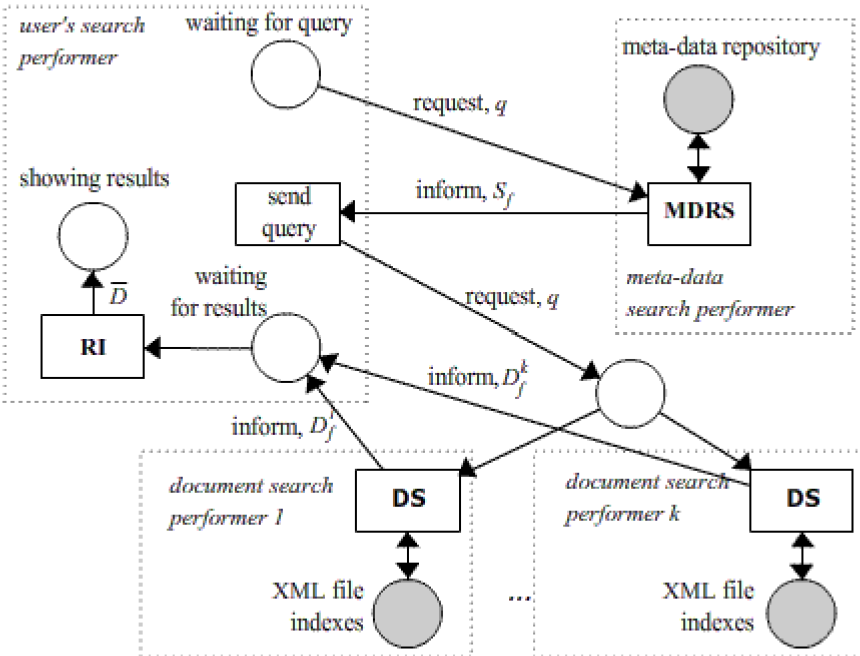


Fig. 2. Document searching related workflow as Petri Net. Data sources are marked as states with gray filling. Workflow starts at "waiting for query" state. Bold texts are acronym names of algorithms presented later in tables 1, 2 and 3.

After receiving the query q, *user's search performer* asks *meta-data search performer* for relevant sources. *The meta-data search performer* compares the query q with representatives from $\overline{R}$ and retrieves the ones that are most similar. Most often representatives are built only from nodes. Therefore similarity can be measured using one of well known tree comparison algorithms [4,5,13]. Let $R_f \subset \overline{R}$ denote a set of retrieved representatives. Later the agent determines locations of sources S_i from the available set S. Obviously, a the location of $S_i \in S$ is determined if there exists a representative $r \in R_f$ such that $r \in R_i$. The resulting set S_f of relevant sources is:

$$S_f = \{s : (s,r) \in SR \wedge r \in R_f\} \tag{6}$$

The set S_f is sent to the *user's search performer* as a response to the received query q. Then the user's agent sends the query q to each of the sources given in S_f. The query is received by *document search performer* agents in each source. A *document search performer* agent performs the local search. Tree [4,5,13], text [9,10] and/or XML [7,16] similarity measures can be used, depending on the domain of documents.

Table 3. *user's search performer*, *meta-data search performer* and *document search performer* role descriptions

user's search performer
Placed in: user's access point **Responsibilities:** Communication with user; Sending search queries to meta-data repository and sources; Search results integration. **Communicates with:** *meta-data search performer* agent to acquire list of relevant document sources; *document search performer* agents to acquire lists of relevant documents. **Required algorithms:** Result integration **(RI)** – integrate results obtained from different sources into one list of results ordered by relevancy. Input: $\overline{D_f}$ (see eq. 7) together with document relevancy information. Output: one list $\overline{D}$ (see eq. 8).
meta-data search performer
Placed in: meta-data repository **Responsibilities:** Searching for relevant document sources based on available meta-data prepared by *meta-data integrator* agent. **Communicates with:** *user's search performer* agent to send him found relevant document sources based on received search query. **Required algorithms:** Meta-data representatives search **(MDRS)** – the algorithm has to compare user's query with available representatives, and pick those best matching. Input: users query q; Output: set of R_f. The resulting set of representatives is then used to pick matching sources set S_f (see eq. 6).
document search performer
Placed in: documents' source **Responsibilities:** Indexing XML document to perform searches quickly; Searching for relevant XML documents. **Communicates with:** *user's search performer* agent to send him list of available relevant documents based on received search query. **Required algorithms:** XML documents index creation **(DIC)** – creates some indexing of XML documents in given source to speed up search process. Input: set D (see eq. 1); Output: indexing of set D for search purposes XML documents index update **(DIU)** – updates the index when a document is changed, added or removed. input: information about changed documents; output: updated indexing Document searching **(DS)** – uses the created index to search for relevant documents based on a given query. Input: user's query q; Output D_f

When the local search is completed the agent retrieves the most relevant documents $D_f \subset D$ and returns them to the *user's search performer* agent.

The *user's search performer* agent collects results as a collection:

$$\overline{D}_f = D_f^1 \cup D_f^2 \cup ... \cup D_f^g, \quad g\text{=card}(S_f). \tag{7}$$

Later the agent has to integrate the results $\overline{D}_f$ to create an ordered list of documents. The way this integration is done, depends on available information sent from sources. At this stage of the process retrieval results are characterized by normalized relevancy, what makes their integration and sorting very easy. The obvious disadvantage of such solution is that results from *document search performer* agents tending to give high relevancy more frequently, are favored.

The resulting list is captured by the following symbol:

$$\overline{D} = (\text{d}_1, \text{d}_2, ..., \text{d}_w), \quad \text{where } w = card(\overline{D}_f) \text{ and } \text{d}_j \in \overline{D}_f \tag{8}$$

where the relevancy of d_i is greater or equal to the relevancy of d_j, $i<j$.

4 Implementation

We implemented the system described above using JADE [1,3,11]. JADE is a set of ready packages written in JAVA. JADE implements FIPA specifications [11], which standardizes many aspects of multi-agent systems: agent communication, agent management and agent discovery. The usage of JADE simplified the system creation, first of all due to pre-implemented communication protocols and aided system distribution.

According to FIPA, agents can register and find services that allow easy discovery of their functionalities. The usage of services allows the system to be dynamic, too. Document sources can be added and removed in real-time without a need to restart or re-implement of the system. Agents in sources offer appropriate services. The *meta-data integrator* agent periodically searches for new or removed services and changes in existing ones. When a change is detected the agent updates meta-data information about it.

For communication our agents use standard FIPA protocols. The FIPA Request Interaction Protocol was used most of the time.

The prototype system was designed to allow quick and easy implementation of each of proposed agents. Basically, thanks to our JADE-based system, agent's creation process is reduced to implementation of algorithms responsible for XML documents or meta-data management described in tables 1, 2 and 3.

Agent designer is free to decide what algorithms to use. The usage of particular algorithms depends also on characteristics of XML documents available in the sources.

Current implementation covers mutual functions of the system, which need to be realized in each complete implementation. These functions are: interaction between agents, common behaviors and service discovery and management. All of them can be treated as meta-actions that control particular algorithms from tables 1, 2 and 3.

Our first implementation does not cover most of the algorithms because their analysis was not the case. We only wanted to see if the system works as predicted on communication and integration level. First of all, for the sake of simplicity, we assumed that XML documents changes don't happen and data is static, because of that we didn't have to implement all "update" algorithms. We used a few simple artificially created XML documents for our testing. Clustering and representatives were prepared manually by human (DC and CRC algorithms). The *meta-data integrator* agent simply gathered representatives from sources and merged them into one list without any modifications (MDI algorithm). There was no special XML documents indexing (DIC algorithm), the *document search performer* agent performed search by direct comparison of source documents with a query. Similarly did the *meta-data search performer* when comparing representatives with a query (MDRS). All XML document comparisons were performed using tree alignment measure [4]. The *user's search performer* agent integrated results by merging the list of received search results and sorting it by relevancy parameter given directly from document sources (Result integration algorithm).

We are planning to explore algorithm related elements in the future research to see how the proposed system deals with different types of XML documents and what algorithms are suited best for different situations.

5 Summary

In this paper we reported partial results from a project within which a generalized multi-agent system for managing distributed collection of XML documents was designed and its prototype was implemented. The system offers ready solutions for all common features, including agent communication, collections distribution, meta-data acquisition and dynamic document sources and collections changes. Hence the system can be treated as a framework and quickly re-implemented in order to manage other types of documents with a usage of different algorithms.

The proposed solution offers joined search functionality within many data sources without the need of strong unification of document indexing process or searching methods. Each source is handled autonomously by its agents. The usage of agent technology allows almost fully independent sources management. The sources integration is done using meta-data repository, where crucial information is gathered and search queries are initially analyzed. The introductory search process handled by agent in meta-data repository makes system at least partially load resistant, because the query is later sent only to a constrained list of sources.

Although the simplest algorithms were used, the system works as predicted. We used small collections of handmade XML documents to perform searches. Most of the time only sources containing relevant documents are queried, while sources without applicable data are omitted. Because the prototype system can be used with various types of documents and algorithms, it is to be treated as a base for further research in this area. In the nearest future we are planning to analyze, choose and implement required algorithms described in tables 1, 2 and 3. In the future experiments the system will be applied to at least two different collections of real-life XML documents. A more through-system behavior analysis is planned.

Due to the assumed generality of the proposed framework we will also consider in our project other types of documents e.g. full text documents with keywords, email messages or shop product lists.

References

1. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE - A white paper. Exp. 3(3) (2003), http://exp.telecomitalialab.com
2. Francesca, F., Gordano, G., Ortale, R., Tagarelli, A.: Distance-Based Clustering of XML Documents. In: Proc. ECML/PKDD 2003, pp. 75–78 (2003)
3. Java Agent DEvelopment Framework (JADE), http://jade.tilab.com
4. Jiang, T., Wang, L., Zhang, K.: Alignment of Trees – an Alternative to Tree Edit. Theoretical Computer Science 143(1), 137–148 (1995)
5. Klein, P.N.: Computing the Edit-Distance Between Un-rooted Ordered Trees. In: Bilardi, G., Pietracaprina, A., Italiano, G.F., Pucci, G. (eds.) ESA 1998. LNCS, vol. 1461, pp. 91–102. Springer, Heidelberg (1998)
6. Koloniari, G., Pitoura, E.: Peer-to-Peer Management of XML Data: Issues and Research Challenges. SIGMOD Record 32(2), 6–17 (2005)
7. Long, J., Schwartz, D.G., Stoecklin, S.: An XML Distance Measure. In: Proc. 2005 International Conference on Data Mining (2005)
8. Niermann, A., Jagadish, H.V.: Evaluating Structural Similarity in XML Documents. In: Proc. 5th Int. Workshop on the Web and Databases, pp. 61–66 (2002)
9. Ro, J.S.: An evaluation of the applicability of ranking algorithms to improve the effectiveness of full-text retrieval. I. On effectiveness of full text-retrieval. Journal of the American Society for Information Science 39(2), 73–78 (1988)
10. Ro, J.S.: An evaluation of the applicability of ranking algorithms to improve the effectiveness of full-text retrieval. II. On the effectiveness of ranking algorithms on full-text retrieval. Journal of the American Society of Information Science 39(3), 147–160 (1988)
11. The Foundation for Intelligent Physical Agents (FIPA), http://fipa.org
12. Touzet, H.: Tree edit distance with gaps. Information Processing Letters 85(3), 123–129 (2003)
13. Zhang, K., Shasha, D.: Simple Fast Algorithms for Editing Distance Between Trees and Related Problems. SIAM J. Computing 18(6), 1245–1262 (1989)
14. Zhang, K., Shasha, D., Wang, J.T.L.: Approximate Tree Matching in the Presence of Variable Length Don't Cares. Journal of Algorithms 16(1), 33–66 (1994)
15. Zhang, K.: Algorithms for the Constrained Editing Distance Between Ordered Labeled Trees and Related Problems. Pattern Recognition 28(3), 463–474 (1995)
16. Zhongping, Z., Rong, L., Shunliang, C., Yangyong, Z.: Similarity Metric for XML Documents. In: Workshop on Knowledge and Experience Management (2003)

Agent Technology for Information Retrieval in Internet

Marcin Maleszka, Bernadetta Mianowska, and Ngoc Thanh Nguyen

Wroclaw University of Technology,
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
{Marcin.Maleszka,bmianowska}@student.pwr.wroc.pl,
Ngoc-Thanh.Nguyen@pwr.wroc.pl

Abstract. Searching for useful information is nowadays difficult because of the information overload problem in a global network consisting of heterogeneous sources. Intelligent information agents, or multiagent systems consisting of those, are one of the solutions to this problem. This paper is a short case study of existing information retrieval systems based on agent approaches. User profiles and searching issues are two main issues described and a functional categorization of independent profiling systems or independent searching systems is proposed, as well as a short description of a hybrid profiling and searching system.

1 Introduction

In today's Internet, a global hypertext network of hundreds of millions of distinct IPs, virtually any information is accessible. However, most problems connected with information retrieval are caused by information overload. Using indexation and clusterization algorithms or other methods to describe collected documents one can find interesting documents in shorter time and on that basis many systems to personalize searching process have been created.

Agent approach to information retrieval arose from more general intelligent agent systems over a decade ago and is now an important part of both areas. Main properties defining an agent are its autonomy from its user, flexibility (taking initiative in a pro-active way, reactively responding to user needs and social interaction with other agents), being located inside an environment it is working with.

In a short few years, many ideas to solve the problem of information retrieval in the Internet have arisen. The World Wide Web is treated as a searching tool - huge databases index the Web using vector space model based on Term Frequency and Inverse Document Frequency (TF*IDF) algorithm [7]. Lycos and WebCrawler are examples of such system, but a huge disadvantage of them is the size of the created database. These algorithms recommend some documents based on calculating some statistic of terms in profile and documents.

Some of the alternate ideas are to create meta-search services. Those provide its user a layer of abstraction over multiple search tools and serve as a Web search interface [7], but may also result in ineffective and wasteful competition for network resources with considerable communication cost. To solve this problem [7] proposes neural network multi-agent system.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 151–162, 2009.

Many approaches to information retrieval using agent technology with genetic algorithms may also be found. In [23] an intelligent Internet searching agent is proposed, based on automatic indexing and hybrid simulated annealing. A bit different approach is to create a search assistant which will be learning how and where to search effectively for information connected with some categories [21]. An agent can dynamically adapt to individual users and can communicate with other agents (groups of agents) connected with group of users.

This paper addresses the issue of information retrieval, describing various tools to overcome it, mainly from the agent systems area. Those are divided depending on their main functional area, be it human-orientation or searching methods. The rest of paper is organized as follows: Section 2 describes agent approach to information retrieval, in Section 3 user profile is presented in many aspects and finally Section 4 focuses on search related problems. Approach combining those is presented as proposition of personalizing agent system in Section 5.

2 Agent System for Information Retrieval

For classification purposes, we divided the existing approaches and arranged them in a tree-like hierarchy, as presented in Fig.1. Such categorization allows one to easily qualify any new system and compare it to most similar ones.

For the basic division, we chose to differentiate between agent systems more focused on personalizing the user profile, agent systems focused on the actual search techniques and not dependent of the user, and hybrid approaches that tend to combine both for better results. Personalization can be obtained by learning based on group of users with common behavior or only on single user's own activities. Searching can be on the highest level divided into three categories: crawling; meta searching, that is selecting the proper information source and combining the results from multiple sources; and single source search, although this one usually works more as a collaborative personalization system. One may note that ideal systems would be the hybrid systems group, which would not only be good at personalizing the user query by filtering or recommendation based on his preferences, but also would use advanced search tactics reinforced possibly by indexing the web or mobility, to present the user with best information in relatively shortest time. Unfortunately such systems are highly complicated and so far do not attain that potential.

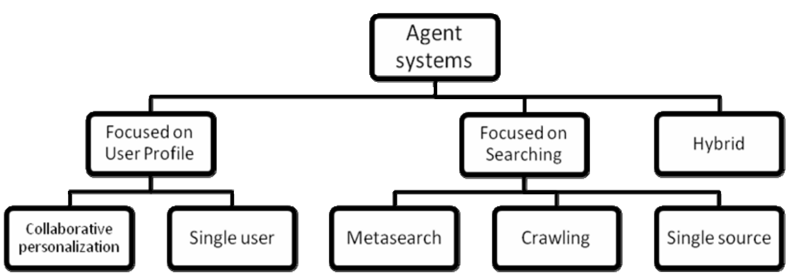


Fig. 1. Classification of information retrieval systems based agent technology

3 Agent Systems Focusing on User Profile

Personal assistant is usually an agent connected with one person or a group of people. His main aim is to find only such documents in the Internet that will be useful to the user, which means that the agent should know the user he is working for. That is why user profiles are built to store information about user interests. In many cases the profile is used to filter documents from the Web or to recommend some documents that may be interesting to the user. An important aspect is to teach the agent how to select interesting articles. The main two points in the user profile approach are how to get to know the user and how to use this knowledge to recommend him documents that will be useful. Systems focused on user profiles can be divided into two main classes: collaborative filtering - systems that consider the behavior of a group of users and systems that focus on a single user (Fig.2).

3.1 Collaborative Personalization

Most of search engines need to have some keywords to start the searching process that can be understood as filtering - documents that contain entered terms will be shown to the user and others will not. Recommendation process requires knowledge about user preferences. When agent system knows his user, his interests, habits, etc. it can look for documents that will fit the profile and be really useful to the user. Similar approach often occurs in information retrieval literature.

The first approach analyzes a web log that contains the access patterns of a group of people in order to discover their general behaviors. Collaborative filtering methods try to classify users into some group of interests to recommend some documents in accordance with the interests and preferences (site assistant in e-commerce systems). On this level there are many ideas how to learn about user preferences and how to classify user to the appropriate group. The most popular method is Web usage mining – it aims to discover interesting patterns of use by analyzing Web usage data. It uses some data mining techniques like association rules [5], clustering and usage pattern discovery [3, 10]. These methods allow agent system to find out what user is interested in and to which cluster he should be classified. Most collaborative filtering methods using data mining techniques fall into two categories: memory-based

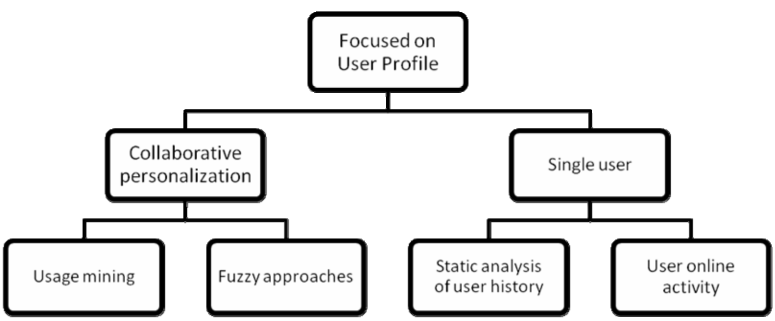


Fig. 2. Classification of systems focused on user profiling

algorithms and model-based algorithms [12]. Memory-based algorithms predict a test user's ratings based on the corresponding ratings of the users in the training database that are similar to the test user. In contrast, model-based algorithms build models that can explain the training examples well and predict the ratings of test users using the estimated models. Both types of approaches have been shown to be effective for collaborative filtering.

A different method of collaborative filtering is fuzzy approach [20]. In this system collaborative filter is a function that takes all past user sessions as input and produces recommendation values for pages not yet accessed by the active user as its result.

Collaborative recommendation technique is a powerful method for leveraging the information contained in user profiles. In contrast to content-based filtering, the agent rates the items chosen by its user and compares the corresponding user preference vector to that of other users projected to the same set of items. It then recommends other items which have been recommended by users who share similar likes and dislikes. For this purpose it has to collaborate with other agents to gain the respective knowledge.

Collaborative filtering is also used in [15]. Authors of [15] mention two options for filtering. The user may construct a software agent that filters search results according to his background and search specification or he can search a pool of existing agents to find one already created by another user with a similar background to do similar search. The second way is supposed to obtain better search result than simply conducting a search using traditional methods.

Although collaborative filtering is popular in e-commerce systems, it has the following drawbacks [4]:

- it is hard to update the user profile of such system incrementally. If the structure of the site changes, the previously learned user profile will be obsolete;
- since collaborative filtering learns the general behaviors of a group of people, it can not acquire the particular interests of each user unless it is given plentiful training examples.

3.2 Personalization Based on Single User Preferences

In contrast to the abovementioned approach, the training data of a single-user personalization system comes from one user only. Therefore, its user profile is specific to a user's interests rather than to the general behaviors of a group of people [4]. Items are recommended to the user according to correlations found between the items' content, for example the presence of certain keywords, features, and the given user preferences in a profile. The latter is usually generated and updated by the agent automatically by observing the user's online activities such as visiting Web pages, dealing with downloaded documents, adding or deleting bookmarks, and printing, as well as affective signals such as eye movement or gestures, and credit assignments to the agent [13]. This method is called content-based filtering.

Recommendation process is popular in some services (news services, e-shop, etc). Service has a database with interesting issues and profiles of its users. When new issue appears and fits to some profiles, it is recommended to this group of users [21]. Personal agent has a little bit more complicated task. It has "only" user profile and the whole Web content. When it finds some documents connected with user query, it filters them in terms of user profile.

In the approach based only on single user we do not have information about other users, theirs preferences, habits, etc. Without this information we could not generate some classes of users and recommend documents which other users marked as useful for them. Here profile is discovered from user's activity in two general ways: a static (demographic) or a dynamic one (see Fig.2).

Static methods are based on history of user activities or can be obtained directly from user information [19]. When the system is started for first time, user needs to fill some form with personal questions about preferences and interests, or is forced to see through some trial set of documents. Having such information the agent system can obtain categories of user preferences and build some hierarchy of them to better the recommendation process in the future. These approaches are really uncomfortable to the user, because he has to spend a lot of time to teach the system exactly what he likes and dislikes.

A better method is to observe user activity in the Web and not to require him to rate the documents. In dynamic approach agent observes the user and saves information about entered requests and pages in the browser. Based on this history it can create some group of user interests and during recommendation process compare new requests with each group to retrieve better information.

In [16] the authors propose two options in recommendation process: first one using user profile and second one by modification of query. When the searching agent has information about its user, it can compare actual query with previous requests. If the similarities between them are not small, agent should search this "totally" new query in Internet, and when similar query exists in the database, agent should use already gathered information, for example where to find best answers for this query. Recommendation by query modification uses information from the profile. From all previous users' queries the system chooses these that are similar to the actual query. On the base of those a new vector of terms is formulated. It is the best representation of this set of queries. This new query is then searched for in the Web.

Very important information can be gathered from user online activities. User preferences can be noted by the time he spends on one document. When user is interested in some area he will drill down through the Internet pages. Based on this data agent can manage personal information, arrange it into some categorized knowledge [21] or even learn how to avoid misconception [5].

Both of those approaches have advantages and disadvantages. Some advantages of static methods are as follows: user interacts with the system directly and has a choice to agree or reject the process of profiling; users can easily adapt the profiles elements about themselves; the system can make some assumptions about a user in advance, and improve the profiling process. Some disadvantages are: it is difficult to ensure that user will voluntarily provide information to the system; answers provided by user may not reflect his own inspiration accurately and the profile is static and not flexible enough to take a user's interest changes into account.

Dynamic methods also have some advantages: the profiling can be processed automatically; the user profile can be continuously updated and adjusted. Some disadvantages include: user profile may not be changed when some interests repeat and others are not taken into account; user profile may not be fully certain and the more general interests of the user can not be traced in real time.

4 Agent Systems for Searching

In this section various agent systems that focus on searching the web will be presented. Note that searching agents may work with profile agents (as described in section 3), for example as a different part of the same agent systems.

The following section was divided according to different topics related to searching agent systems as addressed in various papers. The division between single agent and multi agent systems is not drawn here, as all the mentioned approaches can be used in both types of systems. Usually the single agent systems are derived from classical web tools, and do not often fit the definition of an intelligent agent. Those are e.g. search engine crawlers or single web page assistants. Multi-agent systems are more complex and usually consist of intelligent agents working towards a common goal.

This section divides the search focused agents into three groups (see Fig.3.): crawler agents that aim to index the web, meta-search agents that select the best source for a query and then delegate other agents to check it, and single source assistants that work within a specific domain to speed up its use. While all those approaches may be considered separate, the best results are attained when all three are combined.

4.1 Crawling Agents

Crawling agents derive from web spiders - simple autonomous systems that began their career with the rise of World Wide Web. The idea is to browse the Internet (or Intranet) like a human would do, site by site, following normal links. Client or server based search spiders operate from user home computer or search engine server and start their browsing from preselected homepages. The Internet is browsed by a series of interconnected sites and each visited site may, or may not, be the searched one. However, potentially relevant or important pages may not be connected to a searched group of sites and may not be found. Additionally, in today's Internet the search may take enormous amount of time for a single spider, as it has a growing number of pages to visit. Web spiders may additionally use methods like genetic algorithms [6] to speed up page browsing and rating.

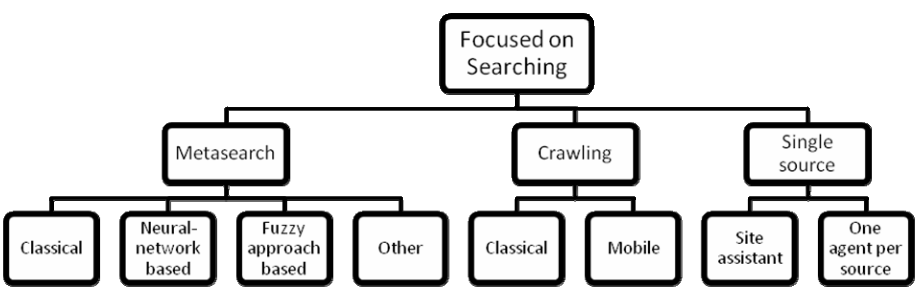


Fig. 3. Classification of systems focused on searching

The basic division of crawling agents lies between mobile and not mobile agents. As an example the author of [14] suggests to complement classical search engine results crawling agent working on-line to retrieve most recent, relevant documents. The adaptive representation of system used in [14] roughly consists of a list of keywords, initialized with the query terms, and of a feed-forward neural net. The keywords represent an agent's opinion of what terms best discriminate documents relevant to the user from the normal sites. The neural net relies on either traditional search engines or a set of personal bookmarks in order to obtain a set of seed URLs pointing to pages supposedly relevant to the query given by the user. The starting documents are prefetched, and each agent in the population is assigned one of these documents and given an initial amount of energy. Energy is the currency that allows agents to survive and crawl – when it is exhausted the agent stop its search.

A more hierarchical approach to crawling may also be possible. In [22] searching agents are divided into two classes: facilitator-agents that manage topics and assign tasks to crawler-agents that perform the actual web search. Subsequently each facilitator-agent has its subgroup of crawler agents that it will contract tasks to and reward them for relevant links. Additionally, crawler-agents may exchange information about visited pages with other crawler-agents connected to the same facilitator-agent. Exchange of such information inside a single topic facilitator-agent is highly desired to raise its utility value.

The other group of crawling agents consists of the mobile ones. Mobile agent idea is based on the possibility to transfer the agent in its current state to a location near its data source in order to reduce network communication load. Mobile agent systems are supposed to provide functionality for agent migration, communication of agents with other agents and with the underlying system, and support for security, transactions and controlling agents [1]. Nowadays support for agent mobility is supported by multiple frameworks, e.g. Jade [11].

Work [9] considers the problems of mobile agents from the routing time point of view. It proposes extended hierarchical query retrieval to obtain query results from remote hosts and tests it feasibility. Additionally it allows for on-line and off-line query retrievals, which are described as, correspondingly, establishing a direct connection with web site by the mobile agent, and downloading query results from web search engines during nighttime and presenting their prefetched results to the user in the daytime.

In general, mobile agents have a slight advantage over classical crawlers, most specifically the lower network load. A huge disadvantage is also the possibility that the target location will not provide support for migrating agents. This becomes a larger problem as the global network grows in size without an adopted standardized agent platform able to receive mobile agents and this situation is not likely to change. Usually proposed solutions are to move the agent as closely to the information source as possible. Still, for standard searches, the time required by the mobile crawlers usually becomes larger than the time required by classical ones.

4.2 Meta Search Agents

The idea of a meta search engine derives from classical yellow pages services and tries to order the information to allow easier access by categorization. More precisely a group of normal search engines is usually tested and for each category, or keyword, the best fitting search engine is selected. For example if the user wants to search for thing to buy, he would rather receive results only from shopping related search engines, than from encyclopedia ones. In this paper we differentiate several method of performing a meta search by single or multi agent systems. First, agents using classical methods, that usually require user attention to select a category for his search or use simple tools, like a relational table assigning each keyword a best browser for use with. Those will not be described in this paper. The second group of meta search agents use neural networks in various configurations to attain the same effect. The third approach is based on fuzzy logics and is usually a generalization of the first one. Other approaches are less popular and differ between every system and cannot be put under a single name, e.g. consensus methods.

In [2, 14] there are some examples of fuzzy search agent systems. An interesting approach is to use many search agents resembling crawlers and a central meta search agent that uses fuzzy algorithms to select proper search agents for each task [2]. In this approach the central agent assigns tasks to search agents based on the fuzzy distance between their knowledge and the asked query. The authors differentiate search agents by the number of solutions given and time to obtain them, for each previous query. Experiments described in [2] show that this technique optimizes the number of queries requested by search agents.

Similarly, [16] proposes a consensus method for integrating answers from different search engines. The searching agents work as crawlers described before, but their knowledge base is actualized based on the consensus of all the agents answers. The consensus itself is calculated by the central managing agent that gathers the answers, evaluates them by checking consistency degree of answer sets and provides the search agents with the distance their answer was from the consensus. Additionally, search agents may answer new queries by recalling an earlier, similar but accurate, query. [16] proposes also to create a single agent being a meta search engine that operates with the described functionality but without communication costs between the agents.

The meta search agent in the final multi agent system may use a neural network meta search engine found in [18] to attain its functionality. Authors of [18] propose to use a feed forward neural network based on CC4 algorithm to satisfy two conditions they set for the meta search engine: good generalization and high speed of learning. The neural network input is translated from result page titles to binary vectors according to a keyword table. After processing, a single neuron returns a binary answer, either a 1 for pages relevant to the query or 0 for irrelevant pages.

The best approach to meta search has not been yet established and different ones work differently in different situations. Classical methods are usually faster in the short term, but offer less precise results, and more sophisticated methods, while requiring time for learning or tuning, usually will work faster and more precisely in trained systems.

4.3 Single Source Agents

Single source agents are simple entities that aim to ease any user operation on the domain it is attached to. The idea and often also the solutions are the most basic of the above, but desirable results may be achieved fast.

In [8] Denzinger and Fuchs experiment with their TECHS approach to cooperative heterogeneous search agents. In this approach they propose to have multiple agents receive the same whole problem. The agents proceed with a selection process, which goal is to select facts from their knowledge base or searched source that could help other agents, but are not likely to be found by these other agents themselves. Over time this leads to creation of cooperating search agent teams. This is an example of single source multi agent search system, which is usually a very good solution in a limited, stable or closed environment. Once the sources become too dynamic, or must be added too often, the system loses its edge.

Mobile system presented in [1] outlines a method for web search with specialized search agents working with a single source only. In this method an user query in an agent is first forwarded to a classical search engine and a list of URLs is retrieved. Web pages unknown to the domain expert are then further analyzed by a special filter and relevant ones are added to agent knowledge base. Finally user query is answered based on information from its expanded knowledge base only.

The other type of single source searching agents, site assistants, are closely related to specific recommendation systems as described in section 3.

As one may notice, the single source agents are a basic approach that has many resemblances to non agent systems. While this point may have not been as closely related to the problem at hand, any classification without it would be incomplete.

5 Conception of a Hybrid System

Hybrid systems, which were mentioned in section 2, combine personalization with advanced search approaches. Such systems may be generally presented as in Fig.4.

The query *q* comes from the interface and is personalized with function *P(q;p)* where *p* is the system's knowledge of the user profile, after personalization, the improved query *q'* is the input to the function *M(q';m)* which selects a proper

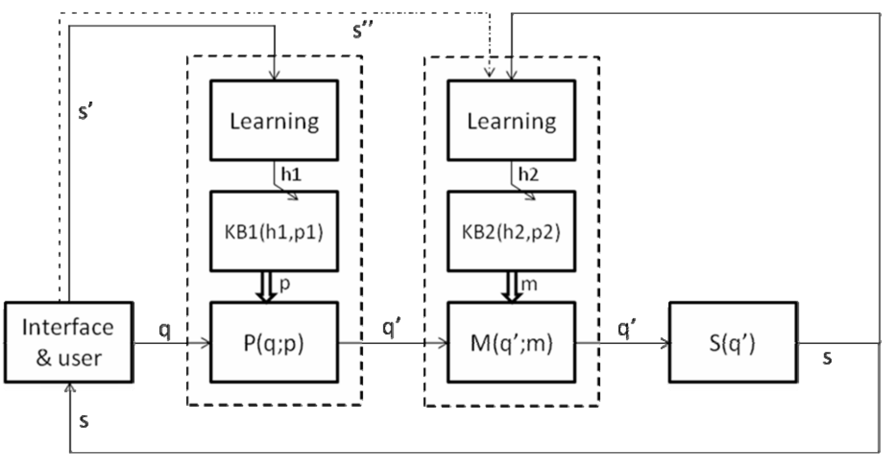


Fig. 4. Schema of a hybrid information retrieval system

source to use for this query based on some knowledge *m*. Note that while M represents a meta search engine, it may as well always select only a single source (no meta search).

Analogically, the actual internet element *S(q')*, which task is to retrieve the data, may be a single source agent, a crawler or a standard search engine. The resulting pages are rated and the page, source and rating tuples are presented both to the user (via the interface part) and to the learning system used for the knowledge base *KB2* used in the meta search process. While the user uses the result, additional knowledge is obtained and user rating (direct, when he rates results; or indirect, when his behavior is just observed) is then used to teach the *KB1* knowledge base used for personalization and sometimes also *KB2* knowledge base used for meta search.

Note that in the above schema, we do not differentiate agents. In fact, the same schema may be applied both to single agent systems, multi agent systems and even non-agent expert systems.

The authors of this paper are currently working on a single agent information retrieval system based on this representation that will be highly generalized and allow to test different approaches of personalization and meta search (the actual network part will be mobile crawling agents with seeds being rated queries in known search engines, as this combination was deemed best for this situation). Viewing the system as a single agent allows personalization on two levels – both in the query improvement, and in meta search. A single meta search knowledge base for multiple users may usually leads to generalization and mistakes (e.g. consider simple query *bike*; one user may like to view different bike pictures with it, while others may like to visit bicycle shops, thus a different search engine would be preferable for each query; a single meta search may only learn to select one of those or, if selecting more, will present unwanted results to the user). As the system is being implemented in JADE environment [11], the description presented above requires small changes.

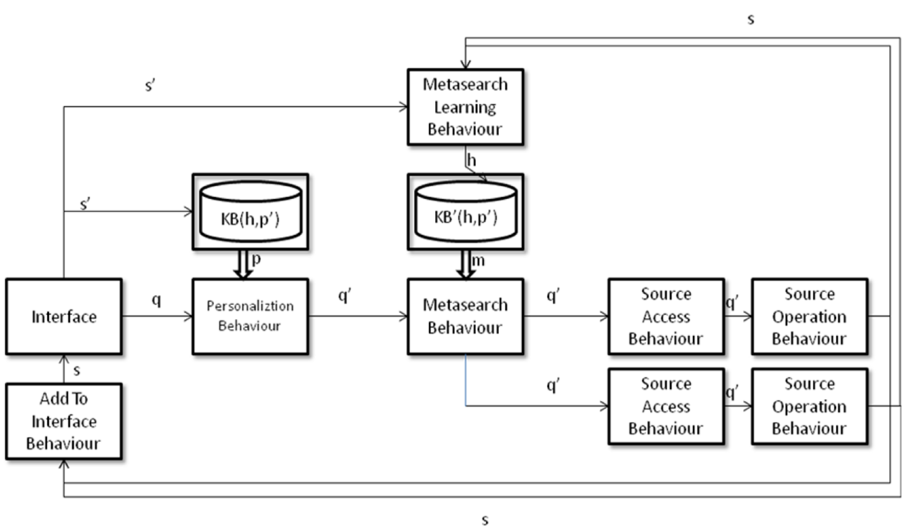


Fig. 5. Schema of a single agent hybrid system in JADE environement

The query retrieval process starts with the user entering his query *q* in the user interface. In general, this query is the exact expression given by the user, with no semantic changes at this level. Eventual modifications may be structural, such as dividing a sentence into terms. The query *q* is analyzed and changed (by adding priorities or new elements) by the Personalization Behaviour. After this the query may no longer be recognizable in the previous form, and as such, is denoted as *q'*.

Query *q'* is then further analyzed by the Metasearch Behaviour, where, based on some premises included in *q'* and additional knowledge in metasearch knowledge base *KB'*, some source is recommended as best suited to answer the query. To operate in the selected source, two additional behaviours are used and those return a set of links relevant to the query denoted here as *s*.

Those answers are then presented to Metasearch Learning Behaviour, which expands the agent source recommendation knowledge in *KB'*, and to the AddToInterface Behaviour, which helps the actual interface in presenting the results in a proper, relevancy based, order.

The interface is further used to observe the user behaviour on the answer links, which allows additional knowledge to be saved in profile knowledge base *KB* and again by the Metasearch Learning Behaviour in the *KB'* knowledge base.

Such approach should be more friendly to the final user both in the functional area and in the non-functional requirements. As mentioned above, the actual results given by the system should also be more related to the single user it is working with. Both the personalization and the meta search part in this system are meant to be interchangeable to create a framework for testing different approaches to areas presented in sections 3 and 4. Additionally, a method for result recommendation is to be tested in similar way between the network sources and the user interface.

References

[1] Baumann, J., Hohl, F., Rothermel, K., Strasser, M., Theilmann, W.: MOLE: A mobile agent system. Software Practice and Experience, 575–603 (2002)

[2] Camacho, D., Hernandez, C., Molina, J.: Information Classification using Fuzzy Knowledge Based Agents. IEEE, Los Alamitos (2001)

[3] Chau, M., Zeng, D., Chen, H., Huang, M., Hendriavan, D.: Design and evaluation of a multi-agent collaborative Web mining system. Elsevier Science, Amsterdam (2002)

[4] Chen, C., Chen, M.: PVA: A Self-Adaptive Personal View Agent. Journal of Intelligent Information Systems (2002)

[5] Chen, C., Hsien, Y., Hsu, S.: Mining learner profile utilizing association rule for web-based learning diagnosis. Elsevier, Amsterdam (2006)

[6] Chen, H., Chung, Y.M., Ramsey, M., Yang, C.C.: A Smart Itsy Bitsy Spider for the Web. Journal of the American society for information science 49(7), 604–618 (1998)

[7] Choi, Y., Yoo, S., Lee, J.: Hierarchically organized neural net agents for distributed web information retrieval. IEEE, Los Alamitos (2000)

[8] Denzinger, J., Fuchs, D.: Enhancing Conventional Search Systems with Multi-Agent Techniques: a Case Study. IEEE, Los Alamitos (1998)

[9] Denzinger, J., Fuchs, D., Selamat, A., Selamat, M.H.: Analysis on the performance of mobile agents for query retrieval. Information Sciences 172, 281–307 (2005)

[10] Girardi, R., Marinho, L.B., Ribeiro de Oliviera, I.: A system of agent-based software patterns for user modeling based on usage mining. Elsevier, Amsterdam (2005)
[11] JADE, Java Agent Development Framework, `http://jade.tilab.com`
[12] Jin, R., Si, L., Zhai, C.: Preference-based Graphic Models for Collaborative Filtering (2003)
[13] Klusch, M.: Information agent technology for the Internet: A survey. Data & Knowledge Engineering 36, 337–372 (2001)
[14] Menczer, F.: Complementing search engines with online web mining agents. Decision Support Systems 35, 195–212 (2003)
[15] Newell, S.: User models and filtering agents for improved Internet information retrieval. Kluwer Academic Publishers, Dordrecht (1997)
[16] Nguyen, N.T.: Advanced Methods for Inconsistent Knowledge Management. Springer, London (2008)
[17] Qiu, M., Huang, H., Yang, L.T., Wu, J.: Intelligent Search Agent for Internet Computing with Fuzzy Approach. In: 11th IEEE International Conference on Computational Science and Engineering (2008)
[18] Shu, B., Kak, S.: A neural network-based intelligent metasearch. Information Sciences 120, 1–11 (1999)
[19] Speretta, M., Gauch, S.: Personalized Search Based on User Search Histories. In: International Conference on Web Intelligence. IEEE, Los Alamitos (2005)
[20] Suryavanshi, B.S., Shiri, N., Mudur, S.P.: A Fuzzy Hybrid Collaborative Filtering Technique for Web Personalization. In: Proceeding of the 3rd Workshop on Intelligent Techniques for Web Personalization (ITWP 2005) (2005)
[21] Tu, H., Hsiang, J.: An architecture and category knowledge for intelligent information retrieval agents. IEEE, Los Alamitos (1998)
[22] Xiang, D., Du, Y.J., Yi, L.Z., Li, K.: Coordination and Communication among Topic-Specific Search Agents. In: Third International Conference on Natural Computation (ICNC 2007) (2007)
[23] Yang, C., Yen, J., Chen, H.: Intelligent internet searching agent based on hybrid simulated annealing. Elsevier Science, Amsterdam (2000)
[24] Zhou, Y., Hu, Y., He, H.: Learning User Profile in the Personalization News Service. IEEE, Los Alamitos (2007)

Intelligent Agent Based Workforce Empowerment

Nazaraf Shah[1], Edward Tsang[1], Yossi Borenstein[1], Raphael Dorne[2], Anne Liret[2], and Chris Voudouris [2]

[1] Univeversity of Essex Colchester, UK
{shahn,edward,yboren}@essex.ac.uk
[2] BT Research Laboratoy Ipswich, UK
{raphael.dorne,anne.liret,chris.voudouris}@bt.com

Abstract. The fast moving and competitive global economy requires that organisations manage their human resources in an efficient and flexible manner in order to tap their full potentials. The management of human resources has crucial role in success or failure of an organisation. Today's services organisations strive to tailor their human re-source management policies that improve their performance by empowering employees. Employee empowerment is a management style that helps organisations to utilise their employee's potential to maximum and be able to react and adapt quickly and efficiently to changes in service provision. This paper explores the possibility of empowering British Telecommunications (BT) field workforce and its potential benefits by using simulator. We investigate the application of a 'pull-based strategy' for empowering BT field workforce and its relation with motivation and productivity of BT workforce. Currently jobs are allocated to BT field engineers by a central scheduler; we refer to it as a 'push strategy'. In this simulation we attempt to assess the impact of empowerment on BT FE motivation and towards overall increase in productivity.

Keywords: Staff Empowerment, Multiagent System, Dynamic Scheduling, Simulation.

1 Introduction

Empowerment has become a popular term in industrial management and service industry in recent years, yet there is no single definition of term empowerment. Wallerstein defines empowerment as follow "*Empowerment is a social action process that promotes participation of people, organizations, and communities towards the goals of increased individual and community control, political efficacy, improved quality of community life and social justice*" [1]. Adrian defines empowerment as a term used to refer to employee task based involvement and attitudinal change in an organisation [2]. Rapport defines empowerment as a mechanism by which people, organisation and communities gain mastery over their life [3]. According to Conger et al. empowerment is a process of enhancing feelings of self-efficacy among organisational members through the identification of conditions that foster powerlessness and through their removal by both formal organisational practices and informal techniques of

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 163–172, 2009.

providing efficacy information [16]. Psychological empowerment is defined as individuals' feelings of control over their lives which they experience as a member of a group and/or an organisation [4]. Wall et al. define empowerment as a management concept that is concerned with added delegation of responsibility to individuals and teams in the management and execution of their jobs [5]. We restrict to this definition in our research of empowering BT workforce.

The roots of empowerment are in theory of X and theory of Y presented by McGregor [6]. Theory X is based on traditional old command and control view while Theory Y is based on integration of individual and organisational goals. These theories represent two different style of management in today's organisations. The theory of X assumes that most people have inherent dislike for work and they dislike responsibility and like to be directed and people work for money and security. These assumptions are still behind management strategies in many today's organisations. This results in a rigid and repressive style of management that produces a depressive work culture.

Motivation is an important success factor behind employee empowerment, without employees' motivation the benefits of empowerment are lost if not contribute to employees' underperformance. Many psychologists have attempted to study what motivates people; one of such work is Herzberg's Motivation Hygiene Theory [7]. In his book [7] he presented the results of his study of the work related motivation of thousands of employees. He concluded that there are two types of motivation, namely *hygiene factors* and motivation *factors*. Hygiene factors are analogous to principles of medical hygiene and are concerned with supervision, interpersonal relation, physical working conditions, salary and security. Deterioration of these factors below a threshold level causes job dissatisfaction and de-motivation among employees.

Many employees' motivational schemes rely directly on performance related pay and bonus schemes. According to Herzberg money earned as a direct reward to individual performance is reinforcement of achievement and recognition.

BT workforce allocation problem's environment is highly dynamic and unpredictable. Research evidence suggests that empowerment increase performance in environments where there are frequent changes in product and service design or unreliability in technology [5]. Empowerment has both individual and group dimensions. This paper is about individual empowerment dimension, where we will assess the effect of empowering individual Field Engineers (FEs) in BT. The same concept can be applied to team work too. BT field engineers are given greater discretion with regards to their preferences on carrying out jobs to meet the organisation goal and their individual goals. The aim of our research is to build a simulator to assess how to raise morale and self satisfaction among BT FEs by empowering them with greater autonomy and responsibility in the management and execution of their daily jobs [9] without compromising BT business objectives.

This paper is organised as follows. Section 2 describes the BT's current job allocation system. In section 3 we provide BT scheduling problem description. Section 4 describes the high level architecture of the simulator. In section 5 we provide the detail of empowering of BT FE using pull strategy. Section 6 describes the simulation results, and finally section 7 concludes the paper.

2 Current Job Allocation System

Currently jobs from customers are assigned to FE by the central scheduler. The scheduler matches skills of the available FEs to the skill required to perform the job. The jobs are assigned to FEs one at a time. The FEs acknowledge the job receipt on receiving job allocation messages and send job end event to the scheduler to indicate successful completion of the job or a failure. The job allocation system is centralised and based on command and control management style and fits into Theory X as explained in section 1.

The objective of the proposed simulator is to assess the applicability of the Theory Y for BT job allocation system. Various empowerment scenarios are used to simulate their effect on the performance of FEs and overall productivity of the system. We elaborate the tradeoffs of these scenarios in an attempt to show what benefits BT could gain at what cost in various situations.

3 BT Workforce Scheduling Problem Description

BT is the largest communication service provider in the UK. BT employs around 30, 000 FE to provide services to businesses and residential customers everyday. These services include network maintenance, fault repair of various kinds and installation of network [8]. The jobs are assigned to FEs by a central scheduler, which matches skills needed to do the job to the available FEs' skills to do it.

The BT workforce scheduling problem is a complex problem; here we present a simplified and a generalised version of this problem. From management point of view BT divides UK mainland into a number of regions and each region is then divided into a number of small areas.

The area under investigation has around 250 jobs which are performed by 86 FEs. For this problem we consider 80 km^2 area divided into six working areas. Travel time (between any two coordinates) in this area is measured as the Euclidean distance divided by speed. We do not consider dynamic travel times. We assume that the speed is 40 km/h on average.

Each FE has a set of potential skills making him capable of performing jobs according to his skill set. FEs are geographically distributed in the region.

Each job has an importance score (importance score defines the priority of the task) and requires one or more skills to perform it. Non appointment jobs are 30 out of 250 jobs, the remaining customers jobs are scheduled in advance. There are around 50 jobs that arrive dynamically during the day, which are required to be scheduled as soon as they enter the system.

A job commitment time is defined as follow.

- Earliest Start Time: the earliest when an FE should arrive on customer site to start working on the job.
- Estimated Duration: the time period during which FE should clear the task (Estimated Duration is obtained from data provided by BT as average of time taken to perform a specific job).
- Latest End Time: marks the time before which task must be completed by an FE.

Incoming jobs during the day has effect on pre-specified compile list of tasks for each FE, due to their different priorities. Such a dynamic environment requires a reactive approach to dynamically incoming jobs. FEs has to perform these jobs as they are assigned to them in real time. We measure the performance of our system based on the total number of jobs completed successfully during a day. The duration of each task depends upon type of that task. A task not completed within its time window or not allocated is marked as a failed task.

3.1 Problem Formulisation

In this subsection we provide the formal description of job, FE, and allocated job.

Job Information

Each job is described by a n-tuple: (Id, Sk, Loc, D, St, Et, P, Rt) Where Id is a unique identification number of the job. Sk is the skill required to do the job. Loc is a location which describes the x and y coordinates of the job. D represent the estimated duration of the job in minutes. P represents importance score of the jobs, which determine priority of the job. St represents earliest start time, the time before which FE cannot start the job. Et represents latest end time, indicates the time before or at which job must be completed to avoid failure. Rt is the time at which job is reported.

Engineer Information

Each FE is described by a tuple: (Id, St, Stloc, Curloc, Ot, Sk, Jobid,) Where id is unique number assigned to engineer. Stloc is start location of the engineer (engineer can start from home or nearest exchange), which is described in term of x and y coordinates. Curloc represent x and y coordinates of engineer's current location. Ot is Boolean variable on overtime availability of the engineer (An integer value represent the amount of overtime allowed in minutes). Sk represents skills set of the engineer (engineers have various combination of skills). Jobid represents the job that engineer is currently committed to do.

Allocated Job

Each job contract is described by a tuple: (FEid, Jid) Where FEid represents identity of the FE whose has chosen to perform the job having Jid as its identity.

4 High Level System Modelling

In our simulation the overall system is modelled as multi-agent system, where each actor in the system is an autonomous agent. We have identified three types of agents in the system, known as *dispatcher agent*, *job generation agent* and *engineer agent*, JADE Agent Framework [11] is used to model and implement various behaviours of these agents. JADE is a FIPA [12] compliant Java based agent development framework. In the following we provide a brief description of each agent.

Dispatcher Agent
It is an interface agent that provides agent based interface to job allocation system. It is responsible for translating requests/response to and from job allocation system. The scheduler agent also maintain directory of all FE agents and their current state. All interactions between job allocation systems and FEs take place via scheduler agent.

Job Generation Agent
Job generation agent acts as a customer relation system. It generates the jobs according to job generation simulation parameters and sent them to scheduler agent. It is capable generating jobs simulating initial demand at the beginning of day and also real time demand during day. The scheduler agent then passes these jobs on to job allocation system for allocation purpose.

Engineer Agent
An Engineer agent represents a FE. These agents simulate various behaviours of FE. There are 86 field engineers in our simulation system each engineer is represented by a Engineer Agent.

5 Field Engineers Empowerment Using a Pull Strategy

Currently scheduled jobs are assigned to FE agent using a push strategy. FE agents have no choice but to accept job allocated by the scheduler. In push strategy there is no incentive or motivation for FE to attempt to increase productivity. Push strategies follow a strict command and control paradigm in which employee role is to comply only [14]. In contrast, a pull strategy follows a more relaxed control approach by providing employee with a power to exercise their preferences and discretion. Berlew emphasised the importance of the pull of the task by employees rather than push from management [15] and its motivational assumptions and their positive effect of performance. In a push strategy commitment to perform a job is initiated by the system, whereas in pull strategy commitment to perform a specific job is initiated by the resource itself rather than the system.

Pull strategy is about distributing control among participants, while push is about maintaining control at one place. We use pull strategy to delegate some of the decision power regarding job assignment to the FEs. Pull strategy helps participants to learn about innovative, efficient and effective ways of performing tasks.

Below we shall describe a manageable pull-based jobs allocation system. We do not claim that this is the optimal strategy. It serves to demonstrate what can be done. It provides a framework for comparison and refinements in the future.

An FE agent requests from the scheduler agent a set of jobs that he could perform next and selects its two preferred jobs from the returned set based on its preferences and inform the dispatcher of its choice. It then waits for dispatcher's approval and start working on job after approval has been received. In order to reduce communication between FE agents and scheduler, FE agents select more than one job as their preferred choice (in current implementation a FE agent selects two jobs, as its first and second preferences) and send their selected priority lists to the scheduler.

One of the following strategies is used by FE agent while selecting choices of jobs from the set of jobs sent by the scheduler agent.

a) The FE agent selects jobs that are closer to its current location.
b) The FE agents selects jobs that is has shorter completion time
c) The FE agents select jobs randomly (this simulates selection criteria private to the engineer.
d) The FE agents select job closer to the centre of their associated working area (this simulates the desire to do jobs close to the engineers' home or other local activities)

We study the effect of these strategies on performance of FE agent. The interaction between FE agent and scheduler agent is governed by FIPA request protocol [16]. Next three behaviours are implemented by the dispatcher in order to enable empowerment using pull strategy.

a) Job Request Receiving Behaviour

The job assignment process is initiated by an FE agent by sending a request to the scheduler agent asking for a best set of jobs available for it. The scheduler agent then checks if there exists any priority job that can be allocated to a job-requesting FE agent; if so, it assigns priority job to the agent (This is a non empowerment scenario). If there is no priority job then the scheduler checks for appointment jobs and send a maximum of five/ten jobs that can be performed by the requesting FE agent. The response from the scheduler agent invokes job selection behaviour (FE agent employs various strategies to select best jobs that it intends to perform) of the FE agent. The FE agent then sends back its choice list to the scheduler agent and this results in invocation of the scheduler agent's *Job Assigning Behaviour* as described below.

b) Job Assigning Behaviour Committed Mode

After receiving FE agent's job choice list, the scheduler allocates the first choice to FE agent provided there is no conflict and remove that job from the job queue, otherwise use conflict resolution strategy regarding FE choices described below.

Pull strategies are inherently prone to conflicts. An FE agent has no knowledge of its peers' choices. One or more FEs may select the same job. It is the job of the scheduler to resolve conflicts. The scheduler accepts FE agent choices using following strategy.

- Accept FE's selected choice as it is if there is no conflict between FEs for that Job.
- If there are conflicts between technicians, the scheduler uses the following strategies to resolve conflicts.
  - Assign job to an FE agent who is closer to the job (this looks after the company's interest of reducing travelling needs);
  - In case of equal distance from the job, allocate job to the FE agent based on first come first served basis (this encourages early commitment, which benefits customer communication).

c) Job Assignment Behaviour Uncommitted Mode

This behaviour is similar to committed choice behaviour except that scheduler can override FE choices. This model tries to strike balance between organisation's objectives and FE preferences thereby creating a win-win environment both for the company and FEs. In this model an FE while working on a job, requests scheduler *N* minutes before completion of the current job to send a set of jobs that FE can do next. The scheduler then sends an ordered set of jobs to FE. FE may reorder the set of jobs based on its own personal preferences or leave the jobs order as it is. FE then sends the ordered set of jobs back to scheduler. If there is no change in jobs order, scheduler assigns the first job to FE, otherwise scheduler uses a weighted objective function to make final decision regarding job allocation that may result in FE not getting his preferred first choice. The weighted objective function takes into account FE personal preference and organisation's objectives.

Jobs in committed mode are allocated using following criteria

- Weighted objective function
  - Jobs completion (Total number of jobs completed by the FE)
  - Service quality (expressed as "preference")
  - Distance (FE's distance from job)
  - Engineers' personal preferences (PP)
- Constraints
  - Matching availability between engineer & job
  - Load balance (LB)
- PP and LB are important to engineers
  - PP may reflect engineers' personal interest or needs
  - Early completion lead to shorter day / day off

In the next Section, we shall simulate both pull and push strategies using the same benchmark and assess the impact of these job allocation strategies on overall performance of the system. In the push configuration, jobs are allocated to FE as explained in section 2. We will also show the performance comparison of uncommitted mode behaviour and pull strategy.

6 Simulation Results

In our simulation, a pull strategy is used as way of empowering BT FE agents; it provides them with the ability to exert their control on the selection of jobs using their preferences. Research studies [2, 4, 5, and 6] on empowerment has shown that such a control on job selection has positive effect on workforce morale. As discussed in section 1 employee empowerment motivates them and could result in increased performance. It is assumed that employee empowerment and performance related pay will increase the performance of FE. We assume that motivated FEs take less time to complete a job, but this assumption can only be truly confirmed by field trial.

In our experiments (committed mode) we study the "worst case" and an "improvement case" on performance. The "worst case" assumes that empowerment has no gain in the completion duration of tasks. The "improvement case" assumes that

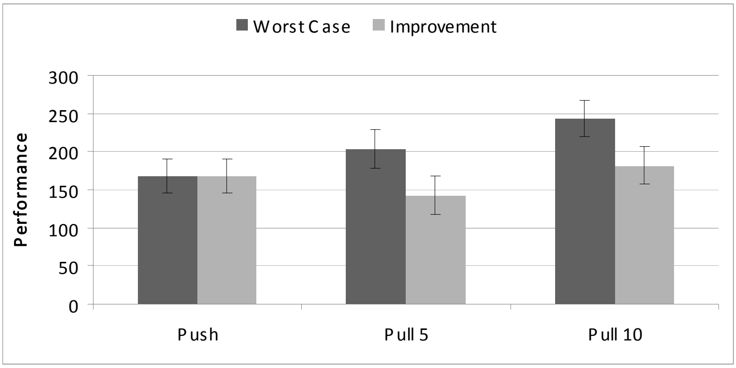


Fig. 1. Performance Comparison of Worst Case and Improvement Case

tasks will take 10% less time to complete. The analysis of past jobs completion data shows that this assumption is reasonable. An empowered FE with incentive for doing more jobs will obviously take advantage such fluctuation in job completion time and try to work smart in order to complete jobs as quickly as possible. The results of both cases are shown in Figure 1. The results shown in the figure are average of 10 runs. There are five problem instances per data point in the graph.

The x-axis represents the Push, Pull 5 (five jobs sent to FE) and Pull 10 (ten jobs sent to FE) options. The y-axis represents the performance in term of number of tasks not being completed. This is a minimisation problem and we want to keep this quantity as small as possible in order to be able to increase job completion rate.

We ran simulation for two configurations of pull strategy. In first configuration FEs are offered a set of five jobs to make their selection from and in second configuration FEs are offered a choice of ten jobs. As shown in figure 3 the "Pull 10" is always more costly than "Pull 5". As expected, when there is no reduction in job completion durations, Push is a better strategy for the company. However, when there is 10% reduction in duration, "Pull 5" is better than Push. There are some scenarios when FEs are assigned a single top priority job, such scenarios effectively turn pull strategy equivalent to push strategy.

The result of uncommitted mode empowerment strategy is shown in figure 2. The x-axis represents day and y-axis represents number of completed jobs. The variation in completed jobs on each day is due to change in job demand and available FEs each day. In this mode FEs have a say in job allocation and scheduler looks after BT's interest. Assuming no gain in task completion time, the uncommitted choice empowerment strategy on average completes 4% less tasks than push strategy. In this scenario the assumption is that BT does not get any improvement in performance as a result of empowerment, as currently we do not have any method of measuring the degree of motivation that will result from application of empowerment. In reality it is believed that FEs will be motivated by having empowerment in job allocation and job execution processes. We do not use any monetary incentive for FEs, instead FE are rewarded by day off or early finish of the day. A motivated FE may complete job faster and may have time off as an incentive. The gain in this strategy comes from following factors.

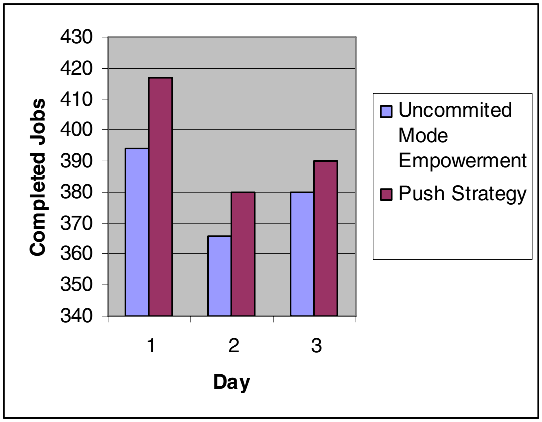


Fig. 2. Performance Comparison of Uncommitted Mode and Push Strategy

- Engineers have a say in job-allocation
- The scheduler looks after BT's interest
- If engineers report availability promptly, more efficient schedules are possible
- If engineers are motivated, they may complete jobs faster. There is incentive to complete jobs early.

The simulation results along with literature review on empowerment indicate that BT may improve performance by introducing empowerment. How much improvement in performance will be achieved by introducing staff empowerment can be in reality measured by conducting field trial.

The simulation data used here is based on jobs and FEs availability in a BT's region over period of three days.

7 Conclusion

In this paper we have discussed various empowerment research efforts and their effect on employee's performance. We have described the overall architecture of dynamic workforce simulator and various behaviours and strategies employed by agents realising this architecture.

We provide simulation result of using pull based (empowerment) and push job allocation strategies and their effect on performance. We have implemented a simulator for simulating BT dynamic scheduling problem. Pull strategy is just one of the functionalities provided by the simulator. The simulator allows the management to assess the impact of pull strategy on overall performance of the system.

This paper is concerned with the individual aspect of BT FEs empowerment. This is our ongoing work on agent based staff empowerment; currently we are conducting experiments on team based empowerment in order to enable FE to exercise empowerment during execution of their jobs through cooperation and coordination with other team members. Our future work will also explore mixed strategies employed by FEs, where some FEs will use pull strategies and other will use push strategies.

References

1. Wallerstein, N.: Powerlessness, Empowerment and Health. Implications for Health Promotion Programs. American Journal of Health Promotion, 197–205 (1992)
2. Wilkinson, A.: Empowerment Theory and Practice. Personnel Review 27, 40–56 (1998)
3. Rappaport, J., Swift, C., Hess, R.: Studies in Empowerment: Steps Toward Understanding and Action, pp. 73–79. Haworth Press, New York (1984)
4. Christopher, R.: Empowerment: the holy grail of health promotion. Health Promotion International 9(1), 39–47 (1994)
5. Wall, T., Leach, D.: What is Empowerment ? Institute of Work Psychology University of Sheffield, Sheffield, UK (2002)
6. Douglas McGregor, Theory of X and Y
7. Herzberg, F., Mausner, B., Snyderman, B.: The Motivation to Work. Transaction Publishers (2007)
8. Lesaint, D., Voudouris, C., Azarmi, N.: Dynamic Workforce Scheduling for British Telecommunications plc. Interfaces, 45–56 (2000)
9. Tsang, E.P.K., Virginas, B., Gosling, T., Liu, W.: Multi-agent based Scheduling for Staff Empowerment. In: Voudouris, C., Owusu, G., Dorne, R., Lesaint, D. (eds.) Service Chain Management, pp. 263–272. Springer, Heidelberg (2008)
10. Spreitzer, G.M.: Psychological empowerment in the workplace: dimensions, measurement, and validation. Academy of Management Journal 38(5), 1442–1465 (1995)
11. JADE, Java Agent Development Framework, `http://jade.tilab.com/`
12. FIPA, Foundation for Intelligent Physical Agent, `http://www.fipa.org/`
13. FIPA Request Interaction Protocol Specification, `http://www.fipa.org/specs/fipa00026/`
14. Block, P.: The Empowered Manager. Jossey-Bass, San Francisco (1987)
15. Berlew, D.E.: Managing Human Energy: Pushing versus Pulling. Executive Power, pp. 33–50. Jossey-Bass, San Francisco (1986)
16. Conger, J.A., Kanungo, R.B.: The Empowerment Process: Integrating Theory and Practice. Academy of Management Review, 471–482 (1988)

Customer Assistance Services for Simulated Shopping Scenarios

Florian Messerschmidt, Andreas D. Lattner, and Ingo J. Timm

Goethe-Universität Frankfurt, Information Systems and Simulation
Institute of Computer Science, P.O. Box 11 19 32, 60054 Frankfurt/Main, Germany
{lattner,timm}@cs.uni-frankfurt.de

Abstract. Providing helpful assistance in shopping scenarios can lead to satisfied customers with the potential for higher customer retention. Different hypotheses addressing the assistance functions and customer types are evaluated by statistical significance tests of agent-based simulation runs. Among others, the results show that the assistance functions lead to reduced shopping durations and a reduced number of forgotten items for our settings.

1 Introduction

Many works in the marketing research for the retail sector take the perspective of the market provider and aim at boosting the sales like, e.g., add-on selling. In this work, we decided to take the view to optimally assist the customers. Therefore, we wanted to develop assistance services for some of the important topics in this area and focused on indoor navigation and assistance for shopping.

Our approach with focus on the customers' needs has also many advantages for the consumer markets. Customers are more likely to go shopping in places where they may feel more satisfied and convenient. These positive perceptions can be increased by assisting them to get all the items they want to buy and not to forget important things. They also have a higher motivation to go shopping when they have little time, because they can do it in shorter time through an assisted navigation. This also gives the opportunity to serve more customers in a period of time, because they will only spend the necessary time for shopping. We decided to base our modeled consumer market on a discounter because it is a major group in Germany [1] and the markets are more comparable within this group. Moreover, it is a common situation in the everyday life.

In section 2 we first give a brief overview on some important aspects of the consumer market design and discuss related work. Then we describe in section 3 how we model the indoor environment in general and present our modeled consumer market. We also discuss the behavior of three different agent types in general as well as the distinct differences between them and introduce the implemented assistance services for navigation and shopping. In section 4 four hypotheses, experimental settings, and the evaluation are presented. Finally, in section 5 we discuss the results.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 173–182, 2009.

2 Related Work

The design of a consumer market is a complex area of intense economic research with influences from psychology. Many organizations like the GFK[1] are doing continuous research on, e.g., where to place which products of what price category, how to group products, and about how customers' tendencies are influencing the layout of consumer markets in general [2,1].

Another important fact in the area of shopping is the customer's use of shopping lists. Studies have shown that a majority of the people uses either a written or memorized form of shopping lists. Also of high interest is the result that only a minority of people (16%) sticks to their list which means that many people deviate from their purchase intentions [3]. In the following we briefly present related works which deal either with the topic of implementing a complete navigation assistance system or the simulation of some relevant aspects.

The e-sisst system was developed by the Fraunhofer Gesellschaft as a navigation guide with some assistance services [4]. e-ssist aimed to give navigational information in a consumer market and to assist the shopping with extra information as well as learning shopping lists. This system only allowed to navigate between two locations, but not to receive an optimized route for several locations.

YPOP is also an indoor navigation system which was recently evaluated in a mall [5]. The focus in this work is the combination of existing techniques or standards (e.g. maps in CAD format, localization by WLAN, etc.) and as well a flexible representation of different contexts. This is done in two ways: The first is to show the surrounding on the navigation device with scaled details depending on the localization quality. The second is that the shown context can be adapted to the situation and user, e.g., a fire fighter will be given different information in an emergency case than a regular customer will get while shopping [5].

The mobile shopping assistant of the "Future Store" [6] is a project by the Metro Group where they provide the customer with the possibility to have an electronic shopping list and use it on their cell phones while shopping. The list is deposited on the servers of the Metro Group and can be accessed via internet during the shopping. Extra information for the products can be also accessed via the cell phone using its camera to read the bar code.

Klügl and Rindsfüser [7] model a complete train station in Bern and use an agent-based simulation approach to analyze the pedestrian flow in detail. They use the observed situation in 2006 as bias to predict future capacity limits for this train station.

The relevance of eGuide / e-sisst and YPOP to our work is that both implemented a complete system which provides an indoor navigation. Furthermore, they provide various assistance services for shopping and giving extra information. Despite the similarities to our work, we followed a different approach for a navigation and shopping assistance. The Future Store is an interesting step towards more assistance for customers and shows that also large companies like the Metro Group recognizes its potential for the customer relationship. In our work

[1] Gesellschaft für Konsum, http://www.gfk.com

we research more extended assistance services. The work of Klügl and Rindsfüser deals with the problem of how to model an indoor environment for the purpose of simulation and presents different concepts which we here combined and modified. We decided to use an agent-based approach because of the natural way to model the scenario.

3 Customer Assistance Services

In the following we first describe the modeling of the consumer market as well as the different customer agent types with their distinct behavior. Subsequently, we present the concepts of the assistance services for navigation and shopping.

3.1 Indoor Model

The indoor model should be able to represent all objects and structures to design shopping scenarios. We decided to use a graph-based representation. The basic principle is that the graph consists of two sets of vertices and edges: One for the representation of paths and access points[2] modeled for the navigation and the other for items[3] and their accessibility.

The accessibility of items from certain access points is realized by a mapping function. Costs (distances) are assigned to the edges in a way that for all paths the triangular inequality is fulfilled. This is necessary to compute routes properly. Access points have a capacity indicating how many agents can use this way point at the same time. The model is represented by a graph $G = (V, E)$, with the set of edges E and the set of vertices V each consisting of two subsets:

The set of vertices V consists of
$V_I :=$ set of items; $V_A :=$ set of access points
with $V_I \cup V_A = V$ and $V_I \cap V_A = \emptyset$
The set of edges E consists of
$E_I :=$ set of edges from items to access points
$E_A :=$ set of edges between access points
with $E_I \cup E_A = E$ and $E_I \cap E_A = \emptyset$
The edges are characterized by
$\forall e \in E_I$ is $e \in V_I \times V_A$; $\forall e \in E_A$ is $e \in V_A \times V_A$
$\forall e := (v1, v2) \in E : v1 \neq v2$

The allocation of items to access points is given by the function *reachable* : $V_I \times V_A$. The reachable relation could be realized by a distance-based function. In our work, we assume this information as given such that for each access point it is known which items can be reached.

In the following, we describe a consumer market which was modeled by using the formalism and which is the setting for later simulations. As an example we

[2] The term "access point" refers to way points where certain items can be accessed.
[3] The term "item" refers to articles that can be purchased.

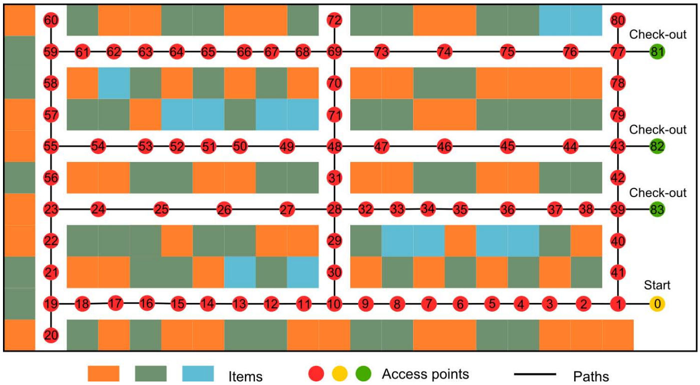


Fig. 1. Modelled consumer market

used a real existing consumer market. We reduced the more than 700 offered items to 76 for the model. The basic layout of the modeled consumer market is also available in simplified form, but the relations of the distances are still the same. This reduction of details brings on the one hand more clarity and on the other hand it reduces the computational complexity of the simulation. Thus, we tried to retain the characteristic structure of the consumer market (Fig. 1).

3.2 Customer Agents

Before presenting a description of the different customer types, we explain the general behavior of the agents. The ability to buy is modeled in the way that an agent is trying to shop at each visited access point all the available items, while items can only be successfully bought once. If the current item to be shopped is on the shopping list, it is bought with a probability of 95% which models to forget a fraction of the items even if they are on the list. If an item is not on the list, it is bought with a probability of 3%, thus, a spontaneous and a planned shopping behavior is modeled.

Another important behavior is the ability to escape a jam situation (i.e., if the agent could not move for a number of time units) which may occur due to the capacity limits for the access points. The used approach is that an agent chooses a random access point and tries to move to it before proceeding with the original route from that access point. The agent discontinues a simulation when it either tried five times in a row to escape a jam situation, or if it waited a total of 50 time units. By discontinuing the agent is immediately removed from the simulation and its occupied capacities are freed.

In our model, we distinguish three different customer agent types: The assisted, the experienced, and the inexperienced buyer. The *assisted buyer* is a buyer who is assisted by services for navigation and shopping. The navigation assistance which computes an optimal route is always active. The shopping assistance presented in the next section can be either enabled or disabled depending on the requirements of the scenario.

The *experienced buyer* is a buyer who has the knowledge about the locations of all the items in the consumer market and does not have to search for the individual items. Because of this it can get to the items on the shortest path, but an optimal arrangement of the items in a shortest route will rarely occur. In reality shopping lists also vary in some items, making it very unlikely to find the best route without assistance. It is also assumed that through regular shopping in the same consumer market a good route is chosen. This means that this route is not the optimum, but only differing from it by a small percentage. In this work the chosen route is a partial permutation of the optimal route. Therefore an optimal order for the items is computed first and then this order is changed through a permutation of 20% of its items (at least one permutation). The computed suboptimal shopping list order is processed by using the shortest paths between the individual items. The assisted and the experienced buyer always choose the check-out with the shortest waiting time.

The *inexperienced buyer* is a buyer who has no knowledge about the locations of the items in the consumer market. This type represents someone who enters for the first time this consumer market. This behavior is modeled by a route planning based on a depth-first search. With the depth-first search paths of different length from the longest to shortest are found. For the route planning every path between two items will be computed by a depth-first search. This creates a random search by which short and very long purchases are made possible. The choice of the check-out is also completely random, without any regard of waiting times. This type of agent cannot use any of the assistance services.

3.3 Assistance Services

In this section two assistant functions are presented, namely navigation and shopping assistance. Due to space restrictions, we leave out a third assistance function that directs the customer to some special offers in the market if long waiting time at the check-out queues is expected.

The navigation enables the agent to obtain an optimal route. Based on the shopping list a sequence is computed which represents a shortest route. It is computed once at the beginning of a simulation and can also be recomputed if there have been any changes to the route during the simulation, e.g., if there is need to escape a jam situation. For every route computation, the current location of the agent is used as start and one of the check-outs as end point. The used algorithm for the computation is presented in more detail in the following. The problem to reach n items on the shortest path (or in general with the lowest costs) relates to the Traveling Salesman Problem (TSP). Since this is an NP-complete problem, the use of exact computation algorithms, such as Branch and Bound algorithms, is limited. To be not restricted on the length of the shopping lists by high calculation complexity, we decided to use an approximate algorithm in this work. A well known algorithm to address the TSP is the 2-opt heuristic [8] where optimization is tried to be achieved by randomly exchanging two edges of a tour. A disadvantage of this heuristic is that the found minimum is not necessarily the global one. Therefore, we used in this work the 2-opt heuristic combined with an optimization through a simulated annealing (SA) algorithm[4]. This enables to escape a local optimum by the possibility to accept a worse tour with a decreasing probability during the computation. The SA algorithm [9] decreases the probability to accept a worse result during the optimization until it drops below a specified limit. Better results are always accepted. We use a combination of 2-opt and SA as discussed in detail by Meer [10]. We modified the criteria for accepting a worse result so that it also depends on the quality of the new result[5]. New tours are generated by the 2-opt heuristic and the decision to accept or dismiss the tour is made by the SA algorithm.

The *shopping assistance* is intended to suggest items for which a purchase intention might exist, although only a partial shopping list is available. This means that some items were forgotten on the list. Missing items should be detected and added to the shopping list as soon as possible. These items should be proposed due to the previously purchased items to be a recommendation based on the current context. A basis for such recommendations is provided by association rules. These rules describe relations between items, such as when the items "A" and "B" are purchased, then item "C" is also purchased with a probability of 70%. These statistical relationships between items can be identified automatically by association rule learning approaches, e.g., [11]. How to learn these association rules is out of the focus of this paper. In this work, association rules are defined without prior learning and used for the recommendation, in order to simplify an adaptation to the evaluation. The agent adds a recommended item to his shopping list with the same probability as the confidence value of the corresponding association rule. If this service is used by an agent, it examines after each successful purchase of an item whether a precondition of an association rule can be found and recommends when appropriate, a further item to be bought.

4 Evaluation

In the following we describe the four different hypotheses which are analyzed in this work and present the used experimental setting. Finally we present the evaluation and discuss the results.

4.1 Hypotheses

Hypothesis 1 (Shortened shopping time through navigation): It is assumed that through better route planning less time for a purchase is needed, compared to the same purchase without navigation. This advantage is neither obvious nor trivial, since each agent plans its route without any information about other agents. So possible jam situations cannot be prevented or foreseen.

[4] Depending on the number of items we chose different values for the annealing.

[5] We defined the quality as currentLength/newLength. Because this value is only calculated in the case of a worse tour quality is always < 1.

Hypothesis 2 (Improved pursuing of the purchase intention): It is assumed that through better route planning an improved pursuing of the purchase intention is achievable, as there are fewer opportunities to buy unwanted items, i.e., that fewer items are purchased spontaneously, for which there is no original purchase intention, and therefore would mean a deviation from the shopping list.

Hypothesis 3 (Less forgotten items): It is assumed that through the shopping assistance, fewer items with an actual purchase intention are forgotten to buy. This means that items with a purchase intention are missing on the list, but should be bought. Therefore, it should be attempted to add these items to the shopping list.

Hypothesis 4 (The existence of different capacity limits for different types of purchase agents): It is assumed that for each agent type a distinct capacity limit for the consumer market exists. This means that for each of the modeled purchase agents a different number can be actively shopping at the same time in the consumer market.

4.2 Experimental Setting

All hypotheses use the modeled consumer market as scenario for the experiments. In the implementation we distinguish between observed and non-observed agents. All agents which are not observed are only used to create a basic setting. They select a random starting access point and have a randomly generated shopping list with a length from 5 to 20 items. Additionally, non-observed agents representing the experienced buyer choose a percentage for permuting the optimal order of the shopping list between 0.2 and 0.4 randomly. Observed agents are regarded in detail for evaluation. All agents are moving at speed 1, which means that they can move 1 unit in distance in 1 unit of time.

For hypotheses 1-3 there is always only one observed agent actively shopping. The number of the non-observed agents is kept for each type and in all constant. With the observed agents of all three types, 50 simulation runs for the combinations shown in table 1 are made. Each scenario has three shopping list variations, this resulting in 5400 simulation runs for the evaluation of each hypothesis. The assisted agent is further called "agent type 0", the experienced agent "agent type 1" and the inexperience agent "agent type 2".

In hypothesis 1 and 2, the assisted agent uses only the navigation assistance. For hypothesis 3 it additionally uses the shopping assistance. We assume that

Table 1. Experimental setting for the scenarios

Scenario	01	02	03	04	05	06	07	08	09	10	11	12
Length of shopping lists	10	10	10	15	15	15	20	20	20	30	30	30
Agents (all)	10	20	30	10	20	30	10	20	30	10	20	30
Agents type 0 (assisted)	3	6	10	3	6	10	3	6	10	3	6	10
Agents type 1 (experienced)	3	6	10	3	6	10	3	6	10	3	6	10
Agents type 2 (inexperienced)	4	8	10	4	8	10	4	8	10	4	8	10

40% of the items were forgotten to add to the shopping list. This means that a list of ten items should actually consist of 14 items and thus these four forgotten items should be bought. The recommended items are determined by a set of manually created association rules. To identify an effect by the shopping assistance, all agents store the number they bought from those forgotten items.

In the experimental setting for hypothesis 4, only observed agents are used. The starting access points are chosen randomly and the shopping lists are also randomly created with a length from 10 to 20 items. Thus, the different types of agents use the identical set of shopping lists in the same scenario. Additionally, all assistance services are enabled for the assisted agents. The simulation starts with 20 agents for each agent type and each setting is repeated 50 times. If more than 20% of the agents quit the simulation on average in the 50 runs, the scenario is viewed as unsuccessful and otherwise as successful. In case of a successful scenario the number of this type of agent is increased by 5 agents and rerun. Otherwise, no further simulation runs are performed for this type.

4.3 Results

Due to the lack of simulation tools which are capable of representing our customer agents as well as graph-based indoor model directly, we implemented a time-discrete simulation-tool ourselves. The data on which the following results are based has been collected using this tool. The hypotheses 1-3 have been evaluated using a one-sided t-test to verify a significant difference between the agent types for a given error level $\alpha = 0.05$. In hypothesis 1, we tested if the assisted agent needed less time for the purchase compared to the other types of agents. The two null hypotheses were $H_0 : T_0 \geq T_1$ and $H_0 : T_0 \geq T_2$ while the alternative hypotheses were $H_1 : T_0 < T_1$ and $H_1 : T_0 < T_2$. T specifies the needed time for the purchase and 0, 1, 2 corresponds to the types of agents (assisted, experienced, inexperienced), i.e., in the first case the null hypothesis (H_0) is that assisted agents need equal or more time for shopping than inexperienced agents while the alternative hypothesis (H_1) is that assisted agents need less time. The results clearly show that the assisted agent is always significantly faster than the experienced and inexperienced agent.

In hypothesis 2, we tested if the assisted agent bought fewer items with no actual purchase intention compared to the other types of agents. The two null hypotheses were $H_0 : K_0 \geq K_1$ and $H_0 : K_0 \geq K_2$ while the alternative hypotheses were $H_0 : K_0 < K_1$ and $H_0 : K_0 < K_2$. K specifies the number of falsely bought items. The results show that the assisted agent always significantly bought fewer items without an actual purchase intention than the experienced and inexperienced agent. This can be explained by the significantly shorter routes of the assisted agent type. With shorter routes there are fewer possibilities to buy extra items without purchase intention. Thus, the assisted agent gains an advantage through its navigation assistance.

In hypothesis 3, we tested if the assisted agent forgot fewer items compared to the other types of agents. The two null hypotheses were $H_0 : V_0 \geq V_1$ and $H_0 : V_0 \geq V_2$ while the alternative hypotheses were $H_0 : V_0 < V_1$ and $H_0 :$

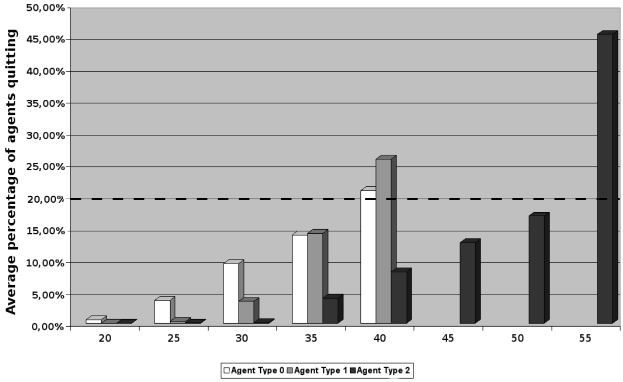


Fig. 2. Fraction of agents quitting the simulation (hypothesis 4)

$V_0 < V_2$. V specifies the number of forgotten items. The results of the statistic tests show that the assisted agent always significantly forgot fewer items with an actual purchase intention than the experienced and inexperienced agent. Two aspects contributed to this good performance of the assisted agent type. First, an identified forgotten item was added to the shopping list with a probability of 80%. This leads also to a higher probability for getting bought by the agent compared to an item which is not on the shopping list. And in case the item was added to the shopping list, the route was recomputed to include the new item.

In hypothesis 4, it could be shown that limits for each different type of agents exist. The observation that more agents of the inexperienced type can be active at the same time is based on the randomized route planning. The navigation is based on a depth-first search which has no fixed routes between two access points. It is expected that these agents will significantly less likely choose the same way, and use all possible paths more evenly. Thus they are more distributed in the consumer market, and therefore fewer blocking situations occur. Contrary the used paths between to access points for agent types 0 and 1 are based always on the shortest paths, i.e., there are preferred ways between access points. There can also be access points which are used in several shortest paths, e.g., crossings. This all leads to a less even distribution over the whole consumer market, resulting in more possible blockades and discontinuing agents. Figure 2 shows the percentage of aborted agents for each scenario.

5 Conclusion

We presented a customer oriented point of view and found hints that by an assistance for navigation and shopping some of the major customer's topics when shopping can be improved. We modeled three types of purchase agents representing archetypes of shoppers and described two assistance services. One used a combination of a 2-opt heuristic with a simulated annealing optimization to compute an approximately shortest route for the purchase. The other assistance service showed how a shopping assistance can be build using association rules.

We showed that by using the navigation assistance a significant decrease in the time needed for a purchase can be achieved. This decrease is especially remarkable because it is even very significant when compared with the only slightly slower experienced agent type. We also showed how the shorter route and respective the less needed time for a complete purchase also lead to fewer falsely bought items without an actual purchase intention. Further results indicate that by using the shopping assistance, forgotten items can be added to the shopping list and thus, decreasing the number of items forgotten to buy with an actual purchase intention. Finally, we showed that due to the different methods of route planning, we also get varying distributions of the agents in the consumer market. This results in a different number of jam situations which lead to an increase in canceling agents by increasing their numbers.

References

1. ZMP/CMA: Nachfrage privater Haushalte nach Nahrungsmitteln 2005. ZMP Zentrale Markt- und Preisberichtstelle für Erzeugnisse der Land-, Forst-Ernährungswirtschaft GmbH (January 2006)
2. Vogler, T.: Transaktionskostentheoretisch fundierte Untersuchung zum Spacemanagement in Supermärkten. PhD thesis, Wirtschafts- und Verhaltenswissenschaftlichen Fakultät der Albert-Ludwigs-Universität Freiburg i. Br (2006)
3. Hayn, D., Empacher, C., Halbes, S.: Ergebnisse einer Literaturrecherche. In: Trends und Entwicklungen von Ernährung im Alltag. Materialienband, vol. 2. Institut für sozial-ökologische Forschung (ISOE), Frankfurt am Main (2005)
4. Kirste, T.: Situation-aware personal assistance - Systemparadigmen & Anwendungsbeispiele, Präsentation des Fraunhofer Instituts für Graphische Datenverarbeitung (November 2004)
5. Tuomisto, J., Rajamäki, J.: Ypop indoor navigation and service information system for public environments. In: ICCOM 2007: Proceedings of the 11th WSEAS International Conference on Communications, Stevens Point, Wisconsin, USA, World Scientific and Engineering Academy and Society (WSEAS), pp. 208–213 (2007)
6. Metro, A.G.: Future store (July 2008), http://www.future-store.org
7. Klügl, F., Rindsfüser, G.: Large-scale agent-based pedestrian simulation. In: Petta, P., Müller, J.P., Klusch, M., Georgeff, M. (eds.) MATES 2007. LNCS, vol. 4687, pp. 145–156. Springer, Heidelberg (2007)
8. Englert, M., Röglin, H., Vöcking, B.: Worst case and probabilistic analysis of the 2-opt algorithm for the TSP: extended abstract. In: SODA 2007: Proceedings of the 18th annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, PA, USA, pp. 1295–1304. Society for Industrial and Applied Mathematics (2007)
9. van Laarhoven, P.J., Aarts, E.H.: Simulated Annealing: Theory and Applications. Springer, Netherland (September 1987)
10. Meer, K.: Simulated annealing versus metropolis for a TSP instance. Inf. Process. Lett. 104(6), 216–219 (2007)
11. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules, 3rd edn. Readings in database systems, pp. 580–592 (1998)

TRES: A Decentralized Agent-Based Recommender System to Support B2C Activities

Domenico Rosaci and Giuseppe M.L. Sarné

DIMET, Università Mediterranea di Reggio Calabria
Via Graziella, Località Feo di Vito
89122 Reggio Calabria, Italy
{domenico.rosaci,sarne}@unirc.it

Abstract. The increasing relevance assumed by the E-Commerce in the Web community is attested by the great number of powerful and sophisticated tools developed in the last years to support traders in their commercial activities. In this scenario, recommender systems appear doubtless as a promising solution for supporting both customers' and merchants' activities. In this paper, we propose an agent-based recommender system, called TRES, able to help traders in Business-to-Consumer activities with useful and personalized suggestions based on interests and preferences stored in customers' profiles, adopting a fully decentralized architecture that suitably introduces in the system both scalability and privacy protection.

1 Introduction

In the last years, E-Commerce (EC) activities are playing a key role in the Web and this is attested by the increasing number of the commercial transactions electronically performed. In particular, a great attention has been reserved for the Business-to-Consumer (B2C) activities, comparable to the retail trade of traditional commerce [7,15]. In the previously introduced scenario, recommender systems [10,14] appear as a promising solution for both merchants and customers. The merchants can be advantaged by the improvement of the performances for their e-Commerce sites; the customers are supported in their decision-making process by means of some useful suggestions about objects, products, or services potentially interesting for them. To carry out such an activity, recommender systems need to exploit a suitable representation (*profile*) of the customer's interests and preferences. This representation can be automatically derived by monitoring and interpreting the large amount of information that users spread during their Web trading activities by considering their purchase histories, Web logs, cookies, etc. and/or by using explicit ratings directly provided by the user. A well-known solution to implement recommender systems is represented by the agent technology in which *software information agents* [8]) autonomously and proactively perform some tasks on the behalf of their human users. The

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 183–192, 2009.

agent-based technology has been widely used in the EC context, also for providing users with recommendations [12], and all these systems exploit the presence of a customer profile, representing how the customers evaluate the different products and product categories, even mediating on heterogenous information spaces [6]. An example of these systems is EC-XAMAS, presented in [9]. However, none of these systems evaluate the customer's interest for a product considering each different stage of the EC in a different way. In other words, if a customer accesses to a product only to read its description, his interest for that product has to be considered very different from the case the customer actually buys the product.

In the past, some behavioural models have been introduced to capture the different phases carried out in an EC process [1,3,4]. These phases involve activities performed by the customer before, during and after the visit of merchant sites. We point out that in this paper we do not deal with the activity of the customer for finding potentially interesting sites, but only with those activities performed by the customer when he accesses to a given site. Our purpose is to support the merchant of a site in generating useful recommendations for the customer, considering the three usual stages performed by the customer himself during the interaction with the merchant, namely: (*i*) a site visiting stage, where the customer visits a description of a product; (*ii*) a negotiation stage, where the customer negotiates the price of a product with the merchant and (*iii*) a buying stage, where the customer actually buys the product, sending an order.

In particular, we base our approach on a multi-agent architecture, called Trader REcommender Systems (TRES). Specifically, in TRES each trader (customer or merchant) is associated with a software agent that constructs and stores an internal profile to take into account his whole Web trading history. The agents exploit their user's profiles in their interaction, to make the merchants capable to generate effective recommendations. The architecture we have adopted is fully decentralized, giving to each merchant the capability to generate recommendations without requiring the help of any centralized computational unit. This characteristic, on the one hand, makes the system scalable with respect to the size of the users' community. In Section 3, we show that the cost of our recommendation algorithm linearly increases with the number of customers present in the system. On the other hand, the privacy of each customer is preserved, since the merchant retrieves information about each customer simply monitoring the customer behaviour in visiting his site.

To verify the effectiveness of TRES for generating suitable suggestions, we have implemented an agent platform, called *Trader Agents Community* (TAC), to support customers and merchants during all the three stages of a B2C process described above. TAC agents exploit their profiles to take care of the traders in a personalized and homogeneous way by means of reciprocal message-based interactions. As a result, customers will be provided during their EC-site visits with personalized Web presentations built on-fly exploiting the generated suggestions. The paper is organized as follows: in Section 2 we present the TAC framework and the activities performed by the agents in order to monitor customers' behaviours. Section 3 describes the TRES recommendation algorithm. In Section 4 some experiments are presented to evaluate our proposal. Finally, in Section 5, we draw some final conclusions.

2 An Overview of the TAC Framework

In this section, the TAC framework will be described in detail. In such a framework each customer C (resp. merchant M) is associated to a personal agent c (resp. m). Below we will describe the knowledge representation model, the structure of each agent and the trading support provided by the agents.

2.1 Representation of Objects and Categories of Interest

The TAC agents support traders in their commercial tasks for products that meet their interests. To this aim, all the TAC agents share the same *Catalogue* $\mathcal{C}$, representing the common agent knowledge, that describes each *product category* (*pc*). Usually, each Web page includes more product instances belonging to different product categories. The TAC agents build their users' profiles by monitoring all the product instances and product categories accessed by the customers in order to measure the associated interests. Moreover, to determine collaborative filtering recommendations, each merchant agent also computes the similarity among its visitors by exploiting the values of interest shown in the offered product instances.

Currently, the TAC catalogue adopts the six digit edition of the *NAICS* coding [11], an official, public, hierarchical classification used in North America to classify businesses in categories. The catalogue is implemented by means of a simple XML-Schema [2] where a product category is described by using the notion of *element* and the product instances are represented by XML *element instances*.

2.2 The Agents

Each generic user (customer or merchant) U is associated to a personal agent a, which manages the *User Profile* (UP) of U. UP stores user information in a tuple $\langle UD, WD, BD \rangle$, where:

- the *User Data* (UD) are personal U's data as name, address, etc.
- the *Working Data* (WD) contains two system parameters called *Memory* (ω) and *Pruning* (σ) thresholds, that will be described in the following, and the current product catalogue $\mathcal{C}$, that is a list of product instances.
- the *Behaviour Data* (BD) of a user contains some parameters that describe the past behaviour of the user. Each parameter has a different meaning if the user is a customer or if he is a merchant. In particular, in the case U is a customer (resp. a merchant), BD is an array that contains, for each product category pc accessed by the customer (resp. in which the merchant offers some products), two components, called pcR and L_{pc}. The first component

Table 1. Parameters contained in the Behavioural Data BD

Name	Customer	Merchant
LVD	last visit date to pi	last access of a customer to pi
PD	Product Data of pi	Product Data of pi
L	List of merchants that sell pi	List of customers that accessed to p
piR	Interest rate for pi	Interest rate for pi

pcR represents the interest in the category pc. The second component L_{pc} is a list of elements, where each element l_{pi} of L_{pc} is associated with a product instance pi belonging to the category pc, visited by the customer (resp. offered by the merchant). In particular, the element l_{pi} contains some data relative to the behaviour of the customer in accessing pi (resp. the behaviour of all the customers that accessed pi) and in particular the parameter piR that represents the interest in that product instance. These data are shown in Table 1.

Note that the meaning of the interest rates pcR and piR is different for the customer and the merchant. In particular, the pcR of a product category pc for the customer represents the customer's interest for pc, while for the merchant it represents the interest shown by all the customers that visited products belonging to pc in the merchant's site. Analogously, the piR of a product instance pi for the customer represents the customer's interest for pi while for the merchant it represents the interest of all the customers that visited pi in the merchant's site.

The behaviour of an agent is as follows. When a customer visits the EC site associated with a merchant, both the customer's and the merchant's agent monitors the customer's behaviour in his/her visit. Then, as a consequence of such a monitor activity, a customer (resp., a merchant) agent could take into account, with respect to a product instance pi belonging to a product category pc, the interest rate piR for that instance and the interest rate pcR for that category. More in detail, to show how an agent computes pcR (resp. piR), we adopt the following formulation:

$$piR = \omega \cdot \frac{\rho_i}{log_{10}(10+N_i)} + (1-\omega) \cdot \frac{piR}{log_{10}(10+\Delta)}$$
$$pcR = \omega \cdot \frac{\rho_i}{log_{10}(10+N_i)} + (1-\omega) \cdot \frac{pcR}{log_{10}(10+\Delta)}$$

This formulation shows that piR (resp. pcR) is computed as a weighted sum of two contributions. The first contribution, weighted by ω, represents the effect of a customer's visit to the involved product instance (resp. category) in the current stage i. This contribution is computed as the ratio between the parameter ρ_i (belonging to [0..1], that measures the importance given by the customer to the stage i ($\rho_i = 1$ means maximum importance), and the value $log_{10}(10+N_i)$, where N_i is the number of times the same stage i has been performed for the same product (resp. category) in the same day. This means that repeated accesses to the same product instance (resp. category) in the same stage are considered with a (logarithmic) decreasing importance, since they are not actually "new" accesses to the product (resp. category). The second contribution, weighted by $1 - \omega$, represents the effect to the past activities of the customer with respect to the involved product instance (resp. category). High values for the ω parameter give more relevance to the most recent access and vice versa. Moreover, Δ is the temporal distance expressed in days between the date of the current visit and the last visit LVD with respect to the visited pi (resp., pc). Note that the logarithmic decreasing we have adopted in the formulas above is an arbitrary choice, fruitfully tested in our experiments.

2.3 The Trading Support

TAC is a message-based framework designed to transfer, in a consistent and efficient way, the business information in order to permit of supporting and monitoring the activities carried out in a B2C process by merchant and customer agents. The different activities occurring within a B2C process require the exchange of several message typologies that will be briefly described below. Note that, in the adopted formulation, the first subscript of a message identifies the sender while the second identifies the receiver. Moreover, we denote by *data* an XML document, whose content is structured in three sections: *(i) Header*, containing sender, receiver, EC stage, and product identifier; *(ii) Products*, that stores the product data; *(iii) Financial*, relative to all the financial information needed to perform a payment.

- $INF_{x,y}(data)$: it requires/provides commercial information about a product;
- $REQ\_INV_{c,m}(data)$: it requires an invoice for a product offered by a merchant m;
- $INV_{m,c}(data)$: it contains the invoice required with $REQ\_INV_{c,m}(data)$;
- $PP_{x,y}(data)$: it is used to negotiate any commercial detail not fixed or specified (i.e., the price for products without fixed price);
- $PO_{c,m}(data)$ (resp., $PO\_A_{m,c}(data)$ or $PO\_R_{m,c}(data)$): it is the purchase order with respect to $INV_{m,c}(data)$ (resp., the notify of acceptance or rejection);
- $MTO_{c,m}(data)$ (resp.,$MTO\_A_{m,c}(data)$ or $MTO\_R_{m,c}(data)$): it notifies that the payment has been performed (resp., received or not received);

EC-site Visit Support stage (s=1). During an EC-site visit a customer sends the message $INF_{c,m}$ to a merchant for requiring commercial information about an offered product and he/she in his/her turn will answer with another message $INF_{m,c}$ containing the required information.

Negotiation Support stage (s=2). In this stage are fixed all the trading details as price, quantity, delivery modality and so on. More in detail, this stage starts with the messages $REQ\_INV_{c,m}$ and $INV_{m,c}$ exchanged between the traders. If the customer accepts the customer's commercial proposal contained in

$INV_{m,c}$, then this stage ends. Otherwise, if it is possible, the two traders use in an alternative manner $PP_{x,y}$ messages to negotiate all the trading details; when an agreement has been achieved, then the $REQ\_INV_{c,m}$ and $INV_{m,c}$ messages will be again exchanged.

Purchase Support stage (s=3). After the "Negotiation" step, if the customer wants to purchase a products he/she sends the message $PO_{c,m}$ to the merchant that can accept/refuse by means of $PO\_A_{m,c}(data)/PO\_R_{m,c}(data)$. If the purchase order is accepted, then the customer performs the payment and informs the merchant with a $MTO_{c,m}$ message; after this, the customer receives a $MTO\_A_{m,c}$ (resp. $MTO\_R_{m,c}$) message to confirm he has received (resp. he has not received) the payment.

3 The Recommendation Algorithm

In this section we present the recommendation algorithm exploited by TRES to generate useful suggestions by using both a *content-based* and a *collaborative filtering* approach. When a customer visits the site of a merchant, the algorithm is executed by the merchant's agent to generate suggestions for the customer. The algorithm exploits the information stored into both the merchant's agent and the customer's agent and generates a list of product instances contained in the merchant's site appearing as the most promising to meet the customer's interests.

The behaviour of the algorithm is represented by the function `Recommend` (Figure 1). The input of this function is the customer's agent c and its outputs are the lists $L3$ and $L4$ of product instances exploited by the merchant m to build an on-fly personalized Web site presentation for c. Within this function, the auxiliary function `extract_pc` is called; it receives as input the BD section of the m's profile and returns the list $L1$ containing those product categories belonging to the catalogue $\mathcal{C}$ of the merchant m. Then m sends the list $L1$ to c by using the function `send`, while the function `receive` waits for the response of c, consisting in the list $L2$. This list includes the v product categories that better

```
void Recommend(customerAgent c, ListOfProductInstancesL3, ListOfProductInstancesL4) {
    ListOfProductCategories L1=extract_pc(m.UP.BD);
    send(L1,c.Ad);
    ListOfProductCategories L2=receive( );
    ListOfProductInstances L3=contentbased_pi(L2, m.UP.BD, y);
    ListsOfCustomersInterests PC[ ]=customersInterests(m.UP.BD, v);
    ListOfProductInstances L4=collaborativefiltering_pi(L2, PC[ ], m.UP.BD, z, x);
    return;
}

ListOfProductCategories categoriesOfInterest(ListofProductCategories L1) {
                        ListOfProductCategories L5=extract_pc(c.UP.BD);
                        ListOfProductCategories L6=intersection_pc(L1, L5);
                        ListOfProductCategories L2=select_pc(L6,v);
                        return L2;
}
```

Fig. 1. The TRES recommendation algorithm

meet the interests of the customer (where v is an integer parameter arbitrarily chosen by c). When $L2$ is received, the function `contentbased_pi` is called and returns the list $L3$ containing the first y product instances having the highest rates for each of the v product categories stored in $L2$ (also y is a parameter arbitrarily set by m).

The next step deals with the generation of collaborative filtering recommendations. First, the algorithm constructs the array of lists PC, where each array element is a list associated to a customer i that accessed the site in the past and contains the v most interesting product categories. To this aim the function `customersInterests` is called, receiving the BD data of the profile of m and the integer v as input. Each array element $PC[i]$ is a list constructed *(i)* by computing for the user i the average a_{pc} of the piR values of all the product instances for each product category pc (this average represents a global measure of the agent interest in pc) and *(ii)* by ordering all the a_{pc} values in a decreasing order and *(iii)* by selecting, for each user, the v product categories having the highest a_{pc} values.

When PC has been computed, the function `collaborativefiltering_pi` is called; it receives in input the list $L2$ (provided by c), the array of lists PC, the BD section of the merchant agent's profile and the two integers z and x, arbitrarily set by m. This function exploits PC to compute the similarity degree between c and the customers that have interacted with m in the past; this measure will be used by m to select the z agents most similar to c. Thus, for each product category in $L2$ considered by c, all the product instances having the higher piR are selected and inserted in the list $L4$. The list $L4$ contains, for each of the z customers most similar to c, the x product instances with the highest rate. More in detail, the similarity between the customer c and a generic customer k that visited in the past the merchant's site is computed as the average of the contributions l_{pc} associated with the product categories common to the list $L2$ provided by the customer c and the list $PC[k]$ associated to the customer k in the merchant's profile. For example, suppose that the product category pc belongs both to $L2$ and $PC[k]$; also suppose that the interest rate of pc in $L2$ is equal to r while the interest rate of pc in $PC[k]$ is equal to s. In this case, the contribution l_{pc} of pc to the similarity between c and k is equal to $|r - s|$. The overall similarity between c and k is computed as the average of all the l_{pc} values.

On the customer agent side, when the list $L1$ coming from the merchant agent m is received, the function `categoriesOfInterest` is executed. This function calls the function `extract_pc` to obtain the list $L5$ containing the product categories of interest for c. After that, the function `intersection_pc` is called and the intersection $L1 \cap L5$ is computed. The function `select_pc` receives as input the list $L6$ and an integer v and then it orders $L6$ in a decreasing order based on the pcR value of the customer's profile; finally, it returns the first v product categories. The resulting list is the new $L2$, returned as the output of `categoriesOfInterest`.

In order to preserve customer's privacy, we observe that TRES faces such an issue in a simple and effective way. Specifically, the customer sends to the merchants only the selection of interesting category it has internally performed, without revealing its profile. In other words, the selection of the interesting category is done on the customer side. On the other hand, the merchant's generates the collaborative filtering recommendations without revealing to the customer any data relative to other customers, operating in this case on the merchant side.

Now we analyze how the time cost required by TRES to perform the recommendation task. In this analysis, we denote by n, p and q, the number of customers, the number of different categories present in the catalogue, and the number of maximum product instances in a category, respectively.

More in detail, the function `extract_pc` computes the product categories of interest for c by comparing the number of items in the site catalogue with those present in the customer's profile presenting a cost $\mathcal{O}(p^2)$. The function `contentbased_pi` examines q product instances for each one of the v product categories contained in $L2$. By considering that in the worst case v is equal to p, the cost of this function is $\mathcal{O}(p \cdot q)$.

The function `customersInterests` considers q product instances for each one of the v product categories selected by c for each of the n (in the worst case) users. The cost of this function is $\mathcal{O}(n \cdot p \cdot q)$.

Finally, the function `collaborativefiltering_pi` computes the similarity between the customer c and the other p customers by analyzing the v product categories selected by c with the product categories stored in the profiles of the other customers. The cost of `collaborativefiltering_pi` is thus $\mathcal{O}(n \cdot p^2 + p \cdot q)$.

By considering that the number of product instances of a seller is usually greater than the number of product categories included in the catalogue, it is reasonable to assume that p is greater or equal to q. In this case, the computational cost of the function `Recommend` results mainly influenced by the cost of the function `customersInterests` and can be assumed as $\mathcal{O}(n \cdot q^2)$.

On the customer side the computational cost of `productOfInterest` is $\mathcal{O}(p^2)$. In fact, the function `extract_pc` and `intersection_pc` have a computational cost $\mathcal{O}(p^2)$, while the function `select_pc` is driven by the computational cost of the used sort algorithm that we can assume as $\mathcal{O}(p \cdot logp)$.

4 Experiments

In this section, we present some experiments devoted to show the effectiveness of TRES to generate useful suggestions for supporting users during their B2C activities. The experiments presented below have been realized by using a TAC prototype, developed in JADE [5]).

For this experiment we have *(i)* built a family of 18 XML EC Web sites by using the NAICS coding as common vocabulary, represented by a unique XML Schema, with 760 product instances belonging to 9 NAICS categories and *(ii)* monitored a set of real users in their B2C activities within the TAC framework. The first 9 sites has been used to obtain an initial profile of the customers' interests and preferences without to exploit any recommendation support and to determine the value of the ρ parameters. Based on such profiles and ρ values

Table 2. Performances of different TRES compared to *EC-XAMAS* in the generation of suggestions (global/content-based/collaborative filtering)

	TRES	*EC-XAMAS*
Pre	0.874/0.808/0.847	0.663/0.648/0.655
Rec	0.835/0.812/0.824	0.613/0.584/0.611
F	0.841/0.804/0.831	0.636/0.605/0.633

the recommendations have been generated by the merchant agents relatively to the other 9 sites.

We have compared the performances of TRES with a classical recommender system approach, namely EC-XAMAS [9]. The main difference between EC-XAMAS and TRES is that EC-XAMAS does not consider the different EC stages to compute the interest rates. Instead, both TRES and XAMAS needs an initialization process to collect user interests

To evaluate the results of the experiments, we have inserted in a list, called A, the product instances suggested by the merchant agent and in a list, called B, the corresponding customer's choices. We have compared the corresponding elements in the two lists in order to measure the effectiveness of the proposed suggestions. In particular, we have adopted some standard performance metrics, namely precision, recall and F-measure [13]. Precision is defined as the share of the product instances actually visited by the customer among those recommended by the system. Recall is the share of the product instances suggested by the system among those chosen by the customer. F-Measure represents the harmonic mean between Precision and Recall.

The performance of the content-based and collaborative filtering components have been considered both in an integrated way and separately for the two approaches. The parameters v, y, z and x of TRES have been set to 2, 3, 2 and 3, respectively (see Section 3). In terms of results (see Table 2) TRES outperforms the approach EC-XAMAS with respect to all the three measures. In particular, the F-measure of TRES is better than EC-XAMAS with respect to global, content-based and collaborative filtering component of about 32 percent.

We argue that the better results of TRES with respect to EC-XAMAS are due to the more sophisticated modelling of the customer's behaviour of TRES, that consider all the stages of the B2C process, while EC-XAMAS model the whole process in a uniform way. This ppropriate modelling is usefully exploited by TRES when generating recommendations, that thus result very effective.

5 Conclusion

This paper presents a recommender system, called TRES, able to act both as content-based and collaborative filtering recommender, to support traders in their B2C activities in a suitable and personalized way, taking into account customers' interests and preferences. The customers' privacy is assured since the customer does not send any private information to the merchant, but autonomously select interesting items from the catalogue sent by the merchant. Some interesting results have been obtained by experimental simulations. The effectiveness of the suggestions generated by TRES are resulted better than a traditional recommender system that does not consider the different trading phases involved in B2C activities. Moreover, the fully distributed architecture makes the performances of TRES highly scalable in heavily accessed B2C environments.

References

1. Engel, J.F., Blackwell, R.D., Miniard, P.W.: Consumer Behaviour. The Dryden Press, London (1995)
2. Extensible Markup Language (XML) Schema (2008), http://www.w3.org/XML/Schema
3. Feldman, S.: The Objects of the E-Commerce, Keynote Speech at ACM 1999 Conf. on OOPLSA, Denver (1999), http://www.ibm.com/iac/oopsla99-sifkeynote.pdf
4. Guttman, R.H., Moukas, A., Maes, P.: Agents as Mediators in Electronic Commerce. Electronic Markets 8(1) (1998)
5. Java Agent DEvelop. framew (JADE) (2008), http://jade.tilab.com/
6. Jung, J.J.: Ontological Framework Based on Contextual Mediation for Collaborative Information Retrieval. Information Retrieval 10(1), 85–109 (2007)
7. Kauffman, R.J., Walden, E.A.: Economics and Electronic Commerce: Survey and Directions for Research. Int. J. Electr. Com. 5(4), 5–116 (2001)
8. Maes, P.: Agents that Reduce Work and Information Overload. Commun. ACM 37(7), 30–40 (1994)
9. De Meo, P., Rosaci, D., Sarnè, G.M.L., Ursino, D., Terracina, G.: EC-XAMAS: Supporting E-Commerce Activities by an XML-based Adaptive Multi-Agent System. Appl. Artif. Intell. 21(6), 529–562 (2007)
10. Montaner, M., Lopez, B., de la Rosa, J.L.: A Taxonomy of Recommender Agents on the Internet. J. on Web Semantics (JWS) 19(4), 285–330 (2004)
11. North America Industry Classifications (NAICS) (2008), http://www.census.gov/naics/2007/index.html
12. Palopoli, L., Rosaci, D., Ursino, D.: Agents' Roles in B2C E-Commerce. AI Commun. 19(2), 95–126 (2006)
13. van Rijsbergen, C.J.: Information Retrieval. Butterworth (1979)
14. Wei, K., Huang, J., Fu, S.: A Survey of E-Commerce Recommender Systems. In: Proc. of the 13th Int. Conf. on Service Systems and Service Management, Washington, DC, USA, pp. 1–5. IEEE Computer Society, Los Alamitos (2007)
15. Zwass, V.: Electronic Commerce and Organizational Innovation: Aspects and Opportunities. Int. J. Electron. Com. 7(3), 7–37 (2003)

Applying Multi-Agent System Technique to Production Planning in Order to Automate Decisions

Mats Apelkrans[1] and Anne Håkansson[2]

[1] Jönköping International Business School,
Department of Informatics, Jönköping, Sweden
Mats.Apelkrans@jibs.hj.se
[2] Department of Computer and Systems Sciences,
Stockholm University, Kista, Sweden
Anne.Hakansson@dsv.su.se

Abstract. Coordinate and deliver information is vital for the financial and operational success of a company. The information is used for understanding and evaluating performance of a manufacturing company and making decisions based on incoming information. Information about orders but also parts to be purchased, assembled for the final product to be delivered, can streamline the production line to provide good quality products in the right time and to right costs at highest profit. For profit, costs are cut by reducing storage and searching for lowest price from established suppliers and providers on web as well as handling production planning automatically. To increase profit, we apply a multi-agent technique to production planning, which can automate business decision-making for the production line. The agents handle incoming orders, the production line, and search for information about the products at the intranet and the extranet. The outcome is decisions about the production line. The multi-agent solution becomes a complement to the production planning brought about by the company's enterprise resource planning system.

Keywords: Intelligent agents, Meta-agents, Multi-agent systems, Business Intelligence, Logistics, Supply chain, ERP.

1 Introduction

The goal for manufacturing companies is to be competitive, which can be achieved by delivering the right products in the right time with good quality and at a competitive price. An important constituent is to produce and deliver information supporting the physical value chain, which is a hard topic studied in the discipline of Business Informatics. Business Informatics is a broad research area and includes topics such as Information Logistics, Enterprise Systems, Enterprise Resource Planning (ERP), Business Intelligence (BI) and Business Processes.

Usually, Business Informatics includes information technology, which is a portion of applied computer science. This technology can be improved by using multi-agent technology where information management, like information logistics can be combined with multi-agent technique. For example, combining the concept of an

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 193–202, 2009.

Information Logistics Process (ILP, discussed in section 1.2) with multi-agent technology, where ILP can be a process of producing information output from given input information [3] using a multi-agent system. To automate the process, ILP needs some kind of knowledge. This knowledge can be provided by intelligent agents since those support the possibility of collecting information and providing the ILP with the knowledge needed for handling ILPs knowledge more intelligible and automatically. Looking at the ILP and multi-agent systems (MAS) we find that there are similarities between the ILP's and AI's behaviour because each intelligent agent's behaviour can be thought of as an ILP handling information between several sources with given input and output [3]. Another automation proposal is to use ILP processes to configure an ERP system [2].

In this paper, we apply a multi-agent technique to production planning in order to automatically make decisions in product line. The intelligent agents work between the Enterprise system within a manufacturing company and the web to search, find and handle information. To reason with the information from intelligent agents, meta-agents are utilized. The meta-agents are responsible for automate the decisions. The meta-agents reason with other agents to handle the orders and plans but also notify the users about the problems in the multi-agent system.

2 Related Work

Looking at the business process perspective, many papers are written in order to optimize the work-flow in a manufacturing situation, e.g., Adam and Sammon [1] say that ERP systems mainly are constructed for push-orders. More advanced systems like SAP R/3 can handle more complicate work-load situations such as mixes between push-, pull-, and rush-orders and even Quotes to Customer (Q2C)–orders. The notion Q2C means manufacturing directly from a quote without having to deal with the time consuming intermediate step like order handling.

In Apelkrans and Håkansson [3] meta-agents were used to solve an information flow problem in a manufacturing process. A combination between theory from Information Logistics Processes (ILP) and meta-agents was conducted to benefit intelligent information handling automatically. The combination of MAS technique with ILP is also utilized to solve the problems in this paper, which is handling pull-orders, push-order and rush-orders automatically and simultaneously.

The importance of agent technology to enlarge the capabilities of ERP systems is discussed and recognized as quite useful in industrial applications [11]. A number of attempts have been achieved in order to enhance ERP systems intelligence by using MAS technology. One example is Intelligent Recommendation Framework (IRF) suggested by Symeonidis et al, [11] *"IRF establishes an efficient, quick and easy way of providing intelligent recommendations to the incoming requests for quotes a customer makes, therefore providing a number of enhancements in an integrated way. Recommendations are* autonomously adapted, without having an impact at IRF run-time performance.". Conclusively, IRF helps the ERP system to take care of new incoming orders in an intelligent way. However in our work, we extend the recommendations to be automated decisions. Furthermore, in order to achieve making decision automatically, our MAS has direct connections to suppliers and customers through the web.

3 Production Planning

For this research we focus on problems arising in a discrete manufacturing company working with push-orders. The company suddenly has to take care of a much more complicated situation due to an incoming rush-order and/or a pull-order (coming from an internet inquiry). Commonly, the push-order production is planned by the company's ERP system through Business Intelligence reports and outside information coming external to the company, like forecasts and season variations. A fundamental piece of product information, especially in discrete manufacturing, is covered by the Bills of Material (BOM) documents, which describe raw materials, parts and components needed to manufacture finished products. When it comes to a demand (pull-) order, it usually has to be accompanied by a BOM document. A BOM document is established from the engineering department (sometimes at the customer otherwise at the producer) during product development.

The outcome from the company's ERP system is usually a master plan for the production, work load for staff and capacity load, which is specifying production line usage. The production line is a set of sequential operations established in a factory whereby materials are put through a refining process to produce an end-product or components, which are assembled to make a finished product. In order to create purchase requisitions, the required raw materials and other items are checked. These purchase requisitions include the quantity and the time for delivery.

Many of the ERP systems supporting companies in their production planning also have a BI part (e.g. SAP R/3). The BI systems include OLAP and data warehouse techniques that create knowledge out of the information stored in the systems database. It could answer question like: which product is sold most in a certain region under a certain period? Moreover, BI systems' tool makes it possible to work out a rather reliable forecast. All this knowledge is summarized in order to plan the production, e.g., for the next 3 months.

3.1 ILP Definitions

The concept of Information Logistics (IL) has been discussed for a number of years and a number of definitions have been established. A recent definition is [10]:

"The main objective of Information Logistics is optimized information provision and information flow. This is based on demands with respect to the content, the time of delivery, the location, the presentation and the quality of information. The scope can be a single person, a target group, a machine/facility or any kind of networked organization. The research field Information Logistics explores, develops and implements concepts, methods, technologies and solutions for the above mentioned purpose"

It is possible to consider IL as a process that manufactures an information product and, hence, to introduce concepts from the production area, like Just-In-Time (JIT). In Apelkrans and Håkansson [2] it is shown that Information Logistics is defined as the discipline with the following properties:

- that will supply the right information at the right time, in the right shape and at the right place
- in a user-friendly way

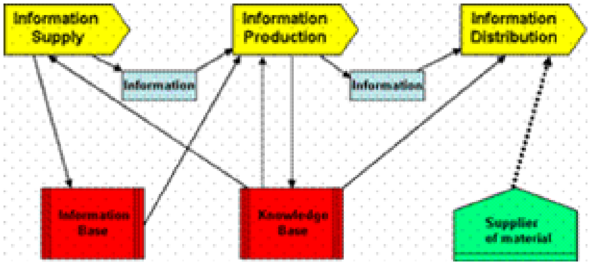


Fig. 1. The ILP Process

- with desired quality
- at the lowest possible cost
- the final product is distributed by some kind of information carrier like paper, card, CD, Smart Card, Internet.

Looking at Information Logistics from a process perspective, we can identify that an information logistics process transforms a given input into some form of output. This input information can be handled, either manually by a user or automatically, by a computer system. The process output is an information product that becomes accessible when the process has ended and delivered to the information receiver who can make use of the information, see Figure 1.

The workflow, which is the chain of processes, is called the Information Logistics Process (ILP). The input to the ILP comes from the application and the output goes to the receiver, which can be either an input to new ILP or stored in a database or presented to a system user. In order to produce the desired output ILP uses the input information and communicates with the knowledge base.

4 Problem Description

The example company used in this research produces goods that are assembled from a number of items (so called discrete manufacturing), some of them manufactured in-house. The company's ERP system is used to give reliable forecasts for their push productions and the company expects to deliver its products in time unless some unexpected events occur. A typical scenario is: "The company's task is to produce 100 batches of products, which have to be delivered within 3 weeks. The forecast says that this volume fits the market demand. Suddenly, a customer inquiry appears from Internet. This customer requires 20 pieces of a very special product, to be used by the customer within 4 weeks. This demand is specified with customer (or company's) own BOM document. At the same time there is an incoming request about managing a rush-order; 20 pieces of a standard product delivered within 2 weeks."

Currently, the ERP system used by the company cannot handle these questions in order to make the best decision, which is either to cancel the order or adjust the master plan for manufacturing or perhaps negotiate on a later deliver of the push order items. Therefore, MAS is applied to support solving those problems.

Sub Questions
For the MAS system, there are several questions that are needed to be processed. For example:

Question	Comments
Is it, at all, possible to take care of these extra orders?	Need to check the production line capacity.
How about machine capacity and staff work load?	The system has limits that need to be checked.
Can we satisfy the demands on reliable production time and delivery dates?	Calculations needed.
Can the company meet the request?	
Is any negotiation necessary with future customers to the batch order?	Some customers may have standard orders, which they always want to buy (solid intervals)
Is there a need for additional purchase orders due to the incoming orders?	Purchase orders for the push-production have already been sent to suppliers.

A number of these questions are usually answered by using the ERP system as decision support. A more cost effective and comprehensible way is to automate the needed decisions through using MAS and its ability to search for solutions via Internet. This is especially important when negotiations with customers and/or suppliers are needed. Furthermore, the MAS can start a 'discussion' with the ERP system's stored information.

As mentioned above, a possible approach to solve the above problems is a combination of ILP processes and MAS utilization. The ILP processes mainly have the responsibility to communicate with the ERP system to solve the questions about capacity, delivery time and procurement needs. The communication needed outside the ERP system is the MAS responsibility, which includes negotiations with customers about possible delays and purchase of additional products that are going to be assembled by the production line.

5 Multi-Agent System Solution

Multi-agent systems have been successfully used for searching and matching information on the web [12]. Commonly, agents are searching for information that corresponds to a request from a user, which can be searching for information about a topic, travel and/or product. From the request, the agents return links to several different pages that are more or less relevant for the user. The work for our agents is finding all the pages that contain the requested products. The request is more than just products; it is the cheapest, to the best quality and at least minimum quantity [3].

The multi-agent system works between the enterprise system, internally, and the web, externally. When the agent-system works with the enterprise system, the agent system has a task environment [9] that is *fully observable*, *static*, *deterministic*, *discrete*, and *episodic*. The environment is fully observable since the multi-agent system has access to all parts of the enterprise system. The agents can easily access all the modules in the system and retrieve the relevant data. The environment is static in the sense that the architecture, structure and data flow do not change, not often anyhow. Thus, even though the content changes, for example, with new products and quantities, customers and addresses, suppliers and products, these do not change the structure of the task environment. Rather, it is only small changes in different databases.

In the enterprise system, the agents in the multi agent system are aware of the next state from the current and, thus, the environment is deterministic. This is due to the unchanging structure. The agents work between the solid modules and can rather easily find the information since they know where to search to perform tasks. This is also due to the discrete set of actions. The enterprise system has a finite number of states that the agents know how to act upon, both knowing what information to be found and where. Hence, the task environment has states that are discrete.

To work with finding information fast and in parallel, we divided the task into atomic episodic, which is performed by several different agents. Thus, instead of assigning the task of finding information from several different modules in the enterprise system, each agent is designed to find one information piece, needed to complete the task. This one task is not necessarily the same task, all the time, so the agents should not be dependent on each other while searching for information since dependencies between agents can jeopardize parallel work. The parallel search is beneficial since it can search for alternative solutions to problems. Finding alternative information and compare the information can be utilized to make the multi-agent system more effective and save money for the company. This facility is especially interesting when it comes to finding suppliers and scheduling production of the manufacturing company where the production of products in the storage can be manufactured while waiting for the ordered products to arrive to the company.

On the web, on the other hand, the agents work in a more complex environment with a vast number of pages and a complex network of links. The task environment (*ibid.*) on the web differs from the ERP system and is *partially observable*, *semi-dynamic*, *stochastic*, *continuous*, and *episodic*. The environment is partially observable where the agents are expected to find the significant information about the required items. This is expected in fully observable environment in which the agents can obtain accurate, up-to-date and complete information about the environment state [12]. However, the vast number of links makes the web cumbersome to work with and makes it difficult to check whether the agents have found all significant pages. Therefore, we regard the web as being partially observable.

The web is highly dynamic since it changes frequently. However, not all the pages are relevant and the agents are only looking for some particular pages with information from supplier companies, which change more seldom. Thus, the environment is semi-dynamic. In fact, some pages are especially requested because these pages are often useful to the manufacturing company. These pages must always be in the set of pages, which result from a request, and the agents have to find these.

The environment can appear to be stochastic, especially in a partially observable task environment with a complex nature [9]. The result of the agents' search for information can differs from another, although the same request. This is a result of a not fully observable and semi-dynamic environment. Even though the locations of the pages are constant, the environment can be stochastic since some of the pages can appear randomly. This will affect the result of the information both positively and negatively. Positively, if a search gives a better result than earlier and negatively if it is the opposite. In our problem space there is sufficient if the agents can find material to purchase that can be delivered in right time to a reasonable price.

Handling a continuous environment is a challenge for the agents. In a continuous environment, the set of quantitative states can vary over time as well as number of states, which both characterize the pages on the web. The location of information is unknown at the time for finding the information. Therefore, the task environment is considered to be continuous. The time and the actions of the agents will differ for every request.

The episodic task environment is beneficial in finding and retrieving information on the web. The task is divided into episodes when the agents' perform one task at the time [9]. The agents' action is to find a page but the found page depends on the request. The request can be more complex than asking for one information piece but can be divided into single parts, each to be handled by an agent. This also support parallelism, which is important when finding information on the web. It speeds up the searching, and finding vast quantities of information about, for example, several suppliers and product components. This supports the manufacturing company to be costs-effective but also efficient for the production.

Finding information on the web, we consider being at extranet for the multi-agent system. The extranet support information about the customer and products that are needed for the production. However, the system also needs contact with the manufacturing company's ERP system. Instead of having a multi-agent system on the web, our multi-agent system has a central base at the company's intranet and when needed, it communicates with the extranet to find information about the products to be assembled [3]. The enterprise system is approached for information about the state of the production, economics and staff of the manufacturing company and the web for more materials to buy. Working with all the tasks simultaneously requires a lot of agents working in the system. To keep track of the agents, we use meta-agents and, thereby, also the modules and web pages at which the information was found. By the meta-agents we can find the same information about the suppliers and products again. This information can be used to teach the agents to find same locations and, thereby, making the agents somewhat intelligent. The meta-agents can also plan actions, maintain individual agents' state information and attempt to control future behavior, classify conflicts and resolve these conflicts [4].

The meta-agents are built on the performance of the intelligent agents to learn the paths the agents are using to find certain information [6], [7]. The meta-agents contain information about all the agents that worked together to find required information piece. By this, the meta-agents can support re-find the same information and speed up the search. Also they can compare the result, supplied by the intelligent agents, and sort out productive performance.

5.1 The Multi-Agent Architecture

On the intranet, the MAS communicate with the company's ERP system to place orders and search in the databases for information about production, product structure consisting of raw material, parts and other components, and customers and suppliers. The information is matched to the product database to calculate what items have to be

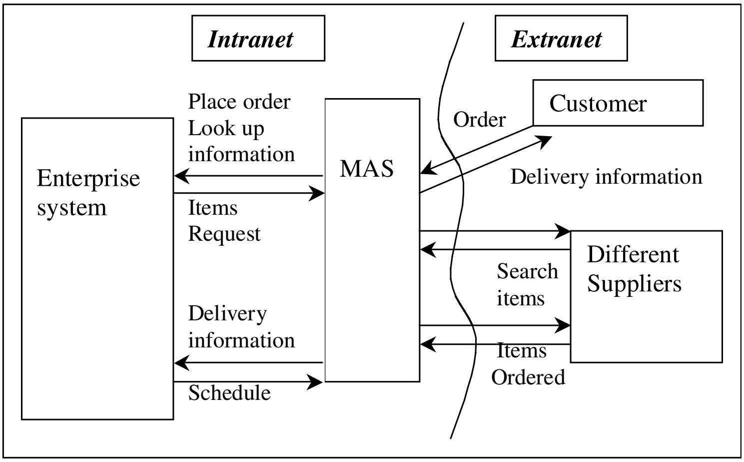


Fig. 2. Communication between customer, MAS, Enterprise system and web

purchased and to customers and suppliers database to check the customers' records and suppliers' material. If the customer is new or its information has changed, the customer database is updated. This is also the case for suppliers. The suppliers are, either found at the Internet or inserted by users. At the extranet, the agents are searching for parts to be purchased from suppliers, i.e., already known in the enterprise system or newly found on the web, and sending information about the items to be delivered to the enterprise system. When this information is straightened, the enterprise system is scheduling the delivery of the products. The schedule is sent to the customer via the MAS.

Handling the information between customer, MAS, Enterprise system and suppliers requires a lot of communication between parts of the system, which, hence, is modular, see Figure 2.

From the enterprise system, the MAS must collect information about the staff and work-load. The MAS checks schedule of the staff and calculate whether it is possible to produce the additional products. Also the delivery dates must be calculated and the production must be check and matched to the products in the storage.

5.2 Automating Decisions

The combination of MAS – ILP approach supports sudden pull- or rush order appears. Decision can be stated by the system to approve (or disprove) the requested order and a master plan for production must be created. This plan shows a number of decisions made in order to fulfill the new order(s) without disturbing the already scheduled push production. Messages are produced and sent to customers and suppliers. Also there will be new documents for staff planning. This requires a system that can perform decisions, automatically.

In the multi-agent system, the meta-agents work at a level on top of agents just as meta-reasoning works on top of rules. Meta-reasoning is a technique that supports the system to reason about its own operation, which can be used to reconstruct the agents' behaviour [8], but also to help in the interaction among agents. The meta-reasoning can be applied in the implementation strategies or plans to respond to requests [5].

The meta-agents supervise both the orders and the master plan. They need to assure that the agents in the multi-agent system perform as expected and that the orders are delivered to the right place in the enterprise system, as well as, to the right people in the organisation. Also the master plan is delivered to right people in the organisation. The meta-agents are in charge if something is going wrong, for example, problems with delivering information. The meta-agents need to assign other agents to handle the orders or plans but also notify the users about the problems in the multi-agent system.

6 Conclusions and Further Work

In this paper, we have applied a multi-agent system technique with intelligent agents and meta-agents to an enterprise system for planning production and at the same time automate decisions. The multi-agent system works between the enterprise system within a manufacturing company and on the web to search, find and handle information. The information is about orders from customers and to suppliers, production, production line and scheduling. Most of the information can be automatically handled by the MAS but under supervision of the manufacturing company. Thus, the company must rely on the decision made by the system. For this the intelligent agents and meta-agents must be visible for and their work evident to the user.

For the research next step is to visualize the agents work at the same time handling security for the agents on the web. The visualization can help the user to control the system's actions. The security is extremely important since the system is working on the web and makes decisions automatically. An intruder could destroy for the company and mess up the orders, item delivery, and production line.

Moreover, it would be extremely interesting to develop some middleware in order to facilitate the communication between the enterprise system and MAS.

References

1. Adam, F., Sammon, D.: The Enterprise Resource Planning Decade, p. 249. IDEA Group Publishing (2004)
2. Apelkrans, M., Håkansson, A.: Enterprise systems Configuration as an Information Logistics Process - A Case Study. In: Cardoso, J., et al. (eds.) Proceedings of 9th International Conference on Enterprise Information Systems, ICEIS 2007, INSTICC, Portugal, pp. 212–220 (2007) ISBN: 978-972-8865-90-0
3. Apelkrans, M., Håkansson, A.: Information Coordination Using Meta-Agents In Information Logistics Processes. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part III. LNCS, vol. 5179, pp. 788–798. Springer, Heidelberg (2008)
4. Chelberg, D., Welch, L., Lakshmikumar, A., Gillen, M., Zhou, Q.: Meta-Reasoning For a Distributed Agent Architecture (2000), `http://zen.ece.ohiou.edu/~robocup/papers/HTML/SSST/SSST.html`

5. Costantini, S.: Meta-reasoning: a survey. In: Kakas, A., Sadri, F. (eds.) Computational Logic: Logic Programming and Beyond. LNCS, vol. 2408, pp. 253–288. Springer, Heidelberg (2002)
6. Håkansson, A., Hartung, R.: Using Meta-Agents for Multi-Agents in Networks. In: Arabnia, H., et al. (eds.) Proceedings of The 2007 International Conference on Artificial Intelligence, ICAI 2007, vol. II, pp. 561–567. CSREA Press, USA (2007a) ISBN: 1-60132-023-X, http://www.world-academy-of-science.org
7. Håkansson, A., Hartung, R.: Calculating optimal decision using Meta-level agents for Multi-Agents in Networks. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part I. LNCS(LNAI), vol. 4692, pp. 180–188. Springer, Heidelberg (2007b)
8. Pechoucek, M., Stepánková, O., Marík, V., Bárta, J.: Abstract Architecture for Meta-reasoning in Multi-agent Systems. In: Mařík, V., Müller, J.P., Pěchouček, M. (eds.) CEEMAS 2003. LNCS, vol. 2691, p. 84. Springer, Heidelberg (2003)
9. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River (2003)
10. Sandkuhl, K.: Information logistics in networked Organisations Issues, Concepts and Applications. In: Cardoso, J., et al. (eds.) Proceedings of 9th International Conference on Enterprise Information Systems, ICEIS 2007, INSTICC, Portugal, pp. 23–30 (2007) ISBN: 978-972-8865-90-0
11. Symeonidis, A., Chatzidimitriou, K., Kehagias, D., Mitkas, P.: An Intelligent Recommendation Framework for ERP Systems. Expert Systems With Applications Journal (2004)
12. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley & Sons Ltd., Chichester (2002)

Applying Constructionist Design Methodology to Agent-Based Simulation Systems

Kristinn R. Thórisson[1,2], Rögnvaldur J. Saemundsson[3], Gudny R. Jonsdottir[1,2], Brynjar Reynisson[1,2], Claudio Pedica[1,2], Pall Runar Thrainsson[2], and Palmi Skowronski[2]

[1] Center for Analysis & Design of Intelligent Agents
[2] Department of Computer Science
[3] School of Business
Reykjavik University
Kringlan 1, 103 Reykjavik, Iceland
{thorisson,rjs}@ru.is

Abstract. Among the benefits of agent-based modeling is parallel development and implementation of components. Integrating large numbers of agents developed by many is, however, a significant challenge. Further, architectural changes can require significant redesign. We have developed *CDM-S*, the *Constructionist Design Methodology for Simulation*, an agent-oriented methodology for developing, implementing and evolving multi-agent systems. CDM-S's strength lies in simplifying modeling and construction of systems with architectural evolution of complex control hierarchies and data flow. We have applied CDM-S in the development of a family of market simulations where companies, employees, banks and consumers are modeled at multiple abstraction levels. These were designed and built by 14 students over a period of 10 weeks. Experience shows CDM-S to be a promising high-level methodology for constructing large multi-agent systems. Here we describe CDM-S and present data on its application in the development process.

Keywords: Design Methodology, Agent-based systems.

1 Introduction

The creation of multi-scale agent-based simulation systems requires integration of a large number of functionalities that must be carefully coordinated to achieve coherent and desired runtime behavior. Typically this work is done according to standard software practices. Development of multi-agent systems is more difficult using these methodologies than standard IT system construction as the characteristics and requirements of these systems are drastically different in many key aspects. Agent-based systems are often built by numerous people over long periods, months, years, sometimes even decades, evolving in the process. Few - if any - methodologies have addressed the many issues that complicate such work. Additionally, agent-based systems often assume or require concurrency throughout.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 203–212, 2009.

We have adopted a methodology that was designed for artificial intelligence, the Constructionist Design Methodology (CDM) [1], to address the special issues encountered in agent-based simulation systems. The result, *Constructionist Design Methodology for Simulation* (CDM-S), has been used in the development of a family of agent-based simulation models. The methodology bears some relation to [2] and [3], although ours has the benefit of having been tested and honed in a broader range of projects [4,5,6]. The models built so far with CDM-S are fairly large, composed of many types of agents, each implementing multiple decision-making policies; the running simulations created to date count up to 100 agents for a single system. These provide a non-trivial test-case for the methodology.

In this paper we present CDM-S and its use in the construction of a family of simulation systems. First we give an overview of the methodology and present the resulting simulation framework at a relatively high level. Then we describe the application of CDM-S to the development process, presenting information on development process and detail selected parts of the process and draw conclusions from the data.

2 CDM-S: Constructionist Design Methodology for Simulation

The original CDM has 9 defined steps; CDM-S simplifies many of these and adapts to simulations. It also adds three new additional steps - 7, 11, 12. Step 7 is one of the keys to successful adaptation of CDM to simulation. The full set of steps of CDM-S is:

1. *Define high-level goals.* Specify the primary motivation and goals behind the system to be developed.
2. *Define the scope of the system.* Specify at a high level what the simulation is intended to do. Ask yourself *What is the set of questions that my system supposed to answer?* - this is the most important question you will ever ask, when building a simulation! Then follow up with these four questions: {a} What is the data? {b} Where is the data? {c} How is it shared? {d} How is it processed/ changed? Use of narratives and story lines are encouraged, as a template of expected behavior of the simulation. Start with an initial write-up of a few high-level examples of system behavior. From this, a more detailed specification of the abilities of the system to answer questions can be built, using information from step 1. This will guide the selection of which agents to include and what their roles should be. It may be useful to think also in this step how the system may be expected to be modified and evolve.
3. *Modularization.* Define the functional areas that the system must serve, and divide this into agents with roughly defined roles. The agents can then be recursively sub-divided using the same approach (principle of divisible modularity). This step is best done in a group meeting where all developers can contribute to the effort. Agents communicate via a pub-sub mechanism and/or blackboard(s). Try to match information exchange (messages) in your model with information exchange in the system(s) to be modeled.

(a) *Agents.* This step operationalizes the role of each agent and defines their interfaces. Agents with highly different functional roles should typically not need access to the same set of data - if they do it means they may have close functional dependencies; consider coupling them or putting them into the same executable with shared access to the same data. Define descriptive names for message types and draw flow charts of agent communication.

(b) *Blackboards.* Blackboards serve both as engineering support and system optimization. Consider using two or more blackboards if {a} there is a wide range of information types in the total set of messages in the system that form natural data groups (e.g. co-existing complex symbolic plan structures versus simple boolean switches) {b} there is a wide range of real-time requirements for the information flow, e.g. high-frequency micro-decisions versus low-frequency plan announcements; {c} the system needs to run on multiple computers to achieve acceptable performance. It is natural that agents with messages containing similar content share a blackboard.

(c) *Unit tests.* Design simple case scenarios that return known results and span short periods of run-time. They are typically run for sub-parts of the system but can also be used to track behavior holistically. Define information snapshots to be published for statistical and monitoring purposes; use monitoring agents to test for "normal" and "obvious" behavior of the system.

4. *Test system against scenarios.*
   (a) *Expected communication (blackboard) throughput.* Network speed and computing power puts natural constraints on the maximum throughput of each blackboard. Make sure the architecture and the hardware setup meets performance expectations.
   (b) *Efficient information flow.* Static or semi-static information that is frequently needed in more than one place in the system may be a hint that the processes using that information should share an executable. (This is not a good idea when the processes sharing the information are of very different nature.)
   (c) *Convenience with regard to programming languages and executables.* If two agents are written in the same language it may be convenient to put them together into one executable. This is especially sensible if {a} the agents use the same set of data, {b} are serially dependent on each other, {c} have similar update frequency requirements, {d} need to be tightly synchronized, or {e} are part of the same conceptual system.
5. *Iterate through 1-4 as often as necessary.*
6. *Assign agent types to team members.* The natural way to assign responsibilities is along the lines of the agents, following people's strengths and primary interest areas. Every agent type gets one Lead that is responsible for that agent working, for defining the messages it post and receives. A good way to assign tasks, especially if the number of team members is small, is to borrow from extreme programming and assign them to pairs: One Lead, chosen for

their primary strength, and one Support, chosen by their interest and/or secondary strength. One individual can serve both roles, in different teams.

7. *Write agent "shells" in a breadth-first approach.* Partial agents are used to achieve a good overview of message flow and interdependencies in the system: Start with mock-ups, before creating full system that do very little but enough to start running the system in a few limited example scenarios. Use case scenarios from step 2 to help implement mock-ups.
8. *Test all agents in cooperation.* Use the unit tests to verify sub-assemblies. This step always takes longer than expected! It is also one of the most overlooked steps, yet it *cannot* be avoided - doing so will only force it to happen later in the process, delaying the ability to accurately estimate the project's full scope, and making it more likely that deadlines are missed and planned functionality has to be canceled.
9. *Build agents to specification.* Build all agents to their *next* function specification. This step is naturally done in frequent alteration with the prior step. Write agents with resilience (graceful degradation): A distributed system can be very fragile; write agents to be resistant to downtime. Give agents temporal knowledge: If agents are unaware of the passing of time they are less likely to represent the behavior of the system they are supposed to simulate. Use the benefits of being able to freely mutate agents (split and merge) as the design and implementation process unravels.
10. *Tune the system with all agents (and agent shells) operating.* This step can be arbitrarily long, depending on the complexity of the interaction between agents and the complexity of the message content being transmitted between them. Computational intensity of some agents may require them to be put on separate computers.
11. *Return to 7 until all agents have reached full specification.*
12. *Expand / evolve the system.* Continue to add agents to the system, as well as modifying existing agents necessary, going back to step 2 or even 1, running through to 11, until the system is abandoned.

3 Application of CDM-S

We will now describe the application of these design principles to the development of a family of agent-based simulation models meant to study the generation, organization, development and evolution of the knowledge embedded in an industry. The original plan was to create models of knowledge evolution in a market economy that would go beyond the current state of the art in terms of detail and predictive power. While we have not yet fully achieved this goal we have made significant headway towards a model with endogenous support for knowledge evolution.

The software we chose for constructing our models is Java™ and Psyclone [7]; portability factored heavily in the choice of both. Python scripts were used for turnkey setup, making it easy for team members to start the simulation using a single command. Like related middleware such as Swarm[1], Psyclone supports architectural re-configuration very well. In Psyclone an implemented architecture can be radically changed with relative ease as to re-routing messages, temporal dependencies, and re-organizing distribution of agents across machines. Semantic interfaces used for specifying data flow provide great flexibility in changing layout after the initial system is built [5]. All parameters in our models allow centralized access, allowing for many variations to be made for comparative runs of models.

3.1 System Modeling

The market economy models consist of agents at multiple levels of abstraction, i.e. individual, firms and other organization, and industry: Multiple types of agents with a number of decision-making policies (for a detailed account see [4]). In most cases the policies are relatively simple; in every case we tried to mirror their natural counterpart to a first approximation. The individuals represent the agent type that is least abstract, in that a single Individual agent corresponds to a single individual in the real world; the Market agent is the most abstract, in that a single Market agent represents thousands of consuming agents in the real world. The firms lie there in between, being partly represented by individuals (the employees) and partly by monolithic rule sets that determine their policies.

The work proceeded in two phases, roughly 5 weeks in duration each. Each phase involved a group of Master's students in the course Agent-Based Modeling and Simulation, who had not used CDM-S or similar methodology before. A total of 14 Master's students worked in parallel on building separate parts of the system. We will focus in particular on the market mechanism in this model, as it had to be re-engineered from Phase 1 to Phase 2, and provides for an interesting case study; the time spent by the teams working on version 1 and 2 are provided and compared. The first phase was implemented by a group of 11 Master's students, the second by a group of 3 Master's students and one undergraduate. In Phase 1 the students started from scratch and used CDM; in Phase 2 they started from the platform designed and implemented by the first group and used CDM-S, which had been revised from among other things lessons learned from Phase 1 (see Figure 1). Both teams were led by the same two instructors.

So, our models include agents at multiple levels of abstraction, i.e. individual, firms and other organization, and market. Similarly, our model can represent how resource constraints, such as the lack of talented individuals or the lack of funding, influence the knowledge buildup in an industry and eventual knowledge substitution. To give the readers a sense of the complexity involved in the models, a short overview of the framework is in order. Further details are given in [4].

Individuals. The first source of complexity stems from our new morphogenetic[2] model of knowledge with highly dynamic properties. Individuals are atomic

[1] http://www.swarm.org/wiki/

[2] Morphogen: An agent that controls the growth and shape of something, e.g. biological tissue.

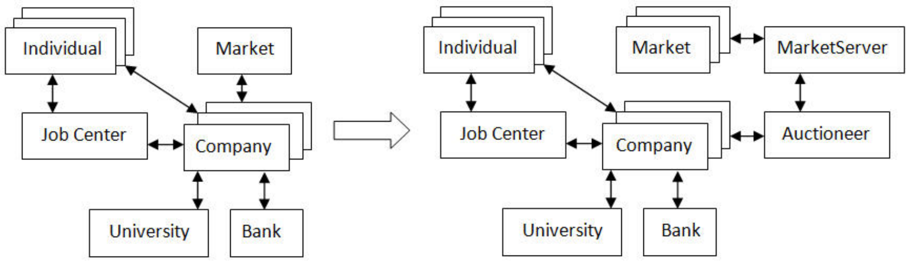


Fig. 1. Agents implemented in Phase 1 are shown on the left and modifications and refactoring in Phase 2 is shown on the right

agents in the system, containing mechanisms for knowledge acquisition, retention and application, salary negotiation and employment search. They can be initialized to have different knowledge (**K**s) in the beginning of a simulation. A learning rate describes how quickly/slowly they learn (acquire new **K**s). They can learn new **K**s by going to special training at the University and their performance on a particular **K** improves over time as they use it to produce products that require that **K**. Individuals evaluate their salaries in connection with the rest of the world (average salaries, employment ads) and decide if they think they are more valuable than their salaries state, using a heuristic comparing their own salary with that of the market for each **K**.

Companies / Firms. Individuals are employed by firms that offer services on the market, based on the knowledge endowments of their employees. Firms compete with other firms on the product/service markets, as well as in factor markets (employees and funding). Decision-making in the firm is based on local policies, including a training policy, an alertness policy and development policy. A firm can add products to its suite of products, but must invest both time and staff on product development before it can start selling.

University. The University controls general knowledge development within all knowledge fields, i.e. the state-of-the-art. It also controls the availability of new knowledge fields, i.e. new inventions. In the University each **K** has an associated "difficulty" rate that determines how intrinsically complex that knowledge is. This interacts with the learning rate of Individuals, which varies. The University holds l_{max} for all **K**s at any point in time.

Job Center. A single agent called Job Center manages all hiring of employees. Companies advertise for individuals with specific levels of **K**s. Unemployed individuals can answer these advertisements and the Job Center finds the most suitable employee through a method that takes into account the salary demands of the individuals. The Job Center also posts statistics such as average salaries.

Bank. The Bank is a simple loaning institution that the companies can apply for a loan from. All companies start with some amount of cash - they will apply for loans if they run out of cash. The bank advertises its current interest rates every day, so the firm can evaluate whether it is favorable or not to take a loan. Global rates change every day and vary from 4% to 20%. As most of the other agents the bank publishes information about its capital and liquid resources.

Market. The market is composed of target groups; each group is initialized with a particular set of values determining its behavior with regard to the products offered by the firms. The market includes various shifting policies and can be set to gradually change its size or shift its product sweet spot. At runtime, customer groups receive product advertisements from companies and make a decision of how much they will buy, according to how well the product meets their needs and the price of the product.

4 Results

Our implemented model in Phase 1 contained 40 Individuals, 5 Firms, a Market, a University, a Job Center and a Bank. This turned out to be impossible to run on a single computer, even our fastest one, so we distributed the system onto 12 computers. This, however, turned out to create bottlenecks in the networking. We used CDM-S steps 3a and 3b repeatedly on the design and worked out a new scheme for combining information flow between companies and individuals into fewer messages, achieving a significant reduction in message traffic. The resulting model ran fairly well on 12 computers.

One of the main lessons learned in Phase 1 was that integrating the agents together in the integration session took more time than expected; frequently going back to fix agent behavior as a result of integration problems. One reasons was in many cases that the values in the messages that were being sent between the agents were not in the same scale. That affected the execution of the system as many wrong decisions were taken by the agents. This shows the importance of having a runnable version of the system with agent shells as early as possible in the design phase and frequently running the whole system together and observing its behavior.

The initial mechanism for the market turned out to be too simplified for the purposes of our simulation and did not ensure realistic distribution of sales between competing firms. To improve on the design one team in Phase 2 focused exclusively on re-engineering the way this mechanism worked, as well as making it more sophisticated. With the introduction of the auctioneer, target groups no longer listened to product advertisements from the firms and decided for themselves what to buy (see Figure 1). Following CDM-S's step 7, agent shells were created early on eliminating integration problems in Phase 2.

To guaranty upward scaling of target groups a new server architecture was introduced in Phase 2. Instead of each target group being an individual executable, a target group server is created that has many target group instances. In the new design the target group server oversaw interrupts and sending and receiving of messages on behalf of the target groups. This reduces the amount of messages that have to be transmitted and decreases the load on the network. The amount of time spent by both market teams is shown in Table 1.

The mutability of our framework makes it possible to build several models with different assumptions and test the outcomes against each other. It also allows us to maintain easy control of all parameters that we are interested in exploring, even at runtime. As suggested by CDM-S, explicit representation of agent interaction through messages, which can be inspected both on paper and at run-time, increased the system's transparency, and helped all team members' understanding of how the various parts of the system work. Typically, in a system with the number and complexity of interactions between agents as found here, one would expect a significant amount of time spent in Phase 2 on reworking the interactions and interconnections between the agents, rather than the agent mechanisms themselves or other parts of the system. However, the results show that most of the time is actually spent on the agents themselves, and then on running the resulting system. This indicates to us that at least one of our goals was achieved through the use of CDM-S: To help balance the time spent on those aspects that typically fall by the wayside, such as running the system under various conditions; CDM-S helped us manage complexity related to the network of agent interactions and allowed us to focus on agent functionality (see Table 1).

Thórisson et al. [1] had reported a total estimated development time of 2 mind-months for a fairly complex interactive multimedia system, which they claimed strongly supported the conclusion that the Constructionist Methodology sped up system creation. Our results here also support this conclusion, but perhaps not quite as strongly. We have reason to believe that the time spent on the relatively complex model, and the results achieved, indicate that CDM-S is a promising design methodology worthy of further study and refinement. However, until other design methodologies provide comparable evaluations, we will not know how

Table 1. Total hours spent by two 3-person sub-teams. In Phase 1 the market and related functions were implemented and in Phase 2 the market was revised with added functionality and robustness. Implementing mock-ups of agent interactions beforehand in Phase 2 contributes directly to valuable time getting devoted to running and tweaking the model (44%).

	Phase 1		Phase 2	
TASK	Hours	%	Hours	%
Defining message types	8	**4%**	7	**3%**
Design / redesign	15	**8%**	14	**7%**
Market agent	90	**51%**	49	**25%**
Auctioneer agent			29	**15%**
Bank agent	22	**12%**	3	**1%**
Monitor	14	**8%**	10	**5%**
Runs and data gathering	30	**17%**	87	**44%**
TOTAL	179	**100%**	199	**100%**

much or how little the CDM-S increases productivity in the creation of complex dynamic systems, over and above alternative approaches.

4.1 Design Process and Teamwork

The iterative steps in CDM-S support teamwork and work distribution in a large group of developers, by among other things facilitating parallel development of agents. This was especially evident in Phase 1, where 11 students were working on the system simultaneously. The breadth-first approach emphasized in step 7 is another key to time-saving. It insures that communication between agents or sub-assemblies work before the details of each agent is implemented, allowing for a runnable version of the full system to be available from early in the implementation time. This also greatly minimized integration problems as the mock-ups serve as test interfaces for other's team's unit tests.

The agent development was part of the class homework so teams were working independently hours on their agents; integration sessions were held every other week. Starting from system goals originally provided by the teams' leaders, architectures were quickly built, requiring only minor adjustments throughout the project. Our original design from Phase 1 proved solid and stayed relatively unchanged, though minor adjustments to single classes where needed. The bank agent had to be redesigned once; thanks to CDM-S this could be done with only minor changes to the surrounding architectural components.

The communication between groups throughout the project was good. Several meetings where held when critical decisions concerning whole of the system needed to be made. The students agreed that the election of a leader for each group proved to be an excellent decision that sped up development processes considerably, although all major decisions where posed to the whole group. Internal communication between group members in our groups was good. The use of CDM-S turned out to be highly beneficial with the relatively large team of developers working in parallel; the agents/messages dual view it provided on the system enabled the collaborating groups to communicate efficiently about interactions between the system's agents, as well as the layout of the architecture, in part and in full.

5 Conclusions

We have presented results from the application of the Constructionist Design Methodology for Simulation, CDM-S, to the creation of a complex, multi-scale model of a knowledge and market economy. The whole project had architectural challenges that were all solved through the use of our CDM-S methodology, which resulted in a system with a high degree of extensibility. The relatively high simulation resolution the models provide is a direct result of using CDM-S. The current distribution of agents in the model along an axis of abstractness can be fairly easily changed through the mutability functionality of the CDM-S. Coupled with the model's scalability we envision being able to for example substitute the Market with several more fine-grain agents that approximate a large group of consumers more closely.

The family of models developed has a clear potential to address several classes of questions regarding the generation, organization and evolution of knowledge in economic settings. The methodology had a large impact on the success of the work. In addition to providing a robust approach to multi-agent systems, the hope is that this work can also lead to more detailed comparisons and evaluations of alternative approaches for building such systems.

Acknowledgments. This research was supported by a Marie Curie European Reintegration Grants within the 6th European Community Framework Programme and by research grants from The Icelandic Centre for Research (Rannís). Psyclone is a trademark of Communicative Machines Ltd., Scotland. The authors would like to thank the team from Phase 1 who built the initial version of the system and contributed the some of the data and architectural details provided here.

References

1. Thórisson, K.R., Benko, H., Arnold, A., Abramov, D., Maskey, S., Vaseekaran, A.: Constructionist design methodology for interactive intelligences. A.I. Magazine 25(4), 77–90 (2004)
2. Giret, A.: A multi agent methodology for holonic manufacturing systems. In: Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M.P., Wooldridge, M. (eds.) AAMAS, Utrect, Netherlands, July 2005, p. 1375 (2005)
3. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing multiagent systems: The gaia methodology. ACM Transactions on Software Engineering and Methodology (TOSEM) 12(3), 317–370 (2003)
4. Saemundsson, R.J., Thórisson, K.R., Jonsdottir, G.R., Arinbjarnar, M., Finnsson, H., Gudnason, H., Hafsteinsson, V., Hannesson, G., Isleifsdottir, J., Jóhannsson, Á., Kristjansson, G., Sigmundarson, S.: Modular simulation of knowledge development in industry: A multi-level framework. In: Proc. of the First Intl. Conf. on Economic Science with Heterogeneous Interacting Agents, Bologna, Italy (June 2006)
5. Thórisson, K.R., List, T., Pennock, C., DiPirro, J.: Whiteboards: Scheduling blackboards for semantic routing of messages and streams. In: Thórisson, K.R., Vilhjalmsson, H., Marsella, S. (eds.) AAAI 2005 Workshop on Modular Construction of Human-Like Intelligence, Pittsburg, Pennsylvania, July 2005, pp. 8–15 (2005)
6. Ng-Thow-Hing, V., List, T., Thórisson, K.R., Lim, J., Wormer, J.: Design and evaluation of communication middleware in a distributed humanoid robot architecture. In: Prassler, E., Nilsson, K., Shakhimardanov, A. (eds.) IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2007) Workshop on Measures and Procedures for the Evaluation of Robot Architectures and Middleware, San Diego, CA (2007)
7. CMLabs: Psyclone Manual (2007)

Agent-Based Framework for Distributed Real-Time Simulation of Dynamical Systems

Grzegorz Polaków and Mieczyslaw Metzger

Department of Automatic Control, Electronics and Computer Science
Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland
{grzegorz.polakow,mieczyslaw.metzger}@polsl.pl

Abstract. A framework is presented, supporting calculation tasks specific to the control theory, allowing for a consolidation of theory-based calculations with real-world control systems by means of hardware sensors and actuators. A networking protocol was designed specifically for this task, employing producer-distributor-consumer data distribution scheme over Ethernet network to enable synchronised execution of spatially distributed mathematical models implemented as intelligent blackboard-based agent system, in order to properly cooperate with sampled real-world continuous dynamical systems. Comprehensive data on internal structure of distributed system is available by monitoring web service, so it is possible to integrate framework's output in other software.

Keywords: multi-agent distributed computation, intelligent blackboard system, real-time networking, modelling and simulation, producer-distributor-consumer.

1 Introduction

In control systems engineering it is often needed to model and simulate dynamical systems consisting of control algorithms and controlled plants to conduct simulated computational experiments, which are otherwise unacceptable in real-world systems due to possible consequences (damage, costs). Usual case is when simulated system is built in modular manner, and tests require dynamical replacement of modules or reconfiguration of connections. Typical example of research is testing dependence of control system performance on various control algorithms and algorithm's parameters. In such computational tests, distinctive modules of system are modelled by sets of recurrent equations, so simulation process consists of repeated simultaneous calculations of such equations involving many variables.

An additional issue, which is control systems research related, is that usually full research process consists of many experiments including both pure computational tests (a mathematical model of a plant controlled by an abstract mathematical controller) and joint experiments (a real-world plant connected to a software controller, a mathematical model of plant controlled with an industrial hardware controller). Therefore, it is highly desired for computation engine to be able to communicate with real-world dynamical systems (by means of sensors and actuators) and control instrumentation (custom I/O systems of programmable logic controllers).

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 213–222, 2009.

Because modelled system is based on recurrence formulas, assurance of proper and timely exchange of variables is essential to keep equations valid. Taking into account that computations are to be performed in spatially distributed hardware of varying platforms – proper architecture and protocol design is fundamental, as iterations of calculation have to be synchronised with sampling periods of underlying hardware.

This paper presents communicational multiagent framework designed to simplify software development for simulation tasks in distributed real-time control systems. Each component of the simulated system is embedded in a separate software agent with clearly defined interface. Framework-based systems can carry out purely computational tests basing on models described with recurrent mathematical formulas, but they also may easily incorporate any real-world instrumentation i.e. hardware sensors and actuators. The framework itself consist of two modules – custom protocol supporting time determined and synchronised communication of calculation agents, and a web based interface enabling visualisation of performed calculations and a dynamic analysis of framework's internal structure.

This work is considered as a significant contribution, as there is not much work done in the field of application of the multiagent systems in the control of continuous time-critical systems. Typically, agent-based systems in industry are used for control of manufacturing processes [1], which (while being complex and spatially distributed) have no strict requirements in the domain of time. Some attempts to the control of continuous processes were made (see [2],[3] for examples), but these are isolated and very specific cases and still no universal approach is developed. This paper, along with the previous works of the authors, tries to establish the concept of agent systems application in the industrial process automation and systems modelling. Such real-time agent systems introduce multiple considerable benefits, especially during a reconfiguration of the systems structure, when compared to the traditional static control systems and modelling methods. However, automation of continuous processes require modified approach, which should take into the account the real-time requirements and timing characteristics of inter-agent communication (an example of work dealing with timing issues of agent communication may be found in [4]). The most popular communication platforms for the multi-agent systems (JADE, RETINA – see [5]) supporting well known ACLs, do not fulfil the uncommon real-time requirement. The framework presented in this paper is designed specifically for the computational tasks during system modelling, and incorporates high-level multiagent concept, while it also fulfils stated low-level real-time requirements.

2 Multiagent Computational Framework

In cases under consideration, each of components of the system is modelled (or represented) by a recurrent equation. Each step of the simulation is performed basing on both current and historical values of variables connecting individual equations. In consequence, whole system is an agent–based blackboard system (see [3] for comparison). In such case, mentioned set of values of variables becomes knowledge table, and each of equations is an agent. The knowledge table is cyclically broadcasted over the network; each broadcast serves as a request for agents to perform an iteration of embedded equations. Results of calculations are immediately sent back and included in the blackboard, so they will be broadcasted to all agents in the next cycle of work. Summarising, the main idea of the presented framework is the use of the producer – distributor – consumer (PDC) data distribution scheme for internal variables of spatially distributed dynamical system, modelled as a set of recurrence equations. Use of the PDC scheme provides accuracy and timeliness of simulated system's data, while keeping the execution of the system synchronised, which allows to integrate simulation of mathematical models with real-world dynamical systems being the subject of the researches.

2.1 Formal Problem Statement

Considered specific class of systems consists of dynamical systems defined by a set of *n* recurrence relations in the form of:

$$x_i|_t = f_i(X|_{t-1}, X|_{t-2}, \ldots, X|_{t-ki}) , \tag{1}$$

where $X = [x_1, x_2, \ldots, x_n]$ is a vector containing all the variables in the system. State of the system Ω is defined by all those previous values of X vector, which are required to determine its future values:

$$\Omega|_t = \{X|_{t-k}, X|_{t-k+1}, \ldots, X|_t\} , \tag{2}$$

where $k = \max_i k_i$. Equations (1) can be booth explicit i.e. stated in the form of mathematical notation, and implicit i.e. embedded in the physical relations of an existing real world dynamical system.

2.2 Architecture and Networking Pprotocol

This paper presents these aspects of the framework which are important from multi-agent systems' point of view (technical networking aspects of prototype implementation are described in [6]). It is proposed that each equation (or set of logically connected equations – see section 2.3) is embedded in self contained application, i.e. agent. The agents encapsulate all the hardware platform-specific properties, and exposes a unified framework-defined interface only. Therefore, agents may be implemented as programs executed at desktop grade PC controlled by nearly any OS (Windows, MacOS, Linux, etc.), but also may be embedded in non-typical hardware platforms such as industrial PLC and even custom made hardware.

All the agents work as equally privileged network nodes, controlled by a super-agent i.e. distributor. The distributor works as a protocol arbiter; it controls the duration of cycle of work and manages the blackboard (current state vector X). Additionally, the distributor holds a list of human readable labels assigned to the each element of the X vector.

Each thread's task consists of network socket listening, awaiting for the X state vector broadcasted by the distributor. In reaction to this transmission, agents assume that new iteration started, and treat the newly received X vector as $X|_{t-1}$. Basing on its value, and on locally stored previous values ($X|_{t-1}, X|_{t-2}, \ldots, X|_{t-ki+1}, X|_{t-ki}$ now treated as $X|_{t-2}, X|_{t-3}, \ldots, X|_{t-ki}, X|_{t-ki-1}$ – where $X|_{t-ki-1}$ is no longer needed and can be discarded) each agent performs iteration of the embedded equation. Resulting values $x_i|_t$, spatially scattered over the network, are sent individually to the distributor, which aggregates them into consistent and properly ordered $X|_t$ vector, and by broadcasting it starts the next network cycle.

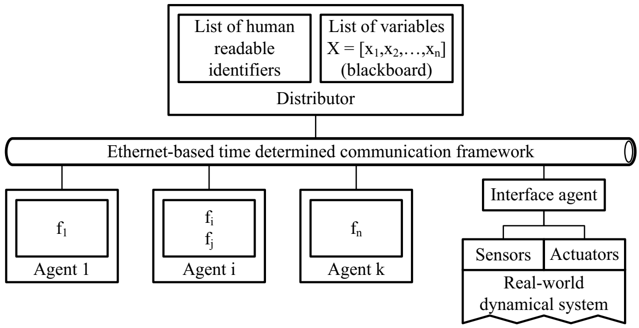


Fig. 1. Main elements of architecture of the framework

In the Fig. 1 the architecture of the framework is shown. Along with *equation agents*, an *interface agent* is shown. Interface agents are a gateway between mathematical models and real world systems. Cycle of work of interface agent is similar to equation agent's one – in answer to global broadcast containing state vector, interface agent writes proper values to actuators and using sensors reads the result, which is then sent back to the distributor as one of $x|_t$ values. The most obvious way of interface agent implementation is as an application executed within the PLC, interacting with the real world physical dynamical system using the industrial control instrumentation.

This way, the system blackboard becomes the field of constant conflict. The blackboard represents the current state of the system or model, and each of the agents constantly tries to modify its content according to its own agenda. In cases of real world objects embedded by interface agents, their agenda is determined by the nature of physical processes taking place. In cases of equation agents, their plans are stated explicitly by mathematical formulas, which may represent control algorithms or another physical processes, such as a plant to be controlled. Each iteration of the system is therefore a single clash between all the agents, result of which shifts the state of the system.

Fig. 2 presents time graph of transmissions during one protocol iteration. For the presentation relatively small n ($n = 4$) is assumed. The cycle starts with the broadcast of state vector X (a); after that, distributed calculations take place (b). When results are available, each of agents sends them to the distributor on its own hand (c).

The protocol runs on top of the standard Ethernet networks with its Internet protocol suite (TCP,UDP,IP). Transmitting the variables is done using UDP protocol, which has possibility of broadcasting (UDP broadcasted messages are mapped on to IP broadcasted datagrams, which are directly transmitted as broadcasted Ethernet frames). It is assumed that used network instrumentation is modern i.e. network is built using twisted-pair cables and switching technology, so it's able to work in full-duplex mode. This allows for full packet collisions avoidance and strict determination of transmission times, which in order means that minimal achievable duration of iteration is determined mainly by calculation times. Additionally, due to using full-duplex mode, return transmissions (c) of $x|_t$ values needs not to be ordered in any special way, because related UDP messages will be queued by Ethernet hardware and delivered to the distributor node timely and reliably ([7],[8]).

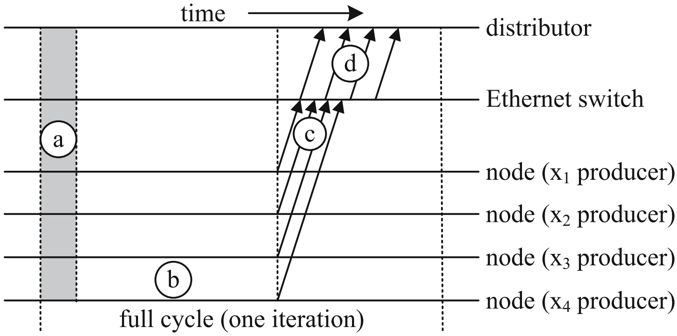


Fig. 2. Data flow during one iteration of the protocol

Duration of cycle of work is predefined with the distributor agent at the configuration stage, and should be chosen according to dynamics of the specific system, with which framework cooperates. Distributor keeps each cycle duration by adjusting waiting time period between stages (d) and (a). Obviously, duration of cycle has to be defined long enough to allow all the calculations to perform. In case when no real-world system cooperates with the framework, cycle duration management may be switched off, so all the calculations are performed as fast as possible, limited only by computation and communication times.

During the framework's cyclical work, the distributor serves also a list of human readable labels for all the elements of the X vectors. The list can be used by all agents during human-machine interaction.

Detailed description of binary formats of all communicational channels in the protocol is available in [9], which describes the framework as a tool for integration of robotic systems. Distributor is developed using National Instruments LabVIEW (software development environment popular in control systems engineering), additionally, a set of function blocks was also developed to help with transformation of LabVIEW developed application into framework-capable agent. However, due to openness of the presented protocol, it is possible to develop an agent using nearly any programming language, from low level C to high level MATLAB (using Instrumentation Control Toolbox).

2.3 Limit of the Protocol

In the presented framework, agents communicate with the intermediate distributor node only. Because each transmission performed by the distributor automatically begins new global iteration, it becomes impossible to implement an equation in which output variable depends on some variables from current iteration (in addition to earlier stated dependence from all previous iteration). I.e. it is formally impossible to implement equation defined as:

$$x_j|_t = f_j(Y|_t, X|_{t-1}, X|_{t-2}, \ldots) , \tag{3}$$

where $Y \subset X \setminus \{x_j\}$ (cases when $x_j \in Y$ are not considered at all, as they require solving, while presented framework serves purely as a computing tool).

Examples of such function include amongst others advanced control algorithms (especially adaptive ones), which consist of several recurrence equations depending in sequence, and all are to be calculated during one iteration. It is possible to implement such cases in the framework by breaking the rule of assigning distinctive thread to each equation. All the sequentially depending equations have to be implemented in one agent – the same thread has to execute all the Fy equations ($y \in Y$). It is possible, because the agent has access to all its internal variables before they are sent to the distributor. In the utmost case, when $Y = X \setminus \{xj\}$, whole system of equations should be implemented in one agent only, and use of the framework becomes pointless. However, nearly none of real life problems from control systems engineering domain is close to such case, as usually equations in the model tend to group into clusters according to the role in the system (plant, controller, disturbance source, etc).

2.4 Advantages of Agent-Based Approach

Employment of the agent-based techniques in the computational engine results in a capability of distributing the computations not only in the physical sense, but also in a logical way, underlining the capability of multiuser approach. The primary computational problem, which is to be modelled and simulated, can be divided and distributed amongst the team of programmers, allowing for cooperative problem solving. This is especially useful when applied to a learning process of control science students. Due to the encapsulation of computations in the distributed agents, which implement common protocol, the simulation is performed in an uniform and consistent way. Furthermore, the agent technology in the simulation and modelling enables the possibility of an extensive model reconfiguration ([10][11]) – it is possible to implement watchdog, results collecting, and visualising agents. remove some agents (i.e. equations) from the pool, replacing the agents with other ones, etc.

A similar effect could be achieved with a simple distribution of processing, however the multiuser approach would be more complicated in such case. Additionally, a simple spatial distribution doesn't cover the issues of transient states and keeping the model equations valid, while the agent-based distribution does. At least, an agentification of the system enables the programmers to implement advanced control schemes, such as competition between various regulator-agents (see [2]).

Agent-based approach results in a specific layered structure of the system, which is illustrated in the Fig. 3. The system consists of two layers: the network protocol and the logical mesh. The logical layer consists of human users, along with their intentions, tasks, and software agents developed to fulfil those tasks. This logical layer is dynamical, and its structure changes accordingly to the tasks performed by current users, displaying self-organising behaviour. Visualisation of such system is a challenging task ([12]), for which a special interface was created, basing on the Web Services standards. This interface is briefly described in the section 3.

2.5 Applications

The framework is used for the simulation of distributed control systems (DCS) on daily basis. These simulations are part of the control science course conducted at our

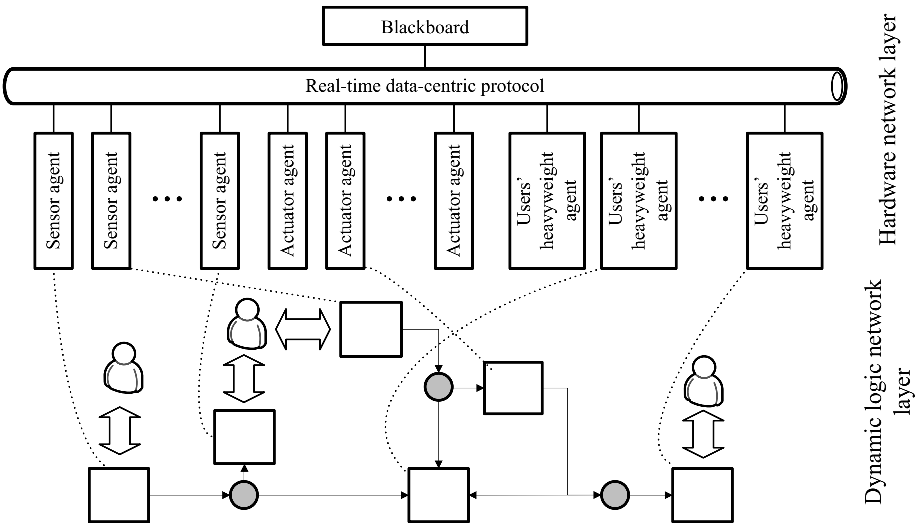


Fig. 3. The layered structure of the system

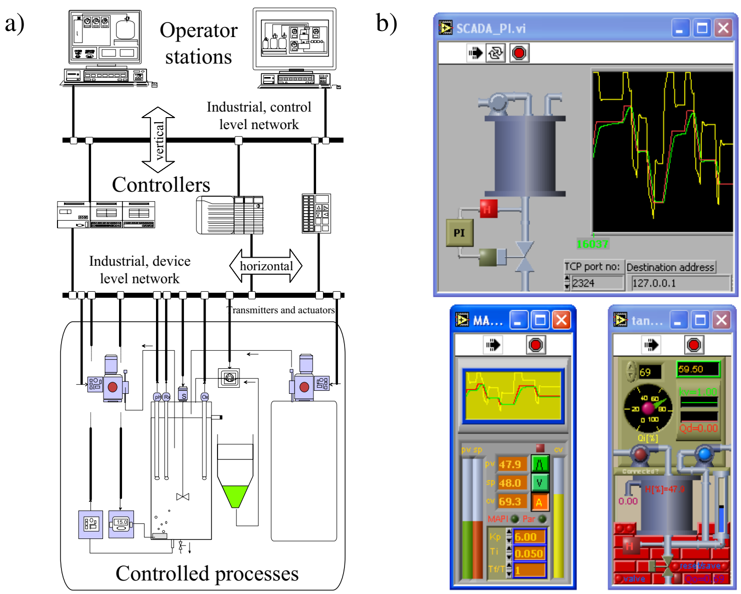


Fig. 4. Distributed control system simulated with the framework: a) structure; b) examples of agent frontends

Department. Typical structure of a DCS is shown in the Fig. 4a). Screenshots of GUIs of example agents is shown in the Fig. 4b). The agents are implemented as independent, self-contained pieces of software. Each of them performs a specific role in the

DCS, the roles of the three pictured agents are: Plant Agent, Controller Agent and Operation Station (SCADA) Agent. They work in real-time and the framework fulfills the role of both vertical and horizontal communication channels. Agents can be started and stopped at any time during the experimentation, the framework registers these events and the other agents adapt to a specific configuration. Such behavior enables teachers to split specific learning tasks among students to encourage cooperation and collaboration.

It should be noted that the various agents have different level of complexity. A control algorithm consists of one-two equations, while a plant contains about a dozen of them. A complexity level of an operation station agent depends on a complexity of data processing needed, from several equations to dozens.

3 Visualisation System

During the framework's exploitation it turned out, that users of the framework have no knowledge on the internal nature of currently simulated system. The only available visualisation was display of current values of state vector X, locally presented by the distributor agent; there was no information on agents present in the system and logical connections between them. Some degree of human-machine interaction was additionally achievable by developing specialised agents interacting with the system and presenting graphical user interface. However, situation in which it was needed to develop such custom agents for each of the possible use of the framework was error prone; therefore it was decided to implement global framework-level visualisation system.

There is much work done in the field of distributed and parallel systems analysis and visualisation (see [13] for example), usual approach is to keep both the system and visualisation modular, distributed and widely accessible. While the presented framework varies from typical grid applications, guidelines can still be applied, so it was decided to implement a standard Web Service for the visualisation task. Using the data served by the Web Service various GUIs can be built according to various guidelines (for example dynamically suited to the user's preference – see [14]) Distributor application was modified to store additional resource i.e. a list of agents currently registered in the system. Each of agents connecting to the distributor declares which variables it consumes and produces; the distributor stores and organises this information. The distributor application was augmented with the embedded web server, the database of variables, agents and connections is served as a single live-updated XML file. Such an open and transparent way of serving database renders it accessible for nearly any programming environment capable of web access and XML parsing.

An example of a system which may use the implemented Web Service is the application for visualisation of self-organising agent systems presented in [15]. This applications employs the Ajax technique to show interactive block diagram visualising the internal structure of agent-based system organising according to unknown rules. In this case, the application shows block diagram of components of currently simulated model. An example of such visualisation is provided in the Fig. 5.

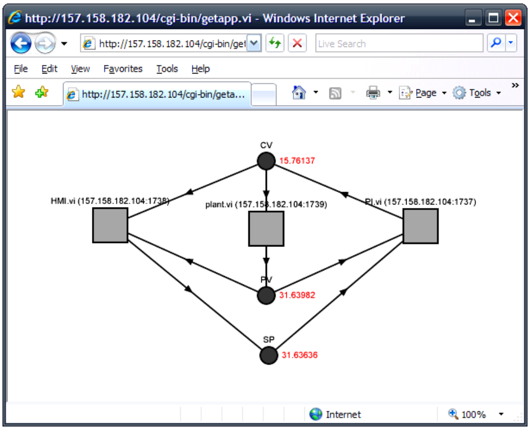


Fig. 5. Achieved visualisation (screenshot of the working monitoring application)

4 Concluding Remarks and Future Work

The agent-based framework presented satisfies all requirements needed in the specific field of control systems research and engineering. It allows for time-synchronised threads execution and supports interconnectivity between analytical computations and real world instrumentation. Remarkable case of the framework's successful practical use is the system described in [15], which required cooperation between several real semi-industrial plants and virtual simulated objects involving multiple users (students, teachers, and researchers).

Future work includes advanced development of the presented visualisation system, as currently it only briefly presents structure of the system and its current state. It is planned to augment it with the agents management capability (structure reconfiguration) and advanced visualisation and interaction options (graphical controls, charts, graphs), effectively transforming it into the complete human-machine interface for the whole framework.

Acknowledgments. This work was supported by the Polish Ministry of Science and Higher Education using funds for 2008-2010 under grant no. N N514 296335.

References

1. Bussmann, S., Jennings, N.R., Wooldridge, M.: Multiagent Systems for Manufacturing Control. Springer, Heidelberg (2004)
2. van Breemen, A., de Vries, T.: Design and implementation of a room thermostat using an agent-based approach. Control Engineering Practice 9, 233–248 (2001)
3. Linkens, D.A., Abbod, M.F., Browne, A., Cade, N.: Intelligent control of acryogenic cooling plant based on blackboard system architecture. ISA Transactions 39, 327–343 (2000)
4. Kang, J.-H., Oh, H.-k., Cho, S.-E., Kim, E.-G.: Performance of Fast TCP in Multi-agent Systems. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS, vol. 4496, pp. 21–30. Springer, Heidelberg (2007)

5. Weyns, D., Van Dyke Parunak, H., Michel, F., Holvoet, T., Ferber, J.: Environments for multiagent systems state-of-the-art and research challenges. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) E4MAS 2004. LNCS(LNAI), vol. 3374, pp. 1–47. Springer, Heidelberg (2005)
6. Polaków, G., Metzger, M.: Agent-Based Approach for LabVIEW Developed Distributed Control Systems. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS, vol. 4496, pp. 21–30. Springer, Heidelberg (2007)
7. Lee, K.C., Lee, S.: Performance evaluation of switched Ethernet for real-time industrial communications. Computer Standards & Interfaces 24, 411–423 (2002)
8. Loeser, J., Haertig, H.: Low-latency Hard Real-Time Communication over Switched Ethernet. In: Proceedings of 16th Euromicro Conference on Real-Time Systems, Catania, Sicily (July 2004)
9. Polaków, G., Metzger, M.: ppPDC Communication Framework - A New Tool for Distributed Robotics. In: Carpin, S., et al. (eds.) SIMPAR 2008. LNCS (LNAI), vol. 5325, pp. 191–202. Springer, Heidelberg (2008)
10. Pěchouček, M., Mařík, V.: Industrial deployment of multi-agent technologies: Review and selected case studies. Autonomous Agents and Multi-Agent Systems 17(3), 397–431 (2008)
11. Mařík, V., McFarlane, D.: Industrial adoption of agent-based technologies. IEEE Intelligent Systems 20(1), 27–35 (2005)
12. Polaków, G., Metzger, M.: Web-based monitoring and visualization of self-organizing process control agents. In: Hummel, K.A., Sterbenz, J.P.G. (eds.) IWSOS 2008. LNCS, vol. 5343, pp. 326–332. Springer, Heidelberg (2008)
13. Baliś, B., Bubak, M., Funika, W., Wismüller, R.: A monitoring system for multithreaded applications. Future Generation Computer Systems 19(5), 641–650 (2003)
14. Sobecki, J., Nguyen, N.T.: Adaptive user interfaces for general access web systems. In: Proceedings of the 14th International Conference on Systems Science. Oficyna Wyd, pp. 70–77. PWroc, Wrocław (2001)
15. Metzger, M., Polaków, G.: Holonic Multiagent-Based System for Distributed Control of Semi-industrial Pilot Plants. In: Mařík, V., Vyatkin, V., Colombo, A.W. (eds.) HoloMAS 2007. LNCS, vol. 4659, pp. 338–347. Springer, Heidelberg (2007)

Learning-Rate Adjusting Q-Learning for Two-Person Two-Action Symmetric Games⋆

Koichi Moriyama

The Institute of Scientific and Industrial Research, Osaka University
8-1, Mihogaoka, Ibaraki, Osaka, 567-0047, Japan
koichi@ai.sanken.osaka-u.ac.jp

Abstract. There are many multiagent Q-learning methods, and most of them aim to converge to a Nash equilibrium that is not desirable in games like the Prisoner's Dilemma (PD). The author proposed the utility-based Q-learning (UB-Q) for PD that used utilities instead of rewards so as to maintain mutual cooperation once it had occurred. However, UB-Q has to know the payoffs of the game to calculate the utilities and works only in PD. Since a Q-learning agent's action depends on the relation of Q-values, the mutual cooperation can also be maintained by adjusting the learning rate. Thus, this paper deals with the learning rate directly and introduces another Q-learning method called the learning-rate adjusting Q-learning (LRA-Q). It calculates the learning rate from received payoffs and works in other kinds of two-person two-action symmetric games as well as PD. Numeric verification showed success of LRA-Q, but, it also revealed a side-effect.

1 Introduction

In this paper, we consider the environment consisting of multiple autonomous actors called *agents* and investigate the use of reinforcement learning in such a multiagent environment. There are many multiagent reinforcement learning algorithms that have been proposed to date, e.g., minimax-Q [4], Nash-Q [3], FFQ [5], etc. Most of them aim to converge to a *Nash equilibrium* without knowing a payoff matrix in advance.

However, Nash equilibria are not desirable in some games. For example, Prisoner's Dilemma (PD) [1,7] is this kind of game. In PD, each player chooses either "cooperation" or "defection". Although it is best for both players to choose cooperation, the only Nash equilibrium is mutual defection because each player can obtain a higher payoff regardless of the opponent's choice if he/she chooses defection. Thus, we want a new reinforcement learning method that avoids Nash equilibria in games such as PD and that effects one of them in games in which it is desirable, without modification.

In the previous paper [6], the author proposed the utility-based Q-learning (UB-Q) for PD. A Q-learning [10] agent chooses an action based on its Q-function having Q-values of actions, and then updates its Q-function using a reward yielded by the action. UB-Q uses a *utility* instead of a reward[1] in order to maintain mutual cooperation once it

⋆ This work was supported by KAKENHI No.18700145 from MEXT, Japan.

[1] Here the terms "reward" and "utility" are an external stimulus and an internal stimulus elicited by the reward, respectively.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 223–232, 2009.

Table 1. Two-person two-action (2×2) symmetric game

$A \setminus B$	B_1	B_2
A_1	r_{11}, r_{11}	r_{12}, r_{21}
A_2	r_{21}, r_{12}	r_{22}, r_{22}

occurred. It is based on the idea that even if mutual defection continues, the agents may produce mutual cooperation because each agent usually uses a stochastic method in choosing actions so as to avoid local optima. If Q-value of cooperation becomes larger than that of defection after the mutual cooperation, each agent will choose cooperation afterward.

However, UB-Q has to know the payoff matrix to calculate the utilities and works only in PD. It means that UB-Q intentionally chooses a worse action when it is in other kinds of games and cannot follow a change of games without notice even if the new game is also PD. Since a Q-learning agent's action depends on the relation of Q-values, the mutual cooperation can also be maintained by adjusting the learning rate. Thus, this paper deals with the learning rate directly and introduces another Q-learning method called *the learning-rate adjusting Q-learning, or LRA-Q*. It calculates the learning rate from received payoffs, which enables it to follow a change of games without notice, and works in other kinds of two-person two-action symmetric games as well as PD.

This paper consists of five sections. In Section 2, we briefly see what two-person two-action symmetric games and Q-learning are. In Section 3, we prove theorems on the condition of the learning rate and introduce LRA-Q. We verify LRA-Q by numeric simulations in Section 4 and conclude this paper in Section 5.

2 Background

2.1 Two-Person Two-Action Symmetric Game

A two-person two-action (2×2) symmetric game is a game in which there are two *equal* players and each player has two actions. It is often shown by a bimatrix called a payoff matrix (Table 1). The players A and B choose their actions from the row of the matrix (A_1 or A_2) and from the column of the matrix (B_1 or B_2), respectively. After choosing actions, each player obtains a payoff in the corresponding entry of the matrix (left for A, right for B). Poundstone [7] discussed such games and pointed out four important ones, i.e., Deadlock, PD, Chicken, and Stag Hunt. (X, Y) stands for the pair of actions of both players such that X and Y are the actions of A and B, respectively.

In this paper, we deal with PD and Deadlock. PD has these relations among the payoffs: $r_{21} > r_{11} > r_{22} > r_{12}$. Each player obtains a larger payoff when he/she chooses A_2 (B_2), usually called "defection", regardless of the opponent's choice. As a result, the action pair becomes (A_2, B_2) and each player obtains a payoff r_{22}. However, it is more desirable for both players to choose the first actions (A_1, B_1), usually called "cooperation", and obtain r_{11}.

On the other hand, Deadlock has these relations among the payoffs: $r_{21} > r_{22} > r_{11} > r_{12}$. This game merely swaps r_{11} for r_{22} and vice versa in the PD payoff relations.

Each player also obtains a larger payoff when he/she chooses A_2 (B_2) regardless of the opponent's choice. Unlike PD, however, the action pair is acceptable because there is no action pair where both players obtain larger payoffs.

2.2 Q-Learning

Suppose an agent senses a state $s_t \in \mathcal{S}$ and chooses an action $a_t \in \mathcal{A}(s_t)$ at a discrete time t. $\mathcal{S}$ is a set of possible states of the environment and $\mathcal{A}(s_t)$ is a set of possible actions in the state s_t. After choosing an action, the agent receives a reward $r_{t+1} \in \mathbb{R}$ and senses a new state s_{t+1}. Q-learning updates an action value function Q by the following rule to make it approach the true value, which is the expected sum of rewards under the optimal policy π^* discounted by $\gamma \in [0, 1)$, i.e., $E_{\pi^*}(\sum_{k=0}^{\infty} \gamma^k r_{t+1+k})$.

$$Q_{t+1}(s,a) = \begin{cases} Q_t(s_t,a_t) + \alpha\,\delta_t & \text{if } (s,a) = (s_t,a_t), \\ Q_t(s,a) & \text{otherwise.} \end{cases}$$

$$\delta_t \triangleq r_{t+1} + \gamma \max_{a \in \mathcal{A}(s_{t+1})} Q_t(s_{t+1},a) - Q_t(s_t,a_t).$$

$\alpha \in [0, 1]$ is a parameter called the learning rate and δ_t is called TD error that approaches 0 when $Q(s,a)$ approaches the true value of the action a in the state s under π^*. For all s and a, $Q(s,a)$ provably converges to the true value under π^* with probability 1 when (i) the environment has the Markov property, (ii) the agent visits all states and takes all actions infinitely, and (iii) α decreases properly [10].

If the true action value function under the optimal policy, Q^*, is known, the agent can choose an optimal action a^* in a state s from Q^* by $a^* = \arg\max_{a' \in \mathcal{A}(s)} Q^*(s,a')$. However, if the agent always chooses such actions during learning, Q may converge to a local optimum because the agent may not visit all states and take all actions. To avoid it, the agent usually uses a stochastic method like ε-greedy [9] to choose actions. ε-greedy method chooses an action having the maximum Q with probability $1 - \varepsilon$ and a random action with probability ε.

3 Learning-Rate Adjusting Q-Learning

Hereafter, we deal with a situation in which two Q-learning agents that do not know the payoff matrix iteratively play either a PD game or a Deadlock game. To continue mutual cooperation in PD, it is necessary to provoke mutual cooperation and maintain it. Here we entrust the former to luck and deal only with the latter. We also does not consider state variables in Q-learning at all. This means that agents do not remember action sequences. Sandholm and Crites [8] showed that normal Q-learning agents produced no mutual cooperation in this context. In the previous paper [6], the author proposed the utility-based Q-learning method for maintaining mutual cooperation provoked fortuitously in PD. In this paper, however, we introduce another approach that encourages the mutual cooperation in PD by adjusting the learning rate of Q-learning.

Without loss of generality, we deal only with the row player of Table 1. According to PD researches, we use the symbols C and D to show the first actions (A_1 and B_1) and the second actions (A_2 and B_2), respectively. The suffixes of the payoffs in the table are also changed. Let the discount factor $\gamma \in [0, 1)$ be constant.

3.1 UB-Q: Utility-Based Q-Learning

Here is the utility-based Q-learning (UB-Q) method [6] for maintaining mutual cooperation provoked fortuitously in PD. When (C, C) occurs, UB-Q uses a utility $r_{cc} + r$ so as to make the Q-value of C ($Q(C)$) larger than that of D ($Q(D)$), such that r satisfies

$$r \geq \frac{r_{dd} - (\alpha r_{cc} + (1-\alpha) r_{cd})}{\alpha}. \tag{1}$$

The learning rate $\alpha \in (0, 1)$ is constant. See [6] for details.

3.2 LRA-Q: Learning-Rate Adjusting Q-Learning

In this section, we introduce the learning-rate adjusting Q-learning (LRA-Q) that satisfies the following two conditions by adjusting the learning rate; (i) when $Q(C) < Q(D)$, one-shot (C, C) makes $Q(C)$ larger than $Q(D)$, and (ii) when $Q(C) > Q(D)$, there is no action pair that makes $Q(D)$ surpass $Q(C)$. The first is identical with the purpose of UB-Q and the second guarantees that the mutual cooperation continues even after at least one of the players takes D by chance.

Initially, we prove a theorem about the first condition.

Theorem 1. *Suppose $Q_t(C) < Q_t(D)$ and the agent received a reward r_{t+1} after one-shot mutual cooperation at t. When $Q_t(C) < r_{t+1} + \gamma Q_t(D)$, we get $Q_{t+1}(C) > Q_{t+1}(D)$ iff the learning rate $\alpha \in [0, 1]$ satisfies*

$$\alpha > \frac{Q_t(D) - Q_t(C)}{r_{t+1} + \gamma Q_t(D) - Q_t(C)}. \tag{2}$$

On the other hand, it is impossible that $Q_{t+1}(C) > Q_{t+1}(D)$ when $Q_t(C) \geq r_{t+1} + \gamma Q_t(D)$.

Proof. $Q_{t+1}(C)$ and $Q_{t+1}(D)$ are calculated from $Q_t(C)$ and $Q_t(D)$ as follows.

$$Q_{t+1}(C) = (1-\alpha)Q_t(C) + \alpha(r_{t+1} + \gamma Q_t(D)) = Q_t(C) + \alpha(r_{t+1} + \gamma Q_t(D) - Q_t(C)),$$
$$Q_{t+1}(D) = Q_t(D).$$

Then,

$$Q_{t+1}(C) - Q_{t+1}(D) = Q_t(C) - Q_t(D) + \alpha(r_{t+1} + \gamma Q_t(D) - Q_t(C)). \tag{3}$$

When $Q_t(C) < r_{t+1} + \gamma Q_t(D)$, from Formula 2 and Equation 3, we get $Q_{t+1}(C) - Q_{t+1}(D) > 0$. Conversely, if $Q_{t+1}(C) > Q_{t+1}(D)$, Equation 3 directly derives Formula 2.

When $Q_t(C) = r_{t+1} + \gamma Q_t(D)$, Equation 3 becomes $Q_t(C) - Q_t(D) < 0$.

When $Q_t(C) > r_{t+1} + \gamma Q_t(D)$, since Equation 3 is a monotonically decreasing function of α, it has the supremum $Q_t(C) - Q_t(D) < 0$ when $\alpha = 0$. □

Next we deal with the second condition. This condition is divided into the two subconditions[2]; (i) when the action pair is (C, D), $Q(C)$ does not fall below $Q(D)$, and (ii) when the action pair is (D, C), $Q(D)$ does not surpass $Q(C)$. Here we consider theorems similar to Theorem 1. Proofs are omitted because they are similar to that of Theorem 1.

[2] The case of (D, D) is ignored here because $r_{cc} > r_{dd}$ in PD.

Theorem 2. *Suppose $Q_t(C) > Q_t(D)$ and the agent received a reward r_{t+1} after (C, D) at t. We always get $Q_{t+1}(C) > Q_{t+1}(D)$ when $r_{t+1} \geq (1 - \gamma)Q_t(C)$. When $r_{t+1} < (1 - \gamma)Q_t(C)$, we get $Q_{t+1}(C) > Q_{t+1}(D)$ iff the learning rate $\alpha \in [0, 1]$ satisfies*

$$\alpha < \frac{Q_t(C) - Q_t(D)}{(1 - \gamma)Q_t(C) - r_{t+1}}. \tag{4}$$

Theorem 3. *Suppose $Q_t(C) > Q_t(D)$ and the agent received a reward r_{t+1} after (D, C) at t. We always get $Q_{t+1}(C) > Q_{t+1}(D)$ when $r_{t+1} \leq Q_t(D) - \gamma Q_t(C)$. When $r_{t+1} > Q_t(D) - \gamma Q_t(C)$, we get $Q_{t+1}(C) > Q_{t+1}(D)$ iff the learning rate $\alpha \in [0, 1]$ satisfies*

$$\alpha < \frac{Q_t(C) - Q_t(D)}{r_{t+1} - Q_t(D) + \gamma Q_t(C)}. \tag{5}$$

Now, we introduce another Q-learning method, called the learning-rate adjusting Q-learning, or LRA-Q. It uses the following learning rate;

$$\alpha = \begin{cases} \alpha_1 & \text{if } (C, C) \text{ and } Q_t(C) < Q_t(D), \\ \alpha_2 & \text{if } (C, D) \text{ and } Q_t(C) > Q_t(D), \\ \alpha_3 & \text{if } (D, C) \text{ and } Q_t(C) > Q_t(D), \\ \alpha_0 & \text{otherwise,} \end{cases}$$

such that α_n $(n = 1, 2, 3)$ satisfies Theorem n and α_0 is the default learning rate. Note that α is truncated if it is not in [0,1]. Other components of Q-learning are not changed.

Next theorem shows that LRA-Q can be used in other kinds of 2×2 symmetric games as well as PD.

Theorem 4. *Suppose the learning rate α satisfies Theorem 1 under $Q_t(C) < r_{cc} + \gamma Q_t(D)$. The possible interval of α is denoted as A. Then the condition s.t. $\inf \mathsf{A} < 1$ is*

$$r_{cc} > (1 - \gamma)Q_t(D). \tag{6}$$

Especially, when we suppose $Q_t(D)$ is a consequence such that (D, D) has continued for a long time, i.e., $Q_t(D) = r_{dd}/(1 - \gamma)$ (see [6] for details), the condition becomes

$$r_{cc} > r_{dd}. \tag{7}$$

Proof. From Formula 2, since $r_{t+1} = r_{cc}$,

$$\inf \mathsf{A} = \frac{Q_t(D) - Q_t(C)}{r_{cc} + \gamma Q_t(D) - Q_t(C)}. \tag{8}$$

If Formula 6 holds, we get

$$r_{cc} + \gamma Q_t(D) - Q_t(C) > Q_t(D) - Q_t(C).$$

Thus $\inf \mathsf{A} < 1$. Conversely, if $\inf \mathsf{A} < 1$, we get Formula 6 directly from Equation 8.

Formula 7 is directly derived by substituting $Q_t(D)$ in Formula 6. □

Recall the payoff relation of Deadlock: $r_{dd} > r_{cc}$. If α_1 always made $Q_{t+1}(C) > Q_{t+1}(D)$, the agent would intentionally choose a worse action. Fortunately, however, Deadlock violates Formula 7, and thus, α is truncated and becomes insufficient to make $Q_{t+1}(C) > Q_{t+1}(D)$. This means the agent chooses D in Deadlock, which is the satisfactory action. Consequently, the theorem tells us that LRA-Q can be used in other kinds of 2×2 symmetric games such as Deadlock as well as PD without modification.

4 Numeric Verification

We conducted three experiments in self-play style. The first was in PD, the second was in Deadlock, and the third was in games that changed from PD to Deadlock and vice versa so as to check if LRA-Q was able to keep up with the change without notice. LRA-Q was compared with a normal Q-learning (Normal) and UB-Q. $Q_0(C)$ and $Q_0(D)$ were both set to 0 and γ was set to 0.5. We used ε-greedy method with $\varepsilon = 0.05$.

4.1 Experiment 1: Prisoner's Dilemma

We used the Axelrod's payoff matrix [1], i.e., $r_{cc} = 3$, $r_{cd} = 0$, $r_{dc} = 5$, and $r_{dd} = 1$. We set $\alpha = 0.25$ for Normal and UB-Q. Since Formula 1 stated that $r \geq (1 - 3\alpha)/\alpha$, r should be larger than 1, and we used 1.2. They were identical with those in [6]. We also set $\alpha_0 = 0.25$ for LRA-Q. The learning rates α_1, α_2, and α_3 do not include the values indicated respectively by Formulae 2, 4, and 5. Thus we set α_1 to the value of Formula 2 plus 0.025, α_2 to that of Formula 4 minus 0.025, and α_3 to that of Formula 5 minus 0.025. It was possible that $\alpha_1 < \alpha_0$. In this case, we used α_0 instead of α_1 because α_0 also satisfied Theorem 1. Similarly, we used α_0 instead of α_2 (α_3) when $\alpha_0 < \alpha_2$ (α_3).

Figure 1 shows typical Q-value transitions of the three methods. $Q(D)$ of Normal was always larger than $Q(C)$ even though (C, C) occurred several times, where $Q(C)$ rose suddenly in the plot. On the other hand, $Q(C)$ of UB-Q became larger than $Q(D)$ at the 2447th iteration. $Q(C)$ of LRA-Q also surpassed $Q(D)$ at the 2109th iteration. Although it is insignificant when $Q(C)$ surpassed $Q(D)$ because it depended on randomness, it is important that, after $Q(C)$ surpassed $Q(D)$, LRA-Q was stabler than UB-Q.

Table 2 shows the statistics of 100 simulations on the number of (C, C)s from the 5001st iteration to the 10000th one. To evaluate UB-Q and LRA-Q, at least one mutual cooperation during the first 5000 iterations is needed. Thus, we take account of the results only from *valid simulations* in which $Q(C)$s of both agents surpassed $Q(D)$s by (C, C) at least once in the first 5000 iterations. The statistics showed that LRA-Q succeeded constantly in achieving mutual cooperation in PD, although Normal failed.

4.2 Experiment 2: Deadlock

We used a payoff matrix that merely swapped r_{cc} for r_{dd} and vice versa in the PD game of Experiment 1, i.e., $r_{cc} = 1$, $r_{cd} = 0$, $r_{dc} = 5$, and $r_{dd} = 3$. In this game, Formula 1 stated that $r \geq (3 - \alpha)/\alpha$. We also set $\alpha = 0.25$ for comparison with Experiment 1. In this case, r should be larger than 11, and we used 11.4. Additionally, since this r was too large, we also conducted an experiment with $\alpha = 1$, in which r should be larger

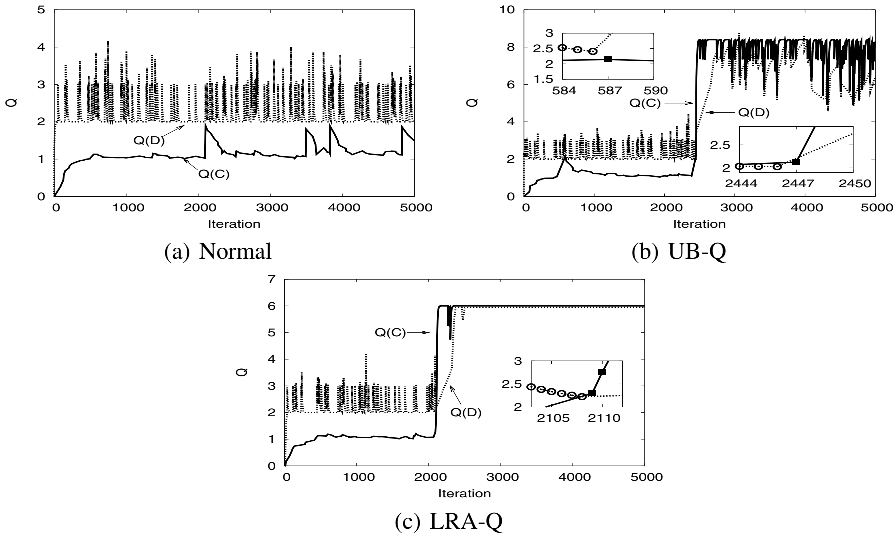


Fig. 1. Typical Q-value transitions in the Axelrod's Prisoner's Dilemma game

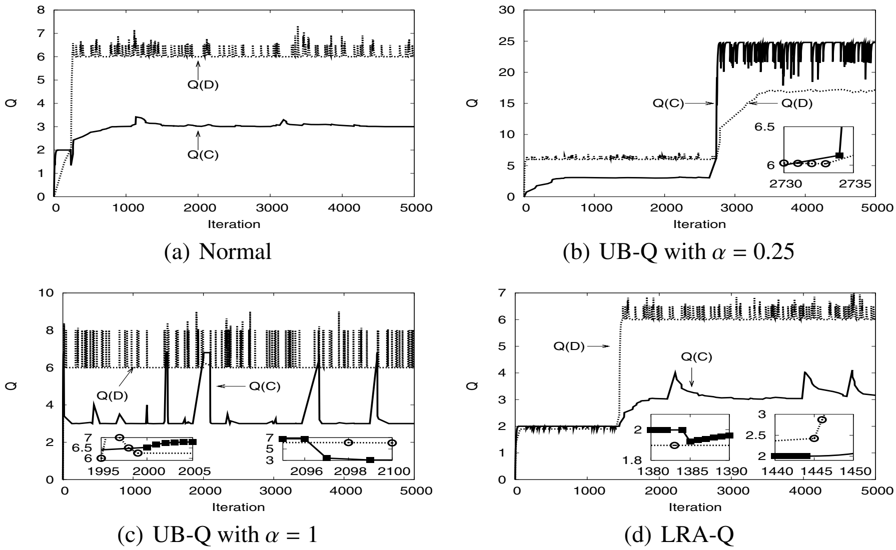
Table 2. Statistics of 100 simulations on the number of (C,C)s from the 5001st iteration to the 10000th one in PD

	Avg	SD	Min	Q1	Med	Q3	Max	# of valid sim
Normal	25.69	72.55	0	2	3	5	389	—
UB-Q	3719.64	1158.34	1	3276	4402	4460	4538	88
LRA-Q	4706.86	467.91	125	4742	4753	4766	4799	98

than 2 and we used 2.4. Settings of Normal and LRA-Q were completely identical with those in Experiment 1.

Figure 2 shows typical Q-value transitions of the four methods. $Q(D)$ of Normal became larger than $Q(C)$ even if $Q(D) < Q(C)$ at the beginning. With $\alpha = 0.25$, $Q(C)$ of UB-Q still surpassed $Q(D)$ even if $Q(D) > Q(C)$ at the beginning. This is an undesirable result of UB-Q. On the other hand, in UB-Q with $\alpha = 1$, $Q(D)$ surpassed $Q(C)$ in most of iterations. Note that, however, from the 2000th iteration to the 2096th iteration, for example, $Q(C)$ temporarily surpassed $Q(D)$. In LRA-Q, even if $Q(C) > Q(D)$ at the beginning, $Q(D)$ successfully surpassed $Q(C)$ after (D,D) occurred at the 1445th iteration. Similar to Experiment 1, it is insignificant when $Q(D)$ surpassed $Q(C)$.

Table 3 shows the statistics of 100 simulations on the number of (D,D)s from the 5001st iteration to the 10000th one. Here we also take account of the results only from valid simulations in which $Q(D)$s of both agents surpassed $Q(C)$s by (D,D) at least once in the first 5000 iterations. Although Normal succeeded constantly in achieving (D,D), UB-Q with $\alpha = 0.25$ failed to learn to take D. LRA-Q also succeeded constantly in achieving (D,D). UB-Q with $\alpha = 1$ also succeeded, but it is in dispute if it succeeded in *learning*, because the Q-value was not updated gradually but *replaced* in one shot.



(a) Normal

(b) UB-Q with $\alpha = 0.25$

(c) UB-Q with $\alpha = 1$

(d) LRA-Q

Fig. 2. Typical Q-value transitions in the Deadlock game

Table 3. Statistics of 100 simulations on the number of (D, D)s from the 5001st iteration to the 10000th one in Deadlock

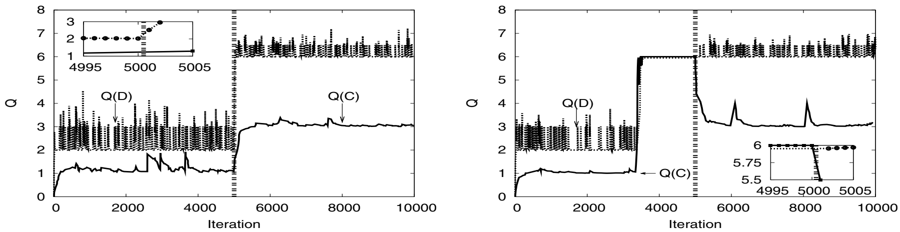
	Avg	SD	Min	Q1	Med	Q3	Max	# of valid sim
Normal	4756.25	14.93	4719	4747	4755	4767	4808	—
UB-Q (.25)	354.62	783.82	0	2	3	6	3029	78
UB-Q (1)	4687.93	54.01	4460	4656	4703	4724	4766	100
LRA-Q	4754.37	14.91	4696	4747	4755	4763	4798	100

4.3 Experiment 3: Changing Games

This experiment was only for Normal and LRA-Q because UB-Q should be notified of the change to recalculate the utility. Both PD and Deadlock settings were identical with Experiments 1 and 2, respectively. Settings of Normal and LRA-Q were also identical with the above experiments, i.e., $\alpha = \alpha_0 = 0.25$ and $\gamma = 0.5$.

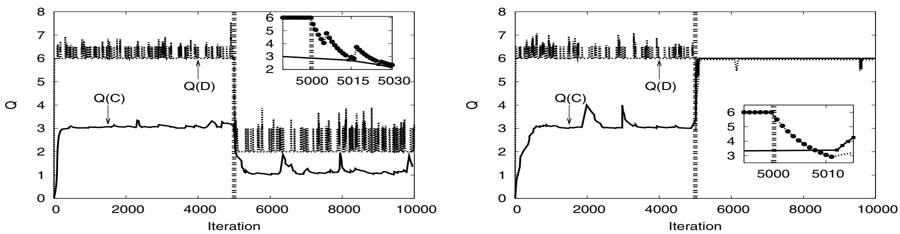
Figure 3 shows typical Q-value transitions of Normal and LRA-Q in a game that changed from PD to Deadlock at the 5000th iteration. The vertical dotted line at the center of each plot shows the time of change. For both Normal and LRA-Q, the first half and the second half were similar to Figures 1 and 2, respectively. We also see that, although LRA-Q had taken C at the end of the first half, it took D after $Q(C)$ suddenly fell below $Q(D)$ at the beginning of the second half. In contrast, Normal merely took D in both halves. Table 4 also shows this fact statistically.

Figure 4 shows typical Q-value transitions of Normal and LRA-Q in a game that changed from Deadlock to PD at the 5000th iteration. This figure also shows that, for both Normal and LRA-Q, the first half and the second half were similar to Figures 2 and



(a) Normal (b) LRA-Q

Fig. 3. Typical Q-value transitions in the game that changed from PD to Deadlock



(a) Normal (b) LRA-Q

Fig. 4. Typical Q-value transitions in the game that changed from Deadlock to PD

1, respectively. However, the statistics shown in Table 5, especially the median and the third quartile of Normal and the first quartile of LRA-Q, were quite different from those in Table 2. This difference means that Normal became more cooperative and LRA-Q became less cooperative. One of the reasons Normal became more cooperative was that, as in Figure 4(a), $Q(C)$ before the change was larger than $Q(D)$ after the change. This plot also shows that, after the change, $Q(D)$ decreased quickly and became similar to $Q(C)$. It was possible that $Q(C)$ surpassed $Q(D)$ if (C,C) occurred at such iterations.

One of the reasons LRA-Q became less cooperative was a side-effect of α_2. Figure 4(b) shows that, also in LRA-Q, $Q(D)$ decreased quickly after the change. If the paces of $Q(D)$ descents of both agents were different, one agent switched its action to C earlier than the other. Suppose the switched agent was the row player A. Then, the action pair became (C,D) and A's $Q(C)$ was updated. At that time, since the condition of α_2 was satisfied and applied, $Q(C)$ did not fall below $Q(D)$. Consequently, A did not return its action to D. From the viewpoint of the column player B, since it obtained

Table 4. Statistics of 100 simulations on the number of (D,D)s from the 5001st iteration to the 10000th one in the game that changed from PD to Deadlock at the 5000th iteration

	Avg	SD	Min	Q1	Med	Q3	Max
Normal	4753.14	14.59	4716	4744	4751	4764	4793
LRA-Q	4751.31	15.66	4705	4741	4749	4761	4797

Table 5. Statistics of 100 simulations on the number of (C,C)s from the 5001st iteration to the 10000th one in the game that changed from Deadlock to PD at the 5000th iteration

	Avg	SD	Min	Q1	Med	Q3	Max
Normal	165.45	208.1	0	4	90	236	1099
LRA-Q	2666.22	2310.6	97	127	4721	4745	4788

larger payoffs r_{dc} afterward, $Q(D)$ became too large to be surpassed by $Q(C)$. Thus, (C,D) continued for a long time. It is clearly an adverse effect of α_2 and needs to be resolved in the future.

5 Conclusion

This work introduced the learning-rate adjusting Q-learning (LRA-Q) based on the theorems on the learning rate in PD games. LRA-Q (i) makes $Q(C)$ surpass $Q(D)$ by one-shot (C,C), (ii) sustains $Q(C) > Q(D)$ after (C,D), and (iii) prevents $Q(D)$ from surpassing $Q(C)$ after (D,C). This work also showed that LRA-Q can be used in other kinds of games such as Deadlock. Numeric verification showed success of LRA-Q, but, it also revealed a side-effect.

In addition to resolving the side-effect, there remain several future works. First, we have to derive a condition of eliciting the first (C,C) or (D,D) from the agents. Second, LRA-Q should be compared with other methods that also modify the learning rate during learning, e.g., WoLF series [2][3]. Third, in this paper we deal only with PD and Deadlock games. LRA-Q should be tested in other 2×2 games, e.g. Chicken and Stag Hunt games, and be extended to more general games, e.g., n-person games.

References

1. Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984)
2. Bowling, M., Veloso, M.: Multiagent learning using a variable learning rate. Artificial Intelligence 136, 215–250 (2002)
3. Hu, J., Wellman, M.P.: Nash Q-Learning for General-Sum Stochastic Games. Journal of Machine Learning Research 4, 1039–1069 (2003)
4. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: Proc. ML 1994, pp. 157–163 (1994)
5. Littman, M.L.: Friend-or-Foe Q-Learning in General-Sum Games. In: Proc. ICML 2001, pp. 322–328 (2001)
6. Moriyama, K.: Utility Based Q-learning to Maintain Cooperation in Prisoner's Dilemma Games. In: Proc. IAT 2007, pp. 146–152 (2007)
7. Poundstone, W.: Prisoner's Dilemma. Doubleday, New York (1992)
8. Sandholm, T.W., Crites, R.H.: Multiagent reinforcement learning in the Iterated Prisoner's Dilemma. BioSystems 37, 147–166 (1996)
9. Sutton, R.S., Barto, A.G.: Reinforcement Learning. MIT Press, Cambridge, MA (1998)
10. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning 8, 279–292 (1992)

[3] Note that WoLF series also aim to converge to a Nash equilibrium, though.

Indirect Alignment between Multilingual Ontologies: A Case Study of Korean and Swedish Ontologies

Jason J. Jung[1], Anne Håkansson[2], and Ronald Hartung[3]

[1] Knowledge Engineering Laboratory
Yeungnam University
Gyeongsan, Korea
j2jung@intelligent.pe.kr

[2] Department of Computer and Systems Sciences
Stockholm University
Kista, Sweden
Anne.Hakansson@dsv.su.se

[3] Department of Computer Science
Franklin University
Columbus, Ohio, USA
Hartung@franklin.edu

Abstract. The existing ontology alignment methods have been trying to automatically obtain semantic correspondences between ontologies. However, they assume that all elements of source ontologies are written by identical languages. In this paper, we have introduced a theoretical idea for building indirect alignment between multilingual ontologies. Thereby, a novel architecture to reuse and compose alignments between ontologies has been designed. For a simple case study, we have collected two multilingual ontologies written by Korean and Swedish languages.

Keywords: Multilingual ontologies, Ontology alignment, Composition.

1 Introduction

Ontology alignment (or mapping) is significantly important to provide semantic interoperability between distributed information systems. When a system is trying to automatically search for a certain information from others and exchange knowledge with others, the essential assumption of such functionalities is that all of the ontologies are mapped with each other, in advance [1]. As shown in Fig. 1, assume that two systems of which knowledge (and resources) are managed by Ontology_1 and Ontology_2 are needed to cooperate with each other. The ontologies have be aligned by any methods (i.e., manually or automatically). As a result, a set of pairs (called correspondences) of ontological elements (e.g., concepts, properties, and so on) are discovered.

One of the representative examples of interoperability is query transformation [2]. A query which is composed of a set of concepts from source's ontology can

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 233–241, 2009.

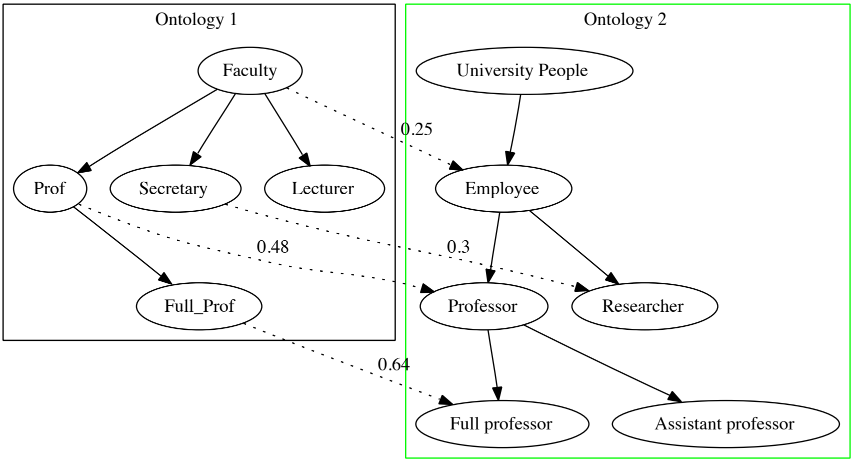


Fig. 1. Alignment between two ontologies

be replaces by referring to the alignment information between two ontologies of source and destination. A query "`Full_Prof`" from Ontology$_1$ can be "`Full professor`".

However, ontology alignment is regarded as an expensive process. Domain experts have to be trained to understand the whole (or partial) semantics and knowledge represented in the ontologies. Even though there have been several algorithms (e.g., GLUE [3,4], PROMPT [5], MAFRA [6], FOAM [7], and OLA [8]) for (semi-)automatically conducting ontology mapping, human experts should be involved to obtain better mapping results.

More serious problem as the motivation of this work is that the ontologies are written in different languages. In this case, such ontology mapping algorithms do not work completely, because most of them are based on string matching between lexicon components (e.g., labels of concepts, properties, and instances) from the ontologies.

In order to solve this problem, we propose a novel idea of *indirect* mapping between multilingual ontologies on distributed ontology environment. Main idea is to share and reuse *direct* mappings between ontologies, which already exist within such a semantic environment.

The outline of this paper is as follows. In the following section 2, we want to define some notations for the rest of this paper. Sect. 3 describes ontology alignment composition approach for building indirect mapping between multilingual ontologies as the main contribution of this work. For enhancing the performance of alignment composition, Sect. 4 shows how to import language corpora and mappings between them. In Sect. 5, we introduce some previous studies related to this work. They are mainly from information retrieval and knowledge management systems communities. Finally, in Sect. 6, we draw a conclusion of this paper.

2 Notations on Ontology Mapping

Main aim of ontology mapping is to find out semantic correspondences between all possible pairs of entities in two ontologies. Here, we introduce our formalisms with notations which is used in this paper.

Definition 1 (Ontology). *An ontology* $\mathbb{O}$ *is represented as*

$$\mathbb{O} := (\mathcal{C}, \mathcal{R}, \mathcal{E}_{\mathcal{R}}, \mathcal{I}_{\mathcal{C}}) \tag{1}$$

where $\mathcal{C}$ *and* $\mathcal{R}$ *are a set of classes (or concepts), a set of relations (e.g., equivalence, subsumption, disjunction, etc), respectively.* $\mathcal{E}_{\mathcal{R}} \subseteq \mathcal{C} \times \mathcal{C}$ *is a set of relationships between classes, represented as a set of triples* $\{\langle c_i, r, c_j \rangle | c_i, c_j \in \mathcal{C}, r \in \mathcal{R}\}$, *and* $\mathcal{I}_{\mathcal{C}}$ *is a power set of instance sets of a class* $c_i \in \mathcal{C}$.

By a certain way of ontology mapping efforts Ψ (e.g., manual, or any automatic methods), we can obtain an alignment result $\mathcal{A}^{\Psi}$ which is a set of correspondences between ontological elements [9].

Definition 2 (Correspondence). *Given two ontologies* $\mathbb{O}_i$, $\mathbb{O}_j$ *with associated entity languages* L_i *and* L_j *and a set of alignment relations* $\mathcal{R}$, *a correspondence is a quadruple:*

$$\langle e, e', r, n \rangle_{e \in \mathbb{O}_i, e' \in \mathbb{O}_j} \tag{2}$$

where $r \in \mathcal{R}$ *and* $n \in [0, 1]$ *is a confidence level.*

Definition 3 (Alignment). *Once we choose a alignment strategy* Ψ *for conducting, alignment between two ontologies* $\mathbb{O}_i$, $\mathbb{O}_j$ *is represented as a set of correspondences;*

$$\mathcal{A}^{\Psi}_{ij} = \{\langle e, e', r, n \rangle | e \in \mathbb{O}_i, e \in \mathbb{O}_j\} \tag{3}$$

For more explanation of this ontology mapping, we suggest to refer to [10].

3 Reusing and Composing Alignments

As mentioned in previous section, multilingual ontologies (e.g., between Korean and Swedish ontologies) must be aligned by human experts. Thus, only possible solution can be that there should be some software tools (e.g., visualization) to help human experts to improve the understandability of ontologies. Once we have either applied ontology mapping algorithms or conducted manual matching between ontologies, the results of ontology alignment should be opened in public.

In order to support reusability of the ontology alignments, we have to construct a certain authorized environment We can assume that the alignments between ontologies should be stored in a *centralized alignment repository*, and be freely available and sharable on distributed ontology environment. Euzenat [11] has been introduced and discussed a variety of alignment processes within an alignment infrastructure. Hence, the alignments stating semantic correspondences between ontologies can be represented by using alignment metadata and alignment ontology[1] for generating in some RDF serialization forms.

[1] It is available on http://www.omwg.org/TR/d7/

Table 1. Example of two multilingual ontologies (i.e., English.rdf and Swedish.rdf) for a flight-book service

```
<rdf:RDF
  xmlns:rdf =  "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:daml = "http://www.daml.org/2000/12/daml+oil#"
  xmlns:service = "http://www.ai.sri.com/daml/ontologies/services/1-0/Service.daml#"
  xmlns:time = "http://www.ai.sri.com/daml/ontologies/sri-basic/1-0/Time.daml#"
  xmlns:p = "http://www.ai.sri.com/daml/ontologies/services/1-0/Process.daml#">

<!--    English ontology, English.rdf    -->
<rdfs:Class rdf:ID="BookFlight">
<rdfs:subClassOf rdf:resource="p:Event"/>
</rdfs:Class>

<rdf:Property rdf:ID="PassengerNumber">
<rdfs:subPropertyOf rdf:resource="p:input"/>
<rdfs:range rdf:resource="#daml:Integer"/>
</rdf:Property>

<!--  Swedish ontology, Swedish.rdf  -->
<rdfs:Class rdf:ID="BokaFlyg">
<rdfs:subClassOf rdf:resource="p:Event"/>
</rdfs:Class>

<rdf:Property rdf:ID="Passagerarantal">
<rdfs:subPropertyOf rdf:resource="p:input"/>
<rdfs:range rdf:resource="#daml:Integer"/>
</rdf:Property>
```

Example 1. Two ontologies for a flight-book service have build by domain experts, and, as shown in Table 1, they are written by English and Swedish languages, respectively. Human expert should figure out the correspondences between ontological entities (e.g., concepts and properties) of both ontologies, as follows.

– $\langle BookFlight, BokaFlyg, \equiv, 0.9\rangle$
– $\langle PassengerNumber, Passagerarantal, \sqsubseteq, 0.7\rangle$

The alignment results can be obtained as shown in Table 2.

In this paper, we focus on composition of alignments, which contain semantic correspondences between multilingual heterogeneous ontologies. Consequently, we can expect that if there are well organized alignments between several ontologies, we can automatically build alignments between rest of multilingual ontologies.

As an example in Fig. 2, we assume the following conditions;

– Three ontologies $\mathbb{O}_i$, $\mathbb{O}_j$, and $\mathbb{O}_k$ are written in Korean, English, and Swedish, respectively.
– Only two alignments *i)* $\mathcal{A}_{ij}$ between Korean $\mathbb{O}_i$ and English $\mathbb{O}_j$ and *ii)* $\mathcal{A}_{jk}$ between English $\mathbb{O}_j$ and Swedish $\mathbb{O}_k$ are manually discovered.

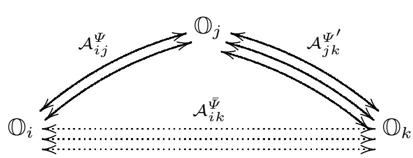


Fig. 2. Indirect alignment by composition

Table 2. Example of two multilingual ontologies for a flight-book service

```
<?xml version=1.0 encoding=utf-8 standalone=no?>
<!DOCTYPE rdf:RDF SYSTEM "align.dtd">
<rdf:RDF xmlns=http://knowledgeweb.semanticweb.org/heterogeneity/alignment
        xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
        xmlns:xsd=http://www.w3.org/2001/XMLSchema#>
<Alignment>
  <xml>yes</xml>
  <level>0</level>
  <type>**</type>
  <onto1>http://ke.yu.ac.kr/English.rdf</onto1>
  <onto2>http://ke.yu.ac.kr/Swedish.rdf</onto2>
  <map>
    <Cell>
      <entity1 rdf:resource='http://ke.yu.ac.kr/English.rdf#BookFlight' />
      <entity2 rdf:resource='http://ke.yu.ac.kr/Swedish.rdf#BokaFlyg' />
      <measure rdf:datatype='&xsd;float'>0.9</measure>
      <relation>=</relation>
    </Cell>
    <Cell>
      <entity1 rdf:resource='http://ke.yu.ac.kr/English.rdf#PassengerNumber'/>
      <entity2 rdf:resource='http://ke.yu.ac.kr/Swedish.rdf#Passagerarantal'/>
      <measure rdf:datatype=&xsd;float>0.7</measure>
      <relation>=</relation>
    </Cell>
  </map>
</Alignment>
</rdf:RDF>
```

Under the circumstance, alignment $\mathcal{A}_{ik}$ between Korean $\mathbb{O}_i$ and Swedish $\mathbb{O}_k$ can be found out not only by human's manual efforts (expensive) but also by automatic composition of the existing alignments $\mathcal{A}_{ij}$ and $\mathcal{A}_{jk}$ (inexpensive).

Definition 4 (Composition). *Given two alignments $\mathcal{A}_{ij}$ and $\mathcal{A}_{jk}$, if there exist a certain bridging entity connecting two multilingual correspondences, the composed alignment $\mathcal{A}_{ik}$ is given by a set of composed correspondences*

$$\mathcal{A}_{ik} = \mathcal{A}_{ij} \cdot \mathcal{A}_{jk} \tag{4}$$

$$= \{\langle e, e''', \mathcal{F}_{Rel}(r, r'), \mathcal{F}_{Conf}(n, n')\rangle | e \in \mathbb{O}_i, e''' \in \mathbb{O}_k\} \tag{5}$$

where

- $e \in \mathbb{O}_i$, $e' \in \mathbb{O}_j$, $e'' \in \mathbb{O}_j$, $e''' \in \mathbb{O}_k$,
- $\langle e, e', r, n\rangle \in \mathcal{A}_{ij}$, $\langle e'', e''', r', n'\rangle \in \mathcal{A}_{jk}$,
- *the bridging entity* $e' \equiv e''$ *(or* $e' \sqsubseteq e''$, $e' \sqsupseteq e''$*) and*
- $\mathcal{F}_{Rel}$ *and* $\mathcal{F}_{Conf}$ *are functions for composing two relations and two confidence values, respectively.*

The bridging entity is not necessarily identical in both correspondences (e.g., $e' = e''$). If there exist meaningful relations (e.g., $e' \sqsubseteq e''$, $e' \sqsupseteq e''$) within a same ontology, we can expect another alignment composition in the future.

Regarding relation composition, we have to consider composition function $\mathcal{F}_{Rel}$. Thanks to Euzenat [9], we can obtain a composition table stating relation algebra for determining the composed relations between the given two relations of correspondences. For example, $\{\equiv\} \cdot \{\equiv\} = \{\equiv\}$ and $\{\sqsubseteq\} \cdot \{\equiv\} = \{\sqsubseteq\}$.

Again, for computing the composed confidence value, $\mathcal{F}_{Conf}$ can be designed in various different ways, e.g.,

- multiplication $\mathcal{F}_{Conf}(n, n') = n \times n'$
- normalization $\mathcal{F}_{Conf}(n, n') = \frac{n \times n'}{2}$, and
- minimization (or maximization) $\mathcal{F}_{Conf}(n, n') = \min(n, n')$ (or $\max(n, n')$).

Example 2. Given three multilingual ontologies (i.e., Korea, English and Swedish languages) for a flight-book service, human experts have found out semantic alignments *i)* $\langle HangGongKwon, BookFlight, \equiv, 0.9\rangle$[2] between Korea and English ontologies, and *ii)* $\langle BookFlight, BokaFlyg, \sqsubseteq, 0.7\rangle$ between English and Swedish ontologies. Composition strategies are chosen with Euzenat [9] for $\mathcal{F}_{Rel}$ and multiplication for $\mathcal{F}_{Conf}$. As a result, we can compose them as

$$\langle HangGongKwon, BokaFlyg, \sqsubseteq, 0.63\rangle. \tag{6}$$

Fig. 2 can be modified as shown in Fig. 3.

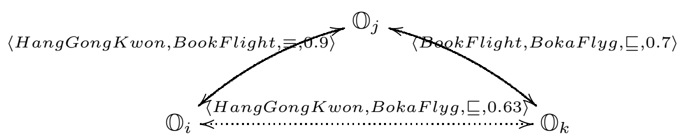


Fig. 3. Indirect alignment by composition

Now, in order to align multilingual ontologies by alignment composition, we have to consider the following two issues;

Similarity between languages. Two languages for the given ontologies are derived from a same root language (e.g., Latin language family). There might be some common vocabularies with same semantics. Therefore, manual alignment between the given ontologies are expensive, and

Composition path. The existing alignments from the centralized alignment repository can be chosen to establish an appropriate linkage (if possible, more that one linkage) which is connecting between the given ontologies. We refer to this linkage as a *composition path.*

4 Importing Language Corpora

As an way of improving align composition-based indirect ontology alignment, we are considering to exploit mapping information between language corpora. In fact, this language corpus has been applied to machine translation (MT) from natural language processing (NLP) communities [12].

[2] HangGongKwon is a Korean work meaning a flight tickets.

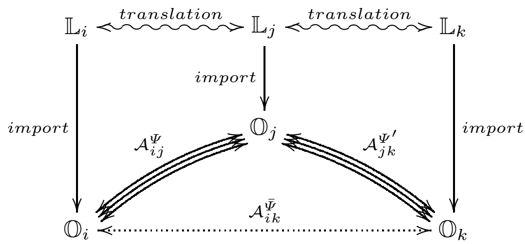


Fig. 4. Indirect alignment by incorporating with language corpora

Definition 5 (Language corpus). *A language corpus is defined as a set of vocabularies. Because they have useful information about i) several word senses (e.g., synonym, antonym, and hypernym), and ii) statistics (e.g., frequencies and co-occurrences) we can retrieve and apply relevant information to various applications.*

In addition, similar to alignment between different ontologies, there are corpus mappings between language corpora, which have been manually developed by experts. This information, particularly, is feasible to not only machine translation but also multilingual ontology alignment, as shown in Fig. 4.

In this paper, main reason for importing language corpora and their mappings is to claim that multilingual ontology alignment should employ semi-automatic approaches. By referring to the corpus mappings, human experts can reduce the mistakes of ontology alignments.

5 Related Work and Discussions

Now, we have to consider the general case where a number of semantic-based information systems are incorporating with each other. To make them automatically interoperable, intelligent agent system should be employed. The multi-agent system can manage the large-scale communications for semantic correspondences. In [13], multiple agents are free to refer to semantic correspondences between external ontologies, and evaluate them whether they agree it or not for building consensual alignments among ontologies. Similar to this work, the proposed approach is also assuming that the ontologies and alignments between ontologies should be opened and accessible.

In this context, we can discuss scalability testing of ontology-based systems. As the number of ontologies are increasing, the cost of manual alignment between ontologies is exponentially increased ($O(N)$). If we can compose the existing alignments, only one manual alignment is required with a new ontologies ($O(1)$).

In viewpoint of ontology engineering, ontology matching is very important to maintain a large number of ontologies over time [14]. In particular, international companies need to fulfill knowledge-based resource management from many different counties [15,16].

Multilingual information retrieval methods have been proposed [17,18,19]. This retrieval systems were applied in digital libraries [20,21]. Furthermore, recently, a special issue for multilingual knowledge management systems has been published [22].

6 Concluding Remarks and Future Work

This paper is a theoretical paper to explain the basic idea of indirect alignment between multilingual ontologies by composing alignments which already exist. In conclusion, a semi-automatic mapping tool should be implemented and exploited to support human experts by recommending candidate alignments between multilingual ontologies.

As future work, we want to figure out correspondence patterns [23] to not only string matching-based alignment but also reasoning-based alignment. In this paper, we have simply made a list of all candidates for composition functions $\mathcal{F}_{Rel}$ and $\mathcal{F}_{Conf}$. For justifying which candidate function is better in which case, we have to conduct experimentations.

Experimental result. We need to conduct experimentations for evaluating the proposed methods.

Dynamic alignments. Ontologies can be evolving over time. It means that ontology alignment might be influenced. Thus, alignment composition should be efficiently updated.

Composing multiple alignments. More generally, the indirect alignment between two ontologies is needed to compose multiple direct alignments.

Acknowledgement

This work was supported by Korea-Sweden Research Collaboration Program funded by Korean Science and Engineering Foundation (208-C-000-280).

References

1. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg (2007)
2. Jung, J.J.: Query transformation based on semantic centrality in semantic social network. Journal of Universal Computer Science 14(7), 1031–1047 (2008)
3. Doan, A., Madhavan, J., Domingos, P., Halevy, A.Y.: Learning to map between ontologies on the semantic web. In: Proceedings of the Eleventh International World Wide Web Conference (WWW 2002), pp. 662–673 (2002)
4. Doan, A., Madhavan, J., Domingos, P., Halevy, A.Y.: Ontology matching: A machine learning approach. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, pp. 385–404. Springer, Heidelberg (2004)
5. Noy, N.F., Musen, M.A.: Prompt: Algorithm and tool for automated ontology merging and alignment. In: Proceedings of the 17th National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, Austin, Texas, USA, July 30 - August 3, pp. 450–455. AAAI Press / The MIT Press (2000)

6. Maedche, A., Motik, B., Silva, N., Volz, R.: Mafra - a mapping framework for distributed ontologies in the semantic web. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS, vol. 2473, pp. 235–250. Springer, Heidelberg (2002)
7. Ehrig, M., Sure, Y.: Foam - framework for ontology alignment and mapping - results of the ontology alignment evaluation initiative. In: Ashpole, B., Ehrig, M., Euzenat, J., Stuckenschmidt, H. (eds.) Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies, Banff, Canada, October 2. CEUR Workshop Proceedings, vol. 156, CEUR-WS.org. (2005)
8. Euzenat, J.: An API for ontology alignment. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 698–712. Springer, Heidelberg (2004)
9. Euzenat, J.: Algebras of ontology alignment relations. In: Sheth, A., Staab, S. (eds.) Proceedings of the 7th International Semantic Web Conference (ISWC 2008), Karlsruhe, Germany. LNCS, vol. 5318, pp. 387–402. Springer, Heidelberg (2008)
10. Jung, J.J.: Ontological framework based on contextual mediation for collaborative information retrieval. Information Retrieval 10(1), 85–109 (2007)
11. Euzenat, J.: Alignment infrastructure for ontology mediation and other applications. In: Proceedings of the First International Workshop on Mediation in Semantic Web Services (MEDIATE 2005), Amsterdam, The Netherlands, December 12, pp. 81–96 (2005)
12. Hutchins, W.J., Somers, H.L.: An Introduction to Machine Translation. Academic Press, London (1992)
13. Laera, L., Blacoe, I., Tamma, V.A.M., Payne, T.R., Euzenat, J., Bench-Capon, T.J.M.: Argumentation over ontology correspondences in mas. In: Durfee, E.H., Yokoo, M., Huhns, M.N., Shehory, O. (eds.) Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, p. 228 (2007)
14. Hu, W., Qu, Y., Cheng, G.: Matching large ontologies: A divide-and-conquer approach. Data & Knowledge Engineering 67(1), 140–160 (2008)
15. O'Leary, D.E.: A multilingual knowledge management system: A case study of fao and waicent. Decision Support Systems 45(3), 641–661 (2008)
16. Jung, J.J.: Taxonomy alignment for interoperability between heterogeneous virtual organizations. Expert Systems with Applications 34(4), 2721–2731 (2008)
17. Oard, D.W., Dorr, B.J.: A survey of multilingual text retrieval. Technical report, Institute for Advanced Computer Studies, University of Maryland, College Park (1996)
18. Sheridan, P., Ballerini, J.P.: Experiments in multilingual information retrieval using the spider system. In: Proceedings of SIGIR (1996)
19. Hull, D.A., Grefenstette, G.: Querying across languages: A dictionary-based approach to multilingual information retrieval. In: Proceedings of SIGIR (1996)
20. Fox, E.A., Marchionini, G.: Toward a worldwide digital library. Communications of the ACM 41(4), 29–32 (1998)
21. Larson, R.R., Gey, F., Chen, A.: Harvesting translingual vocabulary mappings for multilingual digital libraries. In: Proceedings of JCDL 2002 (2002)
22. Yang, C.C., Wei, C.P., Chen, H.: Editors' introduction special issue on multilingual knowledge management. Decision Support Systems 45(3), 551–553 (2008)
23. Scharffe, F., Ding, Y., Fensel, D.: Towards correspondence patterns for ontology mediation. In: Shvaiko, P., Euzenat, J., Giunchiglia, F.F., He, B. (eds.) Proceedings of the 2nd International Workshop on Ontology Matching (OM 2007). CEUR Workshop Proceedings, vol. 304, CEUR-WS.org (2007)

Extracting Relations towards Ontology Extension

Jin-Guk Jung[1], Kyeong-Jin Oh[1], and Geun-Sik Jo[2]

[1] Intelligent e-Commerce Systems Laboratory,
Department of Information Engineering, Inha University,
253 Yonghyun-Dong, Nam-Gu, Incheon, Korea 402-751
gj4024@gmail.com,okjkillo@eslab.inha.ac.kr

[2] Department of Information Engineering, Inha University,
253 Yonghyun-Dong, Nam-Gu, Incheon, Korea 402-751
gsjo@inha.ac.kr

Abstract. Extracting local ontology from domain-specific documents for the purpose of acquiring knowledge or semantic information to extend their ontologies is considered very important. Main components of ontology are concepts and relations between concepts. In this paper, we focus on extracting triples, in which verbs are relations and subjects/objects are concepts, from documents based on natural language. Further, we show that term frequency is the most reliable measure among *tf-idf* and entropy on evaluating relations extracted from documents, particularly the aircraft maintenance manual.

1 Introduction

Ontology is a formal specification for representing knowledge and sharing conceptualization. Main components of ontology are concepts, relations between concepts, attributes, and constraints for restricting use of ontology. Individuals (Instances) are concrete objects such as people and animals compared to ontology as abstract model. In the recent years, ontologies for various domains are developed manually and many researchers exploit them for their machine or application to comprehend semantics of documents and to share domain knowledge in wide research fields like information retrieval and knowledge management.

However, there are some problems such as bottleneck on ontology engineering task and cost of time and paper work for building ontologies manually. To solve these problems many researchers tried to create ontologies from domain texts in past years because they might think that domain documents include the knowledge of the domain [1,2,3,4,5]. Even though automatic methods for extracting ontologies from texts are not complete until now and they are dependant on other natural language processing techniques(NLP), ontology engineers, who are responsible for building domain ontologies, expect some automation on their work with the help of software program to alleviate high time consumption and tedious paper work to construct ontology. In this paper we also assume

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 242–251, 2009.

that domain documents have rich information available for constructing domain ontologies.

We first found that the original aircraft maintenance manual (AMM) given to our project[1] (IAR) exhibits a few information for us who are all non-experts about that domain. The manual is written by the Standard Generalized Markup Language (SGML). We did not find public ontologies for our project. However, we have wanted software agents to maintain efficiently our knowledge management system (KMS). Therefore, we have felt of building domain ontology. At the first time, we attempted to convert the manual to documents represented in the Resource Description Framework (RDF) or the Web Ontology Language (OWL) using methods described in [10]. It means that we only considered information about the document structure and a few other factors. As the semantics information was not enough, it became difficult for us to exploit content of the documents, so that we have considered ontology extension entailing the content of texts.

Our approach to extend the existing ontologies like our initial ontologies is derived from the observation that there is a large amount of concepts and relations between the concepts in the manual. We conducted text preprocessing as one SGML document was transferred into many XML documents per tasks and annotated them. For extracting relations from each sentence, we tried to discover triples from a dependency tree produced by a shallow parser. Finally, we evaluated the triples with some measures, and then we found out that term frequency is the most reliable measurement among *tf-idf* and entropy for a specific domain, particularly the aircraft maintenance.

The remainder of this paper is organized as follows. In the next section we describe some preliminaries and data model using in the paper. The following section describes whole extraction process, how to extract named entities and relations from texts, and how to evaluate them. Section 4 reports on how we have applied it in a specific case and shows the result. We discuss some problems faced during text processing and related work in Section 5, then conclude our work and point out future work in the last section.

2 Preliminaries and Data Model

2.1 Preliminaries

Predefined Ontology: As mentioned in the previous section, we assume that our system had already predefined domain ontology. In fact, we had built this ontology shown in Figure 1 into our system to be described in the next section, and another predefined ontology like *warning*, *figure*, *tool*, and *procedure*, so then extracted ontologies will belong to an appropriate concept as subclass. We use the automatically generated ontology, however, explanation for this process is beyond the scope of this paper. We will describe in details a method for extracting relations from texts in the next section.

[1] http://rafi.inha.ac.kr/eskm/iar/

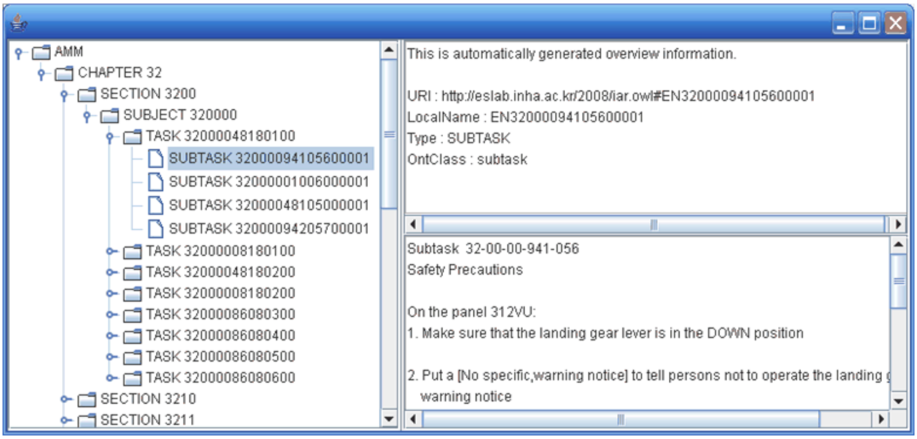


Fig. 1. The predefined ontology preserved the structure of AMM

```
<?xml version="1.0" ?>
<task>
        <warning>
                MAKE SURE THAT THE WHEEL CHOCKS ARE IN POSITION.
                MOVEMENTS OF THE AIRCRAFT COULD BE DANGEROUS.
        </warning>
        <figure>
                32-00-00-991-001            Fig. 201
        </figure>
        <tool>
                D23304000    1 PIN-GROUNDLOCK, NLG
                R 460007174  1 PIN-GROUNDLOCK, MLG
        </tool>
        <procedure>
                Installation of the MLG PIN-GROUNDLOCK, MLG (460007174):
                Make sure that the holes in the MLG lock stay are aligned.
                Make sure that the PIN-GROUNDLOCK, MLG (1) is in a clean and
                serviceable condition.
                Push and hold the button on the PIN-GROUNDLOCK, MLG (1) and
                install it in the holes of the MLG lock stay.
                Make sure that the flag is in view from the ground.
                Installation of the NLG PIN-GROUNDLOCK, NLG (D23304000):
                Make sure that the holes in the NLG lock stay are aligned.
                Make sure that the PIN-GROUNDLOCK, NLG (2) is in a clean and
                serviceable condition.
                Push and hold the button on the PIN-GROUNDLOCK, NLG (2) and
                install it in the holes of the NLG lock stay.
                Make sure that the flag is in view from the ground.
        </procedure>
</task>
```

Fig. 2. Sample document

Text Preprocessing: The other assumption is that our documents source comprises of preprocessed documents as shown in Figure 2. The document collection compiled from the airline maintenance manual of the model, A330. The documents comprise tasks on a chapter. Since the XML document was transferred from the SGML document, we could preserve some information of the document structure such as figures to be referred, tools to be used, and sequential procedure in a task. These tags function as each concept.

2.2 Data Model

For finding semantic relations, we use *graph-based models* or *network-based models*. Therefore, all terms extracted from each sentence are represented as vertices and their interconnecting relations form the edges. The following vocabularies are interchangeable: graphs are networks, vertices are nodes, and edges are links. First, our data model has two layers, which are a *verb layer* and a *noun layer*, for separating concepts/nouns and relations/verbs. Second, each node of the graph corresponds to a verb or a noun and each edge represents a *potential* relation between two nodes, which are connected if they appear in the same context. An edge has weight for indicating strength of relationship between two connected nodes. Some outstanding work using *graph-based* approaches in NLP are described in [11].

3 Extracting Relations

We explain our approach about extracting relations through text processing in this section. First, we describe a procedure for extracting concepts and relations, after then methods for evaluating extracted terms and relations are introduced.

3.1 System Overview

Our system for extracting relations based on linguistic analysis is shown in Figure 3. As mentioned in the previous section, both the preprocessed XML documents and the generated ontology as input go through GATE[2] system [8]. Note that if the ontology already stored into a persistent repository by JENA[3] system, then it can be used as input of GATE. GATE is a framework which provides users with feasibility for building and annotating corpora. It can be freely available for processing texts written in several languages with a baseline set of language processing components. For identifying a subject, a verb, and an object of each sentence, we use MINIPAR [9] which is a shallow parser for producing dependency tree. The input of MINIPAR is each sentence produced by the sentence splitter in GATE. The next process builds network-based model using the identified terms from the dependency tree of MINIPAR. We extract

[2] http://www.gate.ac.uk/

[3] http://jena.sourceforge.net/

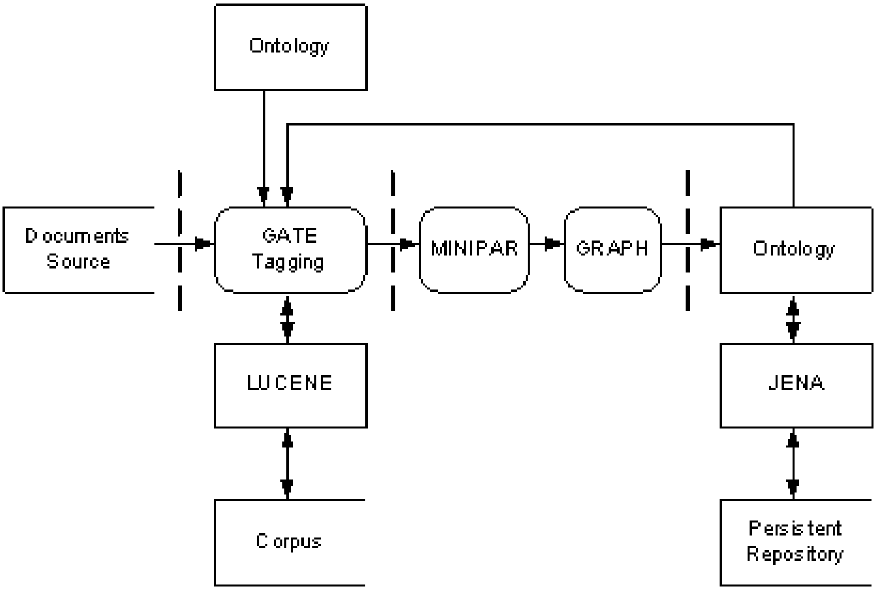


Fig. 3. System Overview

relations from the model and evaluate them with some metrics to be described. These relations produced previously will be evaluated by ontology engineers and will be stored into a persistent repository using JENA if they are accepted.

3.2 Linguistic Analysis

For the linguistic annotation, we exploit GATE, which provides a part-of-speech, lemmatization, and a multi-layered XML-format for a given text, and MINIPAR specifying dependency structure along with grammatical function assignment and phrase structure.

Named Entity Recognition. In order to annotate tools, figures, and parts in *procedure* tag as instances of existing ontology class labels such as *tool*, *figure*, and *part*, we conducted Named Entity Recognition (NER) based on some patterns. The annotations of tools and figures are straightforward. Pattern match technique is used for extracting domain-specific terms concerned with parts. Our pattern-based approach is motivated by the observation that there are many abbreviated entities like *Main Landing Gear (MLG)*. After that only the abbreviation appears. Therefore, we built a dictionary of these instances for further processing. The patterns are as follows:

– Initial cap of tokens with its abbreviation.
– Tokens with its number in a figure.
– Tokens turned into all upper case.
– Tokens turned into all upper case following digits.

Dependency Structure. Each sentence parsed by MINIPAR is represented as a tree. The root of the tree is typically the main predicate of the sentence or a dummy node which has the main predicate as the sole child. Edges in the tree are used to connect each word to its dependency parent.

Relation Extraction. The extracted triples are constructed from selected subject-predicate-object triples, where the subject is chosen as the domain of the relation, while the (direct/indirect) objects as well as the adjuncts define the range.

3.3 Measure

In order to decide significant terms and relations for the AMM domain, we need to quantitatively measure extracted terms whether or not each term is unusual. We exploit the following equations to calculate their ranks on *tf-idf* and entropy. The other is a typical frequency of word occurrence.

$$tf - idf(w) = tf(w) \cdot \log\left(\frac{N}{df(w)}\right), \tag{1}$$

where $tf(w)$ is a number of words occurrences in a document, $df(w)$ is a number of documents containing the word, and N is a number of all documents. $tf - idf(w)$ is relative importance of the word in the document.

$$entropy(w) = -\sum_{i=1}^{N} p_i \cdot \log(p_i), \tag{2}$$

where p_i is a probability of a word occurrences in the ith document, N is same as in the equation (1). Entropy also measures the amount of information associated with a word.

3.4 Evaluation

Evaluation methods for systems of extracting relations were proposed in some work. The task of performance evaluation, however, is one of difficult activities done in a automatically systematic way. Typical methods used on like our work are modified precision and recall adopted from Information Retrieval (IR). To conduct the task with the two metrics, we looked for the gold standard to compare with our extracted relations because two sets of terms are needed for computing precision and recall as you can see in below equations. One is a referred set (R) as the gold standard, and the other is a extracted set (E) from our system. However, we did not have the gold standard constructed by human or system for our domain. Notice that our predefined ontologies in Section 2 are not the gold standard with which to compare.

$$recall(R, E) = \frac{|R \cap E|}{|R|}, \tag{3}$$

$$precision(R, E) = \frac{|R \cap E|}{|E|}, \quad (4)$$

$$f - measure(recall, precision) = \frac{2 \cdot recall \cdot precision}{recall + precision}, \quad (5)$$

4 Experiment and Results

4.1 Building Gold Standard

We assume that our corpus belongs to a specific domain may involve remarkable terms that are relatively more frequent than ones in another collection of general articles such as the Wall Street Journal. To find these terms, we had processed MPQA[4] as a general corpus included 535 news articles collected from various sources [12], meanwhile, we had computed a significance value for the deviation between the relative frequencies to determine which terms are extraordinary one compared to another as described in [3]. We had used these terms ($z > 1.96$ and $z > 2.57$) to constitute a referred set described in Subsection 3.4. Notice that we had excluded some stop-words containing auxiliary and modal verbs from MPQA and our corpus.

4.2 Result

We constructed three groups of triples (subject-predicate-object/complement and YOU-predicate-object/complement) produced by MINIPAR. In our corpus, a lot of sentences are in the imperative mood because of a manual for workers, so that we had attached YOU as subject to the front of predicate-object pairs. A simple sentences has one main verb, a compound sentence has two more verbs, and both of them may have a gerund or an infinitive. Therefore, since we had dealt with different kind of these verbs, three groups of triples were built. Notice that if MINIPAR recommended a semantic subject for a gerund or an infinitive, usually omitted, we extracted it from a parsed tree. The first group (TFS) consisted of triples including only one main verb and connected words(subject and object) mined from each simple sentence, the second group (TFC) constituted TFS and another ones mined from compound sentences, and the third group (TFM) contained all implicative ones.

We first computed some scores on the extracted data by the way mentioned in the previous section in order to identify highly relevant relations. Table 1 showed verb list ordered by each score to demonstrate highly relevant words for our aircraft maintenance manual. For evaluating our system, we also calculated recall and precision on each triple set with the thresholds z, 1.96 (5%) and 2.57 (1%). Table 2 and Table 3 showed that the TFS set yielded a lower recall due to less cases about 81%, while produced the higher F-measure.

[4] Available at http://www.cs.pitt.edu/mpqa/

Table 1. Top 20 Relations in TFM

Rank\Measure	Frequency	TF-IDF	Entropy
1	be	prevent	kill
2	install	have	be
3	do	set	interface
4	prevent	make	depressurize
5	apply	include	notice
6	make	cause	control
7	kill	engage	let
8	remove	turn	access
9	set	install	do
10	depressurize	clean	get
11	use	isolate	examine
12	have	close	apply
13	access	bleed	supply
14	show	move	refer
15	put	spring	measure
16	open	connect	agree
17	control	apply	come
18	notice	operate	pump
19	engage	remove	occur
20	close	obey	start

Table 2. Recall, Precision, and F-measure with Five Percent

Type\Measure	Recall	Precision	F-measure
TFS	67.22	33.61	44.81
TFC	73.89	29.29	41.96
TFM	77.22	27.42	40.47

Table 3. Recall, Precision, and F-measure with One Percent

Type\Measure	Recall	Precision	F-measure
TFS	67.30	29.72	41.23
TFC	74.21	25.99	38.50
TFM	77.99	24.46	37.24

5 Discussion and Related Work

Since we filtered out stop-words, which are usually more frequent but less meaningful, from our corpus and MPQA corpus, we could obtain a very exciting results, for example, as shown in Table 1 we discovered that term frequency is the most reliable metric for our domain. *tf-idf* and entropy metrics are typically used for verifying whether or not words are interesting in the Information Retrieval and Text Mining literature, whereby these functions allow us to compute the amount of information conveyed by a particular term. In other words, these values make sense when regarding information as measuring the unexpectedness of the word. Unexpected terms have a higher information content than expected one. Unfortunately, they were not adequate for our domain because there were some missing words such as *use, show, put,* and *control*. These terms are considered as useful relations by our members joined the IAR project.

We also have shown that TFS set yielded a lower recall due to less cases about 81%, while produced the higher F-measure because of maintaining the precision relatively high at the same time. On each group, The threshold (z) for filtering out general terms took more effect on a precision. It means that removing too many terms with high threshold influences an intersection size of two corpora deeply. Our results compared to others [1,3,6] regarding recall and precision are qualified.

There exist some weakness or problems in our work. First, since MPQA corpus is relatively small we wonder that recall values were produced highly. Regarding the corpus used for building the gold standard on our work, we are planning similar experiments with more general and bigger corpus such as the Wall Street Journal corpus or Reuters than MPQA because it seems to show that different data can have an effect on the semantic extraction operations. Second, the language processor we used contained some incorrect parsing results. Even though most of the relations obtained by triples method were valid, our approach extracted only a small fraction of the total relations from domain texts, because triples in some cases had been obtained from these parse errors. Therefore, using only triples method may not be sufficient to find all relations exist in texts. Third, performing a fully automatic evaluation marks as another limitation. Therefore, this evaluation frequently implies a manual operation by an expert, or by the researchers themselves. In other words, a human effort is still needed to filter out incidental results, to handle semantically incomplete relations or to label some relations, we also attempted to expose the relations to the domain expert.

For constructing ontology some previous work done for general or specific purpose had exploited association rules [4], clustering [7], and formal concept analysis (FCA) methods [6] in order to identify concepts and relations between them from texts. These work were based on statistical and linguistic analysis, some patterns, or dependencies produced by linguistic shallow parser. Also, the work has been done for other specific domain, for example tourism domain and financial domain [1,3,6].

6 Conclusion

Even though automatic methods for extracting ontologies from texts are not complete until now and they are dependant on other natural language processing techniques(NLP), ontology engineers, who will construct domain-specific ontologies, hope for help of software programs on their task because they believe that those alleviate their sufferings like long time to build ontology and stresses to handle it. This work also showed a possibility for our software system to be able to alleviate the sufferings.

For a specific domain, especially the aircraft maintenance, we found out that term frequency is the most reliable measure among *tf-idf* and entropy. We also showed that our performance regarding recall and precision is qualified, roughly 72% and 30% with $z = 5\%$.

For future work, we want to consider prepositional phrases as candidates of relations mentioned in the future work part of [2] because we recognized that there are many cases of parts and devices located on (in) a particular place (parts) like *a sensor on each brake and a Brake Temperature Monitoring Unit (BTMU)*. In this phrase *sensor*, *brake*, and *Brake Temperature Monitoring Unit* are all parts (concepts). A label based on the associated preposition for representing the relationship between the concepts is needed.

References

1. Schutz, A., Buitelaar, P.: RelExt: A Tool for Relation Extraction from Text in Ontology Extension. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 593–606. Springer, Heidelberg (2005)
2. Punuru, J., Chen, J.: Extraction of Non-hierarchical Relations from Domain Texts. In: Proc. CIDM 2007 (2007)
3. Spyns, P., Reinberger, M.-L.: Lexically Evaluating Ontology Triples Generated Automatically from Texts. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 563–577. Springer, Heidelberg (2005)
4. Maedche, A., Volz, R.: The Text-to-Ontology Extraction and Maintenance System. In: ICDM Workshop on Integrating Data Mining and Knowledge Management (2001)
5. Sabou, M.: Extracting ontologies from software documentation: a semi-automatic method and its evaluation. In: Proc. ECAI-OLP 2004 (2004)
6. Cimiano, P., Hotho, A., Staab, S.: Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. Journal of Artificial Intelligence Research 24, 305–339 (2005)
7. Lin, D., Pantel, P.: Concept Discovery from Text. In: Proc. COLING 2002 (2002)
8. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: Proc. ACL 2002 (2002)
9. Lin, D.: Principar - an Efficient, Broad-Corverage, Principle-Based Parser. In: Proc. COLING 1994 (1994)
10. Bohring, H., Auer, S.: Mapping XML to OWL Ontologies. In: Proc. LIP 2005 (2005)
11. Radev, D.R., Mihalcea, R.: Networks and Natural Language Processing. AI magazine (2008)
12. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In: Proc. HLT-EMINLP 2005 (2005)

Deontic Logic Reasoning for Maintaining Ontology Consistency in Agent Networks

Krzysztof Juszczyszyn and Grzegorz Kołaczek

Institute of Computer Science
Wroclaw University of Technology, Wroclaw, Poland
grzegorz.kolaczek@pwr.wroc.pl, krzysztof.juszczyszyn@pwr.wroc.pl

Abstract. In this paper we propose a framework for guiding the processes of ontology alignment and negotiation in an multiagent environment. The information about the structure of the actual communication network and agent roles is used to support decisions about the operations of ontology alignment. The framework is based on the Deontic Logic which is used for reasoning.

1 Introduction

Sharing knowledge within a diverse multiagent environment requires a shared conceptual vocabularies - ontologies, which represent the formal common agreement about the meaning of data [15]. Artificial intelligence defines ontologies as explicit, formal specification of a shared conceptualization [12]. In this case, a conceptualization stands for an abstract model of some concept from the real world; explicit means that the type of concept used is explicitly defined. Formal refers to the fact that an ontology should be machine-readable; and finally shared means that ontology expresses knowledge that is accepted by all the subjects. In short - an ontology defines the terms used to describe and represent an area of knowledge.

Development of dynamic intelligent services is inevitably connected with so-called *semantic technologies* – functional capabilities that enable *both* humans and machines to create, discover, organize, share and process the meanings and knowledge [10]. This is achieved by the use of shared vocabularies or semantic nets. On the level of WWW this implies the adoption of the Semantic Web's XML-based standards for annotating and processing information, usually in the form of web ontologies [3]. Because ontologies are developed and managed independently the semantic mismatches between two or more ontologies are inevitable [1][20][22]. Practical applications show that fully shared vocabularies are exceptional - a number of possible different semantic conflicts was identified by Shaw and Gaines [27], other classifications were addressed in [16]. The vision of Semantic Web allowing agents to publish and exchange ontologies requires strong mechanisms supporting ontology merging and alignment [17].

The aim of this paper is to propose a framework for supporting *semantic interactions* in large multiagent communities. By semantic interaction we will understand the acts of modifying the internal knowledge representation (in the form of semantic network or ontology) as a result of communication between agents.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 252–262, 2009.

2 Agent Networks

Complex network structures emerge in many everyday situations among people (social networks), organizations, software agents, linked documents (WWW) and so on. Previous research has identified the most distinctive properties of such networks [7],[29]:

- Small diameter and average path length (of the order of Log(N) for N network nodes).
- High clustering (probability that the neighbors of any given node will be also each other's neighbors)
- A famous power-law (or scale-free) network node degree distribution.

These properties may be of use when simulating interaction between system components and building evolution models, and they form a basis of many robust and applicable theoretical results. It was shown that they influence the search strategies, communication and cooperation models, knowledge and innovation spreading etc. [28], [29].

Now consider the society of n agents with multiple ontologies. If we want to effectively resolve semantic mismatches within such a community without intervention of human operator, there are several aspects that must be addressed. The architecture of the proposed framework consists of a set of agents (interpreted as software components as the framework is proposed to be applied the Semantic Web environments), each of them is equipped with private vocabulary in the form of semantic net (or ontology). These structures may mutually overlap but there will also be inevitable differences.

Let's now list the parameters needed in our framework. Let $\boldsymbol{A} = \{A_1, A_2, \ldots A_n\}$ be a set of agents and $\boldsymbol{O} = \{O_1, O_2, \ldots O_n\}$the set of their private ontologies. Each agent A_i uses ontology O_i as an formal conceptualization of particular domain of interest. We denote the set of concepts of ontology O_i as C_i={ $c_1^i, c_2^i \ \ldots \ c_{m(i)}^i$ } and the relations between them as R_i. Each agent A_i has also an associated utility function $u_i : \boldsymbol{O} \rightarrow [0,1]$ which is to express the potential attractiveness of the other agents as communication counterparts of A_i. In the simplest A_i may define a list of important concepts and u_i will return the result based only on the set comparison between set of concepts of given ontology and the list of topics A_i is interested in. This means that A_i is willing to communicate with the agents which have knowledge on the specific topics.

The network structure of the system is represented by communication graph defined by n × n matrix G where an entry g_{ij} indicates the presence of directed link from the node (agent) A_i to A_j, and the graphs reflecting the structure of private agents' ontologies. Each of these ontologies O_i may be viewed as a graph $SemNet_i$ with nodes corresponding to concepts from C_i and edges corresponding to relations from R_i. According to [28] we assume that these graphs show small world properties.

People who interact with one another or share common interests form a social network. A social network is defined as a finite set of individuals, by sociologists called actors, who are the nodes of that network, and social ties that are the links (connections) between them [14]. Of course there exist Internet-based social networks which base on the computer networks services and infrastructure, those can be detected by means of analyzing communication activities. In the simplest (and most popular in literature so far) case only mail logs are analyzed but the other communication channels (chat service, blogs, discussion lists) can also be taken into account [9][19]. The set of all discovered relations constitutes the organizational social network. We will use this network as a guidance for maintaining ontological consistency within the system.

Communication-based social network is subject to periodic updates – ongoing communication between users may lead to creation of new relations which are reflected in updated network structure. If we consider that the users use their ontologies to perform data fusion we naturally aim to maintain the consistency of their knowledge models (ontologies), which involves ontology alignment [23][24]. Hence, in our approach, the lifecycle of our environment consists of several steps which are repeated in a cycle:

1. Performing communication activities.
   This means the usual users' activity – exchanging communicates, cooperation, discussion and so on.
2. Updating social network structure.
   Social network's structure is being updated if there appear new communication links. This is being done in the background.
3. Modifying private ontologies.
   The users (as knowledge creators) may modify their domain models (ontologies) by adding or removing concepts, relations or attributes.
4. Checking ontology consistency.
   Ontology consistency is checked along the social links detected in step 2. We assume performing ontology alignment only when needed and only between communicating users.

In our approach we use a Taxonomic Precision (TP), a similarity measure based on the notion of semantic cotopy (see def. 1 below) recently presented and analysed in [11]. The reason to chose this measure was its ability to compare ontologies as whole structures and along multiple dimensions. The TP will serve as a general measure of consistency between two ontologies. The legal values of TP are from the range [0,1]. Note that high value of TP means that the two users may consistently use resources annotated with their private ontologies, perform reasoning tasks etc. Now we introduce the basic definitions needed to formulate the notion of Taxonomic Precision.

Definition 1. The ontology O is a structure $O := (C, root, \leq C)$ where C is a set of concept identifiers and root is a designated root concept for the partial order $\leq C$ on C.

Definition 2. Semantic Cotopy *sc(c,O)* of a concept c from ontology O is a set containing c and all super- and subconcepts of c in O, excluding root concept root(O).

Definition 3. Taxonomic Precision of a concept c and the two ontologies O_1 and O_2 such that $c \in O_1$ and $c \in O_2$ is defined as:

$$tp(c, O_1, O_2) = \frac{|sc(c, O_1) \cap sc(c, O_2)|}{|sc(c, O_1)|} \tag{1}$$

Definition 4. *Global Taxonomic Precision* TP(O_1,O_2)of the two ontologies O_1 and O_2 is defined as:

$$TP(O_1,O_2)=\frac{1}{|C_1|}\sum_{c\in C_1}\begin{cases} tp(c,O_1,O_2) & if \quad c\in C_2 \\ 0 & if \quad c\notin C_2 \end{cases} \tag{2}$$

where:
C_1, C_2 – the sets of concepts of O_1 and O_2 respectively. Note, that the TP is asymmetric, this feature follows frequent approach according to which semantic similarity is asymmetrical; for example: there is an obvious inherent similarity between a concept and its superconcept (like *truck* and *vehicle*) but their domains are different and the first is contained in the second. Hence, its convenient to reflect this fact when defining semantic similarity measures.

5. Perform ontology alignment along the social links.
   If the global taxonomic precision is below some fixed threshold value (which means that the given pair of users have inconsistent domain models) we perform ontology alignment with help of chosen algorithm (for example: one of the listed in [18]). This may also involve ontology learning activities (exchanging/acquiring concepts from each other).

Without any other assumptions, the ontology alignment proposed in step 5 is performed on the peer-to-peer basis. In our approach we will use the Deontic Logic to provide a reasoning mechanism which will be role-sensitive and will restrict the alignment operations only to necessary ones which will help to decrease computational overload and maintain the semantic consistency in the agent community.

3 Roles in the Agent Society

Almost each group of subjects that interacts with each other create its own policy that can be defined either explicitly or implicitly. This policy is intended to influence the behavior of subjects and objects associated with the group. The policy constitutes a set of roles and describes relations between them. The roles are sets of rules that governs the behavior of all subjects that have been activated in it. Each subject active within the group can perform actions in a context of one or several defined in policy roles. In the paper we propose the approach based on the concept of role-based access control [2], [13], [25]. This approach uses roles to distinguish a few the most essential classes of software agents and to set the boundaries of their activities. The analysis of the evolution of relations between communicating software agents resulted in the following types of roles proposition (Table 1).

The question which of the agents may be referred to as *hubs* requires checking the network structure. In general, 5% of network nodes with high degree centrality may be called network hubs. As social and communication networks are mostly scale-free structures, these 5% of nodes group individuals with the number of links exceeding the others by the order of magnitude (or more) [6].

Table 1. List and short description of possible agents' roles

	Name of the role	Short description
1.	*Role_Creator*	This role describes the character of 'active' agents; where the types of the activity can be following: internal ontology modification. The knowledge creator is an agent who changes or creates from scratch his personal ontology. This involves the possibility of obligatory ontology alignment if the change has caused the knowledge inconsistency between agents.
2.	*Role_Hub*	This role brings together all agents that are the hubs in the sense of the communication between agents and the communication graph analysis; they are important in the communication structure and so in information flow. An agent who intensively communicates with the others is taking a hub role. From the point of view of network structure the hub is a node with high degree – there are many edges connecting it with the others.
3.	*Role_Reader*	All agents that play *reader* role are mainly interested in obtaining the knowledge from the other agents, they modify their interior structures (ontology) only as the result of obtaining some 'exterior' signals (the threshold value of the function signalizing the ontology alignment necessity)
4.	*Role_New*	*New* agents are the agents that have joined the network recently and so they do not have long history of interaction with other agents

However, degree centrality is not the only one indicator of the importance of a node in communication network. Intuitively, the nodes which serve as *bridges* (connecting nodes which do not form direct links with each other) are also very important, and their semantic consistency within the network should be maintained with special care. These nodes may be detected by computing so-called clustering coefficient, the measure reflecting the local graph connectivity [7]. The standard form of clustering coefficient is defined according to eq. 3:

$$CC = \frac{2|E(G1(n))|}{\deg(n)(\deg(n)-1)} \tag{3}$$

where:
deg(n) – denotes degree of node *n*,
|*E(G1(n))*| – is the number of edges among nodes in 1–neighbourhood of node *n*,

We also assume that for a node *n* such that $deg(n) \leq 1$ all clustering coefficient are 0. The intuitional meaning of the *CC* is that it represents how many edges exist within 1-edge radius from the node *n* compared to the number of possible edges. *CC* equalling 1 means that the nodes in 1-edge distance from *n* form a full graph.

Summing up, in order to be qualified as a *hub*, a node must belong to the 5% of the nodes with highest degree centrality and show *CC* characteristic for hubs linking different cliques (for social and communication networks it is the value of 0.1-0.5).

Together with roles' definition various types of constraints can be distinguished in dependence on a profile of environment or additional requirements. In the context of role-based access control the most frequently mentioned are the following types of constraints [8], [13], [26]:

- mutually exclusive roles,
- prerequisite roles,
- limitation of the maximum number of subjects (cardinality constraints).

A basic motivation for application of constraints in role-based access control is to reflect the high level policy of an enterprise at the level of access control [25]. The second reason why constraints should be considered in role-based access control model is accordance with one of the basic security principles – the principle of the least privilege.

In the presented approach we describe some discussion about application of constrains in the context of mobile multi agent environment.

4 Application of Deontic Logic in Roles Description

The developers of role-based access control have distinguished several mechanisms to govern the system's and organisational roles. There are three main categories of these mechanisms which are responsible for: definitions of roles, definition of role-entity relations, and definition of role-role relations.

The first step is the identification of a set of entities that may be active within the system and a set of activities. During the next steps, on the base of the system security policy, the relations between the elements of these two sets should be established.

Let us use the earlier defined set of agents *A* and let us denote $B = \{\text{action}_1, \text{action}_2, \ldots, \text{action}_m\}$ as the set of their activities. There are three possibilities for each $\text{action}_k \in B$ in relation to agents from the set *A*: action_k is permitted or action_k is obliged or action_k is forbidden.

In deontic logic it is possible to describe this relation using the modal operators: ***P*** - *it is permitted*, ***O*** - *it is obliged* and ***F*** - *it is forbidden*. According to these operators the sentences above can be formulated in the following way:

- ***P*** action_k,
- ***O*** action_k,
- ***F*** action_k..

Deontic logic is useful in this case because its basic notions are fundamental for normative perspective of system description and describes what is permitted, obligatory and forbidden, for a particular agent. The application of deontic logic allows a formal description and a formal analysis of the above-mentioned notions in the context of the agents behaviour.

We propose to use the formal model based on deontic logic for roles and agents activity description. It is composed of three parts [21]:

(a) *Syntax of the model language.*
It is based on the first-order logic syntax where three additional modal operators are added: ***P*, *O*, *F***.

(b) *Semantic of the model language.*
It is based on the Krippke semantic of possible world where the world accessibility relation is serial.

(c) *The language application rules*:

- action's permissions, obligations, prohibitions and action's requests are formulated in the language of the model,
- all the formulas used in description must be in a form of Horn's clauses,
- if *Reg* is a set of formulas describing permitted, prohibited, and obligatory activities and this set is defined for a particular agent, then this entity may perform all activities described by the formulas that are the logical consequences of the set *Reg*.

4.1 Automation of Reasoning in Role-Based Framework for Ontology Alignment

There are several tools that support the automation of reasoning in the first-order logic. One of them is PROLOG that uses Horn's clauses and the resolution method. This means that the ability to translate formulas of our model into first-order formulas in the form of Horn's clauses would open the application of PROLOG and the resolution method for ontology alignment process.

The following theorem states that it is possible to translate a particular class of role-based access control modal model formulas into form of the first-order Horn's clauses. This theorem makes use of the definition of a semi-functional translation.

The semi-functional translation $T_{sf}\,()$ of a modal logic is a projection that assigns modal formulas and possible world to formulas of the first order logic in the following way [4], [5]:

- $T_{sf}(\phi,x) = P(x)$,
  where ϕ is an atomic proposition and P is the corresponding predicate;
- $T_{sf}\,(\boldsymbol{O}\phi,x) = \forall\, y[R(x,y) \rightarrow T_{sf}(\phi,y)]$
  where R is a possible world accessibility relation;
- $T_{sf}\,(\boldsymbol{P}\phi,x) = \exists f\, T_{sf}(\phi,f(x))]$

where f is a function corresponding to the relation of possible world accessibility.

Theorem 1. $T_{sf}(\phi,w)$ is a conjunction of Horn's clauses iff a formula obtained after deleting all modal operators from the formula ϕ is a conjunction of Horn's clauses, where: ϕ is a formula of role-based access control modal model, $T_{sf}(\phi,w)$ means a semi-functional translation of ϕ, and w stands for a world selected from a set of possible worlds (Krippke model).

Proof. The proof of this theorem is based on the structural induction. The complete proof can be found at [21].

4.2 Validation of Agents' Activities in a Context of Policy Requirements

The policy governing behaviour of the autonomous agents can be described by an identified and defined set of roles. Each agent active within the system can be assigned to one or more roles, and it gets the authorisation to the set of actions that is a logical consequence of its set of roles. In this stage of the research we assume that agents play only one of the four previously defined roles.

Roles are defined by logical formulas. For example, let the role *Role_Creator* be assigned to the *Agent_1*. *Role_Creator* is defined by the following formulas:

Role_Creator:

a) $\forall a \forall o\ Play\_role(a,Creator) \wedge Internal\_ontology(a,o) \rightarrow \mathbf{P} Modify\_ontology(a,o)$

b) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,Creator) \rightarrow \mathbf{P} Communicate\ (a1,a2)$

c) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,New) \rightarrow \mathbf{P} Communicate\ (a1,a2)$

d) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,Reader) \rightarrow \mathbf{P} Communicate\ (a1,a2)$

e) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,Hub) \rightarrow \mathbf{P} Communicate\ (a1,a2)$

f) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,Creator) \wedge Difference\_level(a1,a2,threshold1) \rightarrow \mathbf{O} Align\_ontology(a1,a2)$

g) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,New) \wedge Difference\_level(a1,a2,threshold2) \rightarrow \mathbf{P}\ Align\_ontology(a1,a2)$

h) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,Reader) \wedge Difference\_level(a1,a2,threshold3) \rightarrow \mathbf{O}\ Align\_ontology(a1,a2)$

i) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,Hub) \wedge Difference\_level(a1,a2,threshold4) \rightarrow \mathbf{P}\ Align\_ontology(a1,a2)$

j) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,Creator) \wedge Communicate(a1,a2) \wedge Change\_level(a1,threshold1) \rightarrow \mathbf{O} Inform(a1,a2)$

k) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,Creator \wedge Communicate(a1,a2)) \wedge Change\_level(a1,threshold2) \rightarrow \mathbf{O} Inform(a1,a2)$

l) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,Creator \wedge Communicate(a1,a2)) \wedge Change\_level(a1,threshold3) \rightarrow \mathbf{O} Inform(a1,a2)$

m) $\forall a1 \forall a2 Play\_role(a1,Creator) \wedge Play\_role(a2,Creator \wedge Communicate(a1,a2)) \wedge Change\_level(a1,threshold4) \rightarrow \mathbf{O} Inform(a1,a2)$

Where rule a) says that all autonomous agents playing the role *Role_Creator* are permitted to modify their internal ontology. Rules b)-e) define the permission to communicate between agents playing different roles, rules f)-i) define the obligation or permission to perform ontology alignment when the fixed threshold has been reached. The last rules j)-m) describes the obligation of the agents playing *Role_Creator* to inform other agents about modifications introduced to their internal ontology.

The appropriate sets of rules for *Role_New, Role_Reader, Role_Hub* can be defined in similar way.

Apart from the definition of roles, logical values of several system variables must be set to reflect the current system state. For example:

- *Play_role(Agent_1, Role_Creator)* $\equiv$ *TRUE.*
- *Play_role(Agent_2, Role_Hub)* $\equiv$ *TRUE.*
- *Internal_ontology(Agent_1, Onto_car)* $\equiv$ *TRUE.*

- *Change_level(Agent_1, Max_3)* $\equiv$ *TRUE.*
- …

Where *Play_role*, *Internal_ontology*, *Change_level*, *Difference_level* are the predicate symbols. While a system policy is defined and the values of the system variables are known it is possible to verify the agent's requests. For example, an answer to the question about permission to set up communication between *Agent_1* and *Agent_2* can be looked for. To give an answer to this question an appropriate logical program should be generated. The logical program is a result of semi-functional translation of the formulas defining roles and system variable values. Finally, the logical program is as follows:

- *Plays(x,Agent_1, Role_Creator)*$\Leftarrow$
- *Plays(x,Agent_2, Role_Hub)*$\Leftarrow$
- *Internal_ontology(Agent_1, Onto_car)* $\Leftarrow$
- *Change_level(Agent_1, Max_3)*$\Leftarrow$
- …
- *R(x,f(x))* $\Leftarrow$
- *Modify_ontology(f(x),a,o)*$\Leftarrow$ *Play_role(x,a,Creator),Internal_ontology(x,a,o)*
- *Communicate (f(x),a1,a2)* $\Leftarrow$ *Play_role(x,a1,Creator)*$\wedge$*Play_role(x,a2,Hub)*
- …

The formula describing communication request is also translated and it is a question for the logical program. The access request after semi-functional translation:

- *Communicate(y,Agent_1,Agent_2)*

The final answer of the logical program in this example will be "YES". This means that the action requested by *Agent_1* in *Role_Creator* to communicate with *Agent_2* is admissible in the context of present policy definition.

5 Conclusions and Future Research

Mechanisms that govern evolution of emergent semantic structures in modern web-based multiagent environments are relatively new and not widely addressed research task. Its successful completion has potential to influence novel interconnection architectures (like Semantic Web Grids) in many ways. The most interesting are:

- Creating knowledge and innovation spreading models.
- Developing intelligent search algorithms.
- Formulating the conditions for semantic integrity of distributed systems.
- Support for knowledge-based virtual organizations.

The further development of the proposed framework includes conducting experiments and discovering the rules that govern evolution and behavior of the emerging Semantic Web environment and its underlying semantic network structures.

The Deontic Logic mechanism allows structuring and controlling the actions of ontology alignment with respect of the actual agent roles and the structure of communication network.

References

1. Bailin, S.C., Truszkowski, W.: Ontology Negotiation between Agents Supporting Intelligent Information Management. In: 5th International Conference on Autonomous Agents, Montreal, Canada, pp. 13–20 (2001)
2. Barkley, J., et al.: Role based Access Control for the World Wide Web. In: Proc. NIST-NSA Nat. Computer Security Conf., Baltimore, pp. 23–34 (1997)
3. Berners Lee, T., Hendler, J., Lasilla, O.: The Semantic Web. Sc. American (May 2001)
4. Bolc, L., et al.: Reasoning in non-classical logic. Theoretical basis, Akademicka Oficyna Wydawnicza PLJ, Warsaw (1995)
5. Bolc, L., et al.: Reasoning in non-classical logic. Reasoning automation, Akademicka Oficyna Wydawnicza PLJ, Warsaw (1998)
6. Carley, K., et al.: Toward an interoperable dynamic network analysis toolkit. Decision Support Systems 43, 1324–1347 (2007)
7. Carrington, P., Scott, J., Wasserman, S. (eds.): Models and Methods in Social Networks Analysis. Cambridge University Press, Cambridge (2005)
8. Chen, F., et al.: Constraints for Role-Based Access Control. In: Proc. ACM Workshop on RBAC, Gaithersburg, pp. 382–390 (1996)
9. Culotta, A., et al.: Extracting social networks and contact information from email and the Web. In: Proceedings of CEAS 2004, First Conference on Email and Anti-Spam (2004)
10. Davis, M.: Semantic Wave 2006 - Part 1: Executive Guide to Billion Dollar Markets, Project10X Special Report, Washington (2006)
11. Dellschaft, K., Staab, S.: On How to Perform a Gold Standard Based Evaluation of Ontology Learning. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 228–241. Springer, Heidelberg (2006)
12. Fensel, D., et al.: Ontology Management: Survey, Requirements and Directions. Deliverable 4, IST Project IST-1999-10132, Vrije Universiteit Amsterdam (2001)
13. Ferraiolo, D., Barkley, F.: Specifying and Managing Role-Based Access Control within a Corporate Intranet. In: Proc. ACM Workshop on RBAC, Gaithersburg, pp. 69–78 (1997)
14. Garton, L., et al.: Studying Online Social Networks. Journal of Computer-Mediated Communication 3(1) (1997)
15. Gomez-Perez, A., Corcho, O.: Ontology Languages for the Semantic Web. IEEE Intelligent Systems 17(1), 54–60 (2002)
16. Hameed, A., et al.: Detecting Mismatches among Experts' Ontologies Acquired through Knowledge Elicitation. In: 21st Int. Conf. on Knowledge Based Systems, UK, pp. 9–24 (2001)
17. Hendler, J.: Agents and the Semantic Web. IEEE Intel. Systems 16(2), 30–37 (2001)
18. Kalfoglou, Y., et al.: Ontology Mapping: the State of the Art. The Knowledge Engineering Review 18(1), 1–31 (2003)
19. Ke, J., Minett, J.W., Au, C.-P., Wang, W.S.-Y.: Self-organization and Selection in the Emergence of Vocabulary. Complexity 7(3), 41–54 (2002)
20. Klein, M.: Combining and Relating Ontologies: an Analysis of Problems and Solutions. In: IJCAI Workshop on Ontologies and Information Sharing, Amsterdam, pp. 53–62 (2001)
21. Kołaczek, G.: Model of Role Based Access Control mechanism, Ph.D. thesis, Technical University of Wrocław, Wrocław, Poland (2001)
22. Matheus, C.J., et al.: Derivation of ontological relations using formal methods in a situation awareness scenario, Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2003, pp. 298–309 (2003)

23. Meltzoff, A.N., Prinz, W.: The imitative mind: Development, evolution, and brain bases. Cambridge University Press, Cambridge (2002)
24. Mika, P.: Ontologies are us: A unified model of social networks and semantics. Journal of Web Semantics 5(1), 5–15 (2007)
25. Sandhu, R., et al.: Role-based access control: A multi-dimensional view. In: Proc. 10th Annual Computer Security Application Conf., Greater Orlando, pp. 54–62 (1994)
26. Sandhu, R.: Role hierarchies and constraints for lattice-based access controls. In: Proc. 4th Europ. Symp. Research in Computer Security, Rome, Italy, pp. 20–25 (1996)
27. Shaw, M.L.G., Gaines, B.R.: Comparing Conceptual Structures: Consensus, Conflict, Correspondence and Contrast. Knowledge Acquisition 1(4), 341–363 (1989)
28. Steyvers, M., Tannenbaum, J.: The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. Cognitive Science 29(1) (2005)
29. Watts, D., Strogatz, S.: Collective Dynamics of Small World Networks. Nature 393, 440–442

Multi-Agent System Process Control Ontology

Wim Zeiler[1,2], Gert Boxem[1], Rinus van Houten[1], Perica Savanovic[1], Joep van der Velden[2], JanFokko Haan[2], Willem Wortel[2], Paul Noom[2], and Rene Kamphuis[3]

[1] Technische Universiteit Eindhoven, Vertigo 6.28, 5600 MB Eindhoven, Netherlands
[2] Kropman Building Services, Lage Landseweg 84, 6545 CG Nijmegen, Netherlands
[3] ECN Petten, PO box 1, 1755 ZG Petten, Netherlands

Abstract. An optimal energy infrastructure is essential for adequate indoor comfort. In the design process of energy infrastructures there is no shared understanding (i.e. an ontology) of the design activities to come to an integral design of building and comfort systems.

The purpose of this paper is to achieve an understanding of design activities in the context of designing architecture for Multi agents to optimize process control of the energy infrastructure of a building in relation to its surroundings.

This paper presents a perspective of ontology of design activities in the Flexergy project based on the shared understanding by professionals of what each specific design activity entails. The Flexergy project aims to develop a multi-agent concept for innovative process control of a Flex(ible en)ergy infrastructure for buildings in relation to their environment and their internal processes.

Keywords: Integral design, energy infrastructure, ontology.

1 Introduction

During the last fifteen in the design of buildings attention has enormously increased to comfort in buildings and the consequences for the environment. Key issues of present architecture are preservation of energy resources, occupant comfort and environmental impact limitation. New comfort control technology, such as individual intelligent control, offers new possibilities to further reduce energy consumption of office buildings. Dynamic online steering of individual comfort management and building management could save up to 20% of current energy consumption (Akkermans, 2002). Multi-agent systems (MAS) provide the essential technology for this information infrastructure to connect the end-user to the building systems (Akkermans, 2002):

- large numbers of actors are able to interact, in competition or in cooperation;
- local agents focus on local interests and negotiate with more global agents;
- implementation of distributed decision making by the negotiation processes between the different local or more global oriented agents;
- communication between actors is minimized to generic information exchange between agents.

In two projects, SMART (Smart Multi Agent internet Technology) (Kamphuis et al.2002) and IIGO (Intelligent Internet mediated control in the built environment)

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 263–272, 2009.

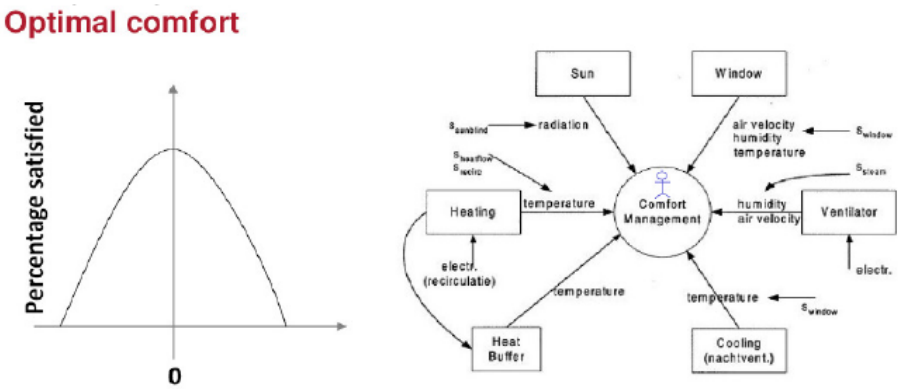


Fig. 1. Different influence factors on the optimal comfort of occupants of a building

(Kamphuis et al. 2005) this technology was developed and tested. To achieve further improved building process control, an multi agent system was developed with agents for: user behaviour, space modelling, weather forecasting and air handlings unit steering, see figure 1.

In this concept local options for influencing indoor comfort are weighed against centrally controlled options. The optimal solution is the solution that brings as much utility as possible for the lowest cost possible. The system also included a forecasting mechanism on every system level with different time scales which play an important role in management of building energy and comfort and these have to be encountered in realizing comfort (Hommelberg 2005), see figure 2.

Although the experimental field tests applied with the multi agent process control systems proved successful and led to a stable it also proved that a more integral approach was needed to further optimize comfort and energy use in a building. All the energy flows such as heat, cold, electricity have to be optimized in connection to each other not only on the level of a specific building but on the level the built environment, this leads to the necessity of agent based comfort optimalizzation in a multi commodity matching market, see figure 2. For such a complex design task the bottom-up approach starting from building segments is not sufficient for building a integral multi-agent process control. A new design approach is needed to structure the different layers and different functional defined tasks to become to a new integral

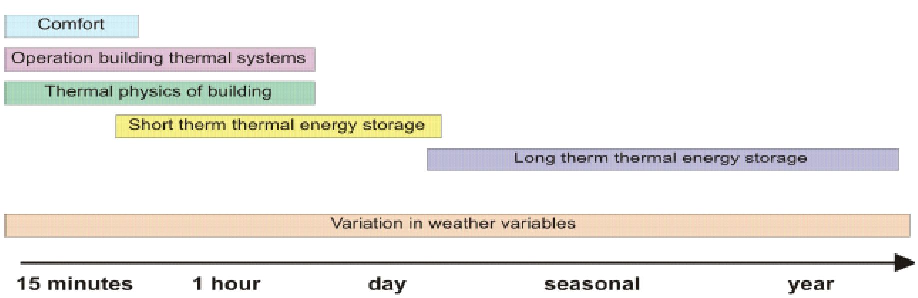


Fig. 2. Different time scale of prediction

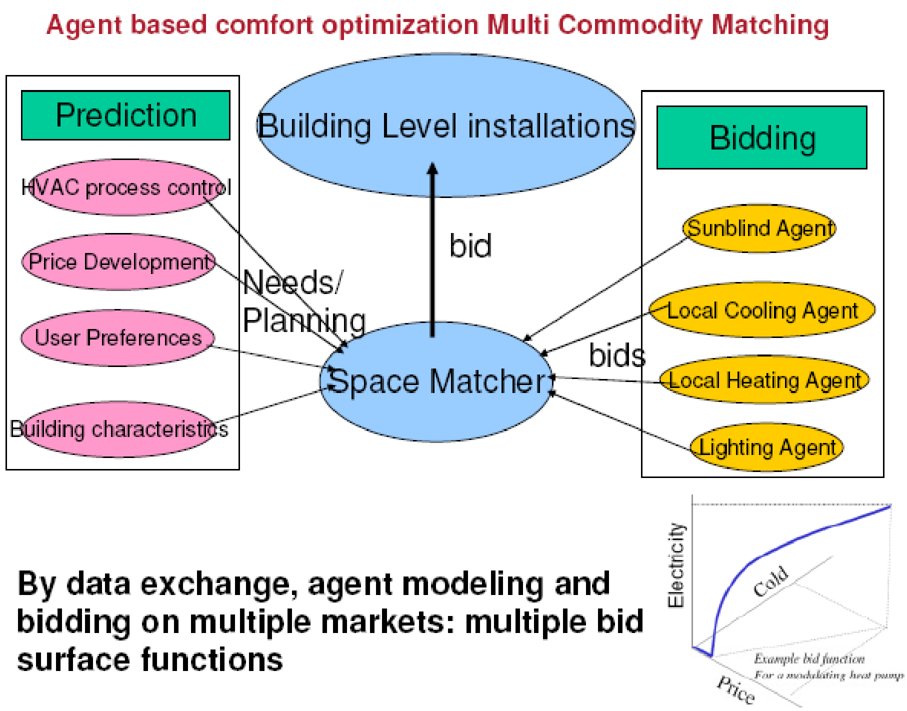


Fig. 3. MAS Multi Commodity matching for comfort optimalization

multi-agent process control system. To achieve this, design knowledge plays an essential role. Gruber et al. (2006) stated that the spectrum of expressiveness and degree of knowledge ranges from simple lists of terms or vocabularies over taxonomies and database schemas up to an ontology (Guarino 1998, Corcho et al. 2003).

Section 2 starts with an introduction to ontology for design followed by a design principle from the mechanical engineering domain. A description of design theory and methodology from the mechanical engineering domain and building services domain as a problem solving method for building MAS system is given. This followed by the presentation of the Integral Design methodology. The knowledge model for design is presented in section 3. Section 4 gives the discussion on our approach. At the moment this Integral Design framework is implemented within the Flexergy project. This project started in 2007 and will continue till 2010.Section 5 gives some discussion about the value of applying insights of ontology research. A short final conclusion is given in section 6.

2 Methodology

The ontology of MAS has to relate to the results of the integral design process. A short introduction of ontology for design (2.1) is followed by the principles of integral design (2.2).

2.1 Ontology for Design

'Ontology' in philosophy means theory of existence in the broadest sense. It tries to explain what is being and how the world is configured by introducing a system of critical categories to account things and their intrinsic relations (Kitamura 2006). In the knowledge engineering community an ontology is viewed as a shared conceptualization of a domain that is commonly agreed to by all parties. It is defined as 'a specification of a conceptualization'(Gruber 1993). "Conceptualization' refers to the understanding of the concepts that can exist or do exist in a specific domain or a community. Ontology is the agreed understanding of the 'being' of knowledge: consensus regarding the interpretation of the concepts and the conceptual understanding of a domain (Dillon et al. 2008). The ontology role is to reflect a community's consensus on a useful way to conceptualize a particular domain (Aparrício et al. 2005). Based on observations from literature, Uschold (1998) identified main categories of uses for ontology's (for further details and examples see Uschold & Gruninger (1996));

- **c**ommunication between people. Here, an unambiguous but informal ontology may be sufficient:
- **i**nter-operability among systems achieved by translating between different modeling methods paradigms, languages and software tools; here, the ontology is used as an interchange format (see Figure 4).

It is necessary to be clear about the purpose of creating a specific ontology because it will determine minimal requirements of the target model (Gruber et al. 2006). Ontology's are formal conceptualizations not made l'art pour l'art, but to help achieve a goal or task by an actor. That task involves knowledge-intensive reasoning to understand the world not just static, but to serve practical purposes of action by the actor in his world (Akkermans 2008). Our world is the design process of energy infrastructure and their process control systems in the built environment.

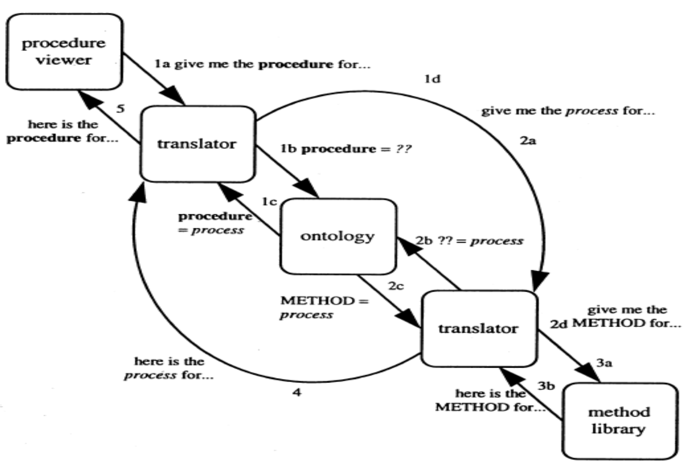


Fig. 4. Interchange format example. This illustrates the use of ontology as an interchange format to integrate different software tools (Uschold 1998).

2.2 Integral Design: A Prescriptive Design Method

Design knowledge sharing is expected to drastically improve the design process of MAS. For example an explicit description of the designer's intentions helps other people to understand the original design more effectively. Even designers themselves can gain deeper insights into the designs themselves (Kitamura et al. 2004). More than two decades of knowledge engineering have shown that there are recurring patterns or stereotypes in the structuring and use of knowledge as an instrument in tasks that involve reasoning and computing. One of these recurring knowledge stereotypes are problem-solving methods; heuristic and stereotypical in the sense that they do not guarantee to solve a given knowledge-intensive problem in general. These problem-solving methods do have demonstrated pragmatic value in solving typical or common cases of knowledge-intensive task that can, moreover, be reused in many different situations (Akkermans 2008). There is a strong analogy between the problem-solving methods and prescriptive design methods.

The design process has been a topic of design research resulting in large numbers of models and theories of design, yet there is no consensus (Sim and Duffy 2003). Based on the Methodical Design by van den Kroonenberg (Blessing 1994) we developed an Integral Design method. The Integral design process can be described at the conceptual level as a chain of activities which starts with an abstract problem and which results through hierarchical abstraction and functional decomposition in a solution. Hierarchical abstraction implies the decomposition of information into levels of increasing detail, where each level is used to define the entities in the level above. The design activity can be divided into four phases: clarification of the task, conceptual design, embodiment design and detail design (Camelo et al. 2007).

3 The Descriptive Method for Integral Design

With the descriptive method for integral design we try to come to an ontology with shares the knowledge for all involved specialists fot our MAS. The method starts with a functional decomposition (3.1), followed by a morphology (3.2). Section 3.3 gives an example for cooling concepts for an office building

3.1 Functional Decomposition

One of the important aspects of the competence of engineering designers is related to their ability in considering functional constraints over the parts of the objects they are designing (Colombo et al. 2007). In order to survey solutions, engineers classify solutions based on various features. This classification provides a mean to decompose complex design tasks into manageable problems. An important decomposition is based on functions. Starting by formulating the need, the program of demands is developed and transformed into functions to fulfil. Functions can be regarded as what a design is supposed to fulfil: the intended behaviour of the object.

Functions have a very significant role in the design process since decomposition is based on functions. Generally, designers think in functions before they are concerned with details. During the design process, and depending on the focus of the designer,

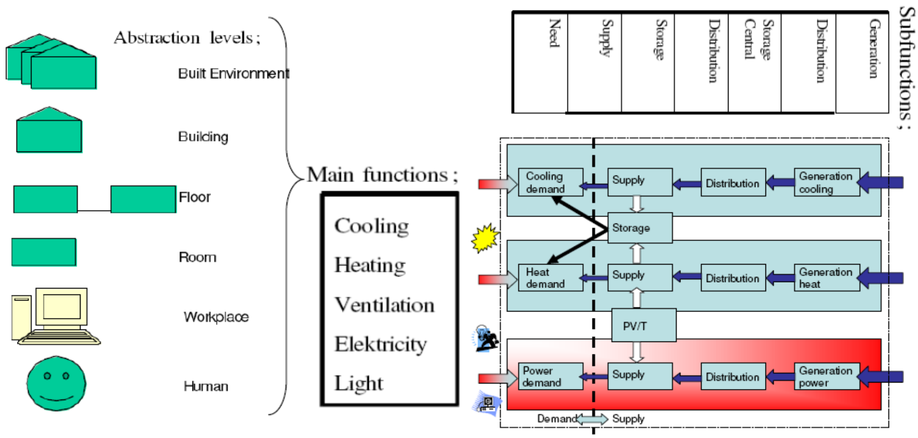


Fig. 5. Abstraction levels, main functions and subfunctions It represents the orderings principle: abstraction levels, main functions and sub functions

functions exist at the different levels of abstraction. The functional decomposition is carried out hierarchically so that the structure is partitioned into sets of functional subsystems, see figure 5.

This functional decomposition provides the means for decomposing complex design tasks into problems of manageable size. This functional decomposition is hierarchically so that the structure is partitioned into sets of functional subsystems. Decomposition is done until simple building components remain whose design is a relatively easy task, but each with each own focus. Functions play a crucial role in a design process, because the results of the design depend on the decomposition of the function (Umeda and Tomiyama 1997).

While there is no common understanding of what a function is, people share the idea that functional knowledge is tightly related to design intention (Kitamura et.al 2004). Functions, as a concept, seems to derive from the designer's intention and it has no clear, unified, objective, and widely accepted definition (Umeda and Tomiyama 1997). Still when designers speak about the 'function' held by an object or by one of its components, they can speak about it because they have sufficient knowledge for associating functions to a suitable object structure (Colombo et al. 2007).

3.2 Morphology

General Morphological analysis was developed by Fritz Zwicky (1948) as a method for investigating the totality of relationships contained in multi-dimensional, usually non-quantifiable problem complexes (Ritchey 1998). The morphological chart gives an overview of necessary functions to fulfill or aspects taken into account and aspect elements or sub-solutions that can be combined together to form a solution.

The Morphological chart is a key element that can improve the effectiveness of the concept generation phase of the design process as it is an excellent way to record information about the solutions for the relevant functions and aspects. The morphological approach aids in the cognitive process of generating the system-level design

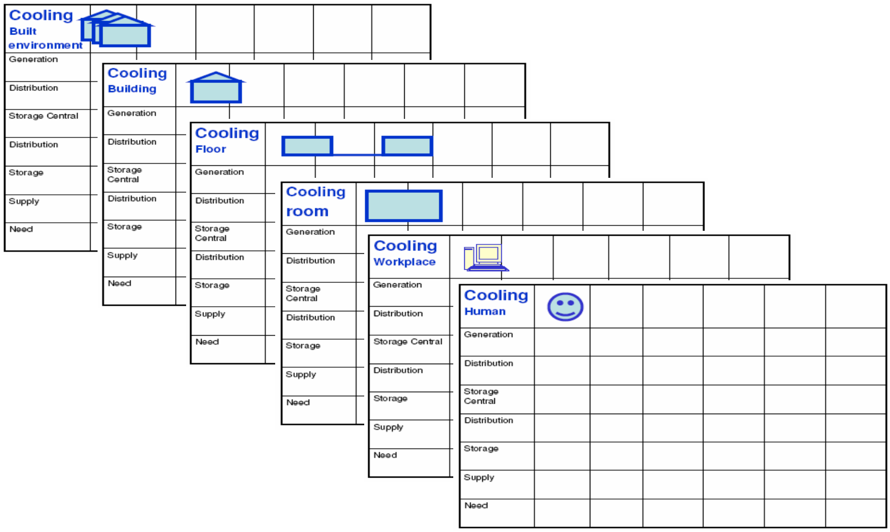


Fig. 6. Combination of functional hierarchical abstraction and morphological charts for the subfunctions of the main function cooling

solutions (Weber and Condoor 1998). It also has definite advantages for communication and for group work (Ritchey 1998).

3.3 Example for Cooling Concepts

Combining the concept of morphological charts with hierarchical functional abstraction levels leads to a structure of different sets of morphological charts for cooling, heating, lighting, power supply and ventilation. In figure 6 an example of the different abstraction level morphological charts is presented. These charts give the solutions for generation, central distribution, central storage, local distribution, local storage and supply to fulfill the need on the specific abstraction level of built environment, building, floor, room, workplace and person.

4 Leading to An Ontology for Integral Design of MAS

By selecting different elements in the morphological charts combination of components form a (sub)system that fulfils the need, in figure 7 such a combination is given. This can be transformed into a PI hydraulic diagram and further to a process control schematic. The necessary agents, to fulfil the complete required functionality, can than be specified more easily. Also the overall structure of all agents is now related the more traditional representations such as PI diagram and process control schematics.

The levels of aspect abstraction in the descriptive model of design can be related to the hierarchical levels used within the functional decomposition and the resulting morphological charts, see figure 7. The morphological charts can be used to generate new possibilities for a flexible energy infrastructure in the built environment.

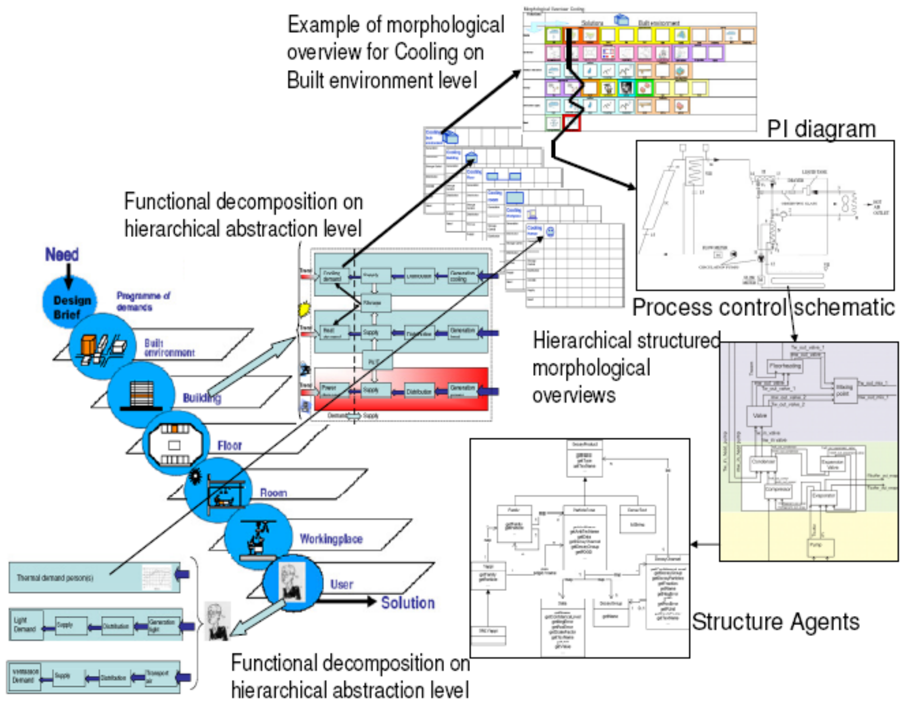


Fig. 7. Set of hierarchical connected functional overviews as basis for the structure of the multi-agent ontology in which the functions act as translator for defining the agents, see for comparison figure 1

The function-oriented strategy allows various design complexity levels to be separately discussed and, subsequently, generated (sub) solutions to be transparently presented. This way interaction with other participants of the MAS design process is aided, and at the same time the design process information exchange is structured. Combining morphological charts for the energy infrastructure in and between buildings makes it possible to integrate in a flexible way the energy flows connected to heating, cooling, ventilation, lighting and power demand, within a building and between buildings and the built environment. This leads to flexibility of energy exchange between different energy demands and sustainable energy supply on the different levels of abstraction in the built environment.

5 Discussion

Our Integral Design ontology can be described at the conceptual level as a chain of activities which starts with an abstract problem and which results in a solution.

In order to survey solutions, engineers classify them according to various features. This classification provides the means for decomposing complex design tasks into problems of manageable size. Decomposition is based on building functions. This functional decomposition is carried out hierarchically so that the structure is partitioned into sets of functional subsystems. Decomposition is carried out until simple building components remain whose design is a relatively easy task. This like the decomposition which is described in the guidelines 2221 and 2222 of the "Association of German Engineers", VDI (Beitz 1985, Pahl et al. 2006).

6 Conclusions

The energy flows of heat, cold and electricity have to be optimized together. This is highly complex and should be done by MAS based on predictions taking into account all changing dynamic parameters. To cope with the complexity of developing such a complex MAS control strategy a design method is used to develop an ontology to structure the process. Using two types of knowledge levels models (Uschold 1998), ontology and problem solving design process model, led to an approach in which the characteristics of the combined models offer an added value to design MAS within the built environment process control domain.

A new integral design method is used to develop a flexible concept for further development and implementation of the new design and control strategies for flexible energy infrastructures for the built environment: **Flex**(ible)(en)**ergy**.

Acknowledgments

Technische Universiteit Eindhoven, Kropman Building Services, Installect consultancy and ECN (Energy research Centre Netherlands) are partners in the Flexergy project, which was partly financial supported by SenterNovem.

References

Akkermans, H.: Being Smart In Information Processing. Technological And Social Challenges And Oppertunities. In: Proceedings IFIP IIP 2002 (2002)

Akkermans, H.: The Business of Ontology calls for a Formal Pragmtics. In: Invited talk 3rd workshop on Formal Ontologies Meet Industry, FOMI 2008, Torino, Italy, June 5-6 (2008), http://e3value.few.vu.nl/docs/bibtex/pdf/HansA2008.pdf

Aparício, A.S., Farias, O.L., dos Santos, N.: Applying Ontologies in the Integration of Heterogeneous Relational Databases. In: Proceedings Australasian Ontology Workshop (AOW 2005), Sydney. Conferences in Research and Practice in Information Technology (CRPIT), vol. 58 (2005)

Beitz, W.: Systematic Approach to the Design of technical systems and products, VDI 2221 0 Entwürf, VDI, Düsseldorf (1985)

Blessing, L.T.M.: A process-based approach to computer supported engineering design, PhD-thesis Universiteit Twente (1994)

Camelo, D., Mulet, E., Vidal, R.: Function and behaviour representation for supporting flexible exploration and generalisation in a functional model for conceptual design. In: Proceedings International Conference on Engineering Design, ICED 2007, Paris, France, August 28-31 (2007)

Colombo, G., Mosca, A., Sartori, F.: Towards the design of intelligent CAD systems: An ontological approach. Advanced Engineering Informatics 21, 153–168 (2007)

Corcho, O., Fernández-López, M., And Gómez-Pérez, A.: Methodologies, tools and languages for building ontologies. Where is their meeting point? Data & Knowledge Engineering 46(1), 41–64 (2003)

Dillon, T., Chang, E., Hadzic, M., Wongthongham: Differentiating Conceptual Modelling from Data Modelling, Knowledge Modelling and Ontology Modelling and a Notation for Ontology Modelling. In: Proceedings Fifth Asia-Pacific Conference on Conceptual Modelling (APCCM 2008), Wollongong, NSW, Australia (January 2008)

Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition 5(2), 199–220 (1993)

Gruber, A., Westenthaler, R., Gahleitner, E.: Supporting domain experts in creating formal knowledge models (ontologies). In: Proceedings of I-KNOW 2006, 6th International Conference on Knowledge Management, Graz, Austria, pp. 252–260 (2006)

Hommelberg, M.P.F.: Careful Buildings, software agents voor een gebouwbeheersysteem, Master thesis Building Services TU/e (dutch) (January 2005)

Kamphuis, I.G., Warmer, C.J., Zeiler, W., Wortel, W., Akkermans, J.M., Jelsma, J.: SMART: Experiences with e-services for Smart Buildings. In: ISPLC Conference, Athens, March 27-29 (2002)

Kamphuis, I.G., Warmer, C.J., Jong, M.J.M., Wortel, W.: IIGO: Intelligent Internet mediated control in the built environment: Description of a large-scale experiment in a utility building setting, ECN rapport ECN-C—05-084 (October 2005)

Kitamura, Y., Kashiwase, M., Fuse, M., Mizoguchi, R.: Deployment of an ontological framework of functional design knowledge. Advanced Engineering Informatics 18, 115–127 (2004)

Kitamura, Y.: Roles of ontologies of engineering artifacts for design knowledge modeling. In: Proc. Of the 5th International Seminar and Workshop Engineering Design in Integrated Product Development (EDIProD 2006), Gronów, Poland, September 21-23, pp. 59–69 (2006)

Pahl, G., Beitz, W., Felhusen, J., Grote, K.H.: Translators and Editors Wallace, K., Blessing, L., Engineering Design, A systematic Approach, 3rd edn. Springer, Heidelberg (2006)

Ritchey, T.: General Morphological Analysis, A general method for non-quantified modeling. In: 16th EURO Conference on Operational Analysis, Brussels (1998)

Sim, S.K., Duffy, A.H.B.: Towards an ontology of generic engineering design activities. Res. Eng. Design 14, 200–223 (2003)

Umeda, Y., Tomiyama, T.: Functional Reasoning in Design. IEEE Expert, 42–48 (March-April 1997)

Uschold, M.: Knowledge level modelling: concepts and terminology. The Knowledge Engineering Review 13(1), 5–29 (1998)

Uschold, M., Gruninger, M.: Ontologies: Principles, methods and applications. The Knowledge Engineering Review 11(2) (1996) (also available as AIAI-TR-191 from AIAI, The University of Edinburgh)

Weber, R.G., Condoor, S.S.: Conceptual design using a synergistically compatible morphologicalmatrix. In: Proceedings Frontiers in Education Conference, FIE 1998, 28th Annual, November 4-7, vol. 1, pp. 171–176 (1998)

Zwicky, F.: Morphological Astronomy. The observatory 68(845), 121–143 (1948)

Relational Learning by Imitation

Grazia Bombini, Nicola Di Mauro, Teresa M.A. Basile, Stefano Ferilli, and Floriana Esposito

Università degli Studi di Bari, Dipartimento di Informatica, 70125 Bari, Italy
{gbombini,ndm,basile,ferilli,esposito}@di.uniba.it

Abstract. Imitative learning can be considered an essential task of humans development. People use instructions and demonstrations provided by other human experts to acquire knowledge. In order to make an agent capable of learning through demonstrations, we propose a relational framework for learning by imitation. Demonstrations and domain specific knowledge are compactly represented by a logical language able to express complex relational processes. The agent interacts in a stochastic environment and incrementally receives demonstrations. It actively interacts with the human by deciding the next action to execute and requesting demonstration from the expert based on the current learned policy. The framework has been implemented and validated with experiments in simulated agent domains.

Keywords: Relational Learning, Learning by Imitation, Agents.

1 Introduction

Learning from demonstration or *learning by imitation* [1,2], represents a promising approach in the area of human-robot interaction. Recent research in other fields define the imitative learning as an essential part of human development [3]. Indeed, humans and animals use imitation as a mechanism for acquiring knowledge. Imitation-based learning [4] and learning by demonstration [5] are exploited to enable an unskilled agent, the *observer*, to learn tasks by simply observing performances of a skilled agent, the *teacher* [6,7]. This approach can be viewed as a collaborative learning based on the interaction between the human and the agent. The agent gathers information about the task in the form of perceptual inputs and action results and estimates the latent control policy of the demonstrator. The estimated policy can then be used to control the agent's autonomous behaviour. As reported in [8], this method can reduce the learning time when compared to classical exploration-based methods such as reinforcement learning (RL) [7].

The goal of this paper is to provide a relational framework allowing an agent to learn and revise in an incremental way a logical representation of a task by imitating a skilled agent. In particular, we propose an incremental policy learning approach based on a relational language used to describe both the demonstration examples and the learnt policy. The agent actively interacts with the teacher by

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 273–282, 2009.

deciding the next action to execute and requesting demonstration. Furthermore, due to the relational description language, it is very simple to use background knowledge about the domain.

Learning from demonstration is strongly related to supervised learning where the goal is to learn a policy given a fixed set of labelled data [9]. From this point of view it is possible to collect the interactions between the teacher and the observer and using them as learning examples. Furthermore, data should be gathered in an incremental way by minimising the number of required labelled data useful to learn the given policy.

When we adopt a Reinforcement Learning (RL) approach, an agent in a certain state receives a rewards (positive or negative) regarding the goodness of the taken action. The agent's goal is to learn a function (*policy*) that maps the state of the world to the action that maximise the overall expected reward. A stochastic environment may require a lot of states needing the agent to take a large number of actions before to learn a good policy. In [10] has been presented a RL framework in which an agent learns observing the actions of other agents (*Implicit Imitation*) without requiring no explicit teacher. This increases the speed of policy learning, but requiring the knowledge or the ability to infer explicit state rewards, that are not always known or easy to infer. In [11] has been investigated how RL can profit from data acquired by demonstrations in the task of pole balancing.

In [12] the authors presented a model to find the intended goal of a demonstration using iterative interactions. They infer the goal of a demonstration without imitating the steps on how to reach the goal by using some psychological observations reported in [13]. The same psychological observations are taken into account in [14] where agents learn new goals and how to achieve the goals. In [15] the agent's behaviour is predicted using a probabilistic distribution of the environment. In [16] has been proposed a framework for imitative learning that uses a probabilistic graphical model to describes a imitative process. In [8] has been used a demonstration based learning algorithm (*confident execution framework*) allowing an agent to learn a policy from demonstrations on how to execute actions. A supervised learning approach is used to learn a policy represented by a Gaussian model. All these approaches still do not use a relational representation formalism able to generalise the learned policies.

2 Logical Background

Here we briefly review the used representation language for the domain and induced knowledge. For a more comprehensive introduction to logic programming and Inductive Logic Programming (ILP) we refer the reader to [17].

A first-order *alphabet* consists of a set of *constants*, a set of *variables*, a set of *function symbols*, and a non-empty set of *predicate symbols*. Each function symbol and each predicate symbol has a natural number (its *arity*) assigned to it. The arity assigned to a function symbol represents the number of arguments the function has. Constants may be viewed as function symbols of arity 0. A *term* is a constant symbol, a variable symbols, or an n-ary function symbol f applied to n terms $t_1, t_2, \ldots, t_n$.

An atom $p(t_1, \ldots, t_n)$ (or atomic formula) is a predicate symbol p of arity n applied to n terms t_i. Both l and its negation $\overline{l}$ are said to be *literals* (resp. positive and negative literal) whenever l is an atomic formula. A *clause* is a formula of the form $\forall X_1 \forall X_2 \ldots \forall X_n (L_1 \vee L_2 \vee \ldots \vee \overline{L}_i \vee \overline{L}_{i+1} \vee \ldots \vee \overline{L}_m)$ where each L_i is a literal and $X_1, X_2, \ldots X_n$ are all the variables occurring in $L_1 \vee L_2 \vee \ldots \overline{L}_i \vee \ldots \overline{L}_m$. Most commonly the same clause is written as an implication $L_1, L_2, \ldots L_{i-1} \leftarrow L_i, L_{i+1}, \ldots L_m$, where $L_1, L_2, \ldots L_{i-1}$ is the *head* of the clause and $L_i, L_{i+1}, \ldots L_m$ is the *body* of the clause. Clauses, literals and terms are said to be *ground* whenever they do not contain variables. A *Horn clause* is a clause which contains at most one positive literal. A *Datalog clause* is a clause with no function symbols of non-zero arity; only variables and constants can be used as predicate arguments.

A *substitution* θ is defined as a set of bindings $\{X_1 \leftarrow a_1, \ldots, X_n \leftarrow a_n\}$ where $X_i, 1 \leq i \leq n$ is a variable and $a_i, 1 \leq i \leq n$ is a term. A substitution θ is applicable to an expression e, obtaining the expression $e\theta$, by replacing all variables X_i with their corresponding terms a_i.

The learning problem for ILP can be formally defined:

Given: A finite set of clauses $\mathcal{B}$ (*background knowledge*) and sets of clauses E^+ and E^- (positive and negative *examples*).

Find: A theory Σ (a finite set of clauses), such that $\Sigma \cup \mathcal{B}$ is *correct* with respect to E^+ and E^-, i.e.: a) $\Sigma \cup \mathcal{B}$ is *complete* with respect to E^+: $\Sigma \cup \mathcal{B} \models E^+$; and, b) $\Sigma \cup \mathcal{B}$ is *consistent* with respect to E^-: $\Sigma \cup \mathcal{B} \not\models E^-$.

Given the formula $\Sigma \cup \mathcal{B} \models E^+$, deriving E^+ from $\Sigma \cup \mathcal{B}$ is *deduction*, and deriving Σ from $\mathcal{B}$ and E^+ is *induction.* In the simplest model, $\mathcal{B}$ is supposed to be empty and the deductive inference rule $\models$ corresponds to *θ-subsumption* between clauses. In particular, a clause c_1 *θ-subsumes* a clause c_2 if and only if there exists a substitution σ such that $c_1\sigma \subseteq c_2$. c_1 is a *generalization* of c_2 (and c_2 a *specialization* of c_1) under θ-subsumption. If c_1 θ-subsumes c_2 then $c_1 \models c_2$.

3 Learning from Demonstration

Here, we assume that the environment is defined by a finite set of states S. For each state $s \in S$, the agent has available a finite set of actions $A(s) \subseteq \mathcal{A}$ which cause stochastic state transition. In particular, an action $a \in A(s)$ causes a transition to the state s'_a when executed in the state s, where $\mathcal{A}$ is the set of all the primitive actions.

The agent is assumed to observe a demonstrator that performs the correct sequence of actions useful to reach a given goal by starting from an initial state of the environment. During each training sequence, the agent records the observation about the environment and the corresponding action performed by the demonstrator. An observation $o \in S$ is represented by a set of ground Datalog literals.

Example 1. The following set of literals represents an observation of the pursuit domain consisting of two agent, prey and predator, moving in a 4x4 grid. The literal *observation(A, D, O)*, denotes that an agent *A* percepts the observation *O* in direction *D*.

{*observation(a, d1, e), observation(a, d2, p), observation(a, d3, e), observation(a, d4, e), direction(d1, named1), direction(d2, named2), direction(d3, named3), direction(d4, named4), north(named1), south(named2), east(named3), west(named4), under(a, e), prey(p), predator(a), wall(w), empty(e)* } .

In this domain, the available actions that can be performed by the demonstrator are *move(X,Y)* , with $X \in \{p, a\}$ and $Y \in \{d1, d2, d3, d4\}$ (directions that an agent can take corresponding to the cardinal directions).

Each training example, $e = \{a, o\}$, consists of an action $a \in \mathcal{A}$ selected by the demonstrator and an observation $o \in S$. Obviously, we assume that the demonstrator uses a good policy π to achieve the goal. Hence, the aim of the agent is to learn this hidden policy $\pi : S \rightarrow \mathcal{A}$ mapping states (observations) to actions.

Classical supervised learning is based on an inductive learning method able to generalize from positive and negative examples labeled by the user. In case of imitative learning, the action taken by the teacher agent may be considered as positive example and all the other possible actions as negative examples. In particular, given a state s, if the teacher takes the action $a_i \in A(s)$, then the observer can assume that a_i is a positive example and all the other actions $a_j \in A(s)$ $1 \leq j \neq i \leq |A|$ as negative ones. A negative example a_j is considerate reliable until the demonstrator performs a_j in a state s. Furthermore, the process of imitative learning is naturally modeled by an agent able to modify its theory in an incremental way, where each new incoming example may give rise a theory revision process. We represent the policy as a set of logical clauses where the *head* of the clause represent the action while the *body* represents the state. In particular, a clause represents an action that may be performed in a given state.

Example 2. In the pursuit domain a learned rule may be the following:
{*move(A, B) :- predator(A), direction(B, D), north(D), observation(A, B, E), wall(E), observation(A, F, G), direction(F, H), south(H), prey(G), observation(A, I, J), direction(I, K), east(K), empty(J), under(A, J)*}

In this example, moving towards direction north is a good choice for the predator. Indeed, there are more chances to cough the prey remaining in the same square.

4 Relational Incremental Learning

INTHELEX (INcremental THeory Learner from EXample) [18] is an incremental learning system for the induction of first-order logic theory from positive and negative examples. The approach used by this system may be defined a multistrategy one since the system realizes the combination of different forms of reasoning in the same symbolic paradigm. It learns theory in form of sets of

Datalog clauses, interpreted according the Object Identity (OI for short) [19] assumption. Under the OI assumption, within a clause, terms (even variables) denoted with different symbols must be distinct.

INTHELEX is a learning system fully and inherently incremental. The learning phase can start by taking in input a previously generated version of the theory or from an empty theory and from the first available example. When the theory is not correct compared with an example, this is rejected and a process of theory revision starts. It can learn simultaneously several concepts, possibly related to each other. It uses a full memory storage strategy, and therefore retains all the available example.

INTHELEX use two inductive refinement operator, one for generalising definitions that reject positive examples (completeness), and the other for specialising definitions that explain negative examples (consistency). In order to perform its task, the system exploits a previous theory (if any) and a *historical memory* of all the past (positive and negative) examples that led to the current theory.

Algorithm 1 reports the tuning procedure used in INTHELEX for refining the theory to be learned. M represents the set of all positive and negative examples already processed, E is the current example to be examined, and T is the theory learned from the examples in M. In particular, a set of examples of the concepts to be learned is incrementally provided by an expert. Whenever a new example is taken into account, it is also stored in the historical memory. The training examples are exploited by the system to modify incorrect hypotheses according to a data-driven strategy. In particular, when a positive example is not covered by the theory, a generalization process produces a revised theory obtained in one of the following way:

1. generalize one of the definitions pertaining to the theory that relate to the concept on the example, by removing some conditions. That ensures the revised theory covers the new example and is consistent with the all negative examples previous examined;
2. add a new definition to the theory to explain the example;
3. add the example as a positive exception.

When a negative example is covered, a specialization process outputs a revised theory by performing one of the following actions:

1. add some conditions to one of the definitions that explains the example. These conditions must be able to characterize all positive example already considered and exclude the current negative example;
2. add a negative condition to one of the definitions to differentiate the current negative example by positive examples previously considered;
3. add the negative example as an negative exception.

The system can be provided with a background knowledge (i.e. some partial concept definitions know to be correct about the domain and hence not modifiable in the same format of a theory rule).

In the scenario of an agent acting in a stochastic world, the *correct actions* of the agent may be considered as positive examples in a supervised learning task.

Algorithm 1. tuning(E,T,M)

Input: E: example; T: theory; M: historical memory;

Add E to M
if E is a positive example not covered from T **then**
    generalize(T, E, M)
else
    if E is a negative example covered by T **then**
        specialize(T, E, M)

In our framework we assume that an agent A aims to learn how to act in a world by imitating an *expert agent* E. Hence, each action taken from E in a given state may be considered as a positive example. All the other possible actions in the same state should be considered as negative examples. In this way A assumes that E acts in a correct way. Given an action-state pair (a, s) and a set $A(s)$ of possible actions that can be taken in state s, all other actions in $A(s) \setminus a$ is assumed to be negative examples until the expert agent does not take any of them in the same state s. In this case it is necessary to retract the previous hypothesized negative example.

Given a goal, each training example $e = \{a, o\}$ belongs to a specific learning class based on the action a. For each class a theory must be learned in order to be able to predict the corresponding action of unseen observations. As reported in algorithm 2, the procedure starts with an empty theory and an empty historical memory. The agent observes the sequence of actions performed by the demonstrator. For each time-step the agent tries to classify the observation

Algorithm 2. IIL

errors $\leftarrow 0$
$T \leftarrow \emptyset$
$M \leftarrow \emptyset$
loop
    $(o, a) \leftarrow$ get an observation-action pair
    $N \leftarrow \{(o, \neg a_i) | a_i \in A(o) \setminus \{a\}\}$
    $c \leftarrow$ classify(o,T)
    if $c \neq a$ **then**
        errors++
        for all $e \in M$ **do**
            if e is negative for the class a with the same body as o **then**
                $M \leftarrow M \setminus \{e\}$
        generalize(T, $a \leftarrow o$, M)
    $M \leftarrow M \cup \{a \leftarrow o\}$
    for all $a_i \in N$ **do**
        if M does not contain the positive example $a_i \leftarrow o$ **then**
            if $c = a_i$ **then**
                errors++
                specialize(T, $\neg a_i \leftarrow o$, M)
            $M \leftarrow M \cup \{a \leftarrow o\}$

(i.e., to predict the corresponding action) by using its learned model. All the actions that are allowed in the domain, but are not performed by the demonstrator are supposed to be negative examples for the class a (the correct action) in the state s. This set N is generated by taking into account all the possible actions that the agent can perform in s. The classification task returns an action c that is compared to the correct action a the demonstrator performs. When the action c does not correspond to the action a a theory revision is needed. All the negative examples for the class a with the same body of o are excluded from historical memory. A generalization process on the current theory with the example e and filtered historical memory starts. At the end of the learning process, the learned theory T represents the optimal policy.

5 Experimental Results

The proposed learning framework has been applied to a pursuit domain, generally used in the field of agent learning. The experiment regards the problem of learning a policy in a domain where a predator should capture a prey. This stochastic environment consists of a 4x4 grid surrounded by a wall, with a predator and a prey. We assume that the predator caught the prey if the prey lands on the same square as the predator at the end of its move. The prey moves with random actions, while the predator follows a *good user-defined* strategy in order to capture the prey. Both predator and prey can move in four directions, specifically north, east, south and west. The action of an agent involves moving in the square immediately adjacent corresponding to the selected direction. In case of the target square is a wall, then the agent remains in the same square. The two agents move alternate turns.

An observation is made up of the agent's perception about the squares surrounding it in the four directions and the square under it. The state of each square may be *empty*, *wall* or *agent*. Starting from an initial state, once captured the prey the sequence of observations does not restart by placing agents in random positions, but it continues from the positions of catch. For example, a predator agent having a wall to the west and the prey to the east, the observation is represented by the following set of literals: { *observation(a, d1, e), observation(a, d2, e), observation(a, d3, p), observation(a, d4, w), under(a, e), direction(d1, named1), direction(d2, named2), direction(d3, named3), direction(d4, named4), north(named1), south(named2), east(named3), west(named4),* } where a stay for predator agent, p for prey agent, e for empty, and w for wall.

Fixed the strategy to capture the prey, we simulated a scenario in which a predator instructs another agent to capture the prey with a minimum number of steps. Hence, given an observation, the action taken from the predator represents a positive training examples, while all the other possible actions are supposed to be negative examples.

We have generated 10 sequences of observations. Each sequence, containing the traces of prey's captures, is made up of 322,5 positive and 967,5 negative

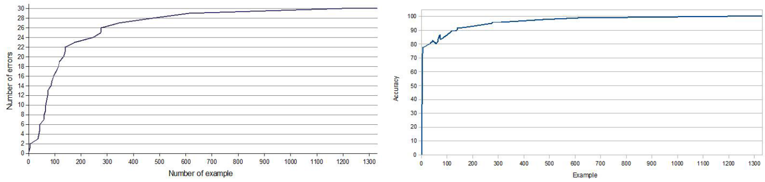


Fig. 1. Errors during the learning phase. The figure on the right represents the prediction accuracy (%) over the entire historical memory for each theory revision.

observation-action pairs on average. Starting from a positive instance each alternative action has been hypothesised to be a negative instance.

A first experiment has been performed on the whole dataset without providing the system with any background knowledge. Over all the 10 sequences the system learned a theory (policy) made up of 9,4 clauses, obtained by 14,8 generalisation and 2,2 specialisation (26,4 errors) on average.

In order to evaluate the behaviour of the learning process, we generated a sequence made up of 1332 observation-action pairs. The Figure 1 reports on the left, the number of errors (a generalisation or a specialisation request) the agent made during the learning phase. In particular, as we can see the number of errors grows until the system learns the correct policy (i.e., the learned classification theory). The Figure 1 on the right, reports the prediction accuracy of the policy, learnt by the agent during the imitation process, on the complete historical memory.

An example of learned rule is { *move(A, B) :- predator(A), direction(B, D), direction(F, H), observation(A, I, J), observation(A, B, E), observation(A, F, G), south(H), north(D), wall(E), prey(G), empty(J), under(A, J).* }

where a predator A moves toward north when at south there is the prey and at north there is a wall.

A new experiment has been performed with the system provided with a background knowledge containing several rules defining the concept of next direction. An example is { *next(A,B):- north(A), east(B).*}. This rule denotes that to north direction follows the east direction. The learned theory abstracts the concept of direction. An example of the learned rule is

{*move(A, B) :- predator(A), direction(B, D), direction(G, F), next(D, H), next(H, F), next(F, E), next(E, D), observation(A, B, I), observation(A, G, K), wall(I), under(A, J), empty(J), prey(K).* }

where the predator A moves in the direction D when it observes that in D there is a wall and opposite there is the prey K at any edge of the grid (D may be one of the for direction). This action applies to four different states.

Over all the same 10 sequences of the previous experiments, the system learned a theory made up of 6,4 clauses, obtained by 12,1 generalization and 3,5 specialization (22 errors) on average. In order to evaluate the behavior of the learning process with the use of a background knowledge, we generated a sequence made up of 1276 observation-action pairs.

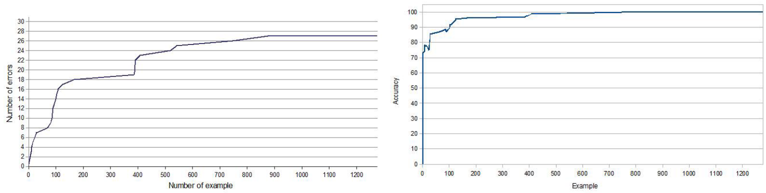


Fig. 2. Errors during the learning phase. The figure on the right represents the prediction accuracy (%) over the entire historical memory at each revision theory with the use background knowledge.

The Figure 2 reports the number of errors (a generalization or a specialization request) the agent made during the learning phase and prediction accuracy of the policy.

It is interesting to note that the theory learned with the use of background knowledge does not represent the actual directions, but the relationship between them. This allows an abstraction on states. Indeed, the policy learned results more compact (a lower number of clauses on average compared with the first experiment) and each clause is applicable to a subset of states.

6 Conclusion

Recently have been studied forms of social learning inspired by the ways people learn and their applications in robotics. In the field of agent and multi-agent systems most of the approaches used to learn does not allow the use of a high-level language to describe the domain and knowledge about it. In this paper we have presented a relational framework that allows to quickly, accurately and incrementally train an unskilled agent to imitate a demonstrator (human or artificial). The proposed technique has been implemented and tested on a dataset belonging to a pursuit domain generally used in the field of agent learning, and the results confirm the validity of the technique. As a future work, we are planning to extend the framework to a domain in which an agent can learn different skills by different teachers autonomously choosing skills and demonstrations.

References

1. Billard, A., Siegwart, R.: Robot learning from demonstration. Robotics and Autonomous Systems 47(2-3), 65–67 (2004)
2. Schaal, S., Ijspeert, A., Billard, A.: Computational approaches to motor learning by imitation. Philosophical Transactions: Biological Sciences 358(1431), 537–547 (2003)
3. Meltzoff, A.: The "like me" framework for recognizing and becoming an intentional agent. Acta Psychologica 124(1), 26–43 (2007)
4. Schaal, S.: Is imitation learning the route to humanoid robots? Trends in cognitive sciences 3(6), 233–242 (1999)

5. Nicolescu, M., Mataric, M.: Natural methods for robot task learning: instructive demonstrations, generalization and practice. In: Proceedings of the second international joint conference on Autonomous agents and multiagent systems (AAMAS 2003), pp. 241–248. ACM, New York (2003)
6. Atkeson, C., Schaal, S.: Robot learning from demonstration. In: Fisher, D. (ed.) Proceedings of the 14th International Conference on Machine Learning (ICML), pp. 12–20 (1997)
7. Smart, W., Kaelbling, L.: Effective reinforcement learning for mobile robots. In: IEEE International Conference on Robotics and Automation (ICRA), vol. 4, pp. 3404–3410 (2002)
8. Chernova, S., Veloso, M.: Confidence-based policy learning from demonstration using gaussian mixture models. In: AAMAS 2007: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1–8. ACM, New York (2007)
9. Bentivegna, D., Atkeson, C., Cheng, G.: Learning from observation and practice using primitives. In: AAAI Fall Symposium Series, Symposium on Real-life Reinforcement Learning (2004)
10. Price, B., Boutilier, C.: Accelerating reinforcement learning through implicit imitation. Journal of Artificial Intelligence Research 19 (2003)
11. Schaal, S.: Learning from demonstration. Advances in Neural Information Processing Systems 9, 1040–1046 (1997)
12. Jansen, B., Belpaeme, T.: A computational model of intention reading in imitation. Robotics and Autonomous Systems 54(5), 394–402 (2006)
13. Wohlschlager, A., Gattis, M., Bekkering, H.: Action generation and action perception in imitation: An instantiation of the ideomotor principle. Philosophical Transaction of the Royal Society of London: Biological Sciences 358 1431, 501–515 (2003)
14. Billard, A., Epars, Y., Calinon, S., Cheng, G., Schaal, S.: Discovering Optimal Imitation Strategies. robotics and autonomous systems. Special Issue: Robot Learning from Demonstration 47(2-3), 69–77 (2004)
15. Jebara, T., Pentland, A.: Statistical imitative learning from perceptual data. In: Proc. ICDL 2002, pp. 191–196 (2002)
16. Verma, D., Rao, R.: Imitation learning using graphical models. In: Kok, J.N., Koronacki, J., de Mántaras, R.L., Matwin, S., Mladenic, D., Skowron, A. (eds.) ECML 2007. LNCS, vol. 4701, pp. 757–764. Springer, Heidelberg (2007)
17. Lavrac, N., Dzeroski, S.: Inductive Logic Programming: Techniques and Applications. Ellis Horwood, New York (1994)
18. Esposito, F., Ferilli, S., Fanizzi, N., Basile, T., Di Mauro, N.: Incremental learning and concept drift in inthelex. Intelligent Data Analysis Journal, Special Issue on Incremental Learning Systems Capable of Dealing with Concept Drift 8(3), 213–237 (2004)
19. Semeraro, G., Esposito, F., Malerba, D.: Ideal refinement of datalog programs. In: Proietti, M. (ed.) LOPSTR 1995. LNCS, vol. 1048, pp. 120–136. Springer, Heidelberg (1996)

A Data-Oriented Coordination Language for Distributed Transportation Applications

Mahdi Zargayouna[1], Flavien Balbo[1,2], and Gérard Scémama[1]

[1] INRETS Institute, Gretia laboratory
Building "Descartes II", 2 rue de la Butte Verte
93166 Noisy le Grand Cedex, France
[2] University of Paris Dauphine, CNRS-Lamsade laboratory
Place Marechal de Lattre de Tassigny,
75775 Paris Cedex 16, France
{zargayouna,scemama}@inrets.fr, balbo@lamsade.dauphine.fr

Abstract. Data-oriented coordination languages rely on a shared space in which agents add, read and retrieve data. They are intuitively well suited for distributed transportation applications, where different actors evolve in a highly dynamic and very constrained environment. However, existing coordination languages can hardly be used for transportation applications, because they cannot express agents complex interaction needs. Indeed, in transportation applications, the interaction needs of the agents are driven by their current context and by *ambient* conditions, expressed in the form of constraints on the values taken by variables. In this paper, we propose LACIOS, a new data-oriented coordination language for designing and implementing distributed transportation applications, and we illustrate our proposal with two examples: a traveler information system and a demand-responsive transport system. LACIOS allows agents to express complex interaction needs, including agents states, system objects values, operators and functions, and is grounded on a model where the multi-agent system design is centered on the environment.

1 Introduction

The transportation domain is a privileged field for the multi-agent community. Indeed, it exhibits characteristics that makes it relevant to model transportation applications in the form of many physically and/or logically distributed interacting components that possess some level of autonomy. In [1], we have identified three recurrent issues when designing and implementing transportation applications: the knowledge processing, the space-time dimension of the problems and the dynamics of the real environment, and we have argued that the design of an environment-centered multi-agent system (MAS) is a solution to these issues. The multi-agent environment contains the recorded descriptions and supports their processing. In this paper, we propose a language to specify environment-centered multi-agent systems and we describe two applications designed and implemented with this language.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 283–292, 2009.

Our language is a data-oriented language, which were initiated by Linda [2], and has known several extensions (e.g. Klaim [3], Mars [4] and Lime [5]). Linda-like models are based on the notion of a shared data repository. Agents communicate by exchanging tuples via an abstraction of an associative shared memory called the *tuplespace*. A tuplespace is a multiset of tuples (tuples duplication is allowed) and is accessed associatively (by contents) rather than by address. Every tuple is a sequence of one or more typed values. Communication in Linda is said to be *generative*: an agent *generates* a tuple and its life cycle is independent of the agent that created it. The tuplespace is manipulated by three atomic primitives: *out* to add a tuple, *rd* to read and *in* to take a tuple (read it and remove it from the tuplespace). The parameter of *out* is a fully instantiated tuple (sequence of values), and the parameter of an *in* or a *rd* primitive is a template: a tuple with potentially one or more formal fields (variables). A tuple and a template match if they have the same arity and if every field in the tuple is either equal to the corresponding value or of the same type of the corresponding variable in the template. The primitives *in* and *rd* are blocking: if no tuple matches their parameter template, the caller agent is suspended until a matching tuple is present in the tuplespace. An additional primitive, $eval(P)$, launches a new agent that will run in parallel with the caller.

LACIOS is a data-oriented coordination language designed to overcome several limitations in existing languages of the literature, and designed to facilitate their use in transportation applications. In this paper, we focus on two limitations: the poor expressive power of templates and the inaccessibility of agent states. In the literature, agents are black boxes: they don't have an observable state described by data. Indeed, data-oriented models describe what the agents *do*, not what they *are*. In the absence of agents' states, agents cannot condition their interaction with their current context, especially in the absence of a rich interaction mechanism.

This paper is organized as follows. Section 2 describes the LACIOS language that we propose and details its syntax. In section 3, we present the traveler information system based on LACIOS. Section 4 presents our proposal of a coordination environment for a demand-responsive transport service. Finally, section 5 concludes the paper.

2 The Coordination Language LACIOS

2.1 Overview

LACIOS is a coordination language that extends Linda for the design and implementation of MAS, defined in a suitable way for transportation applications. For the specification of agent behavior, we adopt four primitives inspired by Linda and a set of operators borrowed from Milner's CCS [6]. A MAS written in LACIOS is defined by a dynamic set of *agents* interacting with an *environment* - denoted Ω_{ENV}, which is composed of a dynamic set of *objects*. Agents can *perceive* (read only) and/or *receive* (read and take) objects from the environment.

Agents are defined by a behavior (a process), a state and a local memory in which they store the data they perceive or receive from the environment.

The distinguishing features of Lacios that we focus on in this paper can be summarized as follows. First, an agent can publish its state, update it and use it to condition its interaction with the environment. Second, the data structure (for exchanged data and for agents' states) is based on typed property-value pairs. Finally, an agent can use complex conditions (using operators and functions) on its own state and on other shared objects in order to access the environment, with a single instruction.

We have proposed an operational semantics, unambiguously defining the behavior of a MAS written in Lacios, which can be found in [7].

2.2 Syntax

Data Structure. For Lacios, we define a standard information system data structure: every datum in the system has a *description*, i.e. a set of *property←value* pairs, and all the properties of the language are typed. We define in the following the notions of a type, a property and a description.

Definition 1. Types. *The types of the language are defined as* $type_1, \ldots, type_{nbt}$. *Every* $type_i$ *is a set such that* $\forall (i,j) \in \{1, \ldots, nbt\}^2, i \neq j, type_i \cap type_j = \{\text{nil}\}$

Remark 1. We assume the existence of the *boolean* type in the language, i.e. $\exists i \in \{1, \ldots, nbt\}$, $type_i = \{\text{true}, \text{false}, \text{nil}\}$

Definition 2. Property. $\mathcal{N}$ *is the property space, it is a countable set of properties. A property* $\pi \in \mathcal{N}$ *is defined by a type* $type(\pi) \in \{type_1, \ldots, type_{nbt}\}$.

A description is composed of properties and their corresponding values.

Definition 3. Descriptions. *DS is the set of descriptions. A description is a function that maps properties to values, i.e.* $d \equiv \{\pi \leftarrow v_\pi \mid v_\pi \in type(\pi)\}_{\pi \in \mathcal{N}}$. *The mapping is omitted when* $v_\pi = \text{nil}$. *We use* $d(\pi)$ *in order to access the value* v_π. *For every description, the set of properties* $\{\pi \mid d(\pi) \neq \text{nil}\}$ *is finite.*

A property evaluated to nil is considered undefined. In Lacios, every description is associated with an *entity*. An entity can be an *object* or an *agent*. An object is defined by its description ($\mathcal{O}$ is the set of objects), while an agent is defined by a description and a behavior ($\mathcal{A}$ is the set of agents).

Definition 4. Entities. $\Omega = \mathcal{A} \cup \mathcal{O}$ *is the set of entities of the MAS. Each entity* $\omega \in \Omega$ *has a description as defined above denoted by* d_ω. *The value of the property* π *of the entity* ω *is denoted by* $d_\omega(\pi)$.

Remark 2. We assume the existence of the type *reference* in Lacios, a value of the type *reference* designates an entity in Ω, i.e. $\exists i \in \{1, \ldots, nbt\}$, $type_i = \Omega \cup \{\text{nil}\}$.

For instance, let o_1 be an object, d_{o_1} could be defined as follows: $\{id \leftarrow$ "$o1$", $destination \leftarrow$ "$Uppsala$", $from \leftarrow$ "$Paris$"$\}$. In this example, $d_{o_1}(from)$ is equal to "$Paris$".

Expressions. Expressions are built with values, properties and operators. We define an operator as follows.

Definition 5. Operators. *Each operator op of the language is defined by:*

- *(i)* $arity(op)$ *The number of parameters of the operator,*
- *(ii)* $par(op) : \{1, \ldots, arity(op)\} \rightarrow \{1, \ldots, nbt\}$, $par(op)(i)$ *gives the index of the type of the* i^{th} *parameter of the operator op,*
- *(iii)* $ret(op) \in \{1, \ldots, nbt\}$*, the index of the type of the value resulting from the evaluation of op.*

For instance, let $type_1 \equiv boolean$. The operator `and` is defined as follows:

$arity(\texttt{and}) = 2$, $par(\texttt{and})(1) = par(\texttt{and})(2) = 1$ and $ret(\texttt{and}) = 1$.

Expressions are used by agents to describe the data they manipulate, either locally or to interact with the environment. An expression may simply be a value, an operator, or a property. If an expression is a property, it refers to a property of the agent that is evaluating it. For instance, when *destination* appears in the behavior of the agent a, it designates the destination of a. If a property *neighbor* of the agent a is of type *reference*, *neighbor.destination* designates the destination of the *neighbor* (an entity) of a.

Definition 6. Expressions. *Exp is the set of expressions. An expression* $e \in Exp$ *is generated via the grammar of table 1.*

Table 1. Syntax of an expression

$e ::=$ nil	
$\mid v$	, with $v \in \mathcal{T} \backslash$nil
$\mid \pi$	, with $\pi \in \mathcal{N}$
$\mid op(e, \ldots, e)$	, with op an operator of the language, and nil doesn't appear in any e
$\mid \pi.e$	, with $\pi \in \mathcal{N}$ and $type(\pi) = \Omega$

We can now associate an expression with a property instead of a value in a description. The result is a *symbolic description* which is transformed into a description when its associated expressions are evaluated.

Definition 7. Symbolic descriptions. *SDS is the set of symbolic descriptions. A symbolic description is a description that maps properties* π *to expressions* e_π*, i.e.* $sds \equiv \{\pi \leftarrow e_\pi \mid type(e_\pi) = type(\pi)\}_{\pi \in \mathcal{N}}$.

Matching. Since we consider a data structure richer than tuples, we also use a matching mechanism richer than templates. The matching in LACIOS materializes what we call a *contextual interaction*, which is the type of interactions that use the state of the agent and the state of objects in the environment to access a set of objects, instead of only one like in Linda templates. To do so, we enhance the expressions' syntax with *entity variables*, which designates objects not known by the agent, but will be discovered during the matching process and will be replaced by objects from the environment before their evaluation.

Definition 8. Variables. *$\mathcal{X}$ is the set of variables. A variable $x \in \mathcal{X}$ is defined by its type $type(x) \in \{type_1, \ldots, type_{nbt}\}$.*

The syntax of an expression becomes:

$$e ::= \ldots \mid x.e \text{ with } x \in \mathcal{X} \wedge type(x) = \Omega$$

For instance, consider the following expression:

$$t.destination = \text{“}Uppsala\text{”} \wedge t.price \leq budget \wedge p.decision = \text{“}accepted\text{”}$$

In this expression, t and p designate two objects, unknown for the moment, where t has to have as *destination* “Uppsala” and a *price* less than the *budget* of the agent, and where p must have as *decision* equal to “accepted” for the expression to be evaluated to true.

Agents' Actions. We define three primitives for LACIOS, two for the interaction with the environment (*add* and *look*) and one for agents' creation (*spawn*).

$$\mu ::= spawn(P, sds) \mid add(sds) \mid look(sds_p, sds_r, e)$$

The primitive $spawn(P, sds)$ launches a new agent that behaves like the process P and that has as a description the result of the evaluation of sds (its transformation to a description ds). The primitive $add(sds)$ adds to Ω_{ENV} an object described by the evaluation of sds.

We choose to use a single primitive to access the environment. The primitive $look(sds_p, sds_r, e)$ allows both object perception and reception (perception and removal from Ω_{ENV}). It blocks until a set of objects becomes present in Ω_{ENV} such that the expression e is evaluated to true; the objects associated with the variables in sds_p are perceived and those associated with the variables in sds_r are received. For instance, the following instruction: $look(\{ticket \leftarrow t\}, \{paper \leftarrow p\}, t.destination = \text{“}Uppsala\text{”} \wedge t.price \leq budget \wedge p.decision = \text{“}accepted\text{”})$ looks for two objects that will be associated with t and p. The object associated with t will be perceived while the object associated with p will be received. After the execution of this instruction, the two objects will be present in the local memory of the caller agent, the latter will have two additional properties of type *reference*: *ticket* that refers to the object associated with the variable t and *paper* that refers to the object associated with p.

3 An Environment-Centered MAS for Traveler Information Systems

In this section, we describe an application based on Lacios. We modeled and implemented a traveler information server [8], called ATIS[1]. The server purpose is to inform online travelers about the status of a part of the transport network that concerns them. Every traveler is mobile and has a specific objective during his connection on the server. Transport Web services are represented with agents in the server and their expertise domains constitute their properties. These transport service providers can give information in response to a request, or they may proactively send information concerning disturbances, accidents, events, etc. The problem in this kind of applications concerns the information flows that are dynamic and asynchronous. Indeed, each information source is hypothetically relevant. An agent cannot know *a priori* which information might interest him, since this depends on his own context, which changes during execution.

3.1 Context

Using Lacios allows us to design an information server parameterized by its users. Indeed, if the information systems are adapted to the satisfaction of punctual needs (request/response), the management of the information followup assumes additional processing. These processing are difficult because the information sources are distributed and the management of the followup assumes a comparison of the available information.

We have defined two categories of agents, the first concerns the agents representing the users (that we call PTA for Personal Travel Agent) while the second concerns the agents representing the services (that we call Service Agent).

3.2 Technical Description

We have implemented a Web server for traveler information, where each Web service has a representant in the server, which is responsible of the convey of messages from the server to the port of the Web service and conversely. The exchange of messages between the server and the services are SOAP [2] messages and the asynchronous communication is fulfilled via the JAXM API [3] for the Web services supporting SOAP, and a FTP server otherwise, used as a sort of mailbox. These details are obviously transparent for the agents evolving in the environment i.e. they interact exactly the same way whatever the transport protocol that is used. Every user is physically mobile and connects to the server via a Mobile Personal Transport Assistant (MPTA) and has during his connection a PTA agent representing him inside the server, which is his interlocutor during his session. The interaction of the MPTA with the PTA agents representing them is a sequence of HTTP requests/responses.

[1] Agent Traveler Information Server.
[2] Simple Object Access Protocol, http://www.w3.org/TR/soap/
[3] Java API for XML messaging, http://java.sun.com/webservices/jaxm/

3.3 Execution Scenario

The context of the example is the following: inside the system, there is an agent representing a trip planning service and an agent representing a traffic service responsible of the emission of messages related to incidents, traffic jams, etc. These agents are persistent, since they are constantly in relation with the system providing the service. On the contrary, PTA agents representing the MPTA in the system are volatile, created on the connection of a user and erased at the end of his session i.e. when he arrives at destination. We have developed a trip planning service which role is to, first receive the trip planning demand (in the form of a SOAP message), then calculates the plan, wraps it in a SOAP message then sends it back to the local agent representing him in ATIS.

Every stop of the network is described by a line number *line* to which it belongs, and a number *number* reflecting his position on the line. A user u is also described by his current position in the network (the properties *line* and *number*). In a basic execution scenario, u has a path to follow during his trip i.e. a sequence of tuples $\{(line, number_{source}, number_{destination})_i \mid i \in I\}$, with I the number of transport means used by the traveler. Every tuple represents a part of the trip, without transfer. To receive his plan, the MPTA connects to the information server, and the agent u representing him is created. Then, the user is asked to specify his departure as well as his destination. Once these information entered, u adds his planning demand in the environment. A demand is an object described by his properties: *emitter*, *subject*, etc. Afterwards, u keeps on listening to messages that are addressed to him, this way: $look(\emptyset, \{message \leftarrow x\}, x.receiver = id)$. The agent representing the trip planning service is listening to messages asking for a plan: $look(\emptyset, \{request \leftarrow x\}, x.subject = \text{“}plan\text{”})$. As soon as he receives the message, he creates a SOAP message addressed to the trip planning Web service and awaits for the response. When he receives the answer, a message is added to the environment addressed to u with the received plan as body: $add(\{emitter \leftarrow id, receiver \leftarrow request.emitter, body \leftarrow plan\})$. The agent u, when he receives the message, analyzes it and displays the result on the user's MPTA. Then, the agent u restrains his interaction to the messages concerning events coming up on his way. To do so, he executes the following action:

$look(\emptyset, \{event \leftarrow x\}, \{x.subject = \text{“}alert\text{”}\}, x.line = line \wedge x.number \geq number \wedge x.number \leq line)$

The agent u is interested by the alerts concerning his transport plan, which is expressed by the preceding *look* action. Let us assume that the agent representing the alert service adds an alert message concerning an accident on the way of u resulting on a serious delay for him. The traveler, via his representing agent u, is notified concerning this alert event. Since the properties *line* and *number* are updated at each move of u (each time he moves from stop to stop), the segment concerned by the alert messages gets gradually reduced until the end of the trip. The use of the environment, the constant update of the properties of the PTA agents, together with the use of *look* actions allowed us to maintain a constant awareness of the traveler about problems occurring during his trip, without relying on continuous requests to the server.

4 Coordination Environment for Demand-Responsive Transportation Systems

We have proposed a demand-responsive transportation system (DRTS) as a MAS in which the agents' activities are coordinated through the environment, based on Lacios.

4.1 Demand-Responsive Transportation Systems

A DRTS is a system designed to answer online customers that desire to be transported from one point in the network to another. Customers specify a time window associated with each point (departure and arrival) inside which they want to be visited. In our application the environment, via the use of Lacios, structures the MAS components temporally and spatially, so that the interaction between agents is driven by their perception of it. The interaction between customers and vehicles is driven by their space-time positions, and the environment is modeled accordingly. A main issue in this application is the dynamics of the environment because when modeled as an MAS, DRTS are open MAS, where agents (e.g. customers and vehicles) join and leave the system freely. In such a dynamic environment, limiting communications is very important, since broadcasting all the available information is very costly. We use an implicit model on which Lacios is grounded, in which communication is decoupled in space and time, so as to offset the loss in information in dynamic environments.

4.2 System Description

We have designed a distributed model for a DRTS, in which two agent categories are defined: Vehicle Agents (VA) and Customer Agents (CA). Both VAs and CAs are generated dynamically: a new CA is associated to each new customer connected to the system, and a new VA is associated to each new vehicle creation (which occurs when no available vehicle can serve a new customer). The descriptions in this system are related to VAs and CAs (see figure 1). A VA is described symbolically by his current position and his remaining available seats. A CA is described by his departure and arrival nodes, his time windows, the vehicle *veh*, and his successor (property *succ*) and predecessor (*pred*) customer in the route of *veh*. A CA that doesn't belong to any VA route has a property *veh* equal to $unknown_{veh}$, and a property *succ* and *pred* equal to $unknown_{cust}$.

In our application the environment structures the MAS components temporally and spatially, so that the interaction between agents is driven by their perception of it. The boolean expressions contained in their *look* actions are defined by VAs so that they will perceive only those customers they can insert in their route. The description of a new CA (with unknown *veh*, *succ* and *pred*) is perceived when: (i) there are two nodes in the route of the same vehicle between which its departure node can be inserted without violating any of the time windows of the customers in that route, (ii) if there are remaining available seats

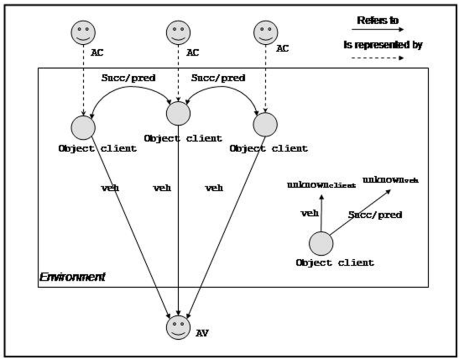


Fig. 1. Symbolic descriptions

when this insertion occurs, (iii) if there are two successor nodes in the route of this same vehicle between which the arrival node of the customer can be inserted, without violating any time window of the route. As a consequence, the use of Lacios allows a new CA to discover the VAs that could be interested by its insertion, without knowing them in advance, while at the same time limiting communication in the system to only those agents that can reach an agreement (an insertion in the route of a vehicle). It is worth noting that the VAs that don't perceive a CA can use their time to be candidates for the insertion of other customers.

The protocol followed in the MAS is a negotiation mechanism between CAs and VAs. When a new customer connects to the system, a CA is created (*spawn*), and is perceived by the available VAs with their current *look* actions (that is, which are not already involved in the insertion of another customer). The syntax of expressions that we introduced for Lacios allows for the translation of the mathematical constraints on the insertion of a customer (vehicle capacity and space-time feasibility) onto Lacios expressions, which is not possible for traditional Linda-like languages. Each VA computes an insertion price for the insertion of this customer, and proposes it to the CA (*add*), which will choose the VA proposing the lowest price.

The interested reader about the complete definition of the agents actions and their expression parameters is invited to refer to [7].

In this application, the use of Lacios structured agent interaction and coordination, and made it more efficient to interact in a dynamic environment where agents appear and disappear without maintaining knowledge about the others and where communications can be disturbed and costly.

5 Conclusion

In this paper, we have described the main part of the Lacios coordination language while the complete definition can be found in [7]. The relevance of

Lacios for transportation applications is assessed by two applications: a traveler information server and a demand-responsive transport service.

In [7], we have implemented a script language on top of Java for the implementation of MAS in Lacios (called Java-Lacios). It consists of a compiler of a program written in Lacios (text file) which generates a Java program. Therefore, system designers and programmers do not have to worry about the creation and synchronization of threads, or the matching process, while building their MAS. This new programming language makes it easier to build environment-centered applications, and is suitable for transportation applications, where the interaction needs of the agents (often based on mathematical constraints) are complex and contextual.

The multi-agent environment might be distributed over several hosts, but until now, this was done in an *ad hoc* way aiming at reducing communication cots between the different hosts. We are currently working on automatic distribution of the environment, following the specification of entities' properties, and based on their clustering in the form of Galois lattices.

References

1. Badeig, F., Balbo, F., Scémama, G., Zargayouna, M.: Agent-Based Coordination Model for Designing Transportation Applications. In: Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems, pp. 402–407. IEEE Computer Society, Los Alamitos (2008)
2. Gelernter, D.: Generative communication in linda. ACM Transactions on Programming Languages and Systems 7(1), 80–112 (1985)
3. De Nicola, R., Ferrari, G.L., Pugliese, R.: Klaim: A Kernel Language for Agents Interaction and Mobility. IEEE Transactions on Software Engineering 24(5), 315–330 (1998)
4. Cabri, G., Leonardi, L., Zambonelli, F.: Reactive tuple spaces for mobile agent coordination. In: Rothermel, K., Hohl, F. (eds.) MA 1998. LNCS, vol. 1477, pp. 237–248. Springer, Heidelberg (1998)
5. Picco, G.P., Murphy, A.L., Roman, G.C.: LIME: Linda meets mobility. In: International Conference on Software Engineering, pp. 368–377 (1999)
6. Milner, R.: Communication and Concurrency, 272 pages. Prentice-Hall, Englewood Cliffs (1989)
7. Zargayouna, M.: Coordination model and language for open multiagent systems. Application to the Dial A Ride Problem. PhD thesis, University of Paris-Dauphine, Paris, France (2007) (in French)
8. Zargayouna, M., Balbo, F., Saunier, J.: Agent information server: a middleware for traveler information. In: Dikenelli, O., Gleizes, M.P., Ricci, A. (eds.) ESAW 2005. LNCS(LNAI), vol. 3963, pp. 3–16. Springer, Heidelberg (2006)

Multi-Agent Based Approach for Quantum Key Distribution in WiFi Networks

Shirantha Wijesekera, Xu Huang, and Dharmendra Sharma

Faculty of Information Sciences and Engineering,
University of Canberra, ACT 2601, Australia
{Shirantha.Wijesekera,Xu.Huang,
Dharmendra.Sharma}@canberra.edu.au

Abstract. Security in WiFi networks has become a major concern as the wireless networks are vulnerable to security threats than wired networks. The 802.11i wireless networks uses 4 way handshake protocol to distribute the key hierarchy in order to encrypt the data communication. In our previous research work [2], we have discussed the use of Quantum Key Distribution (QKD), for key distribution in 802.11 wireless networks. The whole communication flow of our proposed protocol can be split into several key processes. It can be seen that these processes can be implemented efficiently using Software Agents. In this paper we shall discuss the use of Software Agents in quantum cryptography based key distribution in WiFi wireless networks.

Keywords: Quantum Key Distribution (QKD), MAS, WiFi, 802.11i.

1 Introduction

Wireless communications are becoming ubiquitous in homes, offices and enterprises with its ability to provide high-speed, high-quality information exchange between portable devices. The IEEE 802.11 family, also known as WiFi, is one of the most popular Wireless Local Area Networks (WLAN) widely used today across the globe. There are three main participants in WiFi: Supplicant, Authenticator and Authentication Server. Supplicant is the one trying to gain access to the wireless network while Authenticator is the access point which realizes the access control and allows only the Supplicants who are authenticated to use the network. Authentication Server is a centralized server which authenticates the supplicants.

The importance of security in wireless networks cannot be under estimated. Since the communication medium is shared, wireless networks are vulnerable to attacks than wired networks. WiFi networks uses 802.11 and 802.1X for association and authentication process. The authentication of the end users is essential in wireless networks as the wireless medium is accessible openly. A lot of research papers highlighted the security flaws of wireless networks based on 802.11 [10, 11, 12]. Most of those are happening in the form of Denial of Service (DoS) attacks, Main-in-the-Middle (MiM) attacks, session hijacking (SH) etc.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 293–303, 2009.

In our previous [2] and subsequent work, we have come up with a novel protocol to perform the key management in WiFi networks. A careful study of how this protocol works shows that it can be effectively implemented using software agents. Software agents can deliver much needed intelligent behavior to WiFi networks especially in case of adversary attacks. In this paper, we explain how Multi Agent Systems (MAS) can be used to perform the key exchange in WiFi networks.

This paper comprises of seven sections: in section 2, we give a brief recap of our previous work along with short introductions to the use of quantum cryptography in WiFi, 802.11 and 802.1X authentication. Section 3 explains our approach using multi agents to the quantum cryptography based key distribution of WiFi networks. It also explains why Multi Agent Systems are suitable to be used in our QKD based WiFi networks. Section 4 discusses the possible future expansion to our approach while the Conclusions are in Section 5.

2 Integrating Quantum Key Distribution in IEEE 802.11i Networks

The IEEE 802.11 Task Group has come up with an amendment to the IEEE 802.11 standard [3] called IEEE 802.11i [1] in 2004 to address the security flaws encountered in its initial design. IEEE 802.11i separates the authentication and encryption key management. For authentication 802.11i uses IEEE 802.1X [4] and pre-shared key. IEEE 802.1X offers an effective framework for authenticating, managing keys and controlling user traffic to protect large networks. It utilises Extensible Authentication Protocol (EAP) [5] encapsulated over LANs (EAPOL).

In 802.11i, Supplicant first associates with Authenticator using 802.11 Association process. Once this is finished, 802.1X authentication process take place. During this process, the Supplicant and Authenticator choose a suitable EAP type for the authentication supported by both parties. Once this process completes, the 802.11i 4-way handshake process takes place and ensures the 802.11i key hierarchy to establish at both ends. This key hierarchy consists of several keys, namely: Pairwise Master Key (PMK), EAPOL-Key Confirmation Key (KCK), EAPOL-Key Encryption Key (KEK), Group Temporal Key (GTK) and Temporal Key (TK). The 802.11i uses TK to encrypt data for the rest of the communication.

2.1 Use of Quantum Cryptography in WiFi

Based on the laws of physics, quantum cryptography allows the exchange of cryptographic key between two remote parties with unconditional security. Though the use of quantum cryptography in wireless communications is still premature, the "unconditional security" [18] of quantum cryptography offers much needed security for wireless networks. Quantum cryptography is used to produce and distribute a key, known as Quantum Key Distribution (QKD). At present lot of research work and commercial implementations are happening in this area [16, 17, 22, 23]. Several QKD protocols such as SARG04 [6], BB84 [7], B92 [8] and six-state [9] exist as of now. Out of those, BB84 has proven in practical networks. SARG04 protocol is an improved version of BB84 by eliminating Photon Number Splitting (PNS) attacks. As BB84 does, SARG04 protocol operates in two stages: Quantum channel and Classical channel. In the first stage, photon transmission takes place via quantum channel between two parties. Each of these photons represent a binary bit value of the secrete key. During the second stage, the two parties communicate with each other as per the SARG04 protocol to obtain the secrete key. The second stage comprises of four main phases: (1) Raw Key Extraction (Sifting), (2) Error Estimation, (3) Reconciliation and (4) Privacy Amplification. Further investigation of SARG04 protocol is beyond the scope of this paper.

2.2 QKD Based Key Exchange in 802.11i

Figure 1 shows the full 802.11i protocol communication including the quantum key exchange. Flows 1 to 6 illustrate the IEEE 802.11 association and authentication process. During this process, the Supplicant creates an 802.11 association with the Authenticator. Once the IEEE 802.11 association is completed, the IEEE 802.1X

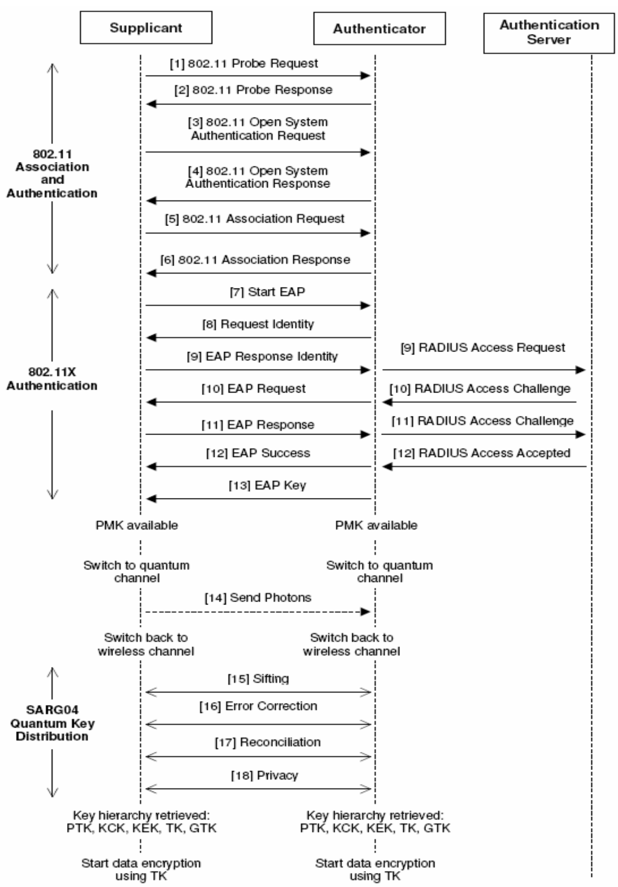


Fig. 1. The QKD based Protocol for Key Exchange

authentication starts with the Supplicant sending EAP-Start message to the Authenticator. This process is shown by flows 7 to 13 of Figure 1. In our work, we choose to use EAP types such as EAP-TLS, EAP-TTLS etc that offer mutual authentication between the Supplicant and the Authenticator.

At the end of this process, i.e. flow 13 of figure 1, both Supplicant and Authenticator are in possession of Pairwise Master Key (PMK). Then the communication switches to quantum channel and the photon transmission takes place from the Supplicant towards the Authenticator. Once the quantum transmission finishes, communication channel switches back to wireless channel. Afterwards the SARG04 protocol takes place as shown in flows 15 to 18 in figure 1 to obtain the final secrete key. From this key, the 802.11i key hierarchy containing PTK, KCK, KEK, TK and GTK can be retrieved. The TK is used to encrypt data for the subsequent data communication. This whole process is not explained in detail in this paper as our focus is on the use of agents for this protocol.

3 Implementation of Agent Based QKD in WiFi Networks

3.1 Why Multi Agent System?

An Agent can be referred to as a sophisticated computer program, which is capable of acting autonomously to accomplish tasks on behalf of its users, across open and distributed environments. Hence agents have individual internal states and goals, and they act in such a manner as to meet its goals on behalf of its users [19]. Multiple agents can work together to form a multi-agent system (MAS), which offer many advantages over a single agent or centralized approach [15]. Some of these advantages are:

- Distributes computational resources and capabilities across a network of interconnected agents.
- Allows interconnection and interoperation of multiple existing legacy systems by adding those using wrappers into an agent society.
- Enhances overall system performance, specifically along the dimensions of computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility, and reuse.

Multi Agent Systems in our quantum based key distribution in IEEE 802.11i networks has various advantages. Firstly, the whole protocol can be subdivided into smaller independent modules: 802.11 Association, 802.1X Authentication, Quantum Communication and SARG04 key extraction. These sub-modules can be represented by individual agents to accomplish the main task required. By this way the workload can be distributed among the sub-modules, rather than handling by a single piece of software (centralised approach). Secondly, there are different varieties of EAP types in use for 802.1X authentication such as EAP-TLS, EAP-TTLS, PEAP etc. Therefore, rather than having separate communication flows for each of them, wrapper agents can be used to implement those different EAP varieties. Thirdly, the system maintenance becomes easy as the agents can work independently. Whenever a new change is required to the protocol, it can be done without effecting to the other modules.

Fourthly, the system is open to extensions due to modularization via agents. For example, imagine a case where a new EAP type introduced to the protocol suite. In such instances, it can be easily incorporated into the agent society via another agent.

Agents also offer the intelligent behavior to the system. This is a special feature where other wireless protocol implementations are lacking. With this feature, the agents can be taught to detect possible adversary attacks.

3.2 Possible Attacks on 802.11 and 802.1X Protocol Standards

Many research papers have shown security vulnerabilities of 802.1X standard [13]. As an example, we shall discuss two such attacks.

Session Hijacking: It was shown that session hijacking is possible on 802.1X [13]. In these types of attacks, an adversary can spoof communication between a legitimate supplicant and the Authenticator till EAP Success message is received. At this point the adversary sends 802.11 MAC disassociate message using Authenticator's MAC address. This causes the legitimate Supplicant to get disassociated from the Authenticator. However, at this moment the Authenticator is not aware that the legitimate supplicant has kicked out, it still remains in Authenticated state. The adversary takes this opportunity to hijack the session.

Denial of Service Attacks: Both 802.11 and 802.1X protocols are subject to DoS attacks [13]. These DoS attacks happen in several ways. Adversaries can send fake EAPOL Logoff, EAPOL Start and EAP failure messages towards Authenticator causing the system to fail.

3.3 QKD Based MAS Application

As discussed in the previous section, the QKD based 802.11i authentication process can be well represented using Multi Agent Systems. In our approach, we split the main functionalities of each of the major phases to be represented by software agents. As identified before, the authentication and key establishment can be split into following main components:

- 802.11 Association and Authentication: for the supplicant to associate with the switch
- 802.1X Authentication: to facilitate mutual authentication between Supplicant and Authenticator
- Quantum transmission: send photons to Authenticator to be used for the key
- Key recovery using SARG04 protocol: recover the final key by removing errors

In our approach, the Agent Society is made up of a main Enterprise as shown in Figure 2 – *The Enterprise*.

Since both Authenticator and Supplicant execute same communication protocols, the same enterprise is used at both ends. However their functionalities will be slightly different as they interact with the communication via "Request – Response" basis.

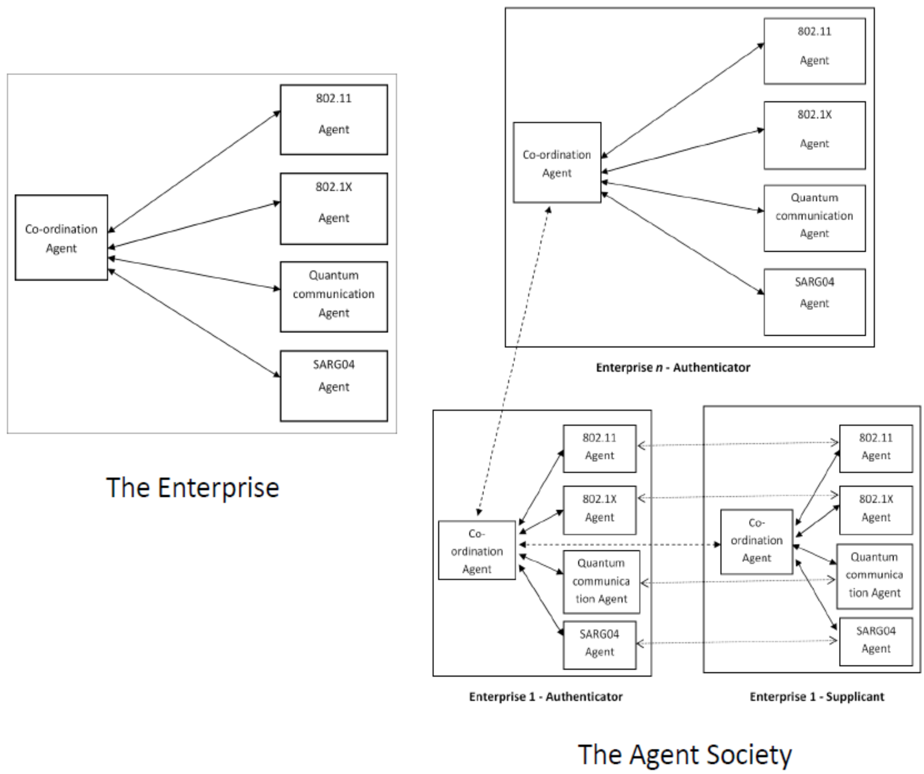


Fig. 2. The Enterprise and The Agent Society

In WiFi networks, the Authenticator caters multiple Supplicants simultaneously. Each time a new Supplicant enters into the network, the Authenticator serves the Supplicant by executing 802.11 and 802.1X protocols. To facilitate the services, Authenticator spawns a new enterprise for each Supplicant that enters into the wireless network. At the same time the Supplicant too creates one instance of this enterprise to proceed with the communication. Single instance of the enterprise is sufficient at the Supplicant's end as it is only dealing with one Authenticator at a time. These enterprises get together makes the overall Agent Society spanning across the WiFi network served by the Authenticator as shown in figure 2-*The Agent Society*.

The functionalities of each agent are described below.

802.11 Agent: The main aim of this agent is to perform the 802.11 Association and Authentication. In doing so, this agent can deliver something present 802.11 standard is not capable of doing. That is, with the use of artificial intelligence, this agent is able to take decisions during various adversary attacks. For an example, during a session hijack attack described in section 3.2, this agent can make a decision upon seeing the disassociate message soon after EAP Success message. This decision could be an intelligent guess to ignore the message or the agent can enquire about the authenticity of that message from its peer 802.11 Agent at Authenticator side.

802.1X Agent: This agent carries out the 802.1X authentication. In this implementation, for simplicity, we only focus on EAP methods that support mutual authentication. This agent is able to support multiple EAP protocols by communicating with different wrapper agents. In addition, it is also able to make decisions on suspicious messages from adversaries similar to what 802.11 Agent does. A wrapper agent serves 802.1X agent to implement the respective EAP protocol for that particular communication. Instance of this agent running at Authenticator communicates with both Supplicant and the Authentication Server.

Quantum Communication Agent: This agent communicates with hardware devices such as photon transmitter and receiver to make the quantum transmission happen.

SARG04 Agent: The main task of this agent is to execute the SARG04 QKD protocol. It executes the 4 phases of SARG04 protocol in order to extract the final secrete key.

Coordination Agent: The coordination agent communicates with all other agents within the enterprise. Coordination agent in each communication session assures that monitoring efforts and management of internal requests with other agents handled consistently within that specific session. It instructs other agents when to start each activity assigned to them. In return, the agents inform the coordination agent on completion of their tasks or alerts if there's anything goes wrong. In addition, coordination agent communicates with peer coordination agents of other enterprises. Further, coordination agents exchange security and any other information between enterprises at Authenticator side. This is especially useful for an enterprise to alert other enterprises about the presence of an adversary. The coordination agent provides an overview of all monitored communications between agents.

The overall operation of this agent society is described below. Figure 3 illustrates the work flow of the protocol.

Whenever a new Supplicant enters into the wireless network, the Authenticator creates a new enterprise to serve the communication. Supplicant too will have its own enterprise created to communicate with the Authenticator. Coordination agent instructs 802.11 agent to perform the 802.11 association and authentication. Once this task is completed, 802.11 agent informs the outcome to coordination agent. It then calls upon 802.1X Authentication agent to perform the authentication. Once the task is done, 802.1X Authentication agent informs coordination agent about the outcome. Next, coordination agent instructs quantum communication agent to carry out the quantum transmission. Upon completing the task, it gives the control back to the coordination agent, which in turn commands the SARG04 agent to go ahead with SARG04 process to obtain the secrete key. Once this is completed, the two sides are authenticated, so they can start communication by encrypting the data using the secrete key.

In this solution, not a single agent is fully aware of the whole communication process. Instead, all agents get together to make the whole communication happen. The coordination agent has some knowledge about the overall flow of the system, but it needs not to be aware of the internal activities of other agents. With this kind of approach, which is quite suitable to be represented as an agent society, modifications can be done effectively.

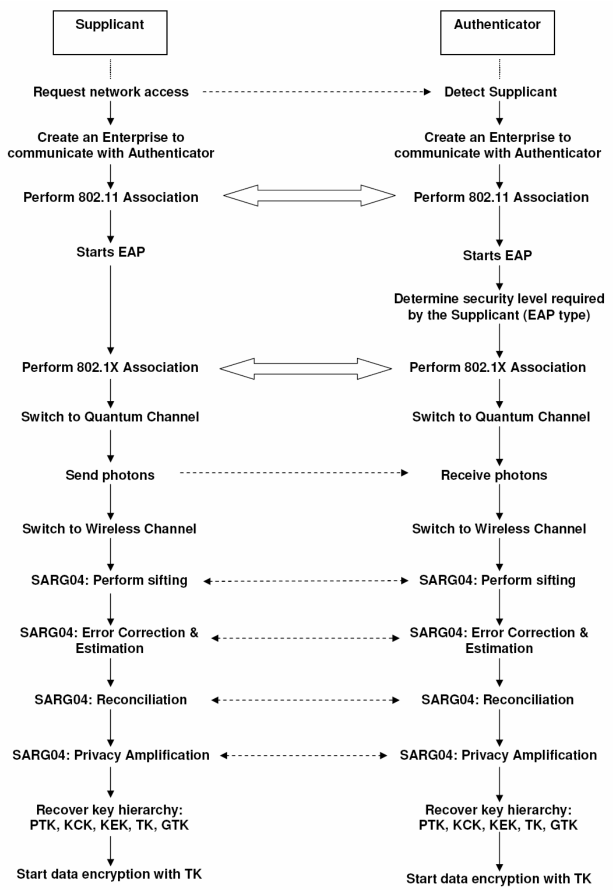


Fig. 3. The Process Flow of Wireless QKD

With this agent society, the attacks described in section 3.2 can easily be detected. For an example, consider the session hijack scenario. In this case, the Supplicant receives MAC Disassociate message soon after EAP Success message. Since this can be treat as a suspicious message (knowing about the session hijack), Agents can use the intelligence of the agent to act on it. Also, in this situation once loosing the association, legitimate Supplicant will try to make a new association with the Authenticator. However Authenticator has an established connection with the Adversary. In this situation Agents reside at Authenticator too can make an intelligent approach to deal with it.

Similarly, the DoS attack described in section 3.2 can be dealt with when 802.1X Agent detects any fake EAP messages.

The software test bed is now being implemented on two computers with one acting as the Supplicant and the other as Authenticator. For simplicity, we use both Authenticator and Authentication Server as one entity.

The Supplicant is running on Windows XP, while the Authenticator/Authentication Server machine is running on Windows Server 2000. We have chosen Microsoft "Native WiFi" product with its user Application Programming Interfaces (API) to be used for software developments. The software development is being done in C++ language. Our main aim is to simulate a lap top setting up the secure communication path with the Authenticator by retrieving secrete key (TK) via the new protocol.

Since the "Native WiFi" software developments are based on C++ platform, we have concentrated on developing MAS application using the same C++ language. We have found that most of the MAS applications only support Java based developments. Therefore we have decided to write our own application for MAS using C++. This works well with MAS, since C++ being an object oriented language, the agents can easily be represented by C++ classes. As of now we are in the process of developing the SARG04 Agent along with Co-ordination Agent. Appleby and Steward of BT Labs have done a similar approach to prototype a mobile agent based system for controlling telecommunication networks [24].

3.4 Performance Analysis of MAS Application

The approach of QKD in WiFi networks provides benefits to the system in many aspects. With the security point of view, this approach brings unconditional security offered by quantum cryptography to the WiFi communication. 802.11i protocol normally requests the secure key, PTK, to be refreshed at regular intervals to guarantee secure communication. With QKD approach, key refresh is not needed as the final key is providing unconditional security, which any other systems cannot offer. This makes the protocol more efficient by saving number of refresh cycles on every communication session.

On the other hand, use of MAS in this solution too brings many benefits to the system. Splitting the main functionalities of the protocol into several key modules in the form of agents, provide improvements in overall protocol efficiency. Since each agent has been assigned a specific task, they come into existence only when needed. Once the task is completed, the agent quits from the operation, thereby releasing the system resources. Maintenance of the system too becomes easy as only the relevant agent is needed to update if any changes are required.

Further, most vulnerable security attacks as described in section 3.2 have been avoided. This too makes the system more robust and efficient.

Overall, above facts justifies how the system performance, resource usage etc are improved with the QKD-MAS approach compared to traditional 802.11i protocol.

4 Future Work

The MAS approach for WiFi key exchange opens up lot of opportunities for future expansions. WiFi networks comprises of several IEEE protocols like 802.11, 802.1X, EAP etc. Due to the fact that communication medium is open air, the message flows of these protocols are subject to various adversary attacks. These attacks can be identified by the intelligent behavior of the agents. Therefore, if a new type of attack is identified, the agents can be taught on to how to deal with them.

In this approach we only considered EAP types which offer mutual authentication. This is mainly because we wanted to simplify our work to facilitate the multi agent implementation. By that way we managed to get an extra security as the two parties can build up trust between each other via mutual authentication. However, in reality, this approach can cater all available EAP types regardless of whether they offer mutual authentication or not. This can be achieved by having multiple wrapper agents for 802.1X Agent. Since the Authenticator has to cater supplicants operating on different EAP types, it can choose the required one for each connection. These EAP types can be added to the system as wrapper agents. With this way, modifications to any of those EAP types can be done with minimal impact to other operations.

In case if a Supplicant is moving from one Authenticator to another, the necessary information for a specific session can be sent to the new Authenticator via a Mobile Agent. Since 802.1X can cache cryptographic keys, such information can be used for this purpose [14]. This eliminates the need to produce new keys for each Authenticator.

5 Conclusion

In this paper we have discussed the use of software agents in QKD based key distribution protocol in WiFi networks. It has been shown that the main protocol flow of this protocol can be broken down to several sub-modules. Each sub-module can be represented by an agent, where those agents work together to achieve a common goal of distributing and recovering final the key. This agent society is particularly useful at Authenticator side as it plays a key role within WiFi networks. As the Authenticator assigns a separate enterprise to look after each Supplicant, the work load can be distributed. This is one of the key requirements for Multi Agent Systems.

This agent approach provides lot of advantages to the wireless communication. Since the key work flows are incorporated into agents, maintenance too becomes easy. Whenever new change to the protocol is needed, it can be done with less effort, without affecting the other agents. It also provides extensibility by allowing different EAP wrapper agents to facilitate different EAP types. This is very useful as different Supplicants require different types of EAP methods. It also provides extensibility by allowing interconnection and interoperation of multiple existing legacy systems (for example, WEP based systems) by adding them using a wrapper into the agent society.

In this work we pay special attention to the security, as the wireless networks are vulnerable to attacks. We can effectively encounter such attacks through the intelligent agent approach achieving reliability and robustness.

It can be seen that this approach is open to extensibility. For example, in case if a Supplicant is moving to a different wireless network, the key information can be transferred to the new network via a mobile agent.

Thus we can conclude that the use of Multi Agent Systems in QKD based WiFi networks offer lot of benefits. There are other research works being done in WiFi area using software agents [20]. We believe our approach using Multi Agent Systems will contribute to develop secure communications for future wireless networks.

Acknowledgments. We would like to take this opportunity to appreciate University of Canberra, Australia for the URG supports.

References

1. IEEE Std 802.11i, IEEE Standard for Information Technology – Telecommunication and information exchange between systems – Local and metropolitan area networks – Specific requirements. Part 11: Wireless LAN Medium Access Control O(MAC) and Physical Layer (PHY) specifications, Amendment 6: Medium Access Control (MAC) Security Enhancements (2004)
2. Huang, X., Wijesekera, S., Sharma, D.: Implementation of Quantum Key Distribution in Wi-Fi (IEEE 802.11) Wireless Networks. In: IEEE the 10th International Conference on Advanced Communication Technology, Phoenix Park, Korea, Proceedings, February 17-20, vol. II, p. 865 (2008) ISSN 1738-9445, ISBN 978-89-5519-135-6
3. ANSI/IEEE 802.11, 1999 Edition (R2003), Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (2003)
4. IEEE Std 802.1X, IEEE Standard for Local and metropolitan area networks, Port-Based Network Access Control (2004)
5. Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., Levkowetz, H.: RFC – 3748, Extensible Authentication Protocol (EAP) (2004)
6. Scarani, V., Acin, A., Ribordy, G., Gisin, N.: Quantum cryptography protocols robust against photon number splitting attacks (2004)
7. Bennett, C.H., Brassard, G.: Quantum cryptography: Public-key distribution and coin tossing. In: Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India, December 1984, pp. 175–179 (1984)
8. Bennett, C.H.: Phys. Rev. Lett. 68, 3121 (1992)
9. Bruß, D.: Optimal Eavesdropping in Quantum Cryptography with Six States. Physical Review Letters 81, 3018 (1998)
10. He, C., Mitchell, J.C.: Analysis of the 802.11i 4-way Handshake
11. De Rango, F., Lentini, D., Marano, S.: Statis and Dynamic 4-Way Handshake Solutions to Avoid Denial of Service Attack in Wi-Fi Protected Access and IEEE 802.11i (June 2006)
12. He, C., Mitchell, J.C.: Security Analysis and Improvements for IEEE 802.11i
13. Mishra, A., Arbaudh, W.A.: An Initial Analysis of the IEEE 802.1X Standard (February 2002)
14. Neal Leavitt, Will IEEE 802.1X Finally Take Off in 2008? IEEE Computer Society, Los Alamitos, pp. 82–85 (March 2008)
15. Multi-Agent Systems, `http://www.cs.cmu.edu/~softagents/multi.html`
16. SECOQC, Development of a Global Network for Secure Communication based on Quantum Cryptography, `http://www.secoqc.net/`
17. Graham-Rowe, D.: Quantum ATM' rules out fraudulent web purchases. New Scientist (2629) (November 2007)
18. Mayers, D.: Unconditional Security in Quantum Cryptography. Journal of the ACM 48(3), 351–406 (2001)
19. Nwana, H.S.: Software Agents: An Overview, Intelligent Systems Research, AA&T, BT Laboratories (1996)
20. Automatic Resumption of Streaming Sessions over WiFi Using JADE, Alvaro Suárez, Member, IAENG, M. La-Menza, Elsa M. Macías, Member, IAENG and Vaidy Sunderam
21. Genesereth, M., Fikes, R.: Knowledge interchange format. Version 3.0 Reference Manual, Technical Report Logic 92-1, Computer Science Department, Stanford University (1992)
22. id Quantique, Quantum Cryptography, `http://www.idquantique.com/`
23. New Scientist, Quantum ATM rules out fraudulent web purchases (November 10, 2007)
24. Appleby, Steward: Mobile Software Agents for Control in Telecommunications Networks. BT Technological Journal 12(2), 1040113 (1994)

A Secure Delegation Model Based on Multi-agent in Pervasive Environments

Kyu Il Kim, Won Gil Choi, Eun Ju Lee, and Ung Mo Kim

Department of Computer Engineering, SungKyunKwan University
2nd Engineering Building 27309, 300 ChunChun-Dong, JangAn-Gu,
Suwon, KyungGi-Do, Korea
{kisado,wonkiler,eunjoo,umkim}@ece.skku.ac.kr
http://dblab.skku.ac.kr

Abstract. In pervasive environments, the connected devices can be aware the status of users and provide the information to users automatically in anytime, anywhere. However these require more security technologies to protect private information. The users must temporarily delegate some or all of their rights to agents in order to perform actions on their behalf. This results in the exposure of user privacy information by agents. We propose a delegation model for providing safety and user privacy in pervasive computing environments. In order to provide safety and user privacy, XML-based encryption and a digital signature mechanism needs to be efficiently integrated. In this paper, we propose access control mechanism based on eXtensible Access Control Markup Language (XACML) in order to manage the services and policies. We also extend SAML, in order to declare delegation assertions transferred to service providers by delegation among agents.

Keywords: Delegation, SAML, XACML, Pervasive Environment.

1 Introduction

Many aspects of Mark Weiser's original vision of ubiquitous computing [3, 4] are rapidly becoming reality. A context aware system has the ability to usefully adapt services or applications to the user's current situation, intention, needs or environments. In pervasive environment, security has become one of the most critical issues. Many organizations use security mechanisms to protect sensitive resource. However existing approaches are inadequate to meet the requirement for privacy safeguard. Traditional system also relies on a static characterization of context, where resource availability is independent of both user current location and access device properties. Nowadays accidents and crime augment in the open system. An attacker in these systems may expose user privacy and service information without authentication. The following issues may require change of current application. In august 2007, he was team manager at the Korea shipbuilding company, who accessed the knowledge management system for about a month. He stole the flow sheet, the plan report and the ship completion report. This information was exposed to the other company. A

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 304–313, 2009.

criminal who owned the grant privilege as a responsible person could easily steal these technologies. The damage amounts to US $500 million.

In pervasive computing environments, user use agents, in order to initiate web services. Agents transfer user's credentials to web service providers which then perform web services on their behalf. Users are required to disclose their privacy information by agents in order to perform web services. This results that user's privacy information is disclosed to the outside. In order to support privacy protection, service confidentiality and assertion integrity, we adopt XML-based encryption and a digital signature mechanism [16]. Security Assertion Markup Language (SAML) is exploited in order to declare delegation assertions transferred to web service providers using a delegation mechanism among agents. An eXtensible Access Control Markup Language (XACML) is exploited in order to manage the services and policies supported by web service providers. OASIS standards, SAML and XACML contain Single Sign-On (SSO) [17], Trust Management, Authentication, and Authorization.

We propose delegation model for providing safety of web service and protection of user privacy with assistance of reliable authority among agents. An important contribution of our model is applicable to delegation in the pervasive environment using all kinds of agents. The paper is organized follows: Section2 describes related work in the field of current mechanism problems. Section3 defines a proposed model. Section4 shows experimental result. Finally, Section5 presents conclusions.

2 Related Work

SAML[1] defined the syntax and processing semantics of assertions made about a subject by a system entity. SAML assertions and protocol messages are encoded in XML and use XML namespaces. They are typically embedded in other structures for transport, such as HTTP POST request or XML-encoded SOAP (Simple Object Access Control) messages. SOAP [18] is a protocol for exchanging XML-based message over computer networks, normally using HTTP. SAML specification defines three different kinds of assertion statements that can be created by a SAML authority. All SAML-defined statements are associated with a subject. The three kinds of statement defined in this specification are:

- **Authentication:** The assertion subject was authenticated by a particular means at a particular time.
- **Attribute:** The assertion subject is associated with the supplied attributes.
- **Authorization Decision:** A request to allow the assertion subject to access the specified resource has been granted or denied.

SAML lacks delegation capabilities. However, SAML provide inherent extensibility to create our delegation assertion. This paper support OASIS standard SAML v2.0. Currently in researched approach Navarro, et al [11] describe SAML assertion for constrained delegation. In order to support delegation, they extend *SubjectStatement* to *SubjectDelegationStatement* which is not supported by the SAML1.1/2.0 Assertion specifications. Wang and Del Vecchio [12] also extend SAML to support delegation. They propose verification rules for delegation assertions relying on a WS-Security X.509 Signature [7]. However, they must verify an ordered delegation chain to prove a trust relationship between a delegater and a web service portal when the indirect delegation is applied. Hu [9] presents a method for creating agent systems, which is based on Public Key Infrastructure (PKI). He considers various kinds of delegation and conditional delegation. However, this approach does not consider mobile agent systems or heterogeneous multi-agent system. Navarro, et al[6] present an access control model for a mobile agent platform, which is based on RBAC [5]. In this model, the Authorization Manager (AM) is exploited, to manage the delegation of authorizations, and issue of authorization certificates. However, they only consider agents complying with its local policy. In our work, we extend the *AttributeStatement* supported by SAML 1.1/2.0 specification to the *AttributeStatement* containing delegation information. We exploit the authentication authority to omit the step of verifying an ordered delegation chain. Our model only requires a delegation assertion to verify a trust relationship.

3 Delegation Model Using Agents

3.1 Delegation Model Based on Multi-agents

Fig1 describes a delegation model to support privacy protection based on multi-agents in pervasive environments. Our model enables users to manage delegation of their own rights. Agents are controlled by the use of services based on delegation rights by the users. They classify agents as to dependent and independent. Two concept delegates equally their right to other agents but independent means agent to work without any requirement. Otherwise, dependent means agent that request requirement to agents. It can't perform the work without support of other agents. Dependent also has hierarchy level to better manage agents because delegation frequency has an aptitude to expose private information. User gives credentials and conditions to User agent. User agent is software that interacts with other agents on behalf of user in order to exploit the resource that he/she wants. UA (*User Agent*) requests the privilege to *privacy control server* (PCS) for web service. PCS means process that determines access privilege over the resource to agent. PCS retrieves service information corresponding to user's retrieval conditions, and returns it with least role as service requester to UA. PCS and UDDI [10] repository will particularly be explain in section3.2. PCS authorizes the roles to UA and UA delegate the assertion to TA (*Task Agent*) of dependent level1. TA delegates the assertion to CA (*Collaboration Agent*) by request of collaboration agent. CA also delegates the assertion until lowest dependent agent by request and lowest dependent agent requests delegation assertion to authentication authority. In order to mange delegation, users are assisted by the authentication authority, which verifies delegation from one delegation agent to other agents [8]. *Authentication Authority* (AA) is authority to authenticate the agents. AA compares the highest level with the lowest assertion. AA issues *Delegation Assertion Ticket* (DAT) if assertion is valid. DAT structure will be explain in detail in section3.3.

The lowest level agent sends DAT to SA (*Service Agent*) through SSO. SA is a web service provider's agent verifying whether delegation assertion is valid and provides the contents to request.

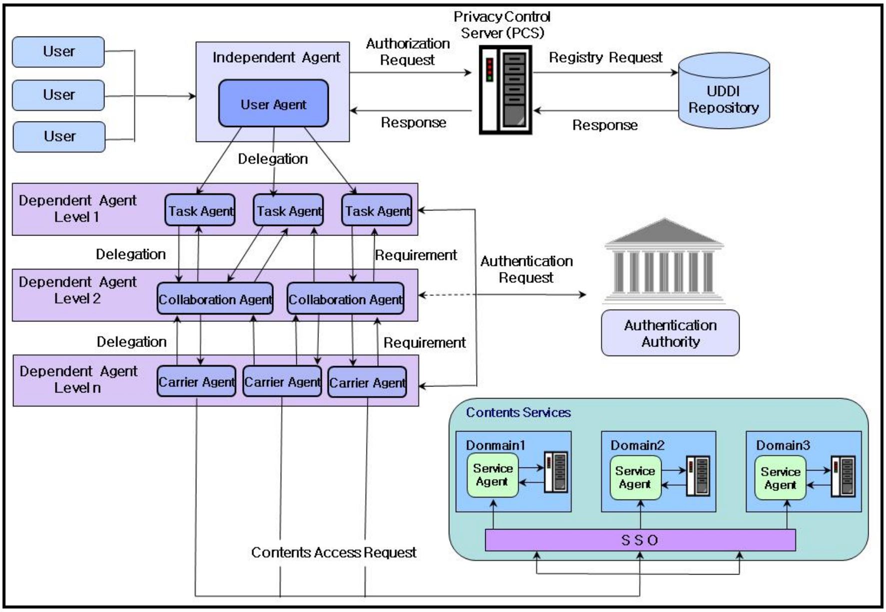


Fig. 1. Delegation Model based on Multi-Agent

3.2 Privacy Control Server

PCS(*Privacy Control Server*) is a system based on the XACML model [2, 14]. XACML is the result of OASIS standardization effort proposing an XML-based language to express and interchange access control policies. Fig2 is extended PCS architecture. PCS defined REA(*Role Enablement Authority*) to support at OASIS. REA means a module to apply RBAC(*Role Based Access Control*). REA assigns the suitable role to user and support necessary factor of role. PCS operates by the following steps. PAP(*Policy Administration Point*) writes policies and policy sets PAP makes them available to the PDP(*Policy Decision Point*) that evaluates the policy.

The access requester sends a request for access to the PEP(*Policy Enforcement Point*). The PEP sends the request for access to the context handler in its native request format. The context handler constructs an XACML request context and sends it to the PDP. Context handler translates the access requests from a native format into a canonical format. The PDP requests any additional subject, resource, action, environment and service attributes from the context handler. The context handler requests the attributes from REA. REA assigns the necessary role that adapts the services retrieved from the UDDI registry according to the retrieval condition of user. UDDI is a platform-independent, XML-based registry for businesses worldwide to list themselves on the Internet. UDDI is an open industry initiative, enabling businesses to publish service listing and discover each other and define how the services or software applications interact over the internet. REA returns the requested attributes to the context handler. The context handler sends the requested attributes and the PDP evaluates the policy. The PDP returns the response context to the context handler. The

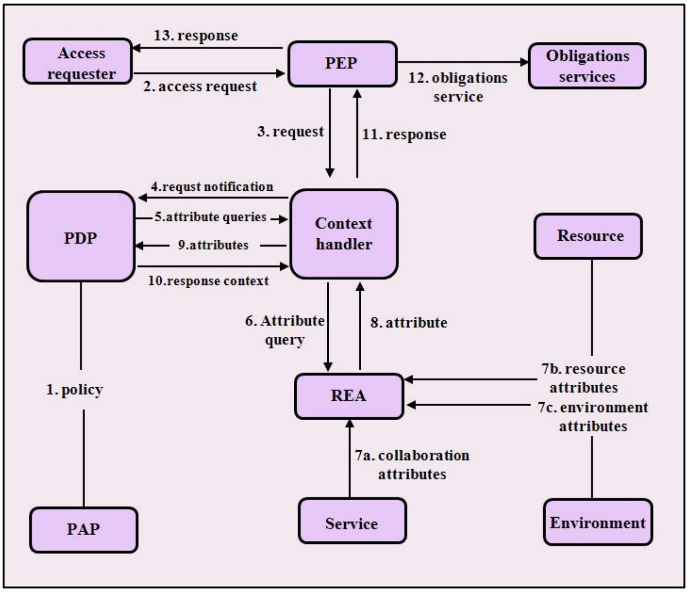
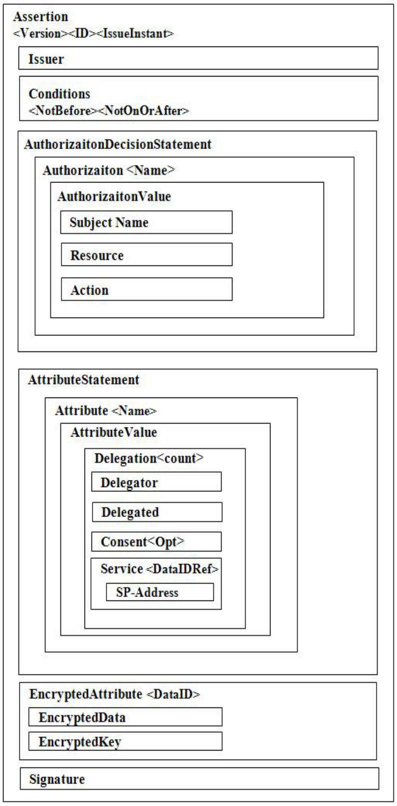


Fig. 2. The extended XACML Architecture

context handler translates the response context to the native response format of the PEP. The context handler returns the response to the PEP. The PEP fulfills the obligations. If access is permitted, the PEP permits access to the resource. Otherwise, it denies access.

3.3 Specification of Delegation Assertion

A SAML specification defines three different types of assertion. However, the schema for delegation is not directly defined by the SAML specifications. Fortunately, by exploiting SAML schema extensibility, we can create our delegation assertion. Our model extends the assertion in order to encode information such as delegation, subject, and input parameters of PCS. The elements and attributes of delegation assertion are described in Fig3 and Fig4 shows delegation assertion exam. Issuer element is the issuer of delegation assertion, the UA. Conditions element indicates the valid duration of delegation assertion using *NotBefore* and *NotOnOrAfter* attributes. In *AuthorizationDecisionSatatement*, *Subject Name* element means the requesting agent name to PCS. *Resource* is the service factor for which authorization is requested. *Action* element is the privilege for which authorization is requested. *Count* attribute is the number of delegation. *Delegation* element indicates whether the agent that receives this delegation assertion, is allowed to delegate the right to another agent. *Consent* element indicates whether the TA obtains of UA's consent to issue delegation assertion. If this is "true", consent has been obtained. *Service* element refers to the *DataIDRef* attribute. *SP-Address* element is the URL of the web service provider. *EncryptedAttribute* element encrypts as public key of agent/authentication authority to request the




```
<Assertion ID="_a75adf55-010d-dadsl42-tcbadb434d"
          IssueInstant="2008-11-05T02:46:02Z" Version="2.0">
 <Issuer>http://www.delegation-authority.com</issuer>
 <Conditions NotBefore="2008-11-05T02:46:02Z"
            NotOnOrAfter="2008-11-05T02:55:00Z">
 <AuthorizationDecisionStatement>
   <Authorization Name="PrivilInfo1">
     <Subject resource ="Hotel Reservation">
        <Name>Use Agent</Name>
        <Action>Read</Action>
 </AuthorizationDecisionStatement>
 <AttributeStatement>
   <Attribute Name="DeleInfo1">
     <AttributeValue>
       <Delgation count="2">
         <Delegator>User Agent</Delegator>
         <Delegated>Task Agent</Delegated>
         <Consent>true</Consent>
         <Service DataIDRef="Serv1">
           <SP-Address>http://www.SP1.com</SP-Address>
         </Service>
         <Service DataIDRef="Serv2">
           <SP-Address>http://www.SP2.com</SP-Address>
         </Service>
       </Services>
     </AttributeValue>
   </Attribute>
   <EncryptedAttribute DataID="Agent1">
     <xenc:EncryptedData> ... </xenc:EncryptedData>
     <xenc:EncryptedKey> ...</xenc:EncryptedKey>
   </EncryptedAttribute>
   <EncryptedAttribute DataID="Serv1">
     <xenc:EncryptedData> ... </xenc:EncryptedData>
     <xenc:EncryptedKey> ... </xenc:EncryptedKey>
   </EncryptedAttribute>
 </AttributeStatement>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
     digital signature of delegation authority </ds:Signature>
</Assertion>
```

Element < > Attribute

Fig. 3. Delegation Assertion Structure **Fig. 4.** Delegation Assertion Exam

assertion. Request agent receives the assertion and decrypts the assertion as their private key. In this paper, we do not explain how to process *encryptedData* and *EncryptedKey* elements. However, this can be processed using the technique described in the XML Encryption Specification [13]. *Signature* element is the digital signature of the UA to support integrity of delegation assertion.

3.4 Delegation Assertion Algorithm

We clear proposed model as algorithm1, 2. In algorithm1, UA request the privilege to PCS. PCS retrieves service information from UDDI registry corresponding to UA's retrieval conditions. PCS return least role to UA and issues the assertion. UA encrypts the assertion as public key of the highest level TA and delegates the assertion to TA. TA delegates the assertion until the lowest level. Agent of lowest level requests verification to trust the assertion to authentication authority (AA). AA compares the highest level with the lowest assertion and issues delegation assertion ticket (DAT). Agents are able to delegate their rights in order to improve the performance of service usage. For example, suppose Alice, she wants to book a flight to Seoul, also wants to

Algorithm 1. Delegation Assertion Ticket Issue algorithm

```
// Agent denotes agent containing User Agent (UA)
// Configuration denotes Privacy Control Server(PCS), Authentication Authority(AA)
// DA: Delegation Assertion and DAT: Delegation Assertion Ticket
// PKI: The Public Key of '!' and PKI!- : The Private Key of '!'
// PCS(r) returns the set of all permissions
// UDDI(c) returns the set all contents

Input: UA's Credential Information
Output: DAT/Reject

UA request the privilege to PCS
For each r_1 ∈PCS(r_i) and c_1 ∈UDDI(c_i)
  If (PCS r_i∩UDDI(c_i) ≠∅) then
   Return Service Information (r_i,c_i)
     For Augment(name_x, resource_y, action_z) ∈UA(Augment)
       If (UA's Augment∩PCS(r_i)=∅) then
        Return Reject
       Else
        Return PCS authorizes the Privilege to UA
          For D_i ← Seniormost Dependent Level
           D_j ← Juniormost Dependent Level
           UA delegate DA to D_i and encrypts DA using PK_Di
           Repeat D_i delegate DA until Juniormost Dependent
           D_j request the authentication to AA
             If mapping(D_i,D_j) = True then
               AA issue DAT signed PK_AA-
             Else
               Return Reject
```

Algorithm 2. Delegation Assertion Verification Algorithm

```
Input: DAT
Output: Success/Fail

Service Provider(SP) gets the DAT from SA.
SP confirms the signature of DA in DAT using PK_DA
If (Signature is not valid) then
   Return Fail
Else
   Decrypts encrypted input data using PK_sp-
   Confirms input data and process it
     If (Result value is successful) then
        Return Success
     Else
        Return Fail
```

hold a hotel reservation for a business trip. Alice will first delegate her right to her agent, UA that performs the task on her behalf. UA delegates its rights to other agents as many as the number of requesting services in parallel. This is more efficient than delegation mechanism such as SPKI/SDSI [15] because SPKI/SDSI must configure an ordered certificate chain to verify the validity of delegation. It results in increasing time complexity on the verifying side. However, in algorithm2, the web service provider is not required to configure a delegation chain, because the issue of DAT is managed by AA. The web service provider only requires DAT for verification. Each result of service is directly transferred to the carrier agent (CA).

4 Experimental Validation

4.1 Security Analysis

We analyze a proposed framework with respect to security analysis as following:

• **Integrity:** integrity mean that assets can be modified only by authorized parties or only in authorized ways. In the proposed framework, delegation assertion is only manipulated by AA, and contains digital signature value as the content of signature element. Signature element also contains information such as digest value and method in order to support the integrity of delegation assertion. We ensure integrity of delegation assertion using these digest information.

• **Replay attack:** A replay attack is a form of network attack in which a valid data transmission is maliciously repeated or delayed by an adversary who intercepts the data. An attacker wants to impersonate the legal request to web service provider using a stolen delegation assertion from legal agent. However, it is impossible because delegation assertion contains digest information in order to prevent fabrication, and is valid during the period defined in *NotBefore* and *NotOnOrAfter* attributes of the condition element.

• **User's privacy:** In the pervasive computing environments, to protect user's privacy is very important because an attacker may expose information related user's privacy to the outside without authorization. In the proposed framework we ensure user's privacy by means of XACML/SAML-based delegation model.

4.2 Experimental Evaluation

The main goal of the experiments was to investigate the performance of our approach in terms of efficiency and scalability. We report experimental results on the security estimation and execution time. The experiments were performed on a 2.40GHz Intel dual core processor machine with 2 GB of RAM. The operating system on the machine was Microsoft Windows 2003 Server and the implementation was built and run in Java 2 Platform, Standard Edition 5.0. In fig5, we specify the number of users as each 10, 20, 30, 40, 80 and 120 and measured verification time of DAT. Delegation chain and proposed technique measured randomly delegation frequency as well as the number of users. In addition to, factor of proposed model is agent's tree height 7 and

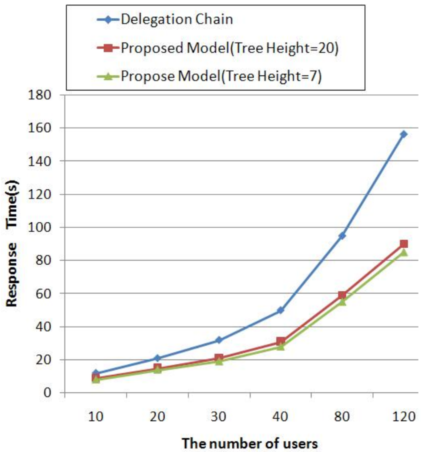


Fig. 5. Verification Time

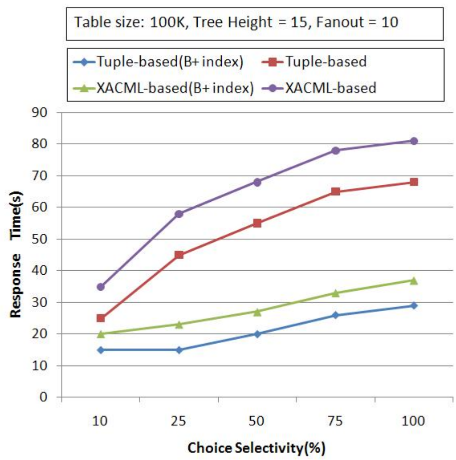


Fig. 6. Selectivity Performance

20. In case of delegation chain, verification time from 10 to 120 people measured each 9.8 and 156.6 second. DAT verification time is average 1.2 second and shows the continuing increase. This result increased the number of SAML token because DAT(*Delegation Assertion Ticket*) increase as well as number of users. But proposed method measured verification time as average 0.7(TH=7) and 0.8(TH=20) second. This point decreased time complexity because our mechanism verifies only once DAT validation like above mention. SAML can encrypt the assertion as full or part. In future, we will bring advanced verification time.

For the second, when agent queries to PCS(*Privacy Control Server*), PCS check in conformity with express policy. Policy compliance check seriously doesn't influenced response time because this conducts bitwise-AND operation in encoding hierarchy. However, tuple-based or XACML-based must search policy purpose with access of real data. This is factor that delays response time. Well-known technique to solve this problem consists of index. Tuple-based or XACML-based that index does not exist causes cost much because of full-scan method. We measured the response time to retrieve the selected data into PCS's policy. Fig6 shows the response time of selectivity against policy search with various labeling schemes. While queries increase, search to apply B+ tree index shows better performance than other labeling.

5 Conclusion

Pervasive environments mean open system so that the convenient life to the users. They require the system that processes the task work instead of the users. At this time, the user must delegate user's privilege to the agents. We proposed a secure delegation model based on multi-agent in pervasive environments. Firstly, we presented a delegation model which is managed by multi-agent with assistance of authentication authority. Secondly, we have described a delegation assertion without any privacy information disclosure. Also this paper exploited a XML-based encryption and digital signature from unrelated agents. Thirdly, we have presented an efficient delegation mechanism without delegation chain. Our model has only required a delegation assertion because delegation assertion ticket is issued by authentication authority having trust relationship with web service providers. We will continually research more advance model about the delegation topic in pervasive environments.

References

1. OASIS, Profile for the OASIS Security Assertion Language (SAML) V2.0 OASIS Standard (March 15, 2005)
2. OASIS, eXtensible Access Control Markup Language (XACML) V2.0 OASIS Standard (February 1, 2005)
3. Weiser, M.: Scientific American. The Computer for the 21st Century, 94–104 (September 1991); reprinted in IEEE Pervasive Computing, 19–25 (March 2002)
4. Syukur, E., Loke, S.W., Stanski, P.: A Policy Based Framework for Context Aware Ubiquitous Services. In: Yang, L.T., Guo, M., Gao, G.R., Jha, N.K. (eds.) EUC 2004. LNCS, vol. 3207, pp. 346–355. Springer, Heidelberg (2004)
5. Sandhu, R.S., Cynek, E.J., Fensteink, H.L., Youmank, C.E.: Role Based Access Control Model. IEEE Computer 29(2) (February 1996)
6. Navarro, G., Ortega-Ruiz, J.A., Ametller, J., Robles, S.: Distributed Authorization Framework for Mobile Agents. In: Magedanz, T., Karmouch, A., Pierre, S., Venieris, I.S. (eds.) MATA 2005. LNCS, vol. 3744, pp. 127–136. Springer, Heidelberg (2005)
7. Welch, V., Foster, I., Kesselman, C., Siebenlist, F.: X.509 Proxy Certificates for Dynamic Delegation. In: 3rd Annual PKI R&D Workshop (2004)
8. Gomi, H., Hatakeyama, M., Hosono, S., Fujita, S.: A Delegation Framework for Federated Identity Management. In: Proceedings of DIM workshop (2005)
9. Hu, Y.J.: Some thoughts on agent trust and delegation. In: Proceedings of the Fifth International Conference on Autonomous Agent (2001)
10. Dai, J., Steele, R.: UDDI Access Control. In: Proceedings of the Information Technology and Application (2005)
11. Navarro, G., Firozabadi, B.S., Rissanen, E., Borrell, J.: Constrain delegation in XML-based Access Control and Digital Rights Management Standards, Communication, Network and Information Security (2003)
12. Wang, J., Del Vecchio, D., Humphrey, M.: Exteding the Security Assertion Markup Language to Support Delegation for Web Services and Grid Services. In: IEEE International Conference on Web Services (2005)
13. XML Encryption Syntax and Proceeding, `http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/`
14. Ardagna, C.A., Damiani, E., De Capitani di Vimercati, S., Samarati, P.: XML-based Access Control Language, Elsevier Information Security Technical Report (2004)
15. Clark, D., Elien, J., Ellision, C., Fredette, M., Marcos, A., Rivest, R.: Certificate Chain Discovery in SPKI/SDSI. ACM Journal of Computer Security (2001)
16. XML Signature, `http://www.w3.org/TR/xmldsig-core`
17. Pfitzmann, B., Waidner, B.: Token-based web Single Sign-On with Enabled Client. IBM Research Report RZ 3458(93844) (November 2002)
18. OASIS, Web Service Security: SOAP Message Security1.0, OASIS Standard (March 2004)

3DE_var: Energy-Efficient Cluster-Based Routing Scheme in Wireless Sensor Networks

Jinsu Kim[1], Junghyun Lee[1], and Keewook Rim[2]

[1] Dept. of Computer Science Engineering, Inha University
[2] Dept. of Computer and Information Science, Sunmoon University, South Korea
{kjspace,jhlee}@inha.ac.kr, rim@sunmoon.ac.kr

Abstract. The traditional cluster-based routing method is a typical method for enhancing an energy efficiency, which selects cluster head in order to send the aggregated data arriving from the nodes in cluster to a sink. However, caused by the frequent information exchange between selected cluster head and nodes of cluster in setup phase of the cluster formation and direct transmission between isolated nodes which are excluded from the all clusters and sink in steady-state phase, it is necessary to minimize the unnecessary energy dissipation using the equal distribution of cluster heads. We propose an efficient cluster-based routing scheme, called 3DE_var. In order to configure the optimum distribution of cluster, it selects new cluster head using the information such as direction obtained from upper level cluster head, distance among nodes in the cluster, residual energy and density. Thus this study prolongs the network lifetime and provides equal opportunity for being cluster head.

Keywords: Wireless Sensor Networks, Cluster-based Routing, Cluster Head Selection, Sleep Mode.

1 Introduction

Wireless sensor networks (WSNs) consist of densely deployed sensor nodes, which have limited computational capabilities, power supply, and communication bandwidth. Sensor nodes which are battery powered recognize phenomena around a certain region and transmit the detected data to sink. Thus energy is the most precious resource of WSNs. Recently, various routing protocols have been proposed for WSNs. One of the methods to improve energy efficiency is a cluster-based routing method which reduces redundant transmissions of duplicate data using aggregation of data. The LEACH [1], one of such representative cluster-based routing methods, can select cluster head effectively per each round using a probability, while it can not guarantee uniform distribution of cluster heads per each round. On the other hand, though the LEACH-C [2] is based on cluster like the LEACH, the LEACH-C can guarantee uniform distribution of cluster heads using selection of cluster head under the control of sink. But it is NP-Hard Problem because it should send to sink the information of each node participating in sensor networks and find the sensor node,

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 314–324, 2009.

which has the shortest average distance, by the distance calculation for each node. To solve such NP-Hard Problem, though the LEACH-C selects cluster head effectively using Simulated Annealing Algorithm [3], the optimum value can not be guaranteed and the communication overhead is required to find the node's location. HEED [4] protocol is the excellent cluster selection method because it uses only each node's own factor. It can select cluster head using only the node's own factor without knowing the energy of all nodes. In addition, it can terminate after being performed during a certain repeat count. However when the residual energy level of nodes is similar, it offers no guarantee on the number of cluster heads because it is performed by 2 times of the selection probability value. It causes huge energy dissipation in the communication of the nodes due to repeated advertisement (ADV) messages transmission.

In this study, we propose a new protocol to minimize the energy dissipation by reducing load which is caused by communication between cluster head and nodes in each round, when the cluster head is selected in a real environment where the distribution of nodes is not uniform. Because sensor nodes around sensor node's location have high possibility to belong to same cluster, we use the information of previous cluster head and the information obtained from each node such as the distance of each node, the residual energy, the density and direction obtained from upper level cluster head, called 3DE_var.

2 3DE_var: Energy-Efficient Cluster-Based Routing Scheme

2.1 The Proposed Setup Phase Details

The detailed setup phase proposed in this paper lists as follows. First, the previous cluster head (CH_{old}) informs the information which selected new cluster head (CH_{new}) to the all nodes in the cluster. Second, the nodes determine their status using this information. That is, the nodes of the cluster which does not change go into sleep mode during a certain interval till the setup phase terminates. Otherwise, the nodes to be deleted prepare to select new cluster heads. Third, the selected cluster head broadcasts ADV message to the nodes, and newly added nodes transmit a join-request message (Join-REQ) message. Fourth, using the received information, it determines TDMA slot similar to the traditional method and makes out a list. It transmits the result to newly added nodes and terminates by preparing the steady-state phase.

This proposed setup phase method is applied to equally the nodes performing communication except the difference of number of them, because the unchanged nodes in the setup phase are in sleep mode till the setup phase terminates. In addition, unchanged nodes use the information such as TDMA slot and CDMA spreading codes which are used in the previous cluster and then it attempts to improve the energy efficiency by reuse as newly added nodes take over the information of the deleted nodes. Table 1 represents the numerical expressions on the necessary energy dissipation amounts in each step of the traditional setup phase and the proposed setup phase. N is the total number of node in the cluster, n is the total number of newly added nodes in the cluster, $E_{Tx}^{'}$ is to transmit the necessary information for succession from CH_{old} to CH_{new}.

Table 1. The comparison of energy dissipation amounts in each step of the setup phase

Step	Traditional Setup Phase ($Setup_{trad}$)	Proposed Setup Phase ($Setup_{prop}$)
1	$E_{Tx} + (N-1)E_{Rx}$	$E_{Tx} + (N-1)E_{Rx}$
2	$(N-1)E_{Tx} + (N-1)E_{Rx}$	$(N-n-1)E_{Sleep} + E'_{Tx} + E_{Tx} + nE_{Rx}$
3	$E_{Tx} + (N-1)E_{Rx}$	$nE_{Tx} + nE_{Rx}$
4	-	$E_{Tx} + nE_{Rx}$

2.2 Calculation on the Number of Unchanged Nodes for Optimum Energy Efficiency

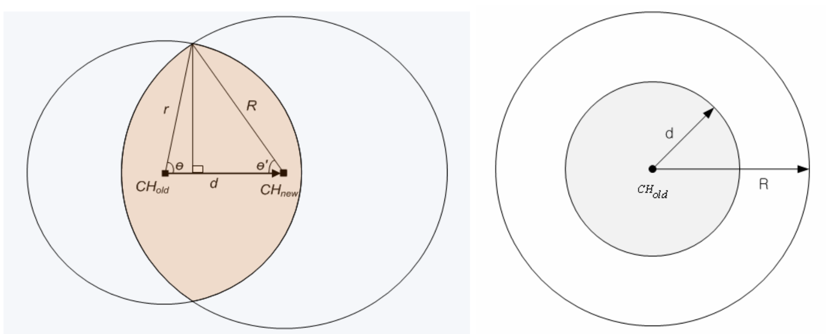
To efficiently consume energy using Table 1 in the setup phase of the traditional method and the proposed method, it is attempted to improve energy efficiency using the changed distance among cluster heads (CH_{old} and CH_{new}) and the number of unchanged nodes independent on the changed distance. That is, it is the case in which the total energy dissipation in the setup phase of the proposed method ($\sum_{step=1}^{4} Setup_{prop(step)}$) is as much as that of in the setup phase of the traditional method ($\sum_{step=1}^{3} Setup_{trad(step)}$), or smaller than that (namely, $\sum_{step=1}^{3} Setup_{trad(step)} \geq \sum_{step=1}^{4} Setup_{prop(step)}$). It can be shown as following equations (1).

$$\frac{(N-n-3)E_{Tx}}{(3n-2N+2)E_{Rx}} + \frac{(n-N+1)E_{sleep}}{(3n-2N+2)E_{Rx}} \geq 1 \quad . \tag{1}$$

where n is the number of new added nodes in the cluster and N is the number of nodes in the cluster, the difference ($E_{Tx} - E_{Rx} = l\varepsilon_{fs} d^2$) between E_{Tx} and E_{Rx} [5] is converted as $E_{Tx} / E_{Rx} \approx 1$, because the number of bits in communication message is not large due to the short distance ($d < d_0$) in the same cluster, and the number of newly added nodes is a constant. In addition, it is shown as following equation (2), when $E_{Sleep} / E_{Rx} \approx 0$ is applied to the equation (1), because the energy dissipation in E_{sleep} is very small compared to E_{Rx} and the number of overlapped nodes ($m = N - n$).

$$m \geq \left\lceil \frac{N+5}{4} \right\rceil \quad . \tag{2}$$

For example, if the number of nodes (N) in the cluster is 20, and the number of unchanged nodes (m) is 7 or more, it means the proposed method in this paper can improve the energy efficiency. The value of m is determined dynamically by the number of nodes in the cluster because the number of nodes in each cluster can be changed in every round over time.



(a) (b)

Fig. 1. (a) The overlapped area according to the cluster head changed from CH_{old} to CH_{new} (b) The effective zone of distance (d) over the cluster radius (R) and the number of nodes in cluster

2.3 Calculation on the Number of Nodes in the Overlapped Area

Whenever a round is changed, the networks should reform cluster through the setup phase. Figure 1 (a) shows the overlapped area which is not changed, when CH_{old} is reformed as CH_{new}. If CH_{old} and CH_{new} have a different radius as r and R respectively, the overlapped area presents the area of nodes of previous cluster which is not changed into new cluster. The overlapped area is calculated as following equation (3). Figure 1 (b) shows the effective zone or effective distance which means that selected nodes as cluster head is efficient in cluster, that is, it will improve energy efficiency.

$$
\begin{aligned}
&Area(r,R,d) = \\
&\left(acos\left(\frac{r^2 - R^2 + d^2}{2dr}\right) r^2 - \frac{(R^2 - r^2 - d^2)\, sin\left(acos\left(\frac{R^2 - r^2 - d^2}{2dr}\right)\right) r}{2d} \right) + \\
&\left(acos\left(\frac{3d^2 - R^2 + r^2}{2dR}\right) R^2 - \frac{(3d^2 - R^2 + r^2)\, sin\left(acos\left(\frac{3d^2 - R^2 + r^2}{2dR}\right)\right) R}{2d} \right)
\end{aligned}
\quad (3)
$$

where, θ and θ' represent degree, d is a distance between CH_{old} and CH_{new}, and both r and R represent radius of cluster head which is assumed that radius of CH_{new} and that of CH_{old} are different. This assumption is suggested to solve the case where nodes are not joined in the cluster, because the number of alive nodes is decreased as time goes by. So these excluded nodes transmit direct to the BS. If it is assumed that the distribution of nodes in the networks is uniform, and the overlapped area calculated by equation (3) is divided by average area ($M^2 / Node_{total}$) where a node is able to be distributed in the networks, the number of nodes distributed in overlapped area is same as the following equation (4).

$$Node_{overlap} = \frac{Node_{total} \times Area(r, R, d)}{M^2} . \tag{4}$$

where M^2 represents the whole area of networks, $Node_{total}$ means the total number of nodes distributed in the networks, and the calculated $Node_{total}$ is compared to m obtained from equation (2).

2.4 Determination of Density and Cost of Nodes in the Cluster

In this paper, the concept of density was used to determinate the cost factor which affects the cluster head selection. The density is dependent on the distance between each node in the cluster and the cluster head. It is calculated by the number of neighbor nodes ($Node_{neighbor}$) in the same cluster and the number of foreign nodes ($Node_{foreign}$) located in other clusters using a certain radius r and R from each node from each node. This paper defines the density as following equation (5).

$$Density_i = \frac{Node_{neighbor}}{Node_{neighbor} + Node_{foreign}} . \tag{5}$$

where i is a node in the cluster, the range of $Density_i$ is normalized between 0 and 1. As time goes by, the nodes consume their energy and finally finish their life cycle. The number of alive nodes ($Node_{alive}$) in the networks decrease and the number of dead nodes ($Node_{dead}$) increase and then the accuracy of density is weakened. However, like the traditional clustering method, the number of alive nodes decreases rapidly from the point of time when that number pass over a certain threshold after dead node's occurrence over time, and then the network lifetime terminates. When the number of alive nodes in the cluster and the residual energy is beyond the threshold respectively, the density for energy efficiency is used by applying the information of each node and the information of CH_{old} to equation (4) and equation (5).

To select cluster head in setup phase of one round, the equal opportunity should be provided to the node in the whole network and use cost to find a node for decrease of energy dissipation. The cost is the information transmitted from each node to cluster head. As for LEACH-C, it is determined using node having the minimum total of the distance between cluster head and nodes among the nodes with beyond average energy. In this paper, it is determined by density and residual energy as main factors which affect cost. Among the information obtained by CH_{old}, the nodes with beyond average energy in the cluster are selected and *CandidateList(see Figure 2)* is made of them. The *CandidateList* is generated by selecting the nodes of which $Density_i$ is beyond the threshold. The threshold is a minimum value (m) which is obtained by equation (2).

2.5 3DE_var Algorithm for Effective Cluster Head Selection

In this paper, the set of cluster head candidates is composed using the information (distance, residual energy, density, cost, and so on) obtained from the nodes to select effective cluster head and the direction information obtained from upper level cluster head. The direction is one of factors on cluster selection determination, which refers to the direction information of superior cluster head (*SCH*) when CH_{old} selects CH_{new} in the hierarchical cluster environment. The reason to refer to superior direction is to minimize the number of nodes which do not included in the clusters and to minimize the overlapped area between clusters. When previous superior cluster head (SCH_{old}) determines new superior cluster head (SCH_{new}), the determined direction information is broadcasted to cluster heads in the networks, and the cluster head receiving the direction information generates list using candidates with the most similar direction and lower cost among the cluster head candidates newly being selected. To provide equal opportunity to every node in the cluster, the node with the lowest cost is not selected, but the node is selected randomly using random function. The reason to select randomly is because if the node with the lowest cost is selected, the possibility of unbalanced selection can be increased and then the possibility of selection of nodes located outer side of cluster is decreased. Therefore only the nodes located in specific area (center of cluster) are selected and locality is occurred. Finally the nodes located far from the center of the cluster generate dead nodes rapidly like MTE (Minimum Transmission Energy) routing protocol [1].

Figure 2 shows 3DE_var algorithm for effective cluster head selection suggested in this paper. In Figure 2, k means k th cluster, and the *num_Cluster(* k *)* means the number of nodes in k th cluster. The *Vec()* converts the node in the cluster and CH_{old} into vector. In addition it returns vector from SCH_{old} and SCH_{new}, and then they are used as two parameters of *Sim()*. The *Sim()* calculates the angle (θ) between two

```
num_Candidate = 0;
cnt = 0;
CandidateList = ∅;
CostList = ∅;
for(i=0; i<num_Cluster(k); i++){
 vecA = Vec( Node_i , CH_old );
 vecB = Vec( SCH_old , SCH_new );
 if(Sim(vecA,vecB)) >= 0)
  if( Energy( Node_i ) >= AvgEnergy(k))
    CandidateList[num_Candidate++] += Node_i ;
}
for(i=0; i<num_Candidate; i++)
 if(numOverlap( CH_old ,CandidateList[i]) >= m )
    CostList[cnt++] += CandidateList[i];
if( cnt > 0){
  ClusterHead = CostList[ rand() % cnt ];
  ProposedSetup();
  if( cnt > δ )
    R = R - α ;
}else {
 R = R + α ;
 TraditionalSetup();
}
```

Fig. 2. 3DE_var algorithm for effective cluster head selection

vectors, and measures the similarity of direction of two vectors. The included angle (θ) between two vectors is obtained as $cos(\theta) = AB / |A||B|$.

In this paper, the threshold of angle, θ between two vectors is set as $\pm 90°$ (namely, $0 \leq \cos(\theta) \leq 1$) based *VecB* vector. The *AvgEnergy()* is a function to calculate the average residual energy of k th cluster. The *Energy()* returns residual energy of each node. The *numOverlap()* returns the number of nodes located in the overlapped area between cluster head and cluster head candidates. m is a minimum value calculated by equation (2). If *CostList* is an empty set, the traditional setup phase is performed. If *CostList* is not an empty set, after select of cluster head using random number generation less than *cnt*, the algorithm proposed in 2.1 to perform the setup phase is conducted.

3 Performance Evaluation

NS-2 [6] network simulator was used in order to evaluate compare the performance of the algorithm proposed in this paper with traditional clustering techniques. Table 2 shows the parameters used in the simulation environment and their corresponding values were presented.

Table 2. The simulation parameters and values

Parameters	Values
Network grid	(0m,0m) ~ (100m,100m)
Sink	(50m, 175m)
Threshold distance(d_0)	87m
Initial cluster radius	25m
E_{elec}	$50 nJ / bit$
ε_{fs}	$10 pJ / bit / m^2$
ε_{mp}	$0.0013 pJ / bit / m^4$
$\varepsilon_{aggregation}$	$5 nJ / bit / signal$
Data packet size	500 Bytes
Broadcast packet size	25 Bytes
Packet header size	25 Bytes
Initial energy	2J/Battery
Number of nodes	100
Number of clusters	5

where d_0 is the threshold distance calculated by $d_0 = \sqrt{\varepsilon_{fs} / \varepsilon_{mp}}$, and each round lasts for 20 seconds.

Figure 3 shows the average number of unchanged nodes counted in the process of selecting the cluster head of LEACH-C from 10 test sets with 100 nodes generated at random, and the cumulative number of nodes was used to optimize energy dissipation. In Figure 3, the average number of nodes changed little for a period and then decreased sharply from 420 seconds when the number of dead nodes increased

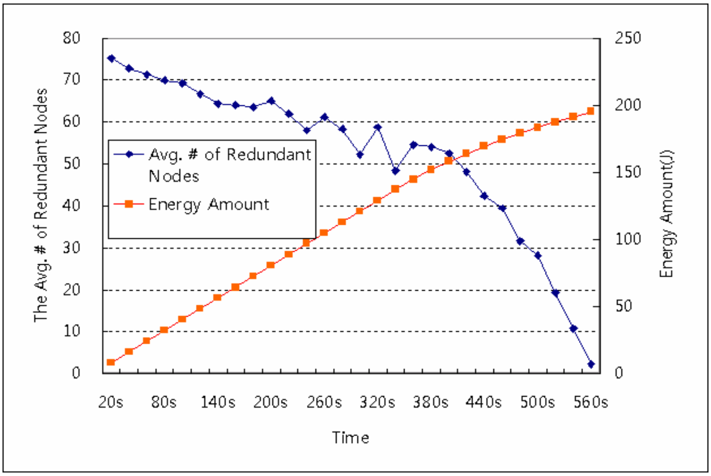


Fig. 3. The average number of redundant nodes and total energy dissipation amount over time

rapidly, and the average energy dissipation at the moment was around 1.64J. Accordingly, this paper sets the threshold to 0.8(1.64J/2J) in the calculation of density using the number of alive nodes and the number of dead nodes explained in 2.4.

Figure 4 is a graph that compares the number of alive nodes changing over time. As in the figure, the points of time when the dead nodes began to appear in LEACH, LEACH-C, HEED, 2DE (except direction information of 3DE_fix), 3DE_fix (fix a radius), and 3DE_var were 390, 360, 380, 380, 380, and 385 seconds, respectively, and the points of them when the last alive node occurred were 547, 580, 590, 613, 619, and 629 seconds. Compared to LEACH, LEACH-C, HEED, 2DE, and 3DE_fix,

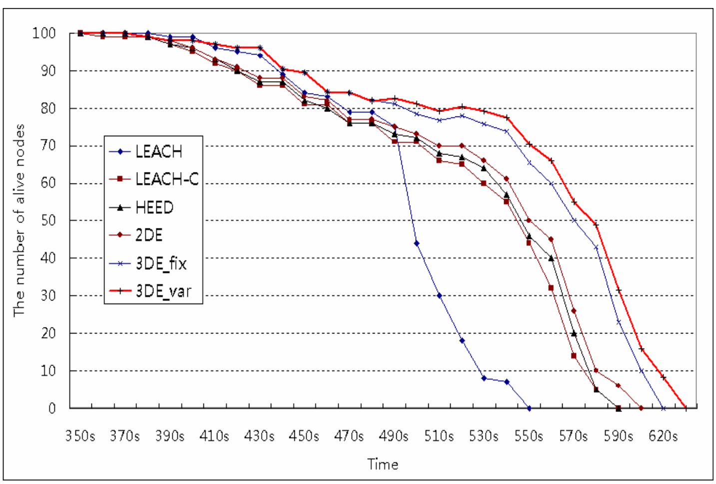


Fig. 4. Comparison with the number of alive nodes over time

the lifetime of the entire sensor networks is increased by around 15.0%, 8.4%, 6.6%, 2.6%, and 1.6%, respectively. Compared to LEACH or LEACH-C, energy dissipation in setting the nodes was reduced and, as a result, overall energy efficiency was enhanced. In addition, HEED wasted energy as it transmitted information through frequent exchange of messages with neighbor nodes for selecting the cluster head, and 3DE_var proposed in this paper used energy efficiently as the sensor network minimized the number of isolated sensor nodes by maintaining balance using direction information and, different from HEED, counted the number of neighbour nodes according to the distance. Although 3DE_var had the occurrence of the dead nodes earlier than LEACH due to the load of information exchange with BS or within the cluster, it extended the lifetime of the network by using energy efficiently through reducing the accumulation of redundant nodes and the number of isolated nodes as it used an adaptive size of radius over time.

Figure 5 and Figure 6 show the number of redundant nodes and isolated nodes over cluster radius from 25m to 75m, respectively. In 3DE_var, α is 1 and the radius R's range varies adaptively from 25m to 85m (d_0). As shown Figure 5, 3DE_var reuses the redundant nodes' information from initial radius (25m) as adaptively varies the radius, thus prolongs the network lifetime. However the others except LEACH and LEACH-C reuse from radius 45m. As shown Figure 6, there are many isolated nodes by radius 35m. The isolated nodes excluded the cluster send the information to sink directly, so these nodes usually died more rapidly than joined nodes in the cluster.

Table 3 shows the threshold of effective distance (d) between cluster radius (if r equals R) and the number of nodes in cluster. These thresholds are calculated from equation (3) and equation (6) as $m \geq Node_{overlap}$.

In the setup phase of each round, it takes time for CH_{old} to send information on deleted nodes to CH_{new} , and for CH_{new} to resend ADV messages to new nodes to be

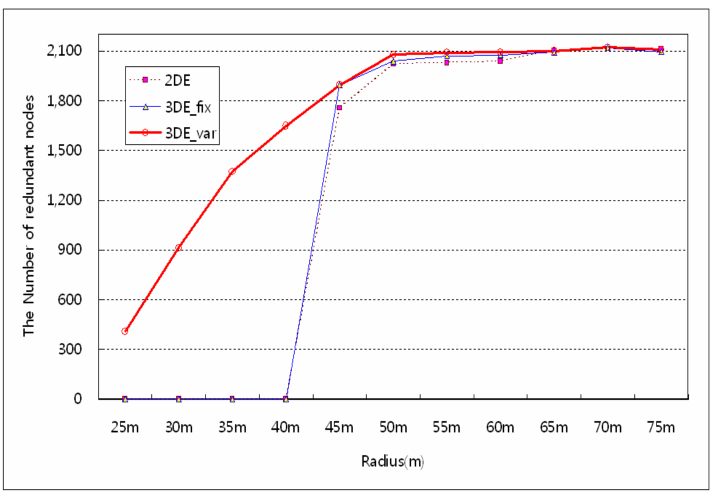


Fig. 5. Comparison with the number of redundant nodes over the cluster radius

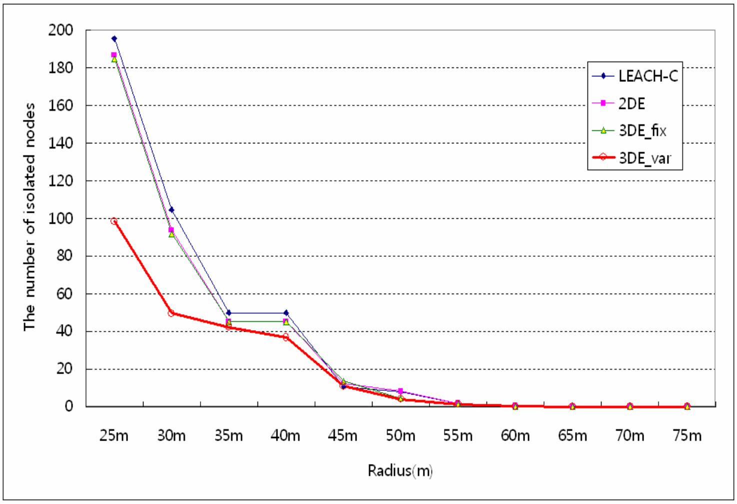


Fig. 6. Comparison with the number of isolated nodes from the whole clusters over the cluster radius

Table 3. The effective distance (d) between Radius (R) and the number of Nodes (N) in cluster

R / N	30m	31m	32m	33m	34m	35m	...	45m
20	0 m	0 m	1 m	4 m	7 m	10 m	...	37 m
30	0 m	1 m	4 m	7 m	10 m	13 m	...	39 m
40	0 m	3 m	6 m	9 m	12 m	14 m	...	40 m
50	1 m	4 m	7 m	10 m	13 m	15 m	...	41 m
60	1 m	5 m	7 m	10 m	13 m	16 m	...	42 m
70	2 m	5 m	8 m	11 m	14 m	16 m	...	42 m
80	2 m	5 m	8 m	11 m	14 m	17 m	...	42 m
90	3 m	6 m	9 m	11 m	14 m	17 m	...	43 m
100	3 m	6 m	9 m	12 m	14 m	17 m	...	43 m

added. Different from traditional clustering methods, however, the proposed method saves processing time as only newly added nodes, instead of all the nodes in the cluster, sends Join-REQ to the cluster head, and does not increase complexity such as long delay time compared to traditional methods because the setup time for cluster formation is very short compared to that in each round. Energy dissipation is reduced because the number of redundant nodes used in cluster formation is counted, and based on the number, whether to use the proposed method or the traditional method is determined by the 3DE_var algorithm alternatively.

4 Conclusions and Future Works

This paper proposed a method of choosing the setup phase to be executed in each round according to situation and selecting the cluster head in consideration of direction information for distributing nodes uniformly to each cluster so that nodes in the network are not isolated, and maximized energy efficiency when a new cluster head had been selected by minimizing unnecessary exchange of information on redundant nodes between the previous cluster head and the newly selected cluster head in each round.

A task for future research is how to solve the problem of extended clustering time in the setup phase as the previous cluster head collects information on nodes to be deleted from the cluster and sends it to the newly selected cluster head. Accordingly, it is necessary to minimize the delay in the setup phase and to overcome constraints such as the use of GPS. Furthermore, we need to design a cluster head selection method in consideration of mobility among nodes within a network.

Acknowledgement

"This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) Support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2009-C1090-0902-0020).

References

1. Heinzelman, W.R., Chandrakasan, A.P., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: Proc. 33rd Hawaii Int'l. Conf. Sys. Sci. (2000)
2. Heinzelman, W.R., Chandrakasan, A.P., Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Trans. Wireless Communication, 660–670 (2002)
3. Murata, T., Ishibuchi, H.: Performance evaluation of genetic algorithms for flowshop scheduling problems. In: Proc. 1st IEEE Conf. Evolutionary Computation, pp. 812–817 (1994)
4. Younis, O., Fahmy, S.: Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach. In: IEEE INFOCOM 2004 (2004)
5. Rappaport, T.: Wireless Communications: Principles & Practice. Prentice-Hall, Englewood Cliffs (2000)
6. NS-2. The Network Simulator ns-2.32 release, http://www.isi.edu/nsnam/ns/

A Routing Agent Using Cross-Layer Method for Collision Avoidance in Ad Hoc Networks

Zhipeng Liu, Chan-Heum Park, Bokman Lee, and Chonggun Kim*

Dept. Of Computer Engineering, Yeungnam University
Gyeongsan, Gyeongbuk, Korea
zhipeng@ynu.ac.kr, shem@yu.ac.kr, neo102030@ynu.ac.kr, cgkim@yu.ac.kr

Abstract. The cross-layer design approach is an important concept in ad-hoc networks which is adopted to solve several open issues. It aims to overcome MANET performance problems by allowing protocols belonging to different layers to cooperate and share network status information while still maintaining separated layers. In this work, we provide a routing agent of cross layer method which is based on the cooperation between Adjacency matrix routing protocol and the IEEE 802.11e MAC protocol. This proposal aims to avoid the congested nodes and guarantee better performance in terms of delay, packet delivery ratio, and so on. In the proposed method, each node maintains an adjacency matrix representing the network topology and MAC conditions with neighbors. The number of retransmissions in MAC layer is introduced as an indication of congestion. When the congestion occurs, the source node works as the routing agent to detect new route by checking MAC information and the adjacency matrix to the destination. The simulation results show that Cross layer give better performance than network layer only methods, especially in the frequent congestion condition.

Keywords: Ad hoc networks, Cross-layer design, Adjacency matrix routing protocol, Dijkstra algorithm, Quality of service.

1 Introduction

Ad hoc networks are defined as the category of wireless networks that utilize multi-hop radio relaying and are capable of operating without the support of any fixed infrastructure [1]. In multihop ad hoc networks, every node acts also as a router and forwards each others' packets to enable the communication between nodes by wireless links. The nodes composing a MANET can freely move and organize themselves arbitrarily: thus, the topology of the network maybe highly dynamic and unpredictable.

Traditional packet-based network architectures assume that communication functions are organized into protocol layers, and that the metadata controlling the packet delivery are injected into protocol headers, one for each protocol layer. Unfortunately,

* Corresponding author.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 325–334, 2009.

the layered open system architecture (OSI) does not seem to remarkably success for ad hoc networks [2]. The strict layering approach reveals to be suboptimal in many application domains of MANETs. The main drawback of the ISO/OSI model is the lack of cooperation among adjacent layers: each layer works in isolation with little information about the network. Moreover, the strict modularity does not allow design joint solutions to the overall network performance [3].

Cross-layer design is an emerging method to support flexible layer approaches in MANETs. Generally, cross-layer approaches attempt to exploit more interactions among layers to achieve better performance [4]. In this paper, we provide a cross layer method which is based on the cooperation between Adjacency matrix routing protocol which is network layer protocol and the IEEE 802.11e MAC protocol. This proposal aims to avoid the link congestion among nodes and guarantee the performance of sensor networks in terms of delay, packet delivery ratio, etc. According to the proposed method, the number of retransmissions in MAC layer is introduced as an indication of congestion. When the collision occurs, the source node can detect that by checking the retransmissions in MAC layer and re-calculate a new route using the adjacency matrix to transfer packets. The simulation results show that our proposal improves the performance, especially in the congestion condition.

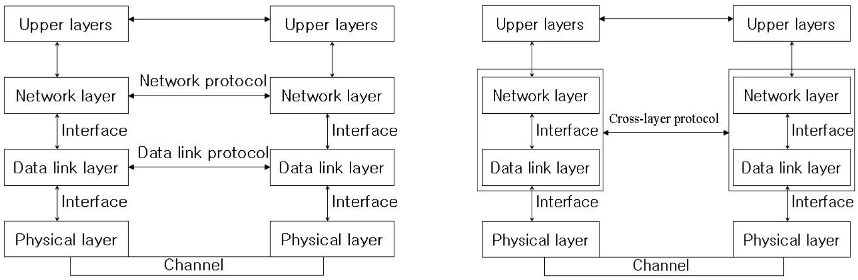
The remainder of this paper is organized as follows. Section 2 introduces the background of the work, including Overview of Adjacency Matrix Routing Protocol and Retransmission mechanism of 802.11 MAC. In the section 3, we provide a description of proposed collision avoidance mechanism. Section 4 presents the simulation scenario in NS2 and performance evaluation. Finally, the conclusion is given in section 5.

2 Background

In our cross-layer scheme, the MAC layer knowledge of the wireless medium around the node is shared with the network layer information for routing. In this section, we briefly describe the definition Cross-layer design, network layer routing protocol and MAC layer protocol in our simulations.

2.1 Definition of Cross-Layer Design

Layered communication approaches typically separate communication tasks into several layers, with a clear definition of the functionality of each layer. In a layered communication stack, interaction among layers occurs through well defined standardized interfaces that connect only the neighboring layers in the stack. Figure 1(a) depicts the traditional layer interaction through standardized interfaces in a strictly layered communication stack. In contrast, cross-layer approaches attempt to exploit a richer interaction among communication layers to achieve performance gains. Several definitions of cross-layer design have been proposed in [11], [12]. Figure 1(b) illustrates the cross-layer routing protocol structure for ad hoc networks. In most cases, cross-layer design is referred to both as a general protocol design methodology as well as a mean to identify protocols designed with this approach.



(a) General protocol structure **(b)** Cross-layer protocol structure

Fig. 1. General and Cross-layer protocol structure of sensor nodes

2.2 Overview of Adjacency Matrix Routing Protocol

In adjacency matrix based routing protocol, the topology agent is the initiator for topology discovery process. In the topology discovery process, the topology agent broadcasts a query message to all neighbouring nodes to collect information about the network topology. The nodes which receive a topology query message deliver it to its successive neighbour nodes. If it has no successive neighbour node, or no acknowledgement received within default wait time or reply wait time expires; it will send back a reply message which contains the Node Address, Node's Adjacency Matrix with its neighbours. On receiving the reply message, the agent node can obtain one-hop connection information of that node. Finally, at the end of the query process, the agent node can get the one-hop adjacency matrix of the whole network topology. After establishing the adjacency matrix, the topology agent advertises the adjacency matrix to its neighbouring nodes, which rebroadcast it to their neighbouring nodes until every node receives the message. On receiving the adjacency matrix, each node independently runs the Dijkstra algorithm over the adjacency matrix to find the shortest path from itself to all other nodes in the network [5], [6], [7].

If a node enters or leaves the network or there is any change of the network topology, all neighboring nodes can detect the change and send a topology update message to the topology agent. The topology agent node has to update the adjacency matrix using the query-reply process and transmit the updated adjacency matrix information to all other nodes.

2.3 Overview of Retransmission Mechanism of 802.11 MAC

Our proposed method is based on the retransmission mechanism employed in IEEE 802.11 which is employed for ARQ (Automatic Repeat Request). A retransmission occurred if an ACK or a CTS frame is not received in the response to transmitting Data or RTS frame. This can be due to either the dropping of a frame or network congestion.

Formally, if the mobile node employs DCF (distributed coordination function) for medium access as illustrated in Figure 2, data transmission is preceded by the RTS/CTS phase to reserve the channel and avoid the hidden/exposed node problem. As shown, when RTS is sent for reserving the channel, the sender waits for a certain time and if it doesn't receive the CTS, it will retransmit an RTS. This process will be repeated till receiving a CTS and having a contention-free channel. The number of

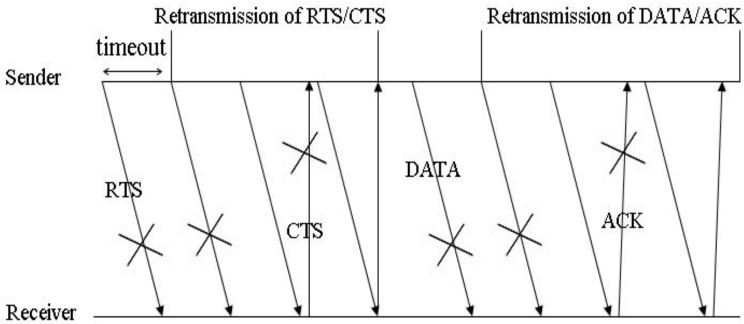


Fig. 2. Retransmission Mechanism of 802.11 MAC on unstable condition

RTS retransmissions gives an indication of the contention level of the link, as well as an estimation of the link quality [8].

The same process is then repeated for DATA frames. Therefore the number of sending frames is a good estimate of congestion of a link. Hence, we propose to use it as an indication to detect the congestion.

3 The Proposed Collision Avoidance Mechanism for Adjacency Matrix Routing

3.1 Link Congestion Condition

In 802.11 variables, ACKFailureCount and RTSFailureCount denote the number of DATA and RTS retransmission [9]. To detect the congestion of a given link, each node uses these two variables to measure the number of frame retransmissions required to send a data packet. Since the default limit for RTS is 7 and for DATA is 4, then the sum of ACKFailureCount and RTSFailureCount exceeding 11 means the congestion occurrence, otherwise not.

Assume that the number of retransmissions on link from i to j to send its kth packet is $Failure_{ij}(k)$, which will be:

$$\mathrm{Failure}_{ij}(k) = \mathrm{ACKFailureCount}_{ij(k)} + \mathrm{RTSFailureCount}_{ij(k)}$$

The $Failure_{ij}(k)$ represents the number of retransmissions of DATA and RTS for a *kth* packet. The route suffers the congestion only while one link of the route encounters the congestion, so we can use the maximum retransmission failures of links over the route to indicate whether the route is congestion or not. Suppose that $\mathrm{MaxFailure}_{S,D}(k)$ represents the maximum retransmissions of links over the route from a source S to a destination D for sending *k*th packet, then it can be obtained by:

$$\mathrm{MaxFailure}_{S,D}(k) = \max\{\mathrm{Failure}_{ij}(k), i, j \text{ is the intermediate nodes of the route from S to D}\}$$

After calculating the maximum retransmission of links over the route, we can detect that the route is congestion or not. In this work, we combine this congestion indication with the routing layer protocol and on detecting congestion, the adjacency matrix routing protocol can share this information and recalculate a new route to transfer the packets.

3.2 Routing Decision Algorithm

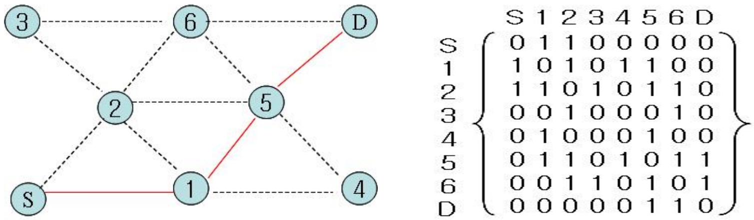
In the network for adopting the adjacency matrix routing protocol, each node maintains the network information as the adjacency matrix broadcast from the agent node. For instance, in an adjacency matrix A= { a_{ij} } for an N-node network, a '1' entry at (i, j) indicates a connection from node i to node j and 0 entry at (i, j) indicates no connection from node i to node j. When a source node originates a new packet addressed to a destination node, the source node employs the Dijkstra algorithm to calculate the route to the destination based on the adjacency matrix, and then places in the header of the packet the shortest path giving the sequence of hops that the packet is to follow on its way to the destination.

In the original adjacency matrix routing protocol, the destination node must send back ACK message to the source node for the transmission of the packets. If the source node doesn't receive ACK message when the timeout reaches, the source node takes the route information as stale and the destination is unreachable, then sends message to agent node to require update of adjacency matrix. But in the mobile ad hoc networks, the failure of packets transmission maybe because of several causes, not just the route breaks. In case of the congestion situation, although the route calculated based on the adjacency matrix is valid, the packets transmission suffers the failure.

In our proposed method, we use the retransmissions of DATA and RTS as the indication of congestion. Each node on the route records the number of retransmissions of DATA and RTS. While the retransmission is more than threshold, the route is regarded to encounter the congestion and the node detecting the congestion should send message to the source node. On receiving the message indicating the congestion, the source node modifies the adjacency matrix, by setting the entry which denotes the path of the congestion route as 0, and recalculating the new route based on the modified adjacency matrix. If the new route also suffers the congestion, the source node repeats the process until the timeout, in which the network information is regarded as stale and source node should send message to agent node for requiring update of adjacency matrix.

3.3 Cross-Layer Routing Example

In Figure 3(a), S and D are the source node and destination node, respectively and the adjacency matrix routing protocol is working in the network. According to the adjacency matrix routing protocol, each node maintains an adjacency matrix shown in Figure 3(b), denoting the network topology, broadcast from the agent node. Based on the adjacency matrix, the source node can run the Dijkstra algorithm to find the route to other nodes in network. For instance, the algorithm can calculate the route S-1-5-D to transfer the packets to destination D [6], [7].

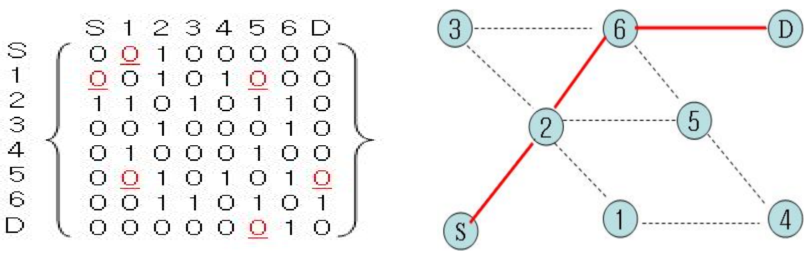


(a) Network Topology (b) Adjacency Matrix

Fig. 3. Network Topology and Adjacency Matrix

In the original adjacency matrix routing protocol, if the source node doesn't receive ACK message from the destination D for some transmission, the source node could consider that the routing table is expired and send the message to agent node for requiring update the adjacency matrix. But the packets loss may be caused by the congestion in the network. So, in this work, we use maximum retransmission of links to detect congestion and recalculate a new route for data transmission when congestion occurs, therefore reducing the propagation of the update and avoiding the congestion route in the network can be achieved.

For example, for a transmission of *k*th packet, $\text{MaxFailure}_{S,D}(k)$ is beyond 11, which means that the route from S to D suffers the congestion. According to our proposed method, the adjacency matrix routing protocol can get this information from MAC layer and recalculate a new route. In case of the congestion, source node S can transform the adjacency matrix as shown in Figure 4(a), which eliminates the congestion route S-1-5-D. Based on the new adjacency matrix, node S runs the Dijkstra algorithm to find a new route shown in Figure 4(b) to delivery the packets to D. Due to eliminating the nodes of the former congestion route, the new route should bring better performance. If the new route also encounters the congestion, the computation can repeat until the timeout is expired. In case of timeout, the source node should send message to the agent node for requiring update of network topology knowledge.



(a). Modified Adjacency Matrix **(b).** Newly calculated route

Fig. 4. Modified Adjacency Matrix and Newly calculated route in case of congestion

4 Simulation

4.1 Simulation Environment

The simulations are implemented in Network Simulator (NS-2) [10], the version of ns-allinone-2.30. The simulation environment as follows:

- Number of nodes: 33;
- Testing area: 1000m x 1000m;
- Mobile node speed: 0m/s, 1m/s, 2m/s;
- Traffic load: UDP, CBR traffic generator;
- Radio transmission range: 250 m;
- MAC layer: IEEE 802.11.
- Simulation time:500 seconds
- Maximum connections simultaneously: 4, 8

The source and destination nodes of connections are chosen randomly from the 33 nodes. The topology for the simulation is obtained from the nam file shown in the Figure 5.

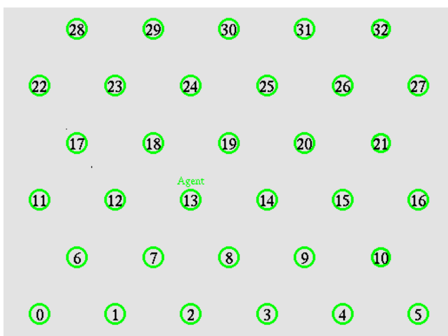


Fig. 5. The Network Topology used in NS-2

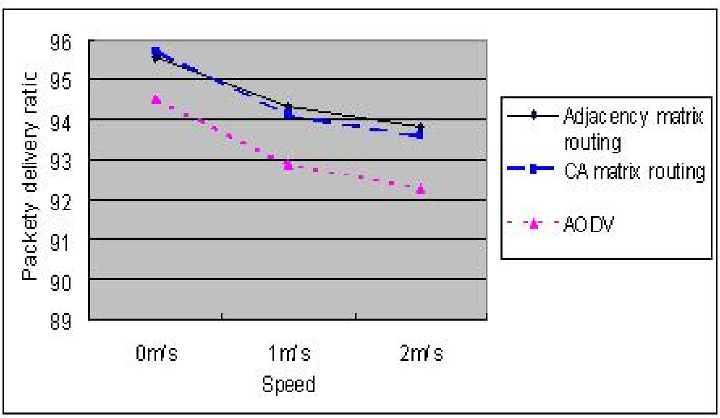


Fig. 6. Packet delivery ratio in 4 maximum connections scenario

4.2 Simulation Results

Figure 6 and 7 show the simulation result, Packet Delivery Ratio with AODV, original adjacency matrix routing and that with proposed congestion avoidance mechanism in case of 4 and 8 maximum connections respectively. Packet delivery ratio is the ratio of successfully arrived packets within the packets sent. As shown in the simulation result, in the scenario with rare congestion, the proposed mechanism just brings the similar performance with original version and better performance than that of AODV. However, in case of 8 maximum connections which mean the frequent congestion, the adjacency matrix routing with the proposed mechanism generates much better performance than original adjacency matrix routing.

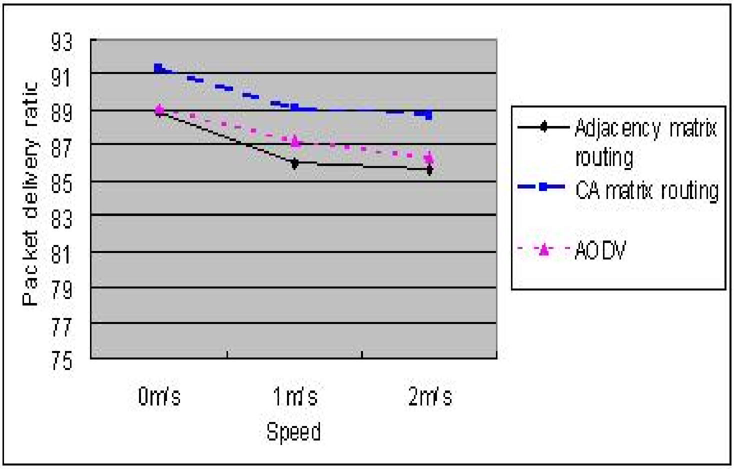


Fig. 7. Packet delivery ratio in 8 maximum connections scenario

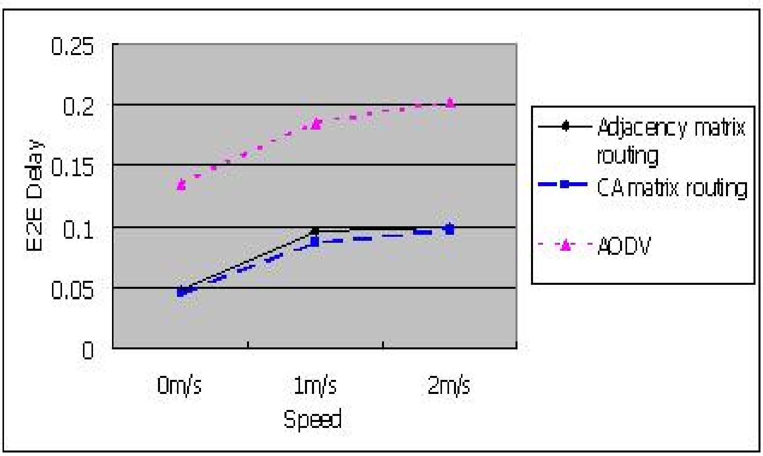


Fig. 8. End to End delay in 4maximum connections scenario

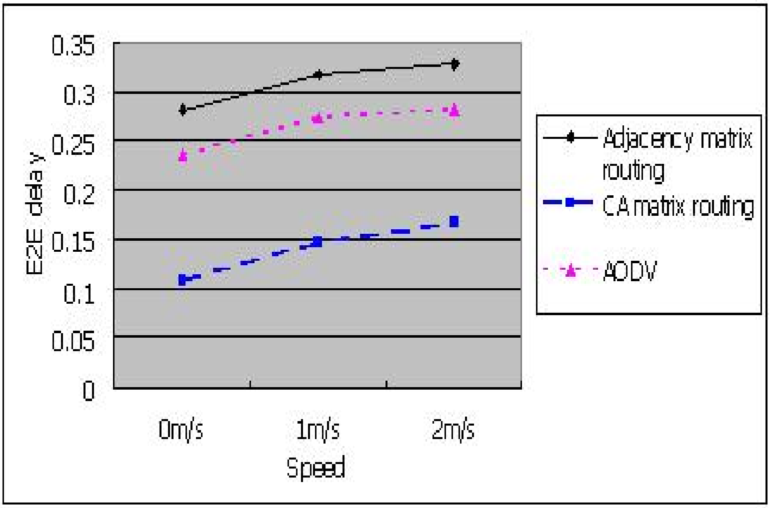


Fig. 9. End to End delay in 4 maximum connections scenario

Figure 8 and 9 present the E2E Delay of the same simulation scenarios except maximum connections. End to End Delay refers to the average time taken for a packet to be transmitted across a network from source to destination. As shown in the simulation result, the proposed congestion avoidance mechanism brings better performance, especially in the environment with frequent congestion.

5 Conclusion

In this paper, we propose a cross layer method which is based on the cooperation between Adjacency matrix routing protocol and the IEEE 802.11e MAC protocol. The topology agent gathers frequently the network topology for delivering the information to all other nodes. The routing agent uses the adjacency matrix routing protocol of network layer and the number of frame retransmissions which can be an indication of congestion from MAC layer to detect the congestion, and recalculate a new route avoiding the congestion nodes for packets transmission. We conduct the simulation with AODV, original and proposed method in the rare and frequent congestion scenarios respectively. The performance evaluation and simulations results show that our proposed mechanism is more efficient in the frequent congestion environment.

Acknowledgments. This research was financially supported by the Ministry of Education Science Technology (MEST) and Korean Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

References

1. Siva Ram Murthy, C., Manoj, B.S.: Routing Protocols for Ad Hoc Wireless Networks. In: Ad Hoc Wireless Networks Architectures and Protocols, ch. 5. Prentice Hall, Englewood Cliffs (2004)
2. Kawadia, V., Kumar, P.R.: A cautionary perspective on cross-layer design. IEEE Wireless Communication Magazine 12(1), 3–11 (2005)
3. Srivastava, V., Motani, M.: Cross-layer design: a survey and the road ahead. IEEE Communication Magazine 43(12), 1112–1119 (2005)
4. Jurdak, R.: Cross-Layer Design. In: Wireless Ad Hoc and Sensor Networks A Cross-Layer Design Perspective, ch. 7. Springer Science+Business Media, LLC (2007)
5. Miller, L.E.: Multihop distance of arbitrary networks (March 2001), `http://w3.antd.nist.gov/wctg/netanal/ConCalc.pdf`
6. Liu, Z., Kim, J., Lee, B., Kim, C.: A Routing Protocol based on Adjacency Matrix in Ad Hoc Mobile Networks. In: ALPIT Proceedings, July 2008, vol. 7 (2008)
7. Liu, Z., Cheon, S., Kim, C.: A Routing Protocol based on Distance Vector in Ad Hoc Networks. In: NCM Proceedings (September 2008)
8. ISO/IEC 8802-11; ANSI/IEEE Std 802.11, 1999 edition. Information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. The Institute of Electrical and Electronics Engineers (1999)
9. Draves, R., Padhye, J., Zill, B.: Comparison of routing metrics for static multi-hop wireless networks. In: Proceedings of ACM SIGCOMM 2004, September 2004, pp. 133–144 (2004)
10. The Network Simulator ns-2 (2006), `http://www.isi.edu/nsnam/ns/`
11. Conti, M., Maselli, G., Turi, G.: Cross-layering in mobile ad hoc network design. Network Computer 37(2), 48–51 (2004)
12. Bisnik, N.: Protocol design for wireless ad hoc networks: the cross-layer paradigm. Technical report, ECSE Department (2005)

Towards Power Efficient MAC Protocol for In-Body and On-Body Sensor Networks

Sana Ullah, Xizhi An, and Kyung Sup Kwak

Graduate School of Telecommunication Engineering
253 Yonghyun-Dong, Nam-Gu, 402-751, Inha University Incheon South Korea
sanajcs@hotmail.com, anxizhi@hotmail.com, kskwak@inha.ac.kr

Abstract. This paper presents an empirical discussion on the design and implementation of a power-efficient Medium Access Control (MAC) protocol for in-body and on-body sensor networks. We analyze the performance of a beacon-enabled IEEE 802.15.4, PB-TDMA, and S-MAC protocols for on-body sensor networks. We further present a Traffic Based Wakeup Mechanism that utilizes the traffic patterns of the BAN Nodes (BNs) to accommodate the entire BSN traffic. To enable a logical connection between different BNs working on different frequency bands, a method called Bridging function is proposed. The Bridging function integrates all BNs working on different bands into a complete BSN.

Keywords: BSN, WBAN, MAC, Traffic, Based, Mechanisms, Bridging, In-body, On-body, Sensor Networks.

1 Introduction

The remote monitoring of body status, and the surrounding environment, are becoming more important for sporting activities, the safety of members of the emergency services, members of the military and health care. The levels of fitness required for the very competitive international sporting events require athletes to be at the very pinnacle of fitness with every muscle used to its utmost. Furthermore, many body functions are traditionally monitored only rarely and separated by a considerable period of time. This can give a very incomplete picture of what is really happening. Consider a patient visiting a doctor for a blood pressure check; he/she may be anxious and thus have elevated pressure resulting in an inaccurate diagnosis. If, however, the patient can be fitted with a simple monitoring system that requires no intervention, then a picture can be built up of how the pressure changes through the day when he/she goes about their normal business. This will give a better picture of what is happening and remove inaccurate results caused by going to visit the doctor. To achieve these requirements, monitoring of movement and body function are essential. This monitoring requires the sensors and wireless system to be very lightweight and to be integrated un-obtrusively into the clothing.

A Body Sensor Network (BSN) allows the integration of intelligent, miniaturized, low power, invasive and non-invasive sensor nodes to monitor body function and the

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 335–345, 2009.

surrounding environment. Each intelligent node has enough capability to process and forward information to a base station for diagnosis and prescription. A BSN provides long term health monitoring of patients under natural physiological states without constraining their normal activities. It can be used to develop a smart and affordable health care system and can be a part of diagnostic procedure, maintenance of chronic condition, supervised recovery from a surgical procedure and to handle emergency events [1].

A number of ongoing projects such as CodeBlue, MobiHealth, and iSIM have contributed to establish a proactive and unobtrusive BSN system [2]-[4]. A system architecture presented in [5] performs real time analysis of sensor's data, provides real time feedback to the user, and forwards the user's information to a telemedicine server. UbiMon aims to develop a smart and an affordable health care system [6]. MIT Media Lab is developing MIThril that gives a complete insight of human-machine interface [7] HIT lab focuses on quality interfaces and innovative wearable computers [8]. IEEE 802.15.6 aims to provide power-efficient in-body and on-body wireless communication standards for medical and non-medical applications [9]. NASA is developing a wearable physiological monitoring system for astronauts called LifeGuard system [10]. ETRI focuses on the development of a low power MAC protocol for a BSN [11].

In this paper, we use the terms BAN Node (BN) and BAN Network Coordinator (BNC) for the sensor node and the network coordinator in a BSN. The rest of the paper is organized into six sections. Section 2 presents discussion on BSN traffic classification. Section 3 and 4 present a brief analysis on in-body and on-body MAC protocols. Section 5 and 6 discuss the Traffic Based Wakeup Mechanism and the Bridging function for a BSN. The final section concludes our work.

2 BSN Traffic Classification

The assorted BSN traffic requires sophisticated and power-efficient techniques to ensure safe and reliable operation. Existing MAC protocols such as SMAC [12], TMAC [13], IEEE 802.15.4 [14], and WiseMAC [15] give limited answers to the heterogeneous traffic. The in-body BNs do not appreciate synchronized wakeup periods because they confine the accommodation of sporadic emergency events. Medical data usually needs high priority and reliability than non-medical data. In case of emergency events, the BNs should be able to access the channel in less than one second [16]. IEEE 802.15.4 Guaranteed Time Slots (GTS) can be utilized to handle time critical events but they expire in case of a low traffic. Furthermore, some in-body BNs have high data transmission frequency than others. We classify the entire BSN traffic into Normal, On-demand, and Emergency traffics as given in Figure 1. The normal traffic is further classified into High, Medium, and Low traffics.

(a)- Normal Traffic: Normal traffic is the data traffic in a normal condition with no time critical and on-demand events. This includes unobtrusive and routine health monitoring of a patient for diagnosis and treatment of many diseases such as gastrointestinal tract, neurological disorders, cancer detection, handicap rehabilitation, and the most threatening heart disease. Some BNs have frequent wakeup periods and thus are designated as high traffic BNs. For example, an ECG node may send data 4 times per

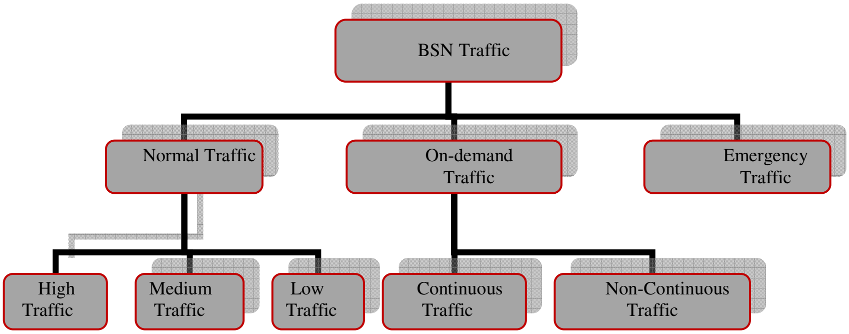


Fig. 1. BSN Traffic Classification

hour, while other BNs may send 4 times a day. The ECG node is thus designated as a high traffic BN. However, the normal traffic classification from high to low and vice-versa depends on the application requirements. The normal data is collected and processed by the BNC.

(b)- On-demand Traffic: On-demand traffic is initiated by the BNC or doctor to know certain information, mostly for the purpose of diagnosis and prescription. This is further divided into continuous (in case of surgical events) and non-continuous (when limited information is required).

(c)- Emergency Traffic: This is initiated by BNs when they exceed a predefined threshold and should be accommodated in less than one second. This kind of traffic is not generated on regular intervals and is totally unpredictable.

3 MAC for On-Body Sensor Networks

On-body sensor networks comprise of miniaturized and non-invasive sensor nodes that are used for various applications, ranging from medical to interactive gaming and entertainment applications. They use Wireless Medical Telemetry Services (WMTS), unlicensed ISM, and UWB bands for data transmission. WMTS is a licensed band designated for medical telemetry system. Federal Communication Commission (FCC) urges the use of WMTS for medical applications due to fewer interfering sources. However, only authorized users such as physicians and trained technicians are eligible to use this band. Furthermore, the restricted WMTS (14 MHz) bandwidth cannot support video and voice transmission. The alternative spectrum for medical applications is to use 2.4 GHz ISM band that includes guard bands to protect adjacent channel interference.

The design and implementation of a power-efficient MAC protocol for on-body sensor networks have been a hot research topic for the last few years. A novel TDMA protocol for on-body sensor network called H-MAC exploits the biosignal features to perform TDMA synchronization and improves the energy efficiency [17]. Other protocols like WASP, CICADA, and BSN-MAC are investigated in [18] – [20]. The

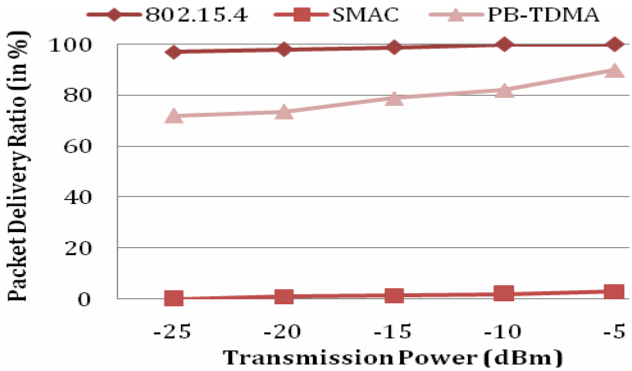


Fig. 2. Packet Delivery Ratio of IEEE 802.15.4, PB-TDMA, and SMAC for On-Body Sensor Networks

performance analysis of a non-beacon IEEE 802.15.4 is adapted to extend lifetime of a node from 10 to 15 years [21]. This work considers low upload/download rates, mostly per hour. Furthermore, the data transmission is based on periodic intervals, which limits the performance to certain applications. There is no reliable support for on-demand and emergency traffics.

We analyze the performance of a beacon-enabled IEEE 802.15.4, Preamble-Based TDMA [22], and SMAC protocols for on-body sensor networks. Our analysis is verified by extensive simulation using NS-2[23]. In case of S-MAC, PB-TDMA, and IEEE 802.11 (DCF) protocols, the wireless physical parameters are considered according to low power Nordic nRF2401 transceiver. This radio transceiver operates in the 2.4-2.5 GHz band with an optimum transmission power of -5dBm. However, in case of IEEE 802.15.4, Chipcon CC2420 radio interface is considered. We use shadowing propagation model throughout the simulations. We consider 9 BNs firmly placed on a human body. The BNs are connected to the BNC in a star topology. The initial BN energy is 5 Joules. The data rate of the BNs is heterogeneous. The simulation area is 1x1 meter and each BN generates Constant Bit Rate (CBR) traffic. The packet size is 128 bytes. The transport agent is User Datagram Protocol (UDP). Simulation results show that IEEE 802.15.4, when configured in a beacon-enabled mode, outperforms SMAC and PB-TDMA as shown in Figure 2. However, the precise location of BNs and the body position influence the packet delivery ratio.

Intel Corporation conducted a series of experiments to analyze the performance of IEEE 802.15.4 for an on-body sensor network [24]. They deployed a number of Intel Mote 2 [25] nodes on chest, waist, and the right ankle. Tabel 1 shows the packet success rate at 0dBm transmit power when a person is standing and sitting on a chair. The connection between ankle and waist cannot be established, even for a short distance of 1.5m. All other connections show favourable performance.

As IEEE 802.15.4 operates in the 2.4 GHz unlicensed band, the possibilities of interference from other devices such as IEEE 802.11 and microwave are inevitable. A series of experiments to evaluate the impact of IEEE 802.11 and microwave ovens on

Table 1. Packet Success Rate at 0dBm Transmit Power

Source \ Destination	Packet Success Rate when a Person is Standing: Chest	Waist	Ankle	Packet Success Rate when a Person is Sitting on an Office Chair: Chest	Waist	Ankle
Chest		99%	84%		99%	81%
Waist	100%		50%	99%		47%
Ankle	72%	76%		77%	27%	

Table 2. Co-existence Test Results between IEEE 802.15.4 and Microwave Oven

Microwave Status	Packet Success Rate	
	Mean	Std.
ON	96.85%	3.22%
OFF	100%	0%

the IEEE 802.15.4 transmission are carried out in [26]. They considered XBee 802.15.4 development kit that has two XBee modules. Table 2 shows the affects of microwave oven on the XBee remote module. When the microwave oven is ON, the packet success rate and the standard deviation is degraded to 96.85% and 3.22% respectively. However, there is no loss when the XBee modules are taken 2 meters away from the microwave oven.

4 MAC for In-Body Sensor Networks

The most challenging task in developing a power-efficient MAC for in-body sensor networks is to accommodate the in-body BNs in a power-efficient manner. Unlike on-body BNs, the in-body BNs are implanted under human skin where the electrical properties of the body affect the signal propagations. The human body is a medium that poses many wireless transmission challenges. The body is composed of several components that are not predictable and will change. Monitoring in-body functions and the ability to communicate with implanted therapeutic devices, such as pacemakers, are essential for their best use.

Zarlink semiconductor has introduced a wireless chip that supports a very high data rate RF link for communication with an implantable BN [27]. The ZL70101 ultra-low power transceiver chip satisfies the power and size requirements for implanted communication systems and operates in 402-405 MHz MICS band [28]-[29]. Other frequency bands such as 916MHz, 1.5GHz, and UWB are also considered for in-body sensor networks [30]-[32]. The use of open air models for implant communication is discouraged in [33]. A Finite-Difference Time-Domain (FDTD) is used to calculate the power deposition in a human head and is measured by the SAR in W/Kg [34]. However, the distance from the implant source has not been discussed. For a duty cycle and transmission rate of 0.05% and 400 kbps, the SubQore radio architecture from Cambridge university consumes a peak current less than 1.7mA [35]. An implantable medical microsystems for interfaces to the central nervous system is presented in [36].

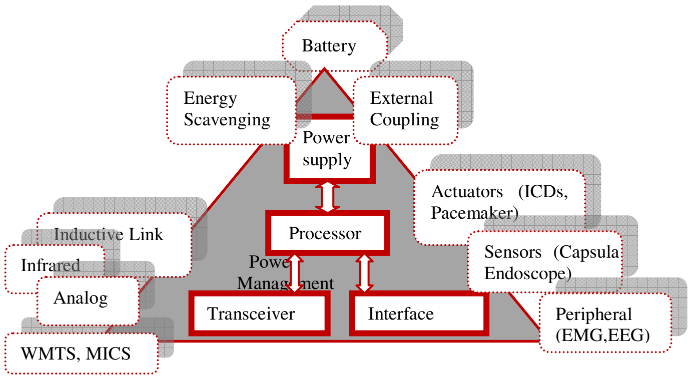


Fig. 3. Implementation issues in the in-body sensor network

The diverse nature of in-body BNs together with the electrical properties of the human body influences the development process of a power-efficient MAC for in-body sensor networks. The data rate of implanted BNs varies, ranging from few kbps in pacemaker to several Mbps in capsular endoscope. Figure 3 explains the in-body sensor networks and implementation issues. In the in-body sensor network, critical traffic requires low latency and high reliability than non-critical traffic. One of the solutions is to adjust initial back-off windows in a traditional CSMA/CA for critical and non-critical traffics. Due to high path loss inside the human body, the use of CSMA/CA does not provide reliable solution in multi-piconets [37]. For a threshold of -85dBm and -95dBm, the on-body BNs cannot see the activity of in-body BNs when they are away at 3 meters distance from the body surface. However, within 3 meters or less distance, the CCA works correctly in the same piconet.

The in-body MAC should also consider the thermal influence caused by the electromagnetic wave exposure and circuit heat. Nagamine *et al.* discussed the thermal influence of the BNs using different MAC protocols [38]. Figure 4 shows the temperature of a BN when ALOHA and CSMA/CA are used.

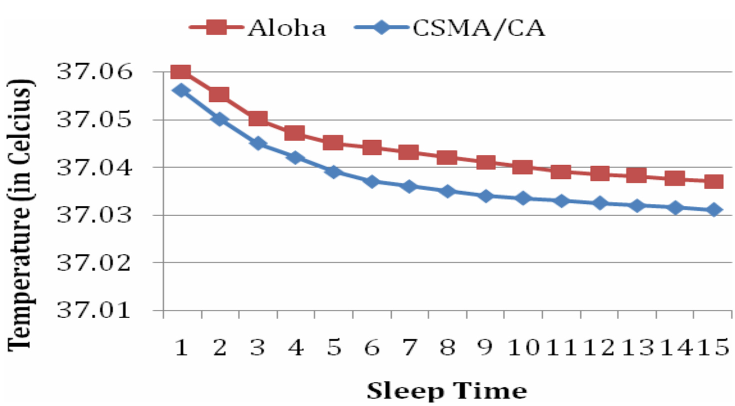


Fig. 4. Saturated Temperature

5 Traffic Based Wakeup Mechanism for a BSN

The heterogeneous BNs require power-efficient and dynamic wakeup techniques for reliable operation. We propose a power-efficient technique called Traffic Based Wakeup Mechanism for a BSN that exploits the traffic patterns of the BNs to accommodate the assorted BSN traffic. The initial wakeup patterns are either predefined (by the company) or created and modified (by the BNC). Table 3 shows the traffic classification and the corresponding solutions.

Table 3. Solutions to the Classified BSN traffic

Traffic / Devices	Normal Traffic			On-demand Traffic	Emergency Traffic
	High	Medium	Low		
BAN Nodes (BNs)	Send data based on the **Traffic-based Wake-up Table**			Receives a **Wakeup Radio Signal** from the BNC and respond	Send a **Wake-up Radio Signal** to the BNC in case of emergency
BAN Coordinator (BNC)	Send data based on the **Traffic-based Wake-up Table**			Send a **Wake-up Radio Signal** to BNs	Receives a **Wake-up Radio Signal** and respond

The wakeup patterns of all BNs are organized into a table called Traffic Based Wakeup Table. The table is maintained and modified by the BNC according to the application requirements. Based on the BNs wakeup patterns, the BNC can also calculate its own wakeup pattern. This could save significant energy at the BNC. The BNC does not need to stay active when there is no traffic from the BN. The designation of normal traffic levels, i.e, high, medium, and low traffics nodes depend on the application. For emergency and on-demand traffics, the BNs and the BNC send a wake-up signal for a very short duration to each other. However, traditional wake-up radio concepts have several limitations when considered in the in-body sensor networks. They are not able to wake-up a particular BN. All BNs wake-up in response to a single wake-up signal, which is not the required environment. The use of different radio frequencies to wake-up a particular BN may provide an optimal solution. However, the use of wakeup radio requires efficient security management schemes.

6 Bridging Function for a BSN

In a BSN, there can be various BNs working on different frequency bands and have correspondingly different Physical Layers (PHYs). The main problem is how to connect different BNs working on different bands in a BSN. In order to accommodate multiple PHYs (radio interfaces) and multiple channels, we introduce a function called Bridging that virtually connects different BNs working on different PHYs.

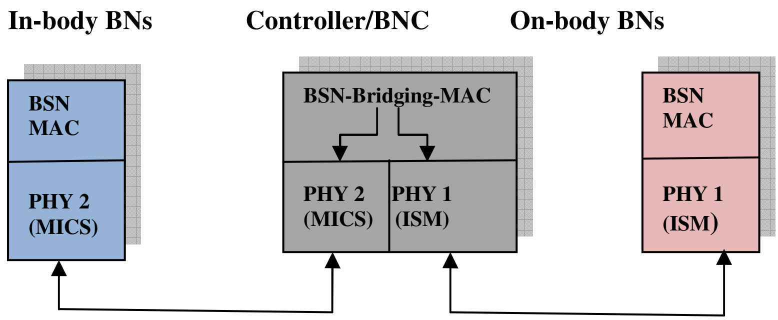


Fig. 5. Protocol Stack of the Bridging Function

Bridging function enables one MAC to support multiple bands and multiple PHYs. One MAC means a common hybrid MAC framework. Furthermore, we use the term Channel from MAC point of view where one channel is a combination of a frequency band and the corresponding PHY technique.

The Bridging function is an enhanced MAC function that establishes logical relationships between different PHY BNs. However, the BNs running Bridging function must have two or more different PHY interfaces in order to work on two or more different Channels at the same time. All necessary information, i.e, Network Info, Channel ID (band/PHY), BNs ID, Connection ID and Connection Type (MAC layer), Source and Destination BNs ID is recorded in a specific table. According to the records in the table, a Channel Mapping is implemented in the intermediate BNs that support at least two bands/PHYs.

The Bridging function is also regarded as link-layer MAC Protocol Data Unit (MPDU) relay. The intermediate BNs receive and store data from one channel, and then forward it on another channel towards the destination. The process is transparent to upper layers and accommodates different PHY techniques. Figure 5 shows the protocol stack of the Bridging function.

The BNs implementing the Bridging function (also called a Bridge) can collect or dissipate data from or to the in-body BNs and communicate with on-body and out-body BNs. The Bridge has two PHY layers, typically unlicensed ISM/UWB Band and a licensed MICS Band. Generally, on-body/out-body BNs are selected for the Bridging function due to their relative larger capabilities and less stringent channel conditions. The Bridging enables the integration of all BNs into a BSN and realizes the interconnectivity among different PHY BNs.

The in-body BNs cannot establish a direct communication with on-body BNs due to power limitations, high path loss, and different PHYs. Therefore, they exploit the Bridging function and forward all the data frames to the on-body BNs via Bridge. Moreover, the in-body BNs do not support peer-to-peer transmission in same network, and thus data transmission between in-body BNs are relayed by the Bridge. Though on-body/out-body BNs can be designated as a Bridge node, but we urge the use of BNC to perform the Bridging function.

7 Conclusions

In this paper, we studied the behavior of several power-efficient MAC protocols including a beacon-enabled IEEE 802.15.4 protocol for on-body sensor networks. We concluded that none of the existing MAC protocols accommodate the assorted BSN traffic requirements in a power-efficient manner. The entire BSN traffic is classified into normal, on-demand, and emergency traffics. A Traffic Based Wakeup Mechanism is proposed for a BSN, which exploits the traffic patterns of the BNs to accommodate the entire traffic classification. We further introduce a Bridging function that integrates all the BNs working on different PHYs into a complete BSN. The proposed wakeup mechanism backed by the Bridging function provides a complete solution towards power-efficient and reliable communication in a BSN.

Acknowledgement

This work was supported by the IT R&D program of MKE/IITA, [2008-F-050-01, Development of WBAN system for In-body and On-body].

References

1. Otto, C., Milenkovic, A., Sanders, C., Jovanov, E.: System Architecture of a Wireless Body Area Sensor Network for Ubiquitous Health Monitoring. Journal of Mobile Multimedia 1(4), 307–326 (2006)
2. `http://fiji.eecs.harvard.edu/CodeBlue` (Date Visited: November 21, 2008)
3. `http://www.mobihealth.org/` (Date Visited: January 20, 2009)
4. `http://www.cs.uoregon.edu/research/wearables/index.html` (Date Visited: July 21, 2008)
5. Jovanov, E., Milenkovic, A., Otto, C., de Groen, P.: A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. Journal of NeuroEngineering and Rehabilitation 2(6) (March 2005)
6. `http://www.ubimon.net` (Date Visited: March 16, 2008)
7. `http://www.media.mit.edu/wearables/mithril/` (Date Visited: Febuary 09, 2008)
8. `http://www.hitl.washington.edu/` (Date Visited: April 25, 2008)
9. `http://www.ieee802.org/15/pub/TG6.html` (Date Visited: December 16, 2008)
10. `http://lifeguard.stanford.edu/` (Date Visited: January 11, 2009)
11. `http://www.etri.re.kr/eng/`
12. Ye, W., Heidemann, J., Estrin, D.: Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. IEEE Communications Letters 12(3), 493–506 (2004)
13. Dam, T.V., Langendoen, K.: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In: SenSys 2003, Los Angeles, pp. 171–180 (2003)
14. IEEE Std.802.15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low data rate wireless personal area networks (WPAN) (2006)

15. El-Hoiydi, A., Decotignie, J.-D.: WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of InfrastructureWireless Sensor Networks. In: The Proceedings of the Ninth IEEE Symposium on Computers and Communication, ISCC 2004, Alexandria, Egypt, pp. 244–251 (2004)
16. Technical Requirement Document, IEEE 802.15.6 (January 2009)
17. Huaming Li, H., Jindong Tan, H.: Heartbeat Driven Medium Access Control for Body Sensor Networks. In: Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments, pp. 25–30 (2007)
18. Braem, B., Latre, B., Moerman, I., Blondia, C., Demeester, P.: The Wireless Autonomous Spanning tree Protocol for multihop wireless body area networks. In: Proceedings of the First International Workshop on Personalized Networks, San Jose, California, USA (2006)
19. Latre, B., Braem, B., Moerman, I., Blondia, C., Reusens, E., Joseph, W., Demeester, P.: A Low-delay Protocol for Multihop Wireless Body Area Networks. In: Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous 2007, pp. 1–8, 6–10 (2007)
20. Li, H., Tan, J.: Medium Access Control for Body Sensor Networks. In: Proceedings of 16th International Conference on Computer Communications and Networks, ICCCN 2007, August 13-16, pp. 210–215 (2007)
21. Timmons, N.F., Scanlon, W.G.: Analysis of the performance of IEEE 802.15.4 for medical sensor body area networking. In: IEEE SECON (2004)
22. Ullah, S., Islam, R., Nessa, A., Zhong, Y., Sup, K.K.: Performance Analysis of a Preamble Based TDMA Protocol for Wireless Body Area Network. Journal of Communications Software and Systems 4(3), 222–226 (2008)
23. http://www.isi.edu/nsnam/ns/ (Date visited: March 26, 2008)
24. Shah, R.C., Yarvis, M.: Characteristics of on-body 802.15.4 networks. In: 2nd IEEE Workshop on Wireless Mesh Networks, WiMesh 2006, September 25-28, pp. 138–139 (2006)
25. Adler, R., et al.: Intel Mote 2: An advanced platform for demanding sensor network applications, Demo session, Sensys (2005)
26. Chen, C., Pomalaza-Ráez, C.: Monitoring Human Movements at Home Using Wearable Wireless Sensors. In: ISMICT 2009, Montreal, February 24-27 (2009)
27. Ullah, S., Higgin, H., Kwak, K.S.: A study of implanted and wearable body sensor networks. In: Nguyen, N.T., Jo, G.S., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2008. LNCS, vol. 4953, pp. 464–473. Springer, Heidelberg (2008)
28. http://products.zarlink.com/productprofiles/ZL70101.htm (Date Visited: December 21, 2008)
29. MICS Band. Australian Communications Agency Paper SP 6/03 (October 2003)
30. Scanlon, W.G., Buirns, J.B., Evans, N.E.: Radiowave propagation from a tissue-implanted source at 418MHz and 916.5MHz. IEEE Trans. on Biomedical Engineering 47(4), 527–534 (2000)
31. Wang, J., Su, D.: Design of an Ultra-wideband system for in-body wireless communications. In: 4th Asia-Pacific conference on Enviromental Electromagnectic, pp. 565–568 (2006)
32. Kim, J., Rahmat-Samii, Y.: Electromagnetic interactions between biological tissues and implantable biotelemetry systems. In: IEEE Inter. Microwave Symposium, pp. 1801–1804 (2005)

33. Gupta, S.K.S., Lalwani, S., Prakash, Y., et al.: Towards a propagation model for wireless biomedical applications. In: IEEE international conference on communications, vol. 3, pp. 1993–1997 (2003)
34. Pandit, V., McDermott, R., Lazzi, G., Furse, C., Gandhi, O.: Electrical Energy Absorption in the Human Head From a Cellular Telephone. In: Proceedings of the 7th conference on Visualization (1996)
35. Cambridge Consultants, New radioarchitecture meets medical implant designers' needs for higher performance and longer life (2005)
36. Wise, K.D., Anderson, D.J., Hetke, J.F., Kipke, D.R., Najafi, K.: Wireless implantable microsystems: high-density electronic interfaces to the nervous system. Proc. IEEE 92(1), 76–97 (2004)
37. Shu, F., Dolmans, G.: QoS Support in Wireless BANs, IEEE P802.15 Working Group for Wireless Personal Area Networks, WPANs (November 2008)
38. Zhen, B., Li, H.-B., Kohno, R.: IEEE body area networks and medical implant communications. In: Proceedings of the ICST 3rd International Conference on Body Area Networks, Tempe, Arizona (2008)

Efficiency of Metaheuristics in PMJS_E/T Scheduling Problem

Marcin Bazyluk and Leszek Koszalka

Wroclaw University of Technology
Chair of Computer Systems and Networks
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
leszek.koszalka@pwr.wroc.pl

Abstract. Influence of the instance size on the efficiency of metaheuristic algorithms for the parallel machine job scheduling with earliness and tardiness penalties problem (PMJS_E/T) has been considered since their emergence. Such size is understood as the number of jobs which need to be scheduled and the number of available machines, capable of their execution. Investigations show that the proper selection of a metaheuristic for a given instance becomes an important issue leading to noticeable improve of quality. This paper tries to determine that dependence and suggests some policy on the selection.

1 Introduction. Related Work

In recent years the problem of job scheduling on parallel machines with earliness-tardiness penalties (PMJS_E/T) has been considered in many scientific and industrial communities. Generally speaking, in a classic example of this type problem we face a collection of independent jobs. Each of these is to be inserted on one of the set of available machines which may be identical or uniform. Specification of a single machine is contained in a vector describing its processing speed of respective jobs.

The problem introduces job weights, used to calculate the penalty for their execution delays. Each job has also its due date, a specific moment in time. If a job is accomplished at its due date, the requirements are met and there is no penalty. Garey *et al.* [6] show that the simplified model with a single machine (SMJS_E/T) and symmetric earliness and tardiness penalties is NP-complete. Yano *et al.* [12] prove NP-completeness of the SMJS_E/T with job weights proportional to their processing times. Sun *et al.* [11] study the case with identical parallel machines (PIMJS_E/T) and a common due date for all jobs and show that it is NP-hard in the ordinary sense when the number of machines is given. Since the problem considered here is a generalization of the three above-mentioned, it is also NP-complete and NP-hard in the ordinary sense.

An example solution to the problem is a schedule of jobs with the point to minimize the total penalty p for all jobs which is calculated with (1) where

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 346–355, 2009.

w_i is the weight of job i, $e_i = max\{0, d_i - c_i\}$ is the earliness of job i and $t_i = max\{0, c_i - d_i\}$ is the tardiness of job i with d_i being its due date and c_i being its completion time (moment in time when processing of a job is finished), I denotes the set of jobs.

$$p = \sum_{i \in I} w_i e_i + \sum_{i \in I} w_i t_i \tag{1}$$

Four metaheuristic algorithms were implemented to deal with such a problem. Three of these are the simulated annealing (SA), the tabu search (TS), and the genetic algorithm (GA) which have become very popular in recent time. This is justified by large body of literature covering wide perspectives of their implementation [3,4,10]. The fourth algorithm was the ant colony optimization (ACO), an own concept for tackling the cosindered problem based on previous applications of ant systems for the graph problems, such as [2,5,7,8]. Complex computational experiments have been conducted on randomly generated instances to compare the performances of the implemented metaheuristics.

2 Problem Formulation

Let us imagine a problem of scheduling N independent jobs on a set of K available parallel machines. Including decisions possible during job scheduling an optimization model was developed to minimize the total cost of processing all jobs. It is a modification of the idea presented in [3].

2.1 Introduced Variables and Parameters

$i, j = 1, 2, \ldots, N$, job indexes
$k \ \ = 1, 2, \ldots, K$, machine index
p_{ik} processing time of job i on machine k, $\qquad p_{ik} = \{1, 2, \ldots, 10\}$
w_i weight of job i, $\qquad w_i = \{1, 2, \ldots, 10\}$
d_i due date of job i, $\qquad d_i = \{0, 1, 2, \ldots, 50\}$
c_i completion time of job i,
e_i earliness of job i, $e_i = max\{0, d_i - c_i\}$
t_i tardiness of job i, $t_i = max\{0, c_i - d_i\}$

$$x_{ijk} = \begin{cases} 1 \text{ if job } j \text{ immediately follows job } i \text{ on machine } k, \\ 0 \text{ otherwise,} \end{cases}$$
$$i = 0, 1, \ldots, N, \ j = 1, 2, \ldots, N, \ k = 1, 2, \ldots, K$$

$$y_{jk} = \begin{cases} 1 \text{ if job } j \text{ is to be executed on machine } k, \\ 0 \text{ otherwise,} \end{cases}$$
$$j = 1, 2, \ldots, N, \ k = 1, 2, \ldots, K$$

For $i = 0$, $x_{0jk} = 1$ means that job j is on the first position of machine k. Having the decision variables above defined we present the recursive formula (2) used for calculating the moment of job completion as the sum of its processing time and of the moment when the job executed directly before it is completed.

$$c_j = \sum_{i=0}^{N} \sum_{k=0}^{K} x_{ijk} c_i + p_{jk} \tag{2}$$

2.2 Optimum Criterion and Restrictions

With the notations defined and parameters values established we are able to present the mathematical formulation of the problem. The objective function is presented in (3). The point is to find its minimal value.

$$T = \sum_{i=1}^{N} w_i(e_i + t_i) \tag{3}$$

The constraints on the introduced parameters are as follow:

– Each job must be processed on only one machine, and when its execution has been started, it must be finished before any other job is inserted on that machine, thus jobs cannot be split into pieces (eq. 4).

$$\sum_{i=1, i \neq j}^{N} \sum_{k=1}^{K} x_{ijk} = 1, \; j = 1, 2, \ldots, N \tag{4}$$

– Each job must immediately follow any other job on a machine (or stay on the first position if i = 0), if it has been chosen to be processed by this machine, so machine idle times are disallowed (eq. 5).

$$\sum_{i=1, i \neq j}^{N} x_{ijk} = y_{jk}, \; j = 1, 2, \ldots, N, \; k = 1, 2, \ldots, K \tag{5}$$

– If a job is processed on a machine, it will be immediately followed by at most one another job, thus machines can process only one job at a time (eq. 6).

$$\sum_{j=1, j \neq i}^{N} x_{ijk} \leq y_{ik}, \; i = 1, 2, \ldots, N, \; k = 1, 2, \ldots, K \tag{6}$$

– The job completion times are finite (eq. 7, where M is a large positive number).

$$\begin{gathered} c_j + M(1 - x_{ijk}) \geq c_i + p_{jk}, \\ j = 1, 2, \ldots, N, \; i = 1, 2, \ldots, N, \; i \neq j, \; k = 1, 2, \ldots, K, \; c_i, c_j > 0 \end{gathered} \tag{7}$$

3 Implementation of Metaheuristics

Four metaheuristics have been selected to deal with the considered problem. They are described in details in the following subsections.

3.1 Simulated Annealing (SA)

As for all local search algorithms, the developed simulated annealing one starts from an initial solution, generated with a modified *Longest Processing Time*, called LPT-MM (*Longest Processing Time - Multi-Machine*) algorithm. Its mechanism is presented in Fig. 1. Starting from a given solution, the algorithm consists of iterative steps to neighbor solutions chosen randomly. There are two possible steps: swap and insert. The execution lasts for a certain amount of time.

In each iteration we may step to a superior or an inferior solution. For the first case the step is unconditional but for the second one the decision to make a step or not depends on a randomly generated number. To make such a decision we define the exponential function in (8), an own concept which uses the linear cooling schedule. C_g is the total global lowest cost, C_n is the total cost of the neighbor solution, T_0 is the initial temperature, t is the current time and t_{max} is the fixed time of calculations. We have introduced the parameter δ which is the temperature decrease factor expressed in [%]. With the value of δ established parameter α for the linear schedule is calculated with (9). Block diagram of the designed simulated annealing is presented in Fig. 2a. Please note that since the initial temperature is only a quotient it is not expressed in any unit but just a number.

$$P = e^{\frac{C_g - C_n}{T_0 (1 - \frac{\delta t}{t_{max} \cdot 100\%})}} \tag{8}$$

$$\alpha = \frac{\delta T_0}{t_{max}} \tag{9}$$

3.2 Tabu Search (TS)

In this paper a deterministic tabu search algorithm generating a complete neighborhood of a current solution from which it afterwards chooses the best one is implemented. Two neighbor types have been selected: swap and insert. To generate a starting solution we propose the LPT-MM algorithm (see Fig. 1).

The tabu search operation consists of consecutive steps in each iteration to the best of the neighbor solutions, even if it is inferior to the base one. To avoid

Step 1 (initialization).

Create an empty job schedule. Sort all jobs $j \in N$ by descending value of $\frac{w_j}{p_j}$.

Step 2 (iterative scheduling).

For each of sorted jobs $i \in N$ do
Find the machine k with the shortest makespan. Place job i on machine k after the last job already placed.

Fig. 1. Mechanism of the LPT-MM algorithm

repetitive steps between the same solutions we define a tabu list which consists of assumed number of rules (Λ) which last solutions visited (checked) fit. The tabu list entries are kept in the list as a FIFO (First In First Out) queue. Each entry is an arc (i, j, k, l) which states that job j immediately follows job i on machine k where jobs i and j are respectively in positions $l - 1$ and l. When $i = 0$ job j is in the first position on machine k and $l = 0$. Block diagram of the designed tabu search is shown in Fig. 2b.

3.3 Genetic Algorithm (GA)

For more information on genetic algorithms we refer to [4]. Due to many different approaches to the considered algorithm which have been presented in the literature, this section states exact parameters of the implemented one. Each iteration simulates one generation of chromosomes, where we are dealing with a population of ancestors and a population of descendants. For the number of ancestors as n we are about to get $n(n-1)$ descendants and their set is generated by mating every possible pair in the ancestors set (chromosomes i, j: $1 \leq i, j \leq n$, $i \neq j$).

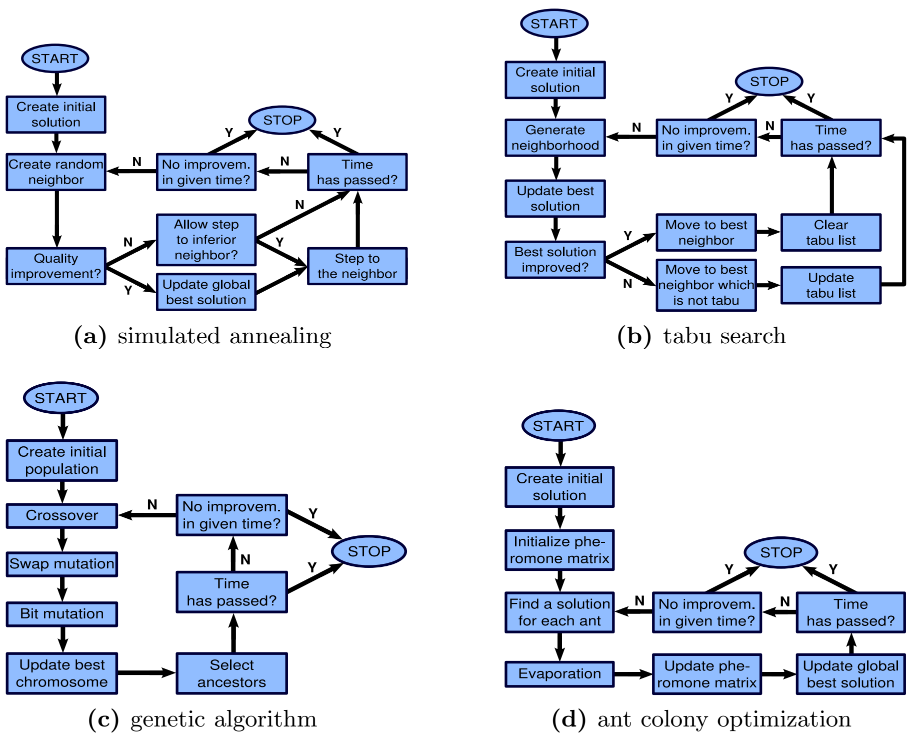


Fig. 2. Block diagrams of the implemented metaheuristics

What was proved experimentally in [1] for the problem examined here simplified by setting the due dates of all jobs to 0, usage of popular PMX crossover in the GA would make it impossible to find satisfying results. Therefore we tried an implementation of a *multi-component uniform order-based crossover* operator (MCUOX) proposed in [10] where a single gene accomodates both an object and the associated selection for that object. Thus for our case each gene corresponds to a job-machine pair. The crossover is executed with a fixed probability X, thus sometimes the descendant chromosome is simply a copy of its first ancestor.

For mutation we implemented two mechanisms: the swap and the bit one with respective probabilities M_s and M_b. First of these randomly chooses two positions on a chromosome and replaces their contents. The second one incorporates reassignment of a random job to a randomly chosen machine in a chromosome.

Probability of selecting chromosomes for crossover from a population of descendants is proportional to the value of their fitness function and for a given chromosome is defined in (10) with G being the size of the population and $F_g = CT_c(g)^{-1}$ being the fitness function of chromosome g, inversely proportional to the objective one where C is a fixed constant. The block diagram of GA is presented in Fig. 2c.

$$P_g = \frac{F_g}{\sum_{g=1}^{G} F_g} \tag{10}$$

3.4 Ant Colony Optimization (ACO)

Ant algorithms are a new promising approach to solving optimization problems based on the behaviour of ant colonies. First ant systems were developed to attack the problems presenting their similarity to ant colonies behaviour such as the TSP [5], QAP [8] or VRP [2]. Later they have been extended for the problem of JS but the literature on this topic is rather limited. An approach with ACO to SMJS was presented in [7]. We have been looking for any article facing the multi-machine case and to my best knowledge it is not available online. Therefore for the needs of this paper we were forced to design the mathematical model on our own.

$$P_{ijk} = \frac{(\tau_{ijk})^{\alpha}(\eta_{ijk})^{\beta})}{\sum_{j\in N, k\in K}(\tau_{ijk})^{\alpha}(\eta_{ijk})^{\beta}} \tag{11}$$

In every generation each of m ants constructs a solution. It iteratively chooses a random job and places it on a random machine at the first empty position. The probability that the ant makes a step, which means placing job j after job i on machine k is defined in (11), where τ_{ijk} is the amount of pheromone on that choice. We also introduce the heuristic value η_{ijk}, as proposed in [9] which is the cost calculated for execution of job j on machine k in this case. Exponents α and β are constants determining the relative influence of respectively the pheromone value and the heuristic on the decision of the ant.

After all ants have finished their paths some of the old pheromone is evaporated by multiplying it with a factor $0 < \rho < 1$ (eq. 12). This avoids the old pheromone from having too strong influence on future decisions.

$$\forall_{i,j \in N, k \in K} \ \tau_{ijk} \leftarrow \tau_{ijk} \cdot \rho \tag{12}$$

Finally m_b best ants, where $m_b \leq m$ add some pheromone to every step they have made on their tours. The amount of pheromone applied to a single step is equal Q/T_c, where T_c is the objective function value of the solution found by the ant and Q is the objective function value of the solution found by LPT-MM, described in Fig. 1 multiplied by a constant value λ (eq. 13). To prevent from reducing some steps to 0 and making the probability of other ones too large we define the minimum (τ_{min}) and maximum (τ_{max}) pheromone values of each step. The block diagram is presented in Fig. 2d.

$$\forall_{i,j \in N, k \in K} (x_{ijk} = 1) \ \tau_{ijk} \leftarrow \tau_{ijk} + \frac{Q}{T_c} \tag{13}$$

4 Investigations

Our point was first to perform the calibration of the internal input parameters for all four metaheuristics. After this job done, we tried to inspect the efficiency variation of the algorithms for different instance sizes. All tests were performed on a workstation with an Intel® Celeron® CPU operating at 2.66GHz clock speed with 1024 Mbytes of physical memory. The computer was equipped with Microsoft® Windows® Server 2003 Enterprise Edition operating system and .NET Framework 2.0.

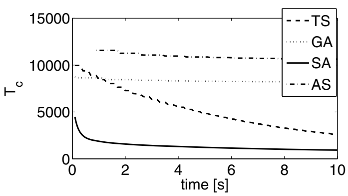
The finetuned parameters of all four considered algorithms for example instances are presented in Tab. 1 (for $t_C = 10s$ and $t_I = 5s$). The calculations time t_C is the amount of time after which the algorithm is stopped unconditionally. The interruption time t_I is the amount of time after which the algorithm is interrupted only if there is no improvement of the objective function of the best solution found thus far.

After the finetuning phase is completed we can proceed to measure the sensitivity of the four developed algorithms to the size of input instances with the calibrated internal parameters which will guarantee their highest possible efficiency. The size of an instance is described by two factors: jobs number N and machines number K. Most conducted research involved 100 repetitions of each algorithm with the same input characteristics to calculate the mean value of the objective function. Only for some larger instances which needed ten minutes and more to finish a single optimization experiment that number was reduced to 50.

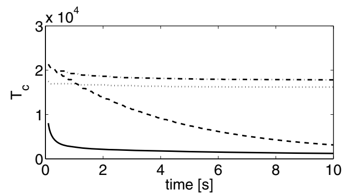
The average evolution of the objective function T_c for different instance sizes and $t_C = 10s$, $t_I = 5s$ is presented in Fig. 3. Decrease of jobs number is accompanied by the approach of tabu search quality to simulated annealing. It finally overtakes the SA for some time for $n = 50$. Since in all conducted experiments the simulated annealing algorithm was predominant over its rivals for the whole

Table 1. Finetuning results

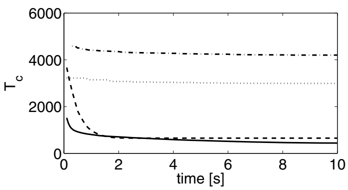
N	K	SA		TS	GA				ACO				
		T_0	$\delta[\%]$	Λ	G	M_s	M_b	X	τ_{min}	τ_{max}	ρ	m	β
150	15	13	85	3	28	0.15	0.14	1.00	100	1200	0.85	10	0.5
	10	16	94	5	30	0.09	0.12	1.00	100	1000	0.80	10	0.5
	5	17	94	8	32	0.14	0.08	1.00	150	1100	0.85	12	0.5
100	15	23	95	8	32	0.08	0.10	1.00	200	800	0.90	24	0.5
	10	18	96	12	38	0.16	0.13	0.95	100	1000	0.81	41	0.5
	5	23	96	14	36	0.12	0.14	1.00	150	1200	0.75	38	0.5
50	15	20	97	11	33	0.10	0.13	1.00	150	1200	0.90	37	0.5
	10	21	95	12	34	0.09	0.13	1.00	100	1100	0.80	36	0.5
	5	20	96	12	39	0.10	0.10	0.97	100	1000	0.78	48	0.5



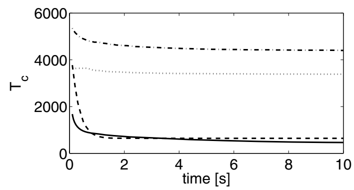
(a) 100 jobs & 10 machines



(b) 100 jobs & 5 machines



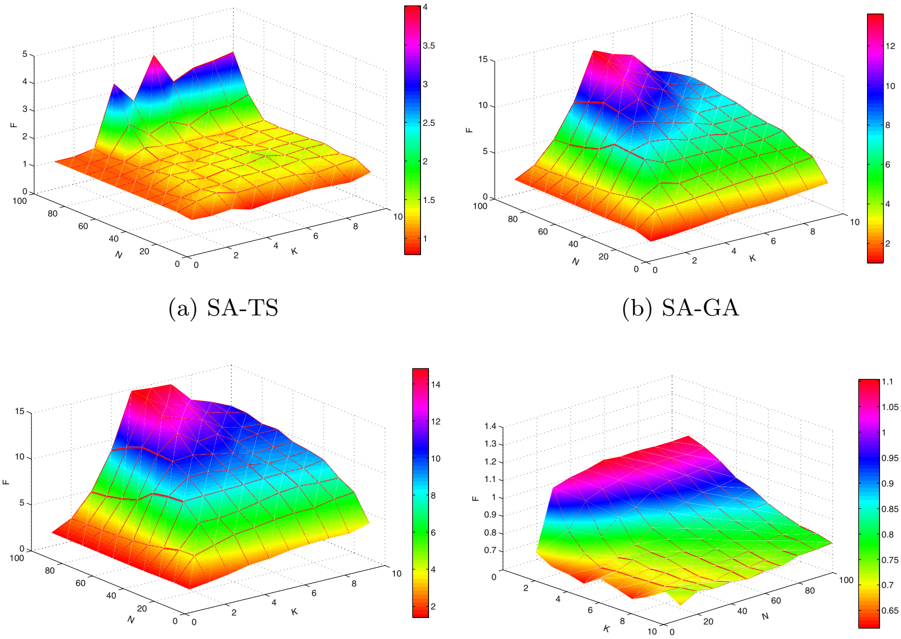
(c) 50 jobs & 10 machines



(d) 50 jobs & 5 machines

Fig. 3. Objective function evolution

time, we will use its current objective function value as a reference point for comparing the three other ones. Hence the quality of TS, AS and GA will be expressed as the proportion of their objective function to the objective function of SA. Superiority of SA over three other algorithms is presented in Fig. 4abc, and superiority of AS over GA in Fig. 4d where F is the proportion mentioned above, m is the number of machines and n is the number of jobs. Figures show that the implemented algorithms are strongly sensitive to the instance size but this sensitivity shows some level of linearity and therefore seems to be predictable. Both local search algorithms proove to be more suitable for the examined range of parameters. The quality of evolutionary ones is comparable.



(a) SA-TS (b) SA-GA

(c) SA-AS (d) AS-GA

Fig. 4. Efficiency margin of the metaheuristics

The proportion of TS to SA quality for 10s of calculations is maintained in the level of approx. 1.3 for all instances except the ones with 90 jobs and more (Fig. 4a). This is caused by rapid growth of calculations for the tabu search.

For the genetic algorithm, its efectiveness comparable to the SA is remarkable only for $K = 1$. However, there is also an interesting increase of quality for $K \geq 2$ (Fig. 4b). Low values of F for $N = 10$ is irrelevant since 10s of calculations is definitely enough to find the optimal or near-optimal solution in this case. The quality of GA starts to raise for $K \geq 5$. The ant colony optimization seems to be dependent upon both jobs and machines number, too. It differs from the GA in the location of the weakest point which is also smoother (Fig. 4c). The margin between the quality of AS and GA shows some continuity (Fig. 4d). Superiority of the ant system emerges linearly for decreasing K and increasing N.

5 Conclusions

In this paper we have considered the PMJS_E/T problem. Received results indicate strong influence of the number of jobs and the number of machines on the effectiveness of all used metaheuristics. This influence shows a high level of linearity in some areas of examined parameters.

Using metaheuristics is very promising - they allow shortening time of calculations while giving satisfactory solutions. On the basis of investigations, authors may recommend the simulated annealing algorithm for solving the problem considered in this paper. Some of the implemented algorithms are based on agent technology (AS), confirming efficiency of that approach for some particular cases.

In the further research authors are planning to concentrate on incorporating the sequence dependent job setup times to make the problem more adaptable to real life situations. Another essential direction of our work will be to introduce machine idle times which could be very important for cost reduction, especially for instances consisting of jobs with relatively large due dates.

References

1. Bazyluk, M., Koszalka, L., Burnham, K.: Using heuristic algorithms for parallel machines job scheduling problem. Computer Systems Engineering, 9–29 (2008)
2. Bullnheimer, B., Hartl, R., Strauss, C.: An improved ant system algorithm for the vehicle routing problem. Annals of Oper. Research 89, 319–328 (1999)
3. Cao, D., Chen, M., Wan, G.: Parallel machine selection and job scheduling to minimize machine cost and job tardiness. Comp. & Oper. Research 32, 1995–2012 (2005)
4. Davis, L.: Handbook of genetic algorithms. Van Nostrand Reinhold, New York (1991)
5. Dorigo, M., Gambardella, L.: Ant colonies for the travelling salesman problem. BioSystems 43, 73–81 (1997)
6. Garey, M., Tarjan, R.: One-processor scheduling with symmetric earliness and tardiness penalties. Mathematics of Operations Research 13, 330–348 (1988)
7. Liao, C., Juan, H.: An ant colony optimization for single-machine tardiness scheduling with sequence-dependent setups. Comp. & Oper. Research 34, 1899–1909 (2007)
8. Maniezzo, V., Colorni, A.: The ant system applied to the quadratic assignment problem. Knowledge and Data Engineering 11, 769–778 (1999)
9. Middendorf, M., Reischle, F., Schmeck, H.: Multi colony ant algorithms. Journal of Heuristics 8, 305–320 (2002)
10. Sivrikaya-Serifoglu, F., Ulusoy, G.: Parallel machine scheduling with earliness and tardiness penalties. Computers & Oper. Research 26, 773–787 (1999)
11. Sun, H., Wang, G.: Parallel machine earliness and tardiness scheduling with proportional weights. Computers & Operations Research 30, 801–808 (2003)
12. Yano, C., Kim, Y.: Algorithm for a class of single machine weighted tardiness and earliness problems. European Journal of Operational Research 52, 167–178 (1991)

An Algorithm for Inconsistency Resolution in Recommendation Systems and Its Application in Multi-Agent Systems

Michał Malski

Institute of Computer Science, The State Higher Vocational School in Nysa,
Armii Krajowej 7 Street, 48-300 Nysa, Poland
malskim@pwsz.nysa.pl

Abstract. This paper presents an algorithm for resolving inconsistency in Web-based recommendation systems and its application in multi-agent systems. This consensus-based algorithm has been presented in the previous paper and describes how it is possible to propose concrete (one of all possible) way of usage of the system (usage path) for individual user. In this work the statistical verification of the algorithm is presented which confirms its effectiveness. Some results of the experiments are also included and a conception for its application in multi-agent systems is presented.

1 Introduction

Recommendation systems are more and more popular in our life. These systems allows to formulate the sequence of products in offers, which are dynamically prepared for individual user basing on his profile, list of bought products in past by this user and another users or other information. For example, using of these systems allows to formulate the sequence of products in offers, which are dynamically prepared for an individual user basing on his profile; the list of bought products in the past by this user etc. Some example with this kind of conception is presented in [6]. Recommendation systems can also be useful in personalization and building interfaces for Web-service. A conception for Tax Declaration Making system has been described in [7], which helps a user to make the tax declaration in the most suitable way. That paper also includes the description of conflicts (inconsistency), which often appears in such systems and the conception for its solution. By employing recommendation systems, creating individual solutions, individual offers, price for a user (or a potential client) the enterprise can build special kind of user profiles.

Conflict resolution methods in recommendation systems are often needed. The reason of this statement is that recommendation systems often use knowledge originating from differences. The Tax Declaration Making system mentioned above is a good example of this case. In this system users are clustered into groups on the basis of their usage paths. A group consists of those users who have similar usage paths. A new user is first classified to a group, and next the system determines the best user path for him on the basis of usage paths of the users belonging to this group. These

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 356–366, 2009.

paths in turn can be different with each other, that we have to deal with some inconsistency which requires to be solved.

In this paper the mentioned above conflict and the algorithm for its solution is presented. In the next section the description of the algorithm is included. A conception for using this algorithm in multi-agent systems is described in Section 3. Section 4 presents statistical verification of the algorithm. This section shows the results of experiments executed using program in which this algorithm is implemented. These experiments and their results are differentiated by the number of positions in the usage path and by the number of given usage paths before solving the conflict problem. On their basis the statistical verification has been performed, which proves the effectiveness of the proposed algorithm.

2 Algorithm for Resolving Inconsistency in Recommendation Systems

In recommendation systems we can group users using their behavior. For example in the Tax Declaration Making system described in [7] we can indicate group of users, who have to fill in the same tax forms. Each member of such a group can have his own way of filling in the form (called *usage path*). A usage path can be represented by a vector whose components are the indexes of the positions in the tax form. For example, assuming that the set of position indexes is $\{1,2,...,5\}$ and two users, the user prefers following sequence $\langle 5,2,4,1,3 \rangle$ while the second user prefers $\langle 4,3,5,2,1 \rangle$.

A problem occurs when a new member is added to the group. This member does not have own usage path so the system try to give him a proposal to use the system. Preparation of new usage path is not simple task, because it should be a path as similar as possible to the usage paths of the "old" members of this group. This similarity can be represented by the distance between new usage paths, but we have to deal with some kind of conflict.

The distance function can be defined similarly as in [7], which is based on the minimal weight of position indexes shifts needed for transforming one sequence into other. For example, for given 2 sequences: $c_1 = \langle 5,2,4,1,3 \rangle$ and $c_2 = \langle 4,3,5,2,1 \rangle$ the weight for shifting index 1 is equal 1; for index 2 – 2; for index 3 – 3; for index 4 – 2; and for index 5 – 2. Thus the distance is equal $d(c_1, c_2) = 1+2+3+2+2 = 10$.

The solution described in [7] of this problem is based on consensus methods. If the universe of all potential usage paths is denoted as U then for given usage paths: $c_1, c_2, ..., c_n$ of all n members of the same group we can determine usage path c^* for new user using following criterion [8, 9]:

$$\sum_{i=1}^{n} d(c^*, c_i) = \min_{c \in U} \sum_{i=1}^{n} d(c, c_i).$$

The criterion requires the new usage path to be one of sequences from U, which guarantees the minimal sum of distances between chosen sequence and given sequences. It is possible to solve optimization problem of by checking each of elements of U, but this solution is not effective and very time-consuming. For usage paths consist of n positions the number of elements of set U is equal $n!$ and for each of these elements we have to calculate the sum of distances to elements of $C = \{c_1, c_2, ..., c_m\}$. More effective algorithm presented in the earlier paper [6] is given below.

Input: Set of positions $A = \{1,2,...,n\}$, set of usage paths $C = \{c_1, c_2, ..., c_m\}$, where sequence $c_i = \left\langle c_i^1, c_i^2, ..., c_i^n \right\rangle$, $c_i^j \in A$ and c_i^j is unique in c_i.

Output: Sequence c^* for which $\sum_{i=1}^{m} d(c^*, c_i) = \min_{c \in U} \sum_{i=1}^{m} d(c, c_i)$.

BEGIN

1. create matrix M of size $n \times n$, which all elements are equal 0
2. for each of elements of C
   for each of elements of A
   add one to element $m_{i,j}$ of matrix M if position of position number i in the element of C is equal j
3. create matrix D (matrix of distances) of size $n \times n$ of elements calculated by formula:
$$d_{i,j} = \sum_{m=1}^{m} \left[m_{i,m} \cdot |m - j| \right]$$
4. n times do:
   - find first minimal element of matrix D, which value is greater or equal 0 and remember its number of row as $i_{\min}$ and number of column as $j_{\min}$
   - calculate $c^*_{j_{\min}} = i_{\min}$
   - change to -1 values of all elements of D in the row with number $i_{\min}$ and in the column with number $j_{\min}$
5. recreate matrix D such like in 3
6. calculate sum of the distances between c^* (appointed in 4) an each of elements of C as following:
$$\sum_{i=1}^{m} d(c^*, c_i) = \sum_{j=1}^{n} d_{(c^j)^*, j}$$

END.

Now it is possible to give an example of solving inconsistency using proposed algorithm. At the beginning of a counting, we have to build a matrix M. It's elements tell us how many times particular position has taken a stance on particular position in a set of particular sequences. In addition to this, the matrix M will be a square matrix of size $n \times n$, if we have n positions. For example of that, imagine that 20 is equal of an element $m_{i,j}$ of a matrix M, as we can see position i has taken a stance on position j twenty times in a set of particular sequences.

We can build the matrix M of size 9×9 and

$$M=\begin{bmatrix} 2 & 0 & 2 & 1 & 2 & 3 & 1 & 0 & 1 \\ 2 & 1 & 2 & 0 & 0 & 2 & 1 & 4 & 0 \\ 1 & 0 & 3 & 2 & 1 & 0 & 2 & 3 & 0 \\ 1 & 2 & 0 & 1 & 2 & 1 & 2 & 2 & 1 \\ 2 & 2 & 1 & 1 & 1 & 3 & 1 & 0 & 1 \\ 0 & 0 & 0 & 3 & 1 & 1 & 2 & 2 & 3 \\ 1 & 1 & 3 & 1 & 2 & 1 & 1 & 1 & 1 \\ 3 & 3 & 1 & 1 & 2 & 0 & 0 & 0 & 2 \\ 0 & 3 & 0 & 2 & 1 & 1 & 2 & 0 & 3 \end{bmatrix}$$

when we have 9 positions $\{1,2,...,9\}$ and example of set of twelve sequences:

$$c_1=\langle 7,4,1,6,3,5,9,2,8\rangle,\ c_2=\langle 8,9,3,6,5,7,4,2,1\rangle,\ c_3=\langle 2,8,1,3,7,9,6,4,5\rangle,$$
$$c_4=\langle 5,8,3,7,4,1,9,2,6\rangle,\ c_5=\langle 1,2,3,9,8,5,4,6,7\rangle,\ c_6=\langle 8,5,2,1,6,4,7,3,9\rangle,$$
$$c_7=\langle 4,5,2,9,8,1,3,7,6\rangle,\ c_8=\langle 2,9,7,6,1,5,3,4,8\rangle,\ c_9=\langle 1,7,5,8,4,2,6,3,9\rangle,$$
$$c_{10}=\langle 3,8,7,4,9,1,5,2,6\rangle,\ c_{10}=\langle 5,9,8,3,7,2,1,6,4\rangle \text{ and } c_{12}=\langle 8,4,7,5,1,6,2,3,9\rangle.$$

The next step what we can do, is preparing matrix D. The number of positions will be represented by rows, moreover the number of position in a sequence will be represented by columns, the same as in matrix M. Below we can see formula to calculate each component of this matrix:

$$d_{i,j}=\sum_{m=1}^{n}\left[m_{i,m}\cdot|m-j|\right].$$

For matrix M we have, for example:

$$d_{1,1}=2*0+0*1+2*2+1*3+2*4+3*5+1*6+0*7+1*8=44,$$

because if we decide to use position 1 on first position into $c*$, so as can be seen we have:

- 2 sequences with position 1 on first position and distance between positions of position 1 is equal 0,
- 0 sequences with position 1 on second position and distance between positions of position 1 is equal 1,
- 2 sequences with position 1 on third position and distance between positions of position 1 is equal 2,
- 1 sequence with position 1 on fourth position and distance between positions of position 1 is equal 3,
- 2 sequences with position 1 on fifth position and distance between positions of position 1 is equal 4,
- 3 sequences with position 1 on sixth position and distance between positions of position 1 is equal 5,

- 1 sequences with position 1 on seventh position and distance between positions of position 1 is equal 6,
- 0 sequences with position 1 on eighth position and distance between positions of position 1 is equal 7,
- 1 sequences with position 1 on ninth position and distance between positions of position 1 is equal 8.

Basing on matrix M from example above, matrix D gets following values:

$$D = \begin{bmatrix} 44 & 36 & 28 & 24 & 22 & 24 & 32 & 42 & 52 \\ 49 & 41 & 35 & 33 & 31 & 29 & 31 & 35 & 47 \\ 49 & 39 & 29 & 25 & 25 & 27 & 29 & 35 & 47 \\ 52 & 42 & 36 & 30 & 26 & 26 & 28 & 34 & 44 \\ 40 & 32 & 28 & 26 & 26 & 28 & 36 & 46 & 56 \\ 68 & 56 & 44 & 32 & 26 & 22 & 20 & 22 & 28 \\ 44 & 34 & 26 & 24 & 24 & 28 & 34 & 42 & 52 \\ 32 & 26 & 26 & 28 & 32 & 40 & 48 & 56 & 64 \\ 54 & 42 & 36 & 30 & 28 & 28 & 30 & 36 & 42 \end{bmatrix}.$$

When we have above matrix which is the matrix of distance, we will be able to calculate sum of distances between particular sequence and a set of sequences. Now we have:

$$\sum_{i=1}^{m} d(c, c_i) = d_{c^1,1} + d_{c^2,2} + \ldots + d_{c^n,n} = \sum_{j=1}^{n} d_{c^j,j}$$

for given $c = \langle c^1, c^2, \ldots, c^n \rangle$ and set of m sequences.

Matrices M and D allow us to describe above example with 12 sequences. All we need to do, is calculating sum of distance between some sequence and set of these 12 sequences. Let $c = \langle 6,1,5,7,4,9,8,2,3 \rangle$, then

$$\sum_{i=1}^{12} d(c, c_i) = d_{6,1} + d_{1,2} + d_{5,3} + d_{7,4} + d_{4,5} + d_{9,6} + d_{8,7} + d_{2,8} + d_{3,9} =$$
$$= 68 + 36 + 28 + 24 + 26 + 28 + 48 + 35 + 47 = 340.$$

The optimization problem is now reduced to the problem of choice of n elements of matrix D, for which the sum will be minimal. Because of the fact that we want to have sequence of all positions as a result, one of the most important things is choosing exactly one element from each row and each column of matrix D, so position can not be repeated and position can not be the same for two or more positions.

The best what we can do is finding first minimal element of matrix D and remembering number of it's row and column. Then into c^* we have to put position with the same number as minimal element's number of row, what is more, we have to do it on position with the same number as minimal element's number of column. We need to remember that before next searching, we have to eliminate minimal element's row and column. These three actions should be repeated n times to determine c^*.

If we choose sequence: $d_{6,7}$, $d_{1,5}$, $d_{7,4}$, $d_{8,2}$, $d_{4,6}$, $d_{5,3}$, $d_{2,8}$, $d_{9,9}$ and $d_{3,1}$ from matrix D, we are able to create $c^* = \langle 3,8,5,7,1,4,6,2,9 \rangle$ and what is more, it is possible to calculate the sum of distance between c^* and set of given 12 sequences:

$$\sum_{i=1}^{12} d(c^*, c_i) = d_{6,7} + d_{1,5} + d_{7,4} + d_{8,2} + d_{4,6} + d_{5,3} + d_{2,8} + d_{9,9} + d_{3,1} =$$
$$= 20 + 22 + 24 + 26 + 26 + 28 + 35 + 42 + 49 = 272.$$

The program, which works according to described algorithm, first calculates c^* as it is described in this Section and next generates all possible sequences of numbers from 1 to 9. Next program checks all 362880 sequences to find out sequence c^*_{op}, which is optimal solution, so grants dependencies below:

$$\sum_{i=1}^{m} d\left(c^*_{op}, c_i\right) = \min_{c \in U} \sum_{i=1}^{m} d(c, c_i).$$

In the ideal situation $c^*_{op} = c^*$, but usually this situation has no place and

$$\sum_{i=1}^{m} d\left(c^*_{op}, c_i\right) < \sum_{i=1}^{m} d\left(c^*, c_i\right).$$

For the above example we have $c^*_{op} = \langle 8,5,7,3,1,4,9,2,6 \rangle$ and $\sum_{i=1}^{12} d\left(c^*_{op}, c_i\right) = 256$.

However, it is possible to be satisfied, because the relative error is small what is presented in Section 4.

3 Inconsistency Resolving Algorithm for the Multi-Agent System

As we can get to know form the literature about multi-agent systems (MAS), modern power of computers is not realized trough individual and separate systems but through connected, distributed and dynamic systems. Owing to this we have at disposal bigger computing power and better capabilities to modeling the real word. The systems of this type work in many domains of life, for example in teleinformatic network management processes [2, 5], in distributed air traffic systems [3, 4], in enterprise to simulate models of production processes and making analyses and simulations, which help to manage the firm [1, 10].

MAS should contain many agents, which interact together to achieve common purposes. In many systems of this type the agent is realizing composite task by dividing this task on many simple task, which transfer to realization to others interacting agents or agents form his environment. In such system each of tasks should be realized by

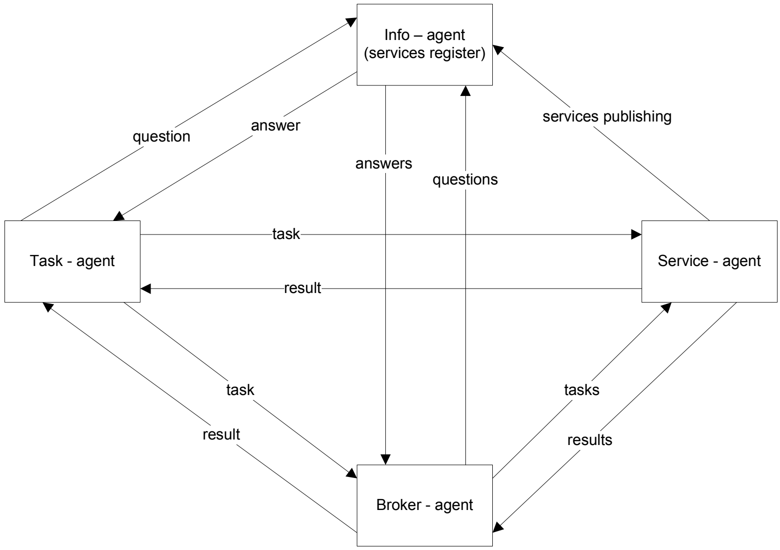


Fig. 1. Service Oriented Multi-Agent System

several agents and results of their working should be compared. In this situation the question is which agents from the set of possible agents can realize concrete task.

In the Service Oriented Multi-Agent Systems (SO MAS) if some agent has a simple task to realization the he sends a request to special agent named info-agent. In the answer info-agent inform agent (which is named task-agent) about agents (service-agents), which can realize his task and then agent can send this task to realization to one or more suitable service-agents. In situation, when task-agent has to realize composite task, he send this task to broker-agent. This agent divide task to many simple task and then works like many task-agents with simple task in way described above. Both situations are presented on the figure blow.

As it is suggested in this paper in MAS one task should be realized by several agents. Each agent returns to the principal agent the result of his realization. Thus inconsistency can occur because the results generated by the agents can be different with each other. In this paper we want to show how the algorithm described in Section 2 can help us to solve mentioned election problem. Let's consider an example.

For given set of five agents $\mathrm{A} = \{1,2,...,5\}$, which can realize the same type of task and given two hundred different tasks of this type, which result is known, we can create set $C = \{c_1, c_2, ..., c_{200}\}$ of sequences $c_i = \left\langle c_i^1, c_i^2, ..., c_i^5 \right\rangle$, $c_i^j \in A$ and c_i^j is unique in c_i. For example, we create c_i by putting 3 as element c_i^1 (first position) if agent number 3 returns the best result after i task realization. By the same way we put 1 as element c_i^5 (last position) if agent number 1 returns the worst result in the same task realization process. After he have C set created it is possible to create matrices M and D, such it is in algorithm. By realizing step number 4 of algorithm we receive new sequence and on first position of this sequence we have number of agent, which results of next tasks should be recognized as the best. This agent is recommended as the best to realize such type of tasks.

In the next part of this section we want to give another example of using this algorithm in connection with MAS. This time we propose to use it in different place. In exemplary situation when we have seven projects (it may be different projects of building of shopping center), we can use can use seven multi-agent systems to calculate these projects. Each of these systems will return as many calculated parameters related with processed project. These parameters will decide about choice from among projects, for example: forecasted cost of realization, cost of materials, time of realization, parameters related with human resources needed to realization, number of retail units, surface and cubature of units, yield of infestation, etc. It is suggested that systems should cooperate during project analyzing to perform results, which can be compare. For example during the cost calculating systems have to use the same source of information about prices. If we have such parameter like time of realization we can create sequence of seven numbers of agent systems order by results related with this parameter in ascending order. In situation when MAS numbered 1, which is analyzing project number 1, return fourth forecasted time of project realization, number 1 will be placed on fourth position in this sequence. If we do the same with another 29 parameters, which decide in process of choosing project, we will have set of 30 sequences $C = \{c_1, c_2, ..., c_{30}\}$. Now we can realize our algorithm and in the effect of his working one more sequence $c*$ will be received. This sequence is special proposition (recommended sequence), which informs investor about suggested sequence of treat projects (which project should become chosen).

The results of experiments executed using program, which works according to described algorithm will be show and statistically verify in the next section.

4 The Statistical Verification of Presented Algorithm

Some results of experiments with described algorithm and preliminary conclusions has been presented in previous paper [6]. This paper presents the results of the statistical verification. For its need we performed 900 tests of program implementing this algorithm. These results have been generated for different size of sequences (s) (different number of positions) and different number of randomly generated sequences. For each pair *s* and *n* the program made 30 tests and each of them returns 3 parameters: the distance between randomly generated sequences and $c*$ calculated by the algorithm; the distance between these sequences and optimal solution c_{op}^* and relative error in percent calculated like in formula presented below:

$$rel.error = \frac{\sum_{i=1}^{n} d(c*, c_i) - \sum_{i=1}^{n} d\left(c_{op}^*, c_i\right)}{\sum_{i=1}^{n} d\left(c_{op}^*, c_i\right)} \cdot 100\% \quad [6].$$

In the table below we present only a part of mentioned experiments.

Table 1. Results of program functioning

s	n	$\sum_{i=1}^{n} d(c^*, c_i)$	$\sum_{i=1}^{n} d(c_{op}^*, c_i)$	rel.error
4	10	44	42	4,762
4	20	146	142	2,817
4	50	370	368	0,543
4	100	710	710	0
4	200	1572	1558	0,899
5	10	72	68	5,882
5	20	148	138	7,246
5	50	382	380	0,526
5	100	764	762	0,262
5	200	1536	1520	1,053
6	10	100	100	0
6	20	218	210	3,81
6	50	526	526	0
6	100	1112	1102	0,907
6	200	2262	2250	0,533
7	10	112	108	3,704
7	20	292	284	2,817
7	50	732	722	1,385
7	100	1522	1504	1,197
7	200	3110	3098	0,387
8	10	168	168	0
8	20	374	354	5,65
8	50	996	988	0,81
8	100	1938	1914	1,254
8	200	4084	4018	1,643

On the basis of the 900 results we can calculate averaged relative error ($\overline{\alpha}$) which is equal 2,649 and standard deviation (S) equal to 2,651 for our sample. For these parameters we can make statistical test to ensure how many percents results of proposed algorithm is worse then optimal solution. The *t-student* test could be suitable test for this situation, but it is applied when the sample is assumed to be normally distributed. In our example relative error does not have normally distribution, what we can check using normality tests, such as the Shapiro-Wilk test or Kolmogorov-Smirnow test. The central limit theorem allows for using this test in situation when the size of sample is greater then 100 and sample does not have normally distribution, so we can create our hypothesis and make test because size of our sample is equal to 900.

The following two hypotheses have been formulated:

H_0 : The averaged error related with proposed algorithm is equal to 3%.

H_1 : The averaged error related with proposed algorithm is less then 3%.

Now we should to calculate t (the statistic value of t student test), according to the formula presented below:

$$t = \frac{\overline{\alpha} - \mu_0}{S}\sqrt{n-1},$$

where $\overline{\alpha}$ and S are known parameters, μ_0 is expected value of averaged relative error and n is the sample size. For our results of experiments we have:

$$t = \frac{2{,}649 - 3}{2{,}651}\sqrt{900-1} = 3{,}970.$$

Usually for t student test it is assuming that the statistical significance level is equal 0,1; 0,05 or 0,01. To be sure that the result of our test is very good the value of this level equal to 0,001 is assumed. The critical value of Student's t distribution is equal -3,301 for 899 degrees of freedom and statistical significance level equal 0,001. Our t value is less than this critical value so we can reject null hypothesis (H_0) and accept hypothesis H_1. As the conclusion we can say that proposed algorithm is alternative for optimal solution because the averaged relative error is less than 3 percent and our algorithm is more effective and faster then algorithm of searching for optimal solution.

5 Conclusions

Nowadays, the dynamically fitting to user's requirements and expectations software and Web-pages are very popular. In this paper it has been shown how the algorithm proposed for such systems can be implemented and how it can help to solve inconsistency problem in Multi-Agent Systems. The results of experiments with this algorithm and statistical verification have been presented.

References

1. Frey, D., Nimis, J., Wörn, H., Lockemann, P.: Benchmarking and Robust Multi agent-based Production Planning and Control. Engineering Applications of Artificial Intelligence - The International Journal of Intelligent Real-Time Automation (EAAI) 16, 307–320 (2003)
2. Fugkeaw, S., Manpanpanich, P., Juntapremjitt, S.: Multi-Application Authentication Based on Multi-Agent System. IAENG International Journal of Computer Science, 37–42 (2007)
3. Gorodetskiy, V., Karsaev, O., Samoilov, V., Skormin, V.: Multi-Agent Technology for Air Traffic Control and Incident Management in Airport Airspace. In: Proceedings of the International Workshop Agents in Traffic and Transportation, Portugal, pp. 119–125 (2008)
4. Hill, J.C., Archibald, J.K., Stirling, W.C., Frost, R.L.: A Multi-Agent System Architecture for Distributed Air Traffic Control. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, pp. 1–11 (2005)
5. Kussul, N., Shelestov, A., Sidorenko, A., Skakun, S., Veremeenko, Y.: Intelligent multi-agent information security system. In: Proceedings of the Second IEEE International Workshop Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Ukraine, pp. 120–122 (2003)

6. Malski, M.: An Algorithm for Inconsistency Resolving in Recommendation Web-based Systems. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS, vol. 4252, pp. 251–258. Springer, Heidelberg (2006)
7. Malski, M.: Resolving inconsistencies in recommendation Web-based systems. In: Proceedings of the 11th System Modelling Control Conference (SMC 2005), pp. 189–194 (2005)
8. Nguyen, N.T.: Methods for resolving conflicts in distributed systems. Monograph. Wroclaw University of Technology Press (2002)
9. Nguyen, N.T.: Consensus System for Solving Conflicts in Distributed Systems. Journal of Information Sciences 147, 91–122 (2002)
10. Peng, Y., Finin, T., Labrou, Y., Chu, B., Long, J., Tolone, W., Boughannam, A.: A Multi-Agent System for Enterprise Integration. Journal of Applied Artificial Intelligence, 323–330 (1998)

Statistical Strength of a Hybrid Cryptosystem on 2-D Cellular Automata with Nonaffine Rotations*

Marian Srebrny[1] and Piotr Such[2]

[1] Dept of Technical Sciences, University of Commerce, Kielce, Poland; and Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
marians@ipipan.waw.pl
[2] University of Commerce; Kielce, Poland
piotr@aprsystem.com.pl

Abstract. In this paper we present some basic evaluation of quality of the multicore (or multiagent) cryptosystem 2DCARotate according to the statistical tests suite recommended by the US National Institute of Standards and Technology (NIST), as well as to a new test by the grade analysis method which has recently been developed at the Institute of Computer Science of the Polish Academy of Sciences. The paper reports some results of these tests for our 2DCARotate based on a nonaffine Boolean function of rotation combined with three cellular automata (viewed as systems of many agents interacting upon neighborhood environment), in comparison with 3DES and AES-Rijndael which are currently the world's best symmetric ciphers. Against such competition our cipher fares surprisingly well.

Keywords: multi-agent security, intelligent multicore computation, cellular automata, symmetric cipher, cryptology, random bit generators, statistical tests.

1 Introduction

This paper is devoted to the security aspect of multi-agent systems. We study a particular powerful model of intelligent multi-agent paradigm based on the classical concept of a cellular automaton, with its cryptographic strength enhanced by using some nonaffine geometric operations. The cellular automata seems well suited as a case study computational infrastructure of the multi-agent collaborative systems paradigm. This paper illustrates how this paradigm based (implemented) on cellular automata can be used to design high quality security software (and FPGA hardware).

The main goal of this paper is to present the results of statistical tests, following the NIST recommendations [8], verifying the quality of a certain encryption system. Statistical testing can be considered an essential preliminary step in validation of encryption algorithms, their correctness and security. We refer the interested reader to [8] for a detailed description of the tests that we have run. The cryptograms output by a good

* The authors gratefully acknowledge insightful conversations with Professors E. Pleszczyńska and W. Szczesny and their advice to test the ciphers using the grade analysis method.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 367–379, 2009.

encryption algorithm should exhibit random sequence properties, since any deviations from random bit distribution could be used as a convenient starting point for various attacks aimed at breaking the cryptosystem. A cipher is considered broken when a decryption method is found that is feasible for unauthorized parties, who do not have the appropriate keys.

Our cryptosystem CA2DRotate combines a tandem of three cooperating 2-dimensional cellular automata with a number of nonaffine transformations of rotating squares of automata agents state bits, in a way that significantly increases the complexity of conceivable attacks against such a cryptosystem. Here we look not only at the three automata as collaborating agents, but also - what seems much more significant - each cell of every such automaton is in fact an autonomous agent reacting upon predefined neighborhood environment. Furthermore, in our approach certain coalitions of the agents are rotated, in such a way that they can be moved to different neighborhood environments.

Introducing a carefully designed nonaffine component transformation seems to add some significant enhancement to the cryptosystems on cellular automata which have been studied so far. Although, we are aware that this idea needs deep considerations, starting with the statistical tests to see whether they show any regularities which can be used to attack the system. In linear cryptanalysis, the attacker tries to approximate a nonlinear operation performed by the cryptosystem with a linear one. This is always possible if one considers an appropriately large (that is, affecting many bits) linear operation; however, if the linear operation is too large then the attack becomes infeasible. The crucial observation is that a cryptosystem may be vulnerable to a linear cryptanalysis attack even if it produces good pseudorandom sequences. Hence, studying the statistical properties of the cryptograms produced by the system is a first preliminary step towards assessing the security properties of the system. Passing statistical test is a necessary, although not sufficient, condition that all cryptographically strong ciphers must meet.

The contribution of this paper is a newly finetuned version of symmetric cryptosystem 2DCARotate, an idea of which was announced in [10], and the experimental results of the tests run on 2DCARotate are presented, with their comparison with the results on four well-known ciphers - DES [5], 3DES [6], AES-Rijndael [1] and Blowfish [9]. The good results achieved by our 2DCARotate cipher are a pleasant surprise for the authors. We have received a number of suggestions since [10] appeared which resulted in major changes and accuracy enhancements. In fact, we have tuned the 2DCARotate version (presented in this paper) following a series of experiments with the statistical tests.

The paper is organized as follows. In Section 2 the necessary background of the cellular automata is reviewed briefly and the design of our 2DCARotate cipher is presented (in a new version tuned after a series of the statistical tests) - the discussion is necessarily brief. In Section 3 we present the results of the NIST [8] recommended tests for 2DCARotate, DES, 3DES, AES-Rijndael and Blowfish, preceded with a very short methodological explanation. In Section 4 the results of grade analysis are presented. The final section 5 gives some conclusions and indications for future work.

2 Cryptosystem on 2D Automata with Rotations

The concept of a cellular automaton, CA, was introduced by John von Neumann [7], as a model of concurrent computation, providing an alternative to Turing machines. Von Neumann also presented an intriguing self-replicating automaton. In the mid-1980s, Stephen Wolfram [17] started to apply cellular automata in cryptology. In the early 1990s, on the basis of Wolfram's work, Howard Gutowitz [2] constructed an interesting cryptosystem based on cellular automata. In [10] we have proposed a blueprint for a new cipher based on 2D cellular automata. The application of 2D cellular automata with rotations to encrypt information is a new solution. The algorithm presented here may be implemented on multicore devices, which would enhance its performance. The use of irreversible rules to generate links additionally strengthens the algorithm. Relatively simple geometric transformations applied to squares yield a nonaffine effect similar to substitutions based on the famous substitution boxes (S-boxes).

2.1 Basic Concepts

A cellular automaton may be viewed as a one- or multi-dimensional array which consists of cells (agents). Cells contain data coded as zeroes or ones. (Viewed as agents, they always are in one of the two possible states: zero or one.)

An automaton changes its states in the process of computation according to the so-called transition rules (protocol). The rules say how to transform a single cell from one state s_i to another state s_{i+1}.

In order to calculate the next state, the rule uses the central cell and adjacent cells which are called the neighborhood environment of a predefined geometric shape. A rule which causes a cell's transition to the next state may be expressed as a transition table. Rules may be reversible or irreversible. When it is possible to recapture uniquely the central bit from (upon) the resulting bit and the current state of the neighborhood, a rule is called reversible. When the same resulting bit is calculated for the same neighborhood and different central cell values, such a rule is called irreversible.

2.2 Encryption with 2D Automata

In this section we briefly give the basic design principles of our 2DCARotate encryption and decryption algorithms. The idea of the algorithms was outlined in [10]. It is a symmetric-key block cipher. The characteristic features of the system are: it uses three automata, two of which are irreversible; a neighbourhood has a special "protruding" toggle bit; a symmetric key which includes identifiers of the transition rule; additional diffusion of information through geometric transformations.

The encryption algorithm is performed on the array tape of 32x8 cells of a 2D automaton, which is called CACrypt. There are also two auxiliary automata: CALink (2x8 bits) and CATop (32x8 bits), which affect the operation of the main automaton, bringing irreversibility to the algorithm. The initial state (i.e., the bits in each cell) are generated from the key, and the next states, computed in the successive steps, are saved in the state vector. This accelerates the performance of the algorithm. The state vector contains 3 states (for each of the CALink, CATop, and CACrypt), what corresponds to the counter of the main loop in the algorithm. In a single run of the algorithm 32 bytes of data, entered to CACrypt, can be encrypted. A pseudocode for this algorithm is as follows. (Some explanations are given below.) The deciphering algorithm is very similar, working in the reverse direction.).

```
procedure Encryption
  //setup of reversible automata
  //generate the state vectors for the CALink
  and CATop automata
  //the main loop of the algorithm
  for k:=1 to 3 do
  begin
    //encryption phase
    for x:=1 to 32 do
      for y:= 1 to 8 do
      begin
       Compute the state of cell (x,y) in
       automaton CACrypt using CALink.
      end
    //toggle phase
    for x:=1 to 32 do
      for y:= 1 to 8 do
      begin
        join the neighbourhood from CATop with
        that from CACrypt for cell (x,y) and
        determine the next state of cell (x,y)
        of CACrypt.
      end
      //diffusion phase
      Rotate the squares.
      Advance by one the state of the CALink
      and CATop state vectors.
  end
```

2.2.1 Symmetric Key

The algorithm uses a symmetric key consisting of two parts. The first part consists of the initial states for the automata CALink, of size 2x8 bits, and CATop of size 32x8. The second part contains the rules for all the three automata:

1. The rules for CALink used to determine the state of the vectors for CALink. These rules are not reversible.
2. The rules for CATop used in the Toggle Phase and to determine the successive states of the automaton CATop. These rules are reversible.
3. The rules for CACrypt used in the Encryption Phase. These rules are reversible. These rules are also used to determine the number of rotation turns and their directions in the Diffusion Phase.

Further details concerning the structure and usage of the key are provided below where we describe the steps of particular phases of our algorithm 2DCARotate.

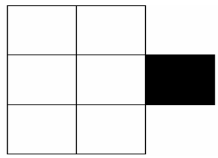


Fig. 1. Neighbourhood used in CACrypt

2.2.2 Encryption Phase

We use a neighbourhood in CAcrypt shaped as shown in Figure 1 below. The cell market black is considered central. Given a neighbourhood and a rule, by a toggle cell (Tc) we mean such a cell that any change of its state causes a change of the central cell. The toggle cell may not exist in a rule, in general. A rule is said to be a right- or left-toggle rule if a toggle cell exists for this rule to the right- or left-hand side, respectively, of the central cell. In Figure 1, the toggle cell is marked black. It is the same cell as the central one.

The key includes 32 rules. Each rule is applied in turn to CACrypt automaton columns. Since the neighbourhood consists of 7 bits, a rule includes 7x128 bits. The rules used here must be reversible. For each neighbourhood combination (excluding the toggle cell), we may have two states of the toggle cell, and thus also two states constituting the result of the rule in question.

The actions the algorithm performs when enciphering a single cell in the CACrypt automaton may be summarised as follows:

1. read the 6 bits from the neighbourhood;
2. based on these 6 bits and the toggle cell, lookup one of the 128 toggle rules;
3. calculate the 7th bit based on the rule - this will be the state of the cell in the next step.

The actions the algorithm performs when deciphering a single cell in the CACrypt automaton may be summarised as follows:

1. read 6 bits from the neighbourhood;
2. based on the 6 neighbourhood bits and the toggle cell, lookup the appropriate rule;
3. calculate the result of the rule thus selected and set the state of the cell accordingly.

For the first column of the CACrypt automaton, the neighbourhood read from the CALink automaton is used. For subsequent columns of the CACrypt automaton, the neighbourhood is shifted analogously to the right.

The size of the CALink automaton is 8x2. Both the initial state of the automaton and the rules are obtained from the key. The rules applied for this automaton are irreversible. Therefore we need to know the states of CALink automaton cells from previous steps during the deciphering process. This is ensured by recording the states of the automaton in its state vector. Before the deciphering process, the CALink automaton is reset to its initial state. The algorithm is run as a precomputation and individual states of the automaton are recorded in the state vector (which is an auxiliary table). During deciphering, this state vector is used to determine the previous state of the automaton.

2.2.3 Toggle Phase

In the second phase of the algorithm, all cells of the CACrypt automaton are toggled. The CATop automaton, whose role is to generate the bits used to construct the neighbourhood, is used to toggle the cells. The neighbourhood consists of 3x3 square of immediate neighbours of bits from the CATop automaton and a single central bit from the CACrypt automaton.

In order to calculate the next state of a cell (for example cell (4,3)) for the CACrypt automaton, the following steps need to be performed:

1. read the states of the cells adjacent to cell (4,3) from CATop;
2. read the state of cell (4,3) from CACrypt;
3. use the rule from the CATop automaton to determine the next state for the neighbourhood thus constructed.

The CATop automaton uses 32 rules, each of length of 512 bits. These rules are built in a way similar to the CACrypt automaton. The mode of operation of the CATop automaton is analogous to that of the CALink one. Similarly as for the CALink automaton, during the deciphering process the state vector consisting of 3 automaton states is first determined, and these states are subsequently used to determine the previous states of the CACrypt automaton.

2.2.4 Diffusion Phase

The third phase of the algorithm's operation consists of rotating 2x2 squares in order to additionally disperse the information. The automaton is divided into a grid of identical 2x2 squares. Within each grid element (single square) grid bits are rotated right or left according to the key. In order to rotate a single square we take the appropriate two bits from the key. For 00 no rotation is performed, for 01 one 90-degree right turn, for 10 – two right turns, for 11 – one left turn.

The rotations define nonaffine boolean functions with high nonlinearity. Our proof of it is beyond the scope of this short version of this paper. We refer the interested reader to [3]. Here, we can only say that in our setting the rotation function is defined on $GF(2)^6$ and has nonlinearity 32.

Such rotations are performed 32 times in a loop. For even loop steps, the grid is constructed in such a manner that the upper left corner of the automaton coincides with the upper left corner of the grid. For odd loop steps, the grid is shifted by one cell down and to the right. In this case, the grid does not coincide exactly with the automaton; "protruding" grid squares are filled with bits from the other edge of the automaton. This makes it possible for a bit from the upper left corner of the automaton to "wander" e.g. to the lower right corner. During deciphering, the rotations are performed in reverse order.

3 Some Results of the NIST Recommended Tests

This section presents the results of selected tests of the quality of our 2DCARotate encryption algorithm treated as a pseudorandom sequence generator and their comparison with the results of four well-known cryptosystems: DES, 3DES, AES-Rijndael and Blowfish. We start with some basic methodological explanation in a way accessible to the reader who may not be a professional in statistics.

In cryptography random (or rather pseudorandom) input data and parameters are used very often. Examples include input material for construction of cryptographic keys, cryptographic protocols, digital signatures, challenges in challenge-response authentication protocols, et cetera.

Current cryptanalysis methods, such as differential and linear cryptanalysis [4], often provide efficient methods for attacking encryption algorithms. In order to thwart these and other attacks, at least to a certain degree, tests concerning the randomness of the cryptograms generated should be conducted first. The most important feature should be the unpredictability of each subsequent ciphertext bit without the knowledge of the text (document) that has been encrypted. By ensuring that the ciphertexts produced by the algorithm have the properties of entirely random sequences, i.e. that the encryption algorithm has the properties of a random generator, we may believe that the algorithm is resistant to cryptanalysis using statistical techniques. It is natural to assume that a good encryption algorithm should output cryptograms that have the properties of random sequences.

On the basis of the suite of recommended tests [8] we selected and performed the most meaningful randomness tests. Based on the test results, the parameters and constants of our encryption algorithm were appropriately "tuned" in order to achieve optimum performance while preserving high-quality randomness properties of the ciphertext generator.

A finite amount of generator output from any generator will sometimes fail a test. Even chunks of random data from a truly random source will sometimes fail a test. The existence of failure samples does not mean that the generator is flawed. A cryptographically strong random number generator will infrequently fail a test.

The tests included in suite [8] are based on the determination of the so-called P-value, which is a synthetic, summarizing measure of information collected using statistical analysis which may constitute evidence that the generated ciphertext sequences do not have the desired randomness properties. P is a real number and may range from 0 to 1. The P-value 1 may be interpreted to mean that the sequence in question is perfectly random. The P-value 0 means that a sequence does not exhibit random features. Below, we assume that the P-value should be higher than 0.01. The P-value is the probability that a perfect random generator would have generated a sequence less random than the ciphertext tested given the kind of non-randomness assessed by the test, e.g. the presence of a recurring pattern. If we assume that the binary generator is a random one and set the acceptability threshold to 0.01, this means that we expect that on average, at most one out of a hundred samples tested may be rejected; i.e., may be non-random with respect to the property tested.

The DES, 3DES, Blowfish and AES-Rijndael ciphers were selected for comparison purposes. Exactly the same principles were adopted and parameters set for testing these ciphers in order to provide a common denominator for comparing their quality with our 2DCARotate cipher. Additionally, the data used for testing were the same as for the 2DCARotate cipher.

For each cipher, tests were run on 32 sequences. During each run, 1,000,000 bits of input data were encrypted according to the principles described in the Testing Strategy section of [8]. Since 1,000,000 bits equals 125 KB of data, and the test should be repeated 32 times within each run, 4 MB of input data were provided. The data were encrypted and then the ciphertext was divided into 32 files of 125 KB each. In order to produce the ciphertexts required to perform the tests, the LockBox [16] library implementing the ciphers tested was used. NIST software was run on the ciphertexts produced in this manner.

The results for the DES, 3DES, Blowfish, Rijndael and 2DCARotate ciphers are presented below. The results present the number of failed tests for individual ciphers. All tests were divided into four groups: single-result tests[1], Aperiodic-Template tests, Random-Excursion tests and Random-Excursion-V tests.

On the basis of test results, it may be claimed that the quality of the 2DCARotate cipher discussed in this paper does not deviate from the quality of such ciphers as DES, 3DES or AES-Rijndael.

In order to facilitate comparisons between ciphers, certain assumptions were made stemming from the significant amount of data required for the comparison. For a run including all possible tests, the Assess[2] program generates 13 results for single-result tests[1], 148 results for the Aperiodic-Template test, 8 results for the Random-Excursion test and 18 results for the Random-Excursion-V test. Altogether, there are 189 results. The Assess program was run 32 times, yielding 6,048 results in total.

In order to present such an amount of data on a single graph, all failed tests were totalled for each of the four test groups. The grand total is shown in the graphs below. In Figure 2, the total number of tests failed are given for individual test groups.

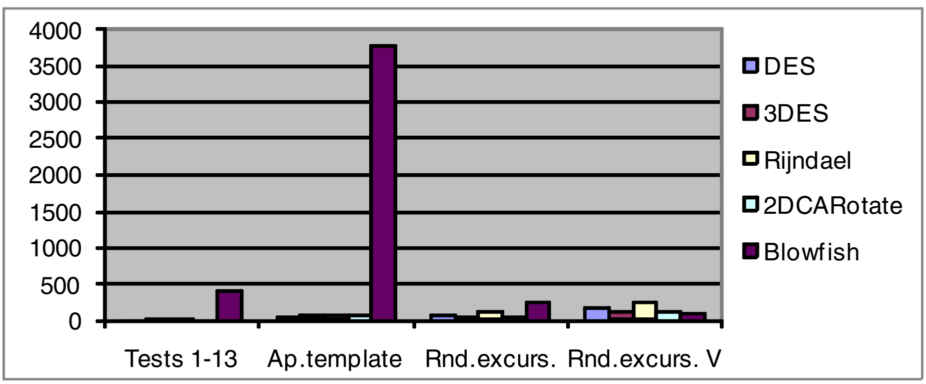


Fig. 2. Comparison of cipher statistical quality

On the Y axis, the number of tests failed by individual ciphers are shown, while on the X axis the labels identifying the individual tests (in four groups) are shown. A more detailed description can be found in [11]. The very poor results of Blowfish are an interesting issue. Its performance against the first group of tests, i.e. single-result tests[1], was very bad. In virtually every one of the 32 cases, it failed as many as 11 out

[1] Tests that return a single numerical value.
[2] The name of the software provided by NIST, which was used to perform the tests.

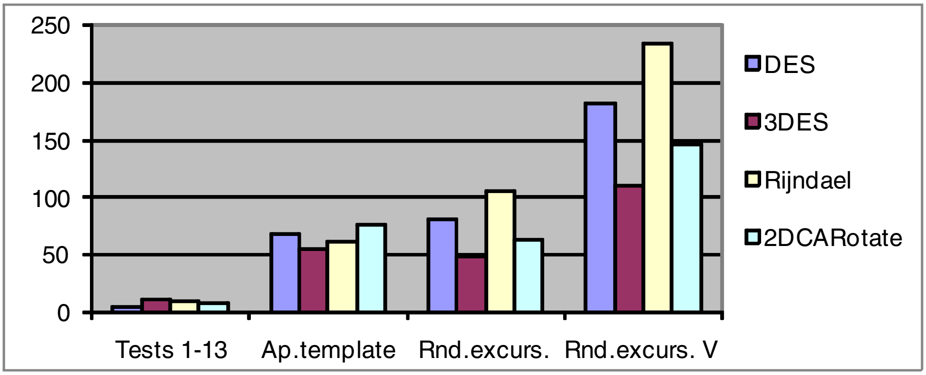


Fig. 3. Comparison of cipher statistical quality (without Blowfish)

of 13 tests. Its results were also very poor for the Aperiodic-Template and Random-Excursion tests. We provide Figure 3 without Blowfish to show a clear comparison between the other 4 ciphers.

4 Cipher Quality Test Using the Grade Analysis Method

In this section we announce briefly the results of pseudorandom sequence and ciphertext tests conducted using a new method recently announced in [14]. It is called the grade analysis method [3].

4.1 Input Data and Processing Parameters

The Szyfry software analyses data from two sources. The first source is the embedded high-quality pseudorandom sequence generator, which is treated as a perfect one. The second source of data are external files containing the ciphertext examined. The data in the files analysed should be in binary format. The files examined in Section 3 above were also used for these tests. These are ciphertext files generated using the DES, 3DES, Blowfish, AES-Rijndael and 2DCARotate algorithms. The input for the ciphertext was a text file compiled from a dictionary of Polish words. Additionally, for comparison, the same calculations were also performed on this file, which will be further called PlainText for the sake of simplicity. The test (described in detail in [14]) consists in the construction of an empirical distribution of grade differentiations within an mxk input table (containing e.g. ciphertext or another examined text) as compared to n random tables of an identical size mxk and presenting it against the background of a set of reference distributions constructed in the same manner for s random input tables. Below, the following parameter values were used: m=k=100, n=500 and s=10, since a decision was made to illustrate the quality of the ciphertexts in question in a simplified manner - as a graphic comparison of the frequency curves

[3] Program Szyfry was developed by P. Bielawski under the supervision of Professor W. Szczesny. It will soon be integrated into the GradeStat suite (http://gradestat.ipipan.waw.pl/). See [13] for a detailed description of this software suite.

(empirical distributions) obtained for these ciphertexts by presenting them against the background of just 10 such curves constructed on the basis of random input tables serving as reference data.

Input data are placed in an m1xk1 table. For testing purposes, the data were put in a 100x100 table. The remaining input parameters of the Szyfry algorithm were set according to the choices suggested by the software for the preliminary analysis scenario, i.e. the size of the m2xk2 table presenting discretised differentiation copulas for the input table against individual random tables (temporary table after processing) was the same as that of the input table, i.e. 100x100, the Spearman ρ coefficient was adopted as the numerical indicator of grade differentiation, the number of iterations (i.e., of comparisons between the input data table and the random table) was set to n=500, the number of GCA (Grade Correspondence Analysis) iterations was set to 50, and the number of randomly generated input tables used to construct reference distributions was set to s=10. The GCA searches for an ordering maximising the value of the coefficient selected (Spearman ρ or Kendall τ). Description of all parameters is beyond the scope of this paper. See [12], [13], [14] and [15].

Seven output data files were generated. The first file contained the data resulting from the analysis of pseudorandom number generator data, further referred to as DaneLos. The next five files contained the results of DES, 3DES, Blowfish, AES-Rijndael and 2DCARotate ciphertext analyses. The final file contained PlainText results. The output data files contained numbers which were subsequently processed using the Excel application. These numbers were in the range from 0 to 1. The results for pseudorandom number generator data were arranged into 10 numerical sequences with 500 values each. The results for ciphertext and PlainText data were also arranged into numerical sequences with 500 values each.

4.2 The Grade Analysis Result

As described in the subsection above, 16 numerical sequences with 500 values each were obtained. For each sequence, the minimum and maximum were determined, i.e. 16 maximum and 16 minimum values were obtained. Among these values, the minimum, amounting to 0.076500653, and the maximum, amounting to 0.211539981, were determined. The range between the minimum and the maximum was divided into 18 subranges starting at 0.07, each with a width of 0.01, i.e. 0.07–0.08; 0.08–0.09…0.23–0.24. For each subrange, the frequency of results for each of the samples analysed was counted.

Figure 4 below presents sample frequency curves for all the ciphers. It is obvious that PlainText and Blowfish data deviate significantly from the data included in 10 random samples and those representing the DES, 3DES, Rijndael and 2DCARotate ciphers. This makes it possible to claim that the degree of randomness in these two samples is low.

In Figure 5, the value ranges on the X and Y axes as well as the subrange widths were changed in order to enable a more detailed analysis of sample and ciphertext randomness. This figure demonstrates that the randomness properties of the DES, 3DES and 2DCARotate ciphers are very good. On the other hand, the Rijndael cipher is a borderline case, similarly as the Los7 sample.

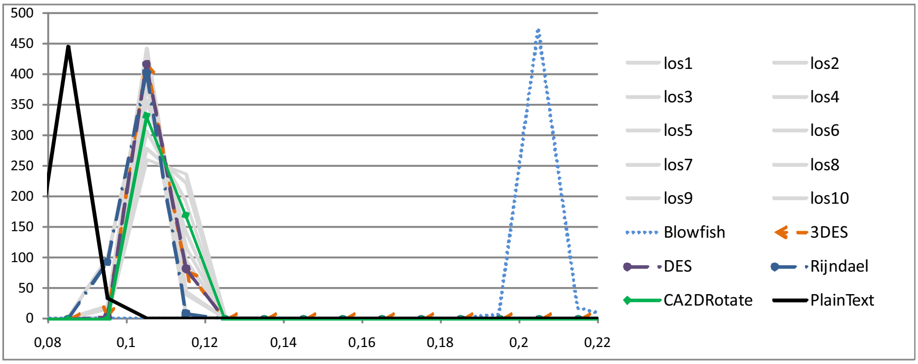


Fig. 4. Graphical analysis of sample frequency curves

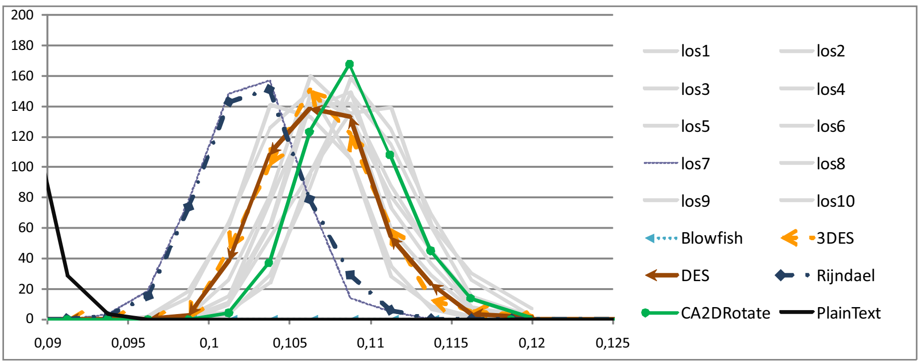


Fig. 5. Analysis of sample frequency curves - selection

5 Conclusion

The application of 2D cellular automata with rotations to encrypt information is a new solution, to the best of our knowledge. The use of irreversible rules to generate the links additionally strengthens the algorithm. The relatively simple geometric transformations, applied to squares of bits, implemented in the algorithm, yield a remarkable nonlinear (nonaffine) effect similar to substitutions, and thus resembling the famous substitution boxes (S-boxes) on which the DES and 3DES algorithms are based. (See e.g. [5] and [6].) The tests conducted and the results presented here demonstrate high statistical quality of our 2DCARotate cipher, which are comparable to the renowned DES, 3DES or AES-Rijndael ciphers. Obviously, in order to present a more thorough analysis, a larger number of random tables (at least 100, and preferably from 500 to 1000) should be used. The presentation could also be improved by using an empirical cumulative distribution function (cumulative frequency curve).

Owing to the simplicity of the 2DCARotate algorithm, which stems from the use of simple mathematical operations, it seems to be a reasonable solution for applications where there is no need to use complicated CPUs as the system agents. This concerns e.g. simple integrated circuits whose design is less complex than that of CPUs. For such circuits, it is easier to implement parallel processing, and cellular automata are very well suited to parallel computations. Cryptosystem 2DCARotate seems highly parallelizable. It can be implemented on multicore devices, which would significantly strengthen its efficiency. It seems especially important these days, with “thousands(!) of cores coming soon”.

There are many collaborating agents in our algoritm/protocol reacting upon their neighborhood environment according to what was called "rules" in the classical literature on the cellular automata, and what can now very well be viewed as rules of intelligent decisions.

Working on a parallel implementation of 2DCARotate system we have got some encouraging results (to be published elsewhere). We used VHDL (Very High Speed Integrated Circuits Hardware Description Language) for prototyping, simulation and debugging of our complex design, treated as a synthesizable multi-agent system. VHDL provides a formal symbolic description of the components or of a hardware circuits and their interconnections. The compilation of the code and the hardware synthesis of different specifications in VHDL were carried out. These formalisms enable to generate software code and/or hardware specifications in VHDL. In our first simulative parallel implementation experiments the encryption algorithm of 2DCARotate runs at about 13 Mb/sec on a programmable device which gives for this implementation 111 MHz internal clock. This is a very slow device as compared with the modern ones available these days, on which we would expect the efficiency of over 80 Mb/sec. Also, our experiments point quite clearly what design components are slow. Their improvement can give some extra speed-up.

It should also be remembered that such ciphers as the DES, 3DES or AES-Rijndael process bytes or double bytes of data during the encryption process. Due to their design principles, ciphers based on cellular automata always work on single bits. However, they only perform very simple mathematical operations which can easily be grouped into bytes or words, in order to get the system easier to implement on modern integrated circuits. We think that such ciphers enhanced with some carefully selected nonaffine components may prove useful in machines employing relatively uncomplicated multicore integrated circuits, which do not include CPUs.

Last but not least, since passing statistical tests cannot be interpreted as a proof of our cipher’s cryptoanalytic robustness, the next area of examination should obviously be the resistance of the proposed cryptosystem against various types of attack. This primarily concerns the methods of linear and differential cryptanalysis.

References

[1] Daemen, J., Rijmen, V.: The Design of Rijndael: AES. Springer, Heidelberg (2002)

[2] Gutowitz, H.: Cryptography with Dynamical Systems, Laboratoire d’Electronique Paris, France (July 25, 1996)

[3] Hardjono, T., Pieprzyk, J., Seberry, J.: Fundamentals in Computer Security. Springer, Heidelberg (2002)

[4] Menezes, A., van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography, 5th printing. CRC Press, Boca Raton (2001), `http://www.cacr.math.uwaterloo.ca/hac/`

[5] Data Encryption Standard, FIPS Publication 46, Washington, DC (1977)
[6] Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, FIPS Special Publication 800-67 (2004)
[7] von Neumann, J.: Theory of self-reproducing automata. Urbana University of Illinois Press (1966)
[8] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S.: A Statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST Special Publication 800-22, with revisions dated May 15 (2001)
[9] Schneier, B.: Description of a New Variable-Length Key 64-Bit Block Cipher (Blowfish). In: Anderson, R. (ed.) FSE 1993. LNCS, vol. 809, pp. 191–204. Springer, Heidelberg (1994)
[10] Srebrny, M., Such, P.: Encryption Using Two-Dimensional Cellular Automata with Application. In: Proceedings of Advanced Computer Systems 2002, pp. 177–190. Kluwer Academic Publishers, Dordrecht (2003)
[11] Srebrny, M., Such, P., Szczesny, W.: Statistical tests of a new cipher related to four renown ones, internal report of the Institute of Computer Science of the Polish Academy of Sciences (December 2007)
[12] Szczesny, W.: Grade correspondence analysis applied to contingency tables and questionnaire data. Intelligent Data Analysis 6(1), 17–51 (2002)
[13] Szczesny, W., Kowalczyk, T., Srebrny, M., Such, P.: Testowanie jakości szyfrów jako generatorów ciągów losowych: wyniki eksperymentów przy wykorzystaniu metod gradacyjnych, Report of the Inst. of Comp. Science, Polish Academy of Sciences (2006)
[14] Szczesny, W., Kowalczyk, T.: Grade differentiation between multivariate data sets, 2008 (to be published); Report of the Inst. of Comp. Science, Polish Acad. of Sciences (2007)
[15] Ciok, A., Kowalczyk, T., Pleszczynska, E., Szczesny, W.: Algorithms of grade correspondence-cluster analysis. The Collected Papers of Theoretical and Applied Computer Science 7(1-4), 5–22 (1995)
[16] TurboPower LockBox software library, `http://sourceforge.net/projects/tplockbox/`
[17] Wolfram, S.: Cryptography with Cellular Automata. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 429–432. Springer, Heidelberg (1986), `http://www.stephenwolfram.com/publications/articles/ca/85-cryptography/1/text.html`

Health Monitor Agent Based on Neural Networks for Ubiquitous Healthcare Environment*

Michael Angelo G. Salvo[1], Romeo Mark A. Mateo[1], Jaewan Lee[1], and Malrey Lee[2]

[1] School of Electronic and Information Engineering, Kunsan National University
68 Miryong-dong, Kunsan, Chonbuk 573-701, South Korea
{masalvo,rmmateo,jwlee}@kunsan.ac.kr
[2] School of Electronics and Information Engineering, Chonbuk National University
664-14 Duckjin-dong, Jeonju, Chonbuk 561-756, South Korea
mrlee@chonbuk.ac.kr

Abstract. Environments such as healthcare systems rely on wireless sensor networks as one of the trends being established in today's industry. Being one of the most researched topics today, ubiquitous healthcare is mostly concerned with providing fast and efficient service. This paper proposes the Health Monitor Agent (HMA) to monitor patients' conditions, provide early detection of serious cases, and take appropriate action in case of emergency. We present this agent to the Ubiquitous and Intelligent Framework. The proposed agent recognizes medical conditions based on symptom patterns from the biosensors in the ubiquitous healthcare environment. The HMA classifies the symptom patterns into the correct medical condition using the multilayer perceptron (MLP). Our proposed algorithm recorded the highest accuracy rate compared to the ZeroR, Simple Logistic, and J48 algorithms.

1 Introduction

People need efficient service in almost everything especially when it comes to a person's health. With the lack of hospital beds in a world being populated more and more each year, some people who have certain conditions opt to just stay home where they feel secured and relaxed. In some cases, it is advisable for some patients to do some activities to be able to relax and forget that they are sick. Others, although bothered with chronic disease, can still go on doing the things they love such as sports. But these patients have to be carefully and not pushing their bodies to the limit. This is where the bio-sensors come in. They range from sensors able to be worn easily, up to sensors embedded into household furniture as discussed in [1]. While other studies of ubiquitous healthcare are limited to home implementation [2], the need for mobility is quickly rising with the advancement of technology today. Most importantly, recent developments suggest that these sensors can be implemented in almost anything and anywhere.

* This research was financially supported by the Human Resource Training Project for Regional Innovation of Korea Industrial Technology Foundation (KOTEF) and Basic Research Program of the Korea Science and Engineering Foundation.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 380–388, 2009.

Providing information wirelessly and automating services based on the context information from user profiles in the environment of the ubiquitous computing concept has emerged as of late. The use of software programs guided by rules on performing their tasks is just one of the many considerations in the environment being discussed.

In [3], Agent technology is mostly used to automate the tasks in the ubiquitous environment. These researches bring up the convenience of living by making the possible interactions to objects in real environment. Though, there are some challenges that most researchers encounter in designing a transparent ubiquitous system such as limited resources that allow only a partial function of system, providing large amount of information, and adding additional features. One of these is the limitation of middleware for mobile and ubiquitous environment that purposely provides the transparency of services to mobile devices as proposed in [4, 5]. This technology provides an abstraction layer between applications and the underlying network infrastructure and it also keeps the balance between the application's QoS requirements and the network lifetime.

In order to achieve higher quality of service for the ubiquitous healthcare environment, we need early detection of illnesses and higher response time in case of emergencies. Some techniques and algorithms are integrated to the agent-based healthcare system for medical diagnosis. Neural network is a common technique for medical diagnosis [6, 7]. Successful application examples show that neural diagnostic systems are better than human diagnostic capabilities. Furthermore, neural network are used to analyze medical images [8, 9]. These research articles survey various approaches and techniques to improve diagnosis in medical images, including mammography, ultrasound and magnetic resonance imaging.

In this paper, we propose the Health Monitor Agent (HMA) to recognize medical conditions. The proposed agent is used for the ubiquitous healthcare system based on the previous framework [10]. The framework considers the distributed software and hardware which provides services and the limited resources to the services by means of mobility middleware. The proposed agent is implemented in the framework to monitor patients and provide necessary services in case of emergency. For the early detection of abnormal health condition, we use the multilayer perceptron (MLP) architecture with two hidden layers. The input data (symptoms) are trained using the backpropagation algorithm. The MLP is compared to the ZeroR classifier, Simple Logistic, and J48 tree. The result was favorable for the MLP architecture.

2 Related Works

2.1 Agent-Based Healthcare System

Healthcare automation using software agent plays a crucial role on disseminating correct information to healthcare proponents and providing immediate medical services. Home healthcare services are previously researched to provide information to physicians the necessary diagnosis to patients and continuous monitoring of patient to acquire immediate response and save lives in critical conditions. Agent-based intelligent decision support is proposed for the home healthcare environment [11] where the multi-agent platform is combined with artificial neural network for the intelligent decision support system in a group of medical specialists collaborating in the pervasive management of care for a patient. Mobile agents are used to serve the collaboration of services for mobile users [12]. An agent is an autonomous, social, reactive and proactive entity, sometimes also mobile. Since telemedicine is grounded on communication and sharing of resources, agents are suitable for its analysis and implementation, and these are adopted for developing a prototype telemedical agent.

2.2 Neural Networks in Healthcare

Artificial neural network (ANN) is a computational system consisting of a set of highly interconnected processing elements, called neurons that process information as a response to external stimuli. A neuron contains a threshold value that regulates its action potential. Neural networks have been applied in the medical domain for clinical diagnosis [6], image analysis and interpretation [13], and drug development [14].

In [15] neural networks were used for automatic detection of acoustic neuromas in MR images of the head. Neural networks were utilized and supported by more conventional image processing operations. The prototype system developed as a result of the study achieved 100% sensitivity and 99.0% selectivity on a dataset of 50 patient cases. A comprehensive list and overview of various research works on the uses of neural networks in the medical domain is presented in [16].

2.3 Ubiquitous and Intelligent Framework for Automated Monitoring of Mobile Patients

This paper uses the previous ubiquitous framework design [10] to propose an agent for the early recognition of medical conditions. In Figure 1, the healthcare environment is shown where there are three location areas interconnected by the overlapping of local wireless range. The ubiquitous healthcare center manages and updates a patient's location and stores health records from each hospital. The remote locations of the patients are updated every time they move to another location by the base station within range. If there are emergency cases detected from the nearest hospital, it sends

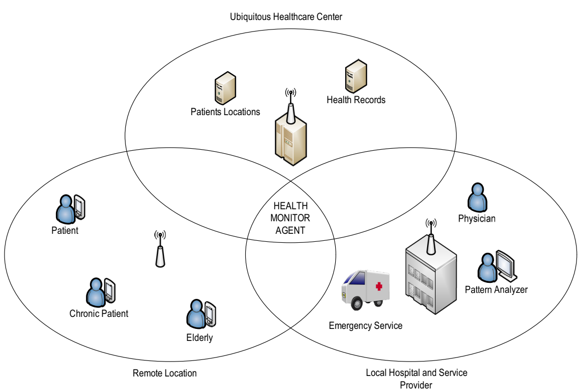


Fig. 1. Ubiquitous healthcare for mobile patients

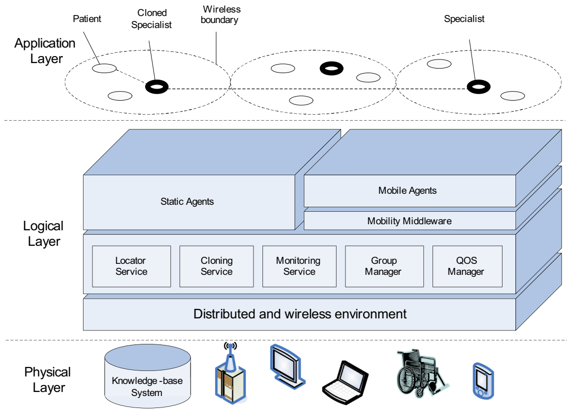


Fig. 2. Ubiquitous and Intelligent Framework

out an ambulance to the location of the patient. That hospital may also gather previous health records of the patient from the healthcare center.

The ubiquitous environment specifications are different from the classical distributed environment. The framework also consists of a middleware that supports mobility in which it focuses on the limited resources of the system like mobile devices. The design of the framework is divided into three layers as shown in Figure 2. First, application layer consists of agents that apply the role of the healthcare systems. The applications layer considers the requirement of the logical layer. The logical layer serves between the application layer and physical layer, the transparent services. The physical layer consists of different hardware like mobile devices, ubiquitous wheel chairs, computers and others. The services from the logical layer are the following:

Locator service. This locates the appropriate physician that monitors the patient using mobile devices. The classification of the appropriate patient to physician agents is the main purpose of the locator service.

Cloning service. This performs the cloning of an agent and deployment of the mobile agent through the base stations. The cloning service communicates to locator for request of service.

Monitoring service. This monitors the activities of every service and agent. All events are monitored and recorded by the monitoring service and updates the knowledge base accordingly.

Group manager. This manages the grouping of agents in the base stations. The group manager decides the migration based on the balanced clustering proposed in [10].

QoS manager. This implements the efficient coordination of the services to perform quality of service.

Health monitor agent (HMA). This collects symptom patterns from the base station near the patient and identifies possible illnesses.

3 Recognizing Medical Conditions Using the Multilayer Perceptron (MLP) Architecture

Each sensor attached to the patient monitors a different condition e.g. heart rate, blood pressure, etc. A threshold value is assigned for each of these conditions. For every reading exceeds the threshold value, meaning it is either below or above the normal expected reading, 1 is assigned in the input pattern. For normal readings, 0 is given. Once all readings are taken from each sensor, the input pattern is trained using the back propagation algorithm.

Back propagation is a supervised learning algorithm where the desired output is known. Input patterns are trained and the error is propagated backwards from the output to the input layer.

This paper proposes the Health Monitor Agent which uses neural network to classify input patterns collected from the sensors. The architecture of the proposed agent is shown in Figure 3. Sensor readings are used for inputs to the input layer. The internal layer consists of 2 hidden layers of the MLP structure while the output layer consists of the classes of conditions. Lastly, we include the decision layer that is connected to the class. This determines the necessary action that is needed to be done based on the current condition of the patient.

Each sensor acts as a neuron. The action of a potential neuron is determined by the weight associated with the neuron's inputs in Equation 1 which patterns are acquired from the sensors. A threshold modulates the response of a single neuron to a particular stimulus confining such response to a predetermined range of values.

$$z = \sum_{i=1}^{n} x_i w_i \tag{1}$$

Equation 2 defines the output y of a neuron as an activation function f of the weighted sum of n+1 inputs. The output is configured by the medical expert and is not discussed in detail by this paper. This equation is done for both hidden layers in the architecture.

$$y = f\left(\sum_{i=0}^{n} x_i w_i \right) \tag{2}$$

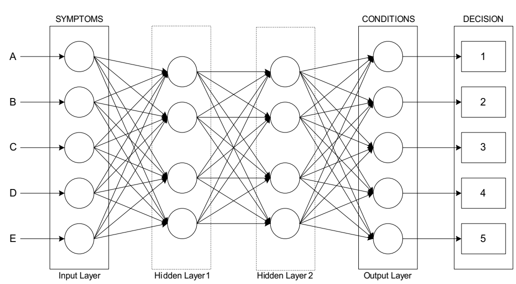


Fig. 3. Network structure of the Health Monitor Agent

The threshold is incorporated into the equation as the extra input in Equation 3.

$$f(x) = \begin{cases} 1 if \sum_{i=0}^{n} x_i w_i > 0 \\ 0 if \sum_{i=0}^{n} x_i w_i \leq 0 \end{cases} \tag{3}$$

The output produced by a neuron is determined by the activation function. This function should ideally be continuous, monotonic, and differentiable. With these features in mind, the most commonly chosen function is the sigmoid which is shown in Equation 4.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4}$$

The accuracy of the response is measured in terms of an error E defined as the difference between the current and desired output in Equation 5.

$$E = \tfrac{1}{2} \sum_j \left(t_{pj} - o_{pj}\right)^2 \tag{5}$$

The error E is propagated backwards from the output to the input layer. Appropriate adjustments are made by slightly changing the weights in the network by a proportion of the overall error E.

After weights are adjusted, the examples are presented again. The error is again calculated and weights are adjusted and the process is repeated until the current output is satisfactory or the network cannot improve further.

We present the input-output pair p and produce the current output o_p. We then calculate the output of the network and calculate the error for each output unit for a particular pair using Equation 6.

$$\delta_{pj} = \left(t_{pj} - o_{pj}\right) f'\left(net_{pj}\right) \tag{6}$$

In Equation 7, we calculate the error by the recursive computation of δ for each of the hidden units j in the current layer. Where w_{kj} are the weights in the k output connections of the hidden unit j, δ_{pk} are the error signals from the k units in the next layer and $f'(net_{pj})$ is the derivative of the activation function. Then we propagate backwards the error signal through all the hidden layers until the input layer is reached.

$$\delta_{pj} = \sum_k \delta_{pk} w_{kj} f'\left(net_{pj}\right) \tag{7}$$

Finally, we repeat the steps for Equations 6 and 7 until the error is acceptably low. After the structure of the neural network is trained, the structure is used for recognizing medical conditions. In detection of such conditions, actions are taken such as dispatching an ambulance or providing immediate precautions.

4 Experimental Evaluation

4.1 Environment

The simulation environment of the proposed system consists of multi-agents created in the JADE agent platform. A single PC was used as environment for the agents because this approximates the expected actual deployment configuration of the ubiquitous healthcare system. We programmed the ANN in Java and put the functionality in a JADE agent.

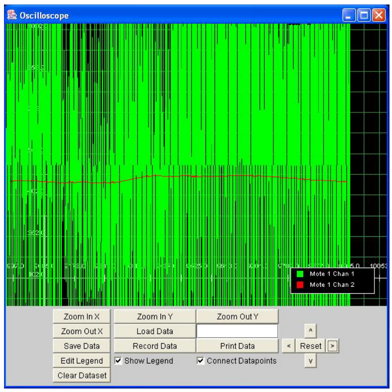


Fig. 4. Screenshot of the oscilloscope readings from the biosensors

Figure 4 shows the readings from the biosensors. These readings are from two sensors in one mote and are used as input data in detecting medical conditions.

Our simulation focused on testing the neural network's ability to classify the symptom patterns into the correct medical condition using two hidden layers in the MLP structure. We then train these symptom patterns using the back propagation algorithm.

To evaluate an improvement in the efficiency of the algorithm, we used a similar simulation environment presented in [13]. 5 health conditions were assumed and each is paired with a set of target outputs. 2 additional conditions were defined which are normal status, where all outputs are zero, and the other when all outputs are 1 indicating that the patient is in a critical situation because all symptoms are present. Each health condition is matched with a set of 3 symptom patterns. We created a test data with 25 pairs of patterns which are different by at least one symptom from the patterns used in the training.

4.2 Results

Based on the previous study [13], three runs of the simulation were made. Thus, also the current simulation was run with the same number of epochs, but with the addition of 10000 and 20000 epochs after the first and third runs respectively. Table 1 shows the performance of the network.

Table 1. Generalization accuracy of the algorithm trained with two hidden layers compared to only one

Generalization Accuracy		
Training Epochs	One Hidden Layer	Two Hidden Layers
5000	88%	90%
10000	89%	92%
25000	92%	94%
30000	93%	95%
50000	96%	99%

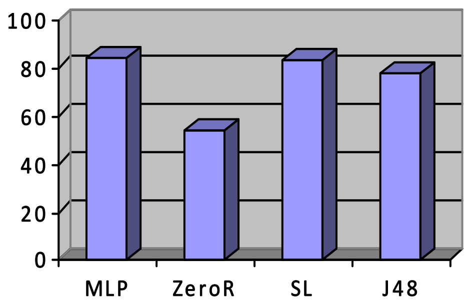


Fig. 5. Bar graphs showing the accuracy of each algorithm

All training patterns were correctly identified by both simulations. Though, it can be inferred that the epochs needed to produce higher rate of generalization accuracy is lower when two hidden layers are used. This is because the first hidden layer serves to partition the input data and the units in the second layer can easily make the final classification.

After determining that 2 hidden layers provide higher generalization accuracy, the MLP was compared to other algorithms. We used a data from the UCI data repository, a popular repository of different data for data mining experiments, which is the heartbeat-c data. Figure 5 shows the accuracy of each algorithm in classifying 303 patterns using 10 folds of cross-validation. The MLP provided the highest accuracy rate with 84.49% followed closely by the Simple Logistic with 83.19%. J48 gave 77.56% while ZeroR was only 54.46% accurate.

5 Conclusion

This paper presented the Health Monitor Agent that classifies symptom patterns into the correct medical condition. Once a medical condition is detected, the HMA suggests appropriate action ranging from providing ambulance to taking immediate precautions depending on the severity of the situation. The agent is applied to the ubiquitous healthcare environment to provide early detection of illness and provide necessary services in case of emergency. The early detection of abnormal health conditions was aided by the multilayer perceptron (MLP), an artificial neural network (ANN). The symptoms patterns were trained using the back propagation algorithm to classify them into the correct illness. Simulation results show that two hidden layers provide higher generalization accuracy than only one hidden layer. Furthermore, the MLP yielded the highest accuracy rate compared to ZeroR, Simple Logistic, and J48.

References

1. Jang, S., Lee, J., Lee, J., Park, S., Hwang, S., Yoon, H., Yoon, Y.: Ubiquitous Home Healthcare Management System with Early Warning Reporting. In: Proceedings of the International Conference on Convergence Information Technology, pp. 2394–2401 (2007)
2. Braecklein, M., Tchoudovski, I., Moor, C., Egorouchkina, K., Pang, L., Bloz, A.: Wireless Telecardiological Monitoring for the Homecare Area. In: Proceedings of the 2005 IEEE, Engineering and Biology 27th Annual Conference, pp. 3793–3795 (2005)
3. Rodríguez, M., Favela, J.: Autonomous Agents to Support Interoperability and Physical Integration in Pervasive Environments. In: Proceedings of AWIC, pp. 307–317 (2003)
4. Bellavista, P., Corradi, A., Stefanelli, C.: Mobile Agent Middleware for Mobile Computing. Computer 34(3), 73–81 (2001)
5. Han, S., Yoon, Y., Youn, H., Cho, W.: A New Middleware Architecture for Ubiquitous Computing Environment. In: Proceedings of STFEUS, pp. 117–121 (2004)
6. Brause, R.: Medical Analysis and Diagnosis by Neural Networks. In: Crespo, J.L., Maojo, V., Martin, F. (eds.) ISMDA 2001. LNCS, vol. 2199, pp. 1–13. Springer, Heidelberg (2001)
7. Joo, S., Moon, W., Kim, H.: Computer-aided Diagnosis of Solid Breast Nodules on Ultrasound with Digital Image Processing and Artificial Neural Networks. Engineering in Medicine and Biology Society 1, 1397–1400 (2004)
8. Giger, M.: Computer-aided Diagnosis of Breast Lesions in Medical Images. IEEE Computational Science and Engineering 5(5), 39–45 (2000)
9. Verma, B., Zakos, J.: A Computer-aided Diagnosis System for Digital Mammograms based on Fuzzy-neural and Feature Extraction Techniques. IEEE Transactions on Information Technology in Biomedicine 5(1), 46–54 (2001)
10. Mateo, R., Salvo, M., Lee, J.: Balanced Clustering using Mobile Agents for the Ubiquitous Healthcare Systems. In: Proceedings of the International Conference on Convergence and Hybrid Information Technology, pp. 686–691 (2008)
11. Cervantes, L., Gerardo, B., Mateo, R., Lee, J.: Recognizing Medical Conditions from Symptom Patters Using Neural Networks. In: Proceedings of the 14th KSII Fall Conference, pp. 251–255 (2006)
12. Della Mea, V.: Agents Acting and Moving in Healthcare Scenario: A Paradigm of Telemedical Collaboration. IEEE Transaction on Information technology in Biomedicine 5(1), 10–15 (2001)
13. Antkowiak, M.: Artificial Neural Networks vs. Support Vector Machines for Skin Disease Recognition. Master's Thesis, Department of Computing Science, Umea University, Sweden (2006)
14. Weinstein, J., Kohn, K., Grever, M.: Neural Computing in Cancer Drug Development. Predicting Mechanism of Action, Science, 447–451 (1992)
15. Dickson, S.: Investigation of the use of Neural Networks for Computerized Medical Image Analysis. PhD Thesis, Department of Computer Science, University of Bristol (1998)
16. Sordo, M.: Introduction to Neural Networks in Healthcare. Open Clinical (2002), http://www.openclinical.org/docs/int/neuralnetworks011.pdf

Understanding Users' Subject Interests in the Web Site Based on Their Usage of Its Content: A Novel Two-Phase Clustering Framework

Ahmad Ammari and Valentina Zharkova

School of Computing, Informatics, and Media, University of Bradford
a.n.ammari@Bradford.ac.uk, v.v.zharkova@Bradford.ac.uk

Abstract. In order to understand the behavior of website users, a deep analysis of content and usage data can reveal valuable knowledge about the main subjects these visitors are truly interested in. Preprocessing and clustering the highly unstructured content of web pages should be addressed very carefully in order to provide effective results. In this paper, a novel proposed two-phase self organizing feature map clustering framework to segment web users based on their subject interests in the diverse content of a University website is described. Also, the overall noise and dimensionality reduction of the sample web site content is properly addressed through the formulation of a comprehensive ten-step preprocessing procedure, which provided very promising experimental results when applied to the input web pages in the first phase of the proposed framework.

Keywords: Clustering, Data Preprocessing, Self Organizing Feature Maps, Web Content Mining, Web Usage Mining.

1 Introduction

1.1 Understanding Web Users' Behavior

In the current era of information revolution, it is strongly believed that satisfying the informational needs of customers can govern the revenues of any organization. Meeting customers' expectations in the easiness, clarity, and simplicity of the navigation process in an organization's website is crucial to the effectiveness of the customer acquisition policies. The importance of well understanding of the web users behavior during their website surfing is now changing the principles, arts, and techniques of website design and information presentation.

However, most websites today are still struggling to present effectively the ideal structure of their web pages, which can provide maximum support to visitors in finding the most relevant information with minimal time and effort [14]. The serious need to further study the complex behavior of website visitors in order to create more attractive web content, predict their preferences, and recommend the pages that truly match their needs, is still challenging web researchers [11].

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 389–399, 2009.

1.2 Addressing the Existing Limitations

Clustering huge data is a very popular mining function that has been widely applied in customer segmentation, gene and protein analysis, product grouping, finding taxonomies, and text mining [5]. This is a descriptive, unsupervised learning-based tool that has been extensively used to partition web datasets into subsets (clusters) to give the insight into which cases of the data under study are more similar to one another than to the other cases within the original dataset. Since the preprocessing of web page contents usually produces data of high dimensionality, such as the term-document matrix (TDM) representation of web document collections [2], Self Organizing Feature Maps (SOFM) – originally developed by Kohonen [8] – could still be one of the most promising neural network – based algorithms in the clustering of highly dimensional data to be shown in a low-dimensional structure. Unfortunately, the studies that utilized the SOFM as their mining model to discover useful patterns in web users behavior from a combination of web content and usage data on the basis of term frequencies presented only a one-level clustering model [14], [15]. This approach can generate a relatively very huge vocabulary size and thus enormously increase the computational costs of running the SOFM algorithm [9] especially when addressing very large websites.

Moreover, many clustering-based web mining studies are used to undermine the importance of presenting a clear procedure to preprocess content data used in clustering web pages. Web content preprocessing is a fundamental process during which various refinement activities, such as data cleaning and dimensionality reduction [1], should be applied on the naturally unstructured data of web pages as this is a crucial condition for any data mining model, including clustering, to well perform [3].

To overcome the above limitations, this paper proposes a two-phase SOFM-based clustering Web Mining Framework to better understand user behavior segmentation and boost up quality of providing the required information on the web. The paper also presents a stepped clustering preparation procedure to web content data.

The rest of this paper is organized as follows: In Section 2, our procedure to preprocess web content data is explained. In Section 3, the proposed two-phase SOFM clustering framework to help understand user web interests is described. In Section 4, our experimental results are discussed. Finally, conclusions are drawn in Section 5.

2 Data Preprocessing

2.1 Web Content Preprocessing

Web page content preprocessing tasks follow the well known linguistic preprocessing [3] requirements used extensively in text mining, with a special *iterative* data cleaning phase due to the hybrid mixture of elements. The extraction and reformatting of useful content from such mixed pool is a tedious and error-prone process yet unfortunately has not been addressed extensively in the literature of web data preparation [4], [10], [17]. To address such a problem, we have proposed and applied the following web content preprocessing procedure:

1. Selecting top $\boldsymbol{n}$ frequently visited web page URLs from the University website under study using web-log analyzer statistics. Any page URL having views $\boldsymbol{v}$ within the range $200 \leq \boldsymbol{v} \leq 10000$ during the studied period were selected for content parsing.
2. Filtering the selected URLs by discarding those referring to removed pages, multimedia objects (images, video files … etc), and unauthorized access. A Perl-based page fetch checking tool was created and used in the filtration.
3. Mapping each URL of a determined web page $\boldsymbol{j}$ to a unique number, $\boldsymbol{j} \in \{\mathbf{1}, \mathbf{2} \ldots \boldsymbol{D}\}$, where $\boldsymbol{D}$ is the total number of determined pages.
4. Parsing the content of pages referred to by the filtered URLs into one page collection. A Perl HTML parsing tool was created and used.
5. Creating and applying text filters to remove unneeded elements such as markup tags, spacing and line breaks between the markup elements and their attributes, embedded client-side scripts, forms, frames, and multimedia objects. We used the Perl scripting language to implement these filters, utilizing its powerful regular expression (*regex*) features.
6. Removing stop words, which are irrelevant terms with respect to the main subject of the pages, such as determiners, conjunctions, and prepositions.
7. Stemming the remaining words to their corresponding stem words. We applied Porter's stemming algorithm [13] due to its simplicity and high performance.
8. Formulating the term – page matrix $(\boldsymbol{S} \ \chi \ \boldsymbol{D})$ with each page $\boldsymbol{j} \in \{\mathbf{1}, \mathbf{2} \ldots \boldsymbol{D}\}$ is represented as a vector of the weights of the relevant stem words $\boldsymbol{i} \in \{\mathbf{1}, \mathbf{2} \ldots \boldsymbol{S}\}$ in the page collection, where each weight ω_{ij} for the $\boldsymbol{i}^{th}$ stem word in the $\boldsymbol{j}^{th}$ page is determined using the *tf-idf* weighting scheme as defined below:

$$\omega_{ij} = \frac{freq_{ij}}{\max\limits_{1 \leq k \leq S} freq_{kj}} \times \log \frac{D}{n_i} \tag{1}$$

   Where $\boldsymbol{freq_{ij}}$ is the number of occurrences of stem word $\boldsymbol{i}$ in web page $\boldsymbol{j}$ and $\boldsymbol{n_i}$ is the total number of occurrences of the stem word $\boldsymbol{i}$ in the web page collection.
9. Strengthening the term – page matrix by removing *outlier stem words*. Outlier stem words could be classified into three subtypes:

   a. Stem words that have too low weight values.
   b. Stem words that have too high weight values.
   c. Stem words that their number of non-zero weights in the page collection is too low.

   Stem words having too low weight values are those that exist in most of the pages in the collection because their $\boldsymbol{n_i}$ values are too high, whereas stem words that either have too high weight values (their $\boldsymbol{n_i}$ values are too low) or their non-zero weights are seldom in the collection are those that exist too rarely in the site. All of the three types are considered *noisy* because they tend to increase the fuzzy distinction between the web pages [12] in the page collection causing these pages to lack discrete cluster-memberships. Count-based heuristics were used for the filtration of stem words.

10. Further strengthening the term – page matrix by removing *outlier web pages*. Outlier pages are those that have relatively little number of non-zero weights of stem words in their weight vectors, thus they tend to increase the sparseness of data and reduce the self-descriptiveness of pages in the collection. Descriptive statistics were used for the filtration of web pages.

2.2 Web Usage Preprocessing

An important preprocessing task that should be done before trying to mine useful patterns from web usage data is to identify user sessions from web server log files in a process that is called session identification, or sessionization [15]. Semantic technologies can be exploited to improve session discovery algorithms [6]. A good survey of popular sessionization techniques is presented in [4].

Our framework not only takes into account the URLs accessed in each user session, but also the duration of time the user spends on each page during his session. To determine the page visit duration, each two subsequent page access times are subtracted, then a missing value imputation technique is used to determine the duration of the last page accessed in the session.

3 A Novel Two-Phase SOFM Clustering Framework

3.1 Using SOFM as a Clustering Algorithm

Self-organizing feature maps (SOFM) [8] learn to classify input vectors – in our case the web page vectors – according to how they are clustered in the input space. The neurons in the layer of an SOFM are arranged initially in physical positions based on a topology function. During training iterations, the distances between training vectors and neurons in the map are calculated with a distance function. In each iteration, SOFM identifies a winning neuron, or a Best Matching Unit (BMU), which has the least distance with the presented input vector [3]. The initial weight W of the BMU as well as the weights of all neurons $\boldsymbol{n_i}, \boldsymbol{n_{i+1}}, \ldots, \boldsymbol{n_k}$ determined by a neighborhood function $\boldsymbol{\Phi}$ are updated to $\hat{\mathrm{W}}$ as defined below:

$$\hat{\mathrm{W}}_{\mathbf{n_i}} = \mathrm{W}_{\mathbf{n_i}} + \boldsymbol{\Phi} \cdot \boldsymbol{\alpha} \cdot (\boldsymbol{\rho} - \mathrm{W}_{\mathbf{n_i}}) \tag{2}$$

Where $\boldsymbol{\alpha}$ is the learning rate and $\boldsymbol{\rho}$ is the weight of the input vector. The neighborhood function $\boldsymbol{\Phi}$ attains a maximum of 1 for the BMU neuron and keeps decreasing as neurons get far from the BMU till it reaches 0 for neurons outside of the neighborhood area of the BMU. The neighborhood area depends on a neighborhood radius $\mathbf{r}$ whose value decreases from iteration to another during the training session. SOFM clustering of text documents has achieved good results in many research projects. The most popular example is WEBSOM [7], [9] which is partly used to organize large collections of internet newsgroups.

In our framework, there are two phases in the SOFM-based clustering process, representing two web mining categories: web content mining, and web usage mining, respectively.

3.2 Phase-1: Web Content Clustering

In order to present user-oriented web content, it is very important to understand web user segmentation based on the semantic subjects each group of pages presents in their content. To obtain such categorization, a phase-1 content clustering is applied on the preprocessed term – page matrix for cluster identification. Since each page is thought of as a vector in the dimensional space, it is wise to map the cosine of the angle between two page vectors, $\boldsymbol{P_j}$ and $\boldsymbol{P_k}$, into a distance function ***Dis*** to be fed into the SOFM algorithm.

$$Dis\left(P_j, P_k\right) = 1 - \frac{P_j \cdot P_k}{|P_j| \cdot |P_k|} = 1 - \frac{\sum_{i=1}^{S}(\omega_{ij} \cdot \omega_{ik})}{\sqrt{\sum_{i=1}^{S}(\omega_{ij})^2 \cdot \sum_{i=1}^{S}(\omega_{ik})^2}} \tag{3}$$

Another popular distance measure usually used in SOFM clustering projects is the Euclidean distance [3] defined below:

$$Dis\left(P_j, P_k\right) = \sqrt{\sum_{i=1}^{S}(\omega_{ij} - \omega_{ik})^2} \tag{4}$$

3.3 Page - Cluster Membership

Groups of pages discovered by the first clustering phase are mapped into cluster identification numbers representing the main topics of the web site. While some clustering algorithms such as K-means [3] forces each case in the training set to be assigned to only one cluster, it is more realistic to use the fuzzy batch version of the SOFM algorithm [16], where in each training iteration, the BMU and their neighbor neurons in the map can smoothly move toward an average position of all of the input training vectors. The choice of using a fuzzy SOFM algorithm not only aligns with the common sense fact that a web page may present content related to more than one main topic, but also adheres to the fact that SOFM performs better when aiming to visualize high dimensional data [1]. *Defuzzification* can then be done by assigning each web page crisply to the cluster in which its membership value is the highest.

3.4 Mapping User Page Sessions to User Cluster Sessions

After assigning each page to the cluster that it mostly belongs to, a *user cluster* session table can now be formulated, as shown in Table 1. This table can then be used to create the final input vectors, shown in Table 2, used to train phase-2 SOFM clustering, in which detected clusters are assigned *interest weights* based on the total time spent on each of them during each session. For example, in the sample shown in Table 2, the user in the first session spent a total of 56 and 240 seconds on visiting web pages that belong to clusters B and C, respectively. The user in the second session spent a total of 96, 300, and 200 seconds on visiting pages that belong to clusters A, B, and D, respectively. These time totals represent weights for the user interests in each cluster, that is, in each of the main subjects the website is consisted of

Table 1. Sample user cluster sessions

Session ID	Page ID	Cluster ID	Visit Time (sec)
1	9	B	56
1	20	C	120
1	35	C	120
2	44	A	96
2	9	B	100
2	8	B	200
2	10	D	200

Table 2. Sample user session vectors

	Total time spent on clusters (sec)			
Session ID	Cluster A	Cluster B	Cluster C	Cluster D
1	0	56	240	0
2	96	300	0	200

We argue that this approach of representing sessions in terms of cluster interest weights rather than page interest weights has three major benefits to the overall behavior understanding process:

- First, it clarifies which subjects, rather than which pages, interest the users, which allows building subject-oriented, not only page-oriented, recommender systems.
- Second, it greatly reduces the dimensionality of the training user sessions used as input vectors in the phase-2 clustering step since the number of page clusters detected in phase-1 clustering is indeed lower than the number of pages in the page collection. This reduction in the vector dimensions is one of the optimal requirements for the SOFM algorithm [16].
- Third, it strengthens the assumption of a previous research [14] that the time spent on viewing specific web content is proportional to the interest the user has in this content, which could be criticized by having the possibility of a user who may spend a lot of time on a specific page searching for interesting content but fail to find any. But now, with our representation of the user interest, even if such a possibility exists, what is taken into account is not how long the user spends on a specific page, but rather how long he spends on a specific cluster, that is, a specific subject.

3.5 Phase-2: Web Usage Clustering

Finally, to group users based on their subject interests, a phase-2 SOFM clustering step is performed. Motivated by the benefit of having a reduced dimensionality of the input vectors in this clustering phase compared to the input vectors of phase-1 clustering, it is proper to use a simpler form of a distance measure than the cosine measure. Hamming distance [3] between two user session vectors $\boldsymbol{US_j}$ and $\boldsymbol{US_k}$, could be a good choice:

$$Dis\left(US_j, US_k\right) = \sum_{i=A}^{D} \left|us_{ji} - us_{ki}\right| \tag{5}$$

Where $\boldsymbol{us_{ji}}$ and $\boldsymbol{us_{ki}}$ are the time weights of cluster $\boldsymbol{i} \in$ {A, B, C, D} in sessions $\boldsymbol{US_j}$ and $\boldsymbol{US_k}$, respectively.

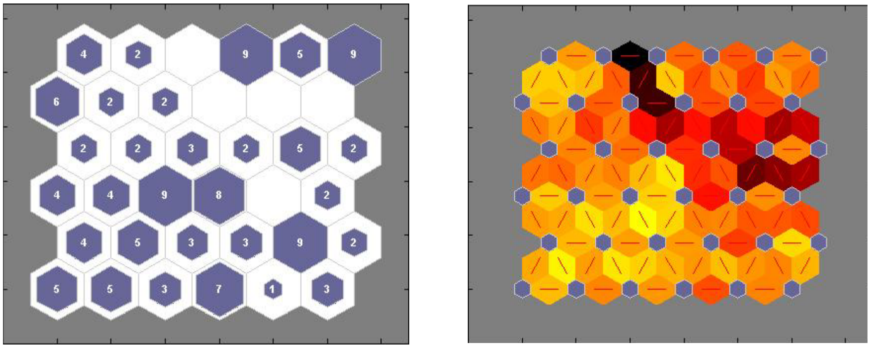
4 Experimental Results

We began testing our framework by performing phase-1 clustering of web pages in a web page collection. We chose to study the web behavior of the Bradford University School of Informatics website users because this site is sufficiently large in both the number of contained web pages and the number of records in its web log files. Table 3 lists a summary of the data properties that have been manipulated by our proposed web content preprocessing procedure. The numbers in the second column of the table refer to the preprocessing step numbers as described in Section 2.1.

Perl was used to implement the web content preprocessing parsers, filters, and weight matrix creators as Perl unparalleled portability, speed, expressiveness, API wealth, and powerful regular expression support had made it the language of choice to solve complex computing problems. Matlab new neural network toolbox (v6) was used to perform the SOFM page clustering simulations as it supports two new data visualization functions: *plotsomhits*, which plots the sample hits of the SOFM, showing the number of times each neuron in the network map had become a BMU during the training session, and *plotsomnd*, which plots the neurons neighbor distances during and after training the layer with the input vectors. Patches between neurons in the distances diagrams are colored from black (large distances) to yellow (small distances) to show how close each neuron's weight vector is to its neighbors.

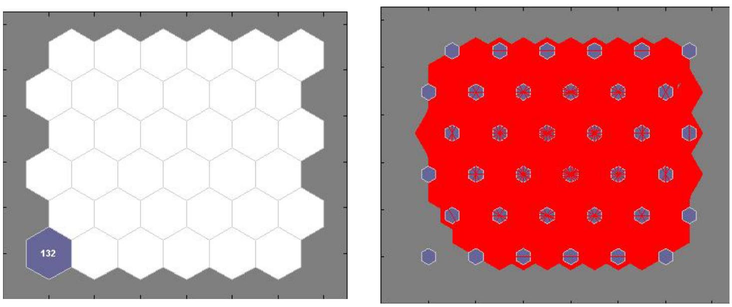
Table 3. Web Content Data Properties

Property	Preprocessing Step No.	Before Preprocessing	After Preprocessing	Tools used
No. of collected URLs	1, 2	10000	219	Fetch Checker & *Regex*-based Filters
URL Format	3	URL addresses	List of {ID, URL} pairs	Structured Program
Web content	4	Separate pages	One page collection	HTML Parser
	5	Content with noisy elements (code, blank lines …etc)	Cleaned content	*Regex*-based Filters
Extracted terms	6, 7, 8	Unstructured strings of terms	Term – Page weight matrix	*Tf-Idf* weighting scheme
Stem words	9	Term – Page matrix with noisy stem words	Noisy stem-free term – page matrix	Heuristics
Page vectors	10	Term – Page matrix with noisy page vectors	Noisy page-free term – page matrix	Heuristics
Term – Page Matrix Dimensions	6 – 10	D = 219 Pages S = 3380 Stem words	D = 132 Pages S = 374 Stem words	All tools used in steps 6 - 10



(a) Sample Hits (b) Neighbor Distances

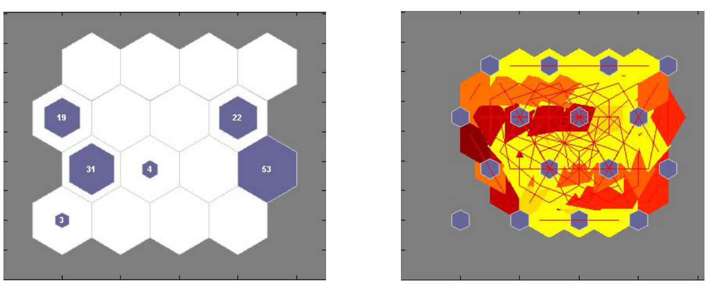
Fig. 1. A 6 X 6 SOFM trained using the *Euclidean* distance measure



(a) Sample Hits (b) Neighbor Distances

Fig. 2. A 6 X 6 SOFM trained using the *Cosine* distance measure

Fig. 1 shows the sample hits (a) and neighbor distances (b), for a 36-neuron SOFM layer trained using the Euclidean distance with a term-page matrix of 132 pages that have 374 stem word weights each.



(a) Sample Hits (b) Neighbor Distances

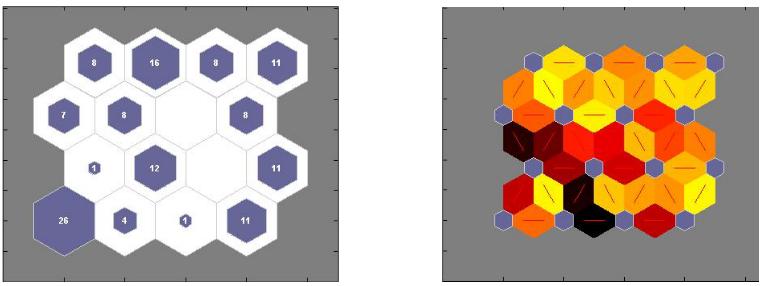
Fig. 3. A 4 X 4 SOFM trained using the *Cosine* distance measure

While this Euclidean-based SOFM was able to recognize 3 to 4 page clusters in such a high dimensional space, a cosine distance-based SOFM having the same number of neurons clearly failed to recognize more than one cluster, as shown in Fig. 2.

Further dimensionality reduction of the input vectors (length = 216 word weights each), realized by further removal of noisy non-zero weight stem words from the term-page matrix, let the cosine distance-based algorithm to recognize around 3 clusters of web pages with fuzzy distances between their *centroids*, as shown in Fig. 3.

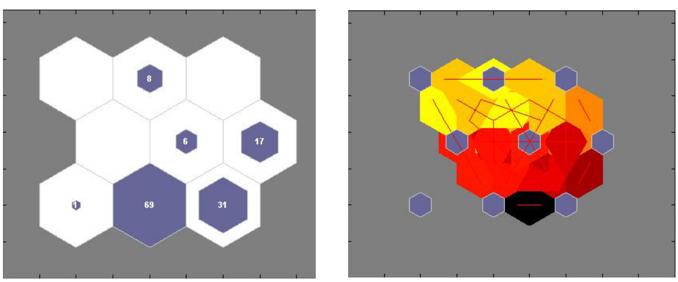
A Euclidean-based clustering simulation for the same sample, however, managed to recognize a little more number of clusters, as depicted by Fig. 4.

The two applied distance measures did not agree on the number of detected page clusters until the number of neurons in the SOFM layer was further reduced to 9 neurons. As can be seen in Fig. 5 and Fig. 6, both Cosine and Euclidean-based SOFM networks were able to depict 5 clusters representing 5 main subjects for the page collection with reduced stem words representation.



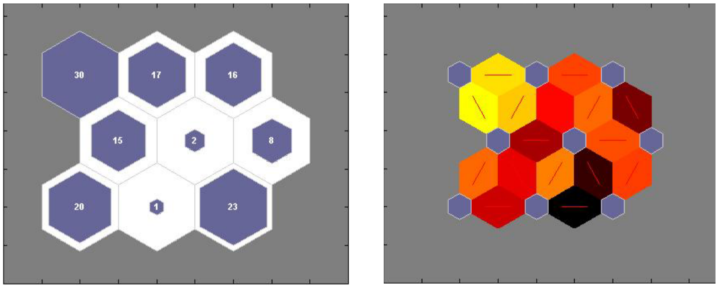
(a) Sample Hits (b) Neighbor Distances

Fig. 4. A 4 X 4 SOFM trained using the *Euclidean* distance measure



(a) Sample Hits (b) Neighbor Distances

Fig. 5. A 3 X 3 SOFM trained using the *Cosine* distance measure



(a) Sample Hits (b) Neighbor Distances

Fig. 6. A 3 X 3 SOFM trained using the *Euclidean* distance measure

5 Conclusion

The efficient preprocessing of the unstructured mixture of elements in web content resources is indeed a tedious and error-prone process that needs to follow a standard and comprehensive approach. A procedural approach to preprocess the web content data was introduced in this paper. Moreover, clustering web pages in a hybrid collection of content, such as a website, is the first step toward understanding user web interest segmentation. SOFM-based clustering can provide a realistic overall picture to the main subjects that these web pages are all about. Using the cosine angle measure to cluster word-weighted pages in a fuzzy environment like an SOFM has revealed a significant unresponsiveness to very high dimensional vectors in the input space. This was a very supportive to let us produce a novel two-phase SOFM clustering approach that not only reduces the overall dimensionality of the final input user sessions to be grouped by similarity thus improve overall clustering results, but also perfectly aligns with the assumption that the time spent on viewing a web topic by certain users is proportional to how much they are really interested in that topic. Finally, it was observed that as the number of neurons decreases in the SOFM layer, both Cosine and Euclidean-based algorithm training tend to agree on the number of detected clusters, that is, the number of the subjects that our web resource content is consisted of. This can be highly taken into account when the cluster validity issue is raised in web mining-related clustering projects.

Future work will be conducted in three directions: First, the second clustering phase of the framework will be performed after further discarding of anomaly pages, assigning each remaining page in the page collection to a detected page cluster based on its highest cluster membership value, and preparing the user cluster session table to be input to the phase-2 SOFM. Second, a cluster validity approach will be proposed in order to measure the quality of the generated clusters in both clustering phases of the framework. Third, repeated tries to cluster web pages with different map topologies than the used hexagonal topology should be performed and compared with existing clustering results.

References

1. Abonyi, J., Feil, B.: Cluster Analysis for Data Mining and System Identification. Birkhauser Verlag, Berlin (2007)
2. Akerkar, R., Lingras, P.: Building an Intelligent Web, Theory and Practice. Jones and Barlett, London (2008)
3. Cios, K.J., Pedrycz, W., Swiniarski, R.W., Kurgan, L.A.: Data Mining, A Knowledge Discovery Approach. Springer Science+Business Media, New York (2007)
4. Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. Knowledge and Information Systems 1(1) (1999)
5. Hornick, M.F., Marcadé, E., Venkayala, S.: Java Data Mining: Strategy, Standard, and Practice. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, San Francisco (2007)
6. Jung, J.: Semantic Preprocessing of Web Request Streams for Web Usage Mining. Journal of Universal Computer Science 11(8), 1383–1396 (2005)
7. Kaski, S., Honkela, T., Lagus, K., Kohonen, T.: WEBSOM – Self-organizing maps of document collections. Neurocomputing, 101–117 (1998)
8. Kohonen, T.: Self-organizing Maps, 3rd edn. Springer, Berlin (2001)
9. Kohonen, T., Kaski, S., Lagus, K., Honkela, T.: Very Large Two-Level SOM for the Browsing of Newsgroups. In: Vorbrüggen, J.C., von Seelen, W., Sendhoff, B. (eds.) ICANN 1996. LNCS, vol. 1112, pp. 269–274. Springer, Heidelberg (1996)
10. Li, Y., Zhang, C., Zhang, S.: Cooperative strategy for Web data mining and cleaning. Applied Artificial Intelligence 17, 443–460 (2003)
11. Mobasher, B., Cooley, R., Srivastava, J.: Creating Adaptive Web Sites Through Usage-Based Clustering of URLs. In: Proceedings of IEEE Knowledge and Data Engineering Exchange (1999)
12. Nurnberger, A., Borgelt, C.: Fast Fuzzy Clustering of Web Page Collections. In: Proceedings of PKDD Workshop on Statistical Approaches for Web Mining, Pisa, Italy (2004)
13. Porter, M.: An algorithm for suffix stripping. Program (14), 130–137 (1980)
14. Velásquez, J.D., Yasuda, H., Aoki, T.: Using Self Organizing Feature Maps to acquire knowledge about visitor behavior. In: Proceedings of the knowledge-based intelligent information and engineering systems, Oxford, UK, pp. 951–958 (2003)
15. Velásquez, J.D., Yasuda, H., Aoki, T., Weber, R.: A new similarity measure to understand visitor behavior in a web site. IEICE Transactions on Information and Systems E200-D(1) (2004)
16. Vesanto, J.: Using SOM in Data Mining. Licentiate's thesis, Helsinki University of Technology (2000)
17. Zhang, S., Zhang, C., Yang, Q.: Data Preparation for Data Mining. Applied Artificial Intelligence 17, 375–381 (2003)

Toward Semantic Mobile Web 2.0 through Multiagent Systems*

Gonzalo A. Aranda-Corral, Joaquín Borrego-Díaz, and Fernando Gómez-Marín

Departmento Ciencias de la Computación en Inteligencia Artificial
Escuela Técnica Superior de Ingeniería Informática
Universidad de Sevilla

Abstract. In this paper, a solution for semantic annotation of Consumers Generated Content, based on the actions of a multiagent system, is presented. This solution is designed in the context of an spin-off which was born from Minerva project for mobile services[1]. Our proposal aims to build a Mobile Web 2.0 platform for the dissemination of interesting events, captured by users through mobile phones. The multiagent system creates an agent for each event, which suggests and manages several semantic features linked to an Ontology related to events.

1 Introduction

Nowadays, we observe a key convergence into the future of telecommunications. On the one hand, the great success of mobile networks implantation provide us with a global and ubiquitous net and, on the other hand, the exploding of social nets in Web 2.0 provides to users with tools for generating content of any kind. The convergence of the two global networks (see fig. 1 (up), extracted of the book[4]) leads new kinds of personal communication and public sharing contents, example of mobile lifestreaming.

Despite this convergence -seems to be the "ultimate" convergence- more work on knowledge interoperability among heterogeneus devices (mobile phones, Web 2.0 mashups, Semantic Web or Intranets) will be needed. This convegence is called *Metaweb* in [10]. As we will argue in section 2, the special features of mobile devices make hard the convergence of Mobile Web 2.0 with the Semantic Web. These may lead a new digital divide between Mobile Nets and Semantic Web or Intranets. This problem should be (at least partially) solved before massive user content generation.

In this paper, we want to show how to solve the problem for synthetizing sound metadata for user contents by means of using a multiagent system, which controls and advices the semantic annotation (see 1, down). In this way, the

* Supported by the project 2C/040 *WeTeVe* of Minerva Platform of mobility services, http://www.proyectominerva.org/

[1] A coordinated project of *Andalusian Consejería de Innovación, Ciencia y Empresa*, Vodafone, *Universidad de Sevilla*, *Parque Científico y Tecnológico Cartuja'93*, *Sevilla Global S.A.M.*, ETICOM and AICIA.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 400–409, 2009.

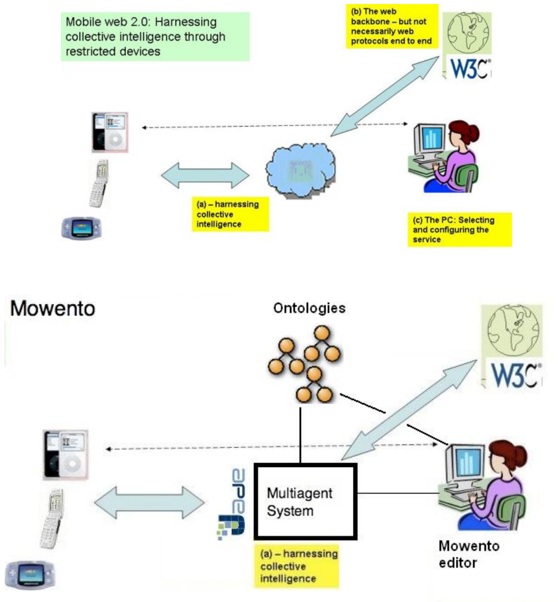


Fig. 1. Mobile Web 2.0(up) and Semantic Mobile Web 2.0 in Mowento (down)

system (called *Mowento*) bridges the gap between user tagging and annotation with respect to an ontology. Mowento is designed for testimonial documents publishing on events witnessed by the user at the time is going. These events are captured using mobile devices (photos and short videos). In section 4 we will describe this platform in more detail. These digital documents are sent to Mowento Platform, via MMS, to be published (in different ways, depending on user's account) into the WWW, as well as, an alert on this new event is broadcasted -via SMS- to Mowento neighborhood's user. The key engineering choice of Mowento is the use of a multiagent system (implemented on JADE[2]) to execute several semantic tasks on the documents. Each document, which is received in Mowento (via Vodaphone's Red Box platform), induces the creation of an agent associated to the document, and it will replace some of classical user Web 2.0 tasks. Here, we are focused on tagging, although there exists others semantic tasks, such as, visual ontology debugging inspired in [2].

2 Features of Mobile Web 2.0

"Mobile Web 2.0" (MW2.0) designation adresses to a new realm of platforms. Tools and social networking which share the main features of Web 2.0, but it also has specific characteristics. Some of these are related to mobile devices usability and user's behaviour. Briefly:

[2] http://jade.tilab.com/

2.1 Identity in MW2.0

Users in Web 2.0 are identified by its avatar and its history into the social net. In case of mobile phones, users aim to achieve a similar identity, althoug its main nick is the phone number. Thus, MW2.0 platforms and tools, must to be designed to grant this requirement. An important example is the management of personal tags.

2.2 Tagging and Mobile Devices

Tagging is a social method to categorize or to classify documents. In Web 2.0, tagging is a task that different Web sites consider it in many different ways, mainly [9]:

- For managing personal information
- As social bookmarking
- To collect and to share digital objects
- For improving the e-commerce experience

In case of MW2.0, and specifically Mowento (which is designed for fast publishing of documents related on events), third motivation is very important. However, tagging from mobile devices can be a tedious task. Therefore, an ontology on events have to be used as a consensus ontology for tagging digital objects by a mobile application. User mobility is not considered as a key feature for ontological commitments, because such complex information can be provided by the system/platform (using geolocalization and *post-tagging* systems).

2.3 Services in MW2.0

MW2.0 services can be created in two different ways [6]:

- By mobile extension of an existing Web 2.0 service.
- By a pure MW2.0 service specifically dedicated to mobile network and based on user-generated content. New tools for generated knowledge management must be designed.

2.4 Mobile Phones and Sharing Content

Semantic annotations of digital documents from mobile devices have several limitations, some of them related to local features of telecomms:

1. Mobile phones are tedious tools for writing content. Thus, mobile applications should simplify tagging task. Nevertheless, this barrier turns on an advantage if mobile application shows us, as available tags, classes from Mowento's ontology.

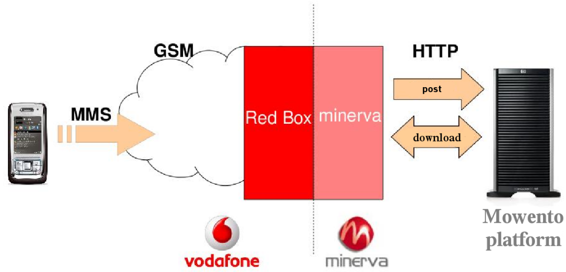


Fig. 2. MMS channel of Mowento

2. The user appreciate the inmediate generation of digital document on an event and its fast publication. This reason suggest the use of MMS channel instead of Mobile Internet (see 2). Moreover, in Spain, the relative higher cost of Mobile Internet services discourage this channel, if the application wants to have a great scope.

Regarding to document diffusion, MW2.0 platfoms must allow different levels of advertising and sharing, from private (own) use, personal use (shared with a trusted network), public, and even collaborative. That is, other users can add, transform or refine the information. Propagation through the WWW is a classic Web 2.0 service, enabled by WWW Mowento Platform, but distribution of this information in a mobile-based network is a more complex task than Web 2.0[8]. At mobile-based networks are important the *weak ties* because they connect social neighborhoods. Thus, a *micro-dissemination* of a document among the user's neighborhood ensures a greater impact than the publication in the WWW platform only. This decision is supported by the well-known thesis in Mobile Data Industry: *the content is not the king*, it is the contact [7]. In case of Mowento, the contact is the key bridge for micro-dissemination.

3 Semantic Web and Mobile Web

The convergence between Semantic Web and Mobile Web 2.0 depends on the specific management of ontologies. Ontologies and tags/folksonomies are two knowledge representation tools that must be reconciliated in metaweb projects. An useful bridge between these two kinds of representations could be Formal Context Analysis (FCA) [3]. FCA is a mathematical theory that formalizes the concept of "concept" and allow to compute concept hierarchies out of data tables, and it is also used for ontology mining from folksonomies [5].

In case of Mowento, FCA is used as an ontological representation of tags. Rougly speaking, ontology class X, in Mowento's ontology, is intended as *the class of digital objects with tags X*. In fact, Mowento's ontology is extracted from several experiments with our mobile application and an arbitrary set of tags. Attribute exploration method is used to refine this ontology, as well as, to obtain a system for (post)suggesting tags. This decision was useful to solve the tedious task of tagging by means of mobile phones, because the ontology, that is offered in a mobile application, is an hierarchy of tags extracted from concept lattice, given by FCA. In this way, the interoperability among sets of tags from different users it is (partially) solved.

Nevertheless, it is necessary to augment and improve the set of tags ,by means of suggestion tags method, related to original user's tags. Mowento use the stem basis [3], associated to the ontology of tasks, as knowledge basis for a *production system of suggested tags*. These will be implemented as a new agent's behaviour, associated to the document at the platform (see section 5).

4 Mowento's Services

Mowento platform has been thought to facilitate publication of audiovisual documents and communication between members of a mobile social network. In this way, the main focus were to implement a fast and simple solution where generated content management should be as automatic as possible.

Some experimentation steps has been done previously, which they have been useful to define basic features to implement this initial stage. The solution that is described here (see fig. 3) can be divided into five different components, as follows.

4.1 Mobile Application

Mobile application is a key tool due to audiovisual documents are generated at the moment. So the user should find it attractive, fast and easy enough, to create and to send documents to the platform. Members joining on the network depends on the experience they have on these first steps, so mobile app will be decisive for the project success or failure.

Speed and simplicity are the main characteristics to be considered. The user of mobile device is not at a very comfortable environment to spend time writing a full description for the document, or doing complex steps which require special

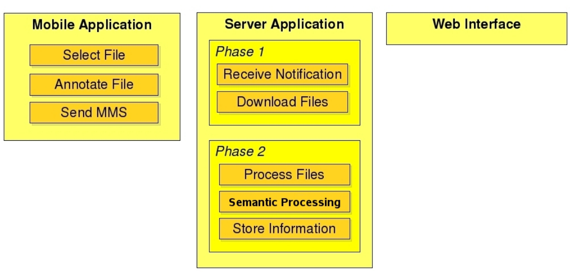


Fig. 3. Mowento service's structure

attention. Basically, actions should be "Shot And Send" (SAS action). So a mobile application on this point is required.

There are a huge number of mobile devices types and due to one of principal aims of this application is to be accessible as much people as possible. For this, the first prototype was decided to be implemented in Java. It has been used J2ME, Java version for mobile platforms. It is the most extended language, in this kind of devices, and will permit us a fast development.

The features implemented for this first prototype have been the picking of the selected file, that could be a short video or a photograph, with the current limit of 300KB due to MMS limitations. Any other kind of file have not been considered for this project. It is also implemented the basic annotation of the document. The fields, which are provided, are title, small description, tags set and date. Default values are available and empty values are admitted for a fast sending. Finally, sending the full audiovisual document by MMS.

4.2 Server Application

The next application which is needed, is on the server side, for automatic reception, management and organization of user's audiovisual documents, sent by MMS. Our first prototype developed is based on a traditional solution for web platforms called LAMP. We mean Linux as operative system (Ubuntu), Apache as a web server, MySQL as database and finally PHP for dynamic web pages. The LAMP system is a robust and tested enough by a huge community of developers in a great number of web projects, and also is open source and free. These features made it our choice. The full system run on a dedicated standalone server.Thanks to Minerva project, we have got access to the Vodafone's Red Box service for SMS/MMS handling. Minerva offers a simple interface to this service for research projects like ours.

The application server has two phases: First one is the MMS reception: each MMS sent to our platform is notified by Minerva using a POST message, by HTTP protocol. So when the server application receives the notification of a MMS, it captures all the related information in the message about that MMS. Using them, all of attached files from Minerva are downloaded to our platform by HTTP protocol, too.

Second one, is the downloaded file processing: all the information collected about the file is saved into the database, afterwards it makes the needed transformations to a suitable formats for its publication on the web -also creates photographs and videos' thumbnails- and translate from native video format to flash video format (FLV). For this first implementation, only the jpg format for images and 3gp for video have been taken into account.

4.3 Web Interface

A web interface it has been developed for publication and social network's audiovisual documents access. Although the final and main mission of this part is to encourage participation and interaction among network's members, it requires other web 2.0 features which are not implemented yet.

This interface have been developed using Symfony[3], a web application framework for PHP projects. It is an open source framework with a huge and detailed documentation. This kind of tool will permit a fast grown of the web interface. We will be able to implement more complex features at every version without making serious changes on basis.

Last version of this interface can be tested at the site http://mowento.cs.us.es. Currently, this version permits users login, viewing and searching audiovisual documents and a limited interactions among users. A user can comment the contents, vote both of them, content and comments, as well as he can make his favorite content and user's lists. Features that will permit us a first basic interaction between them. Tagging and post-tagging systems are transparent up to now, because the ontology is centered on academic events, at this moment.

4.4 Multiagent System

In order to implement all semantic features on Mowento, we will use a multiagent system, with a restricted set of semantic behaviours. In a first step, this multiagent system will be composed of two types of agents. The first agent group will have some generalized behaviours and it will be responsible for classifying the semantic tasks to execute. The second group will be a very specialized group to perform a few, very specific, semantic tasks. The first specific semantic task will be *Post-tagging* audiovisual contents, to help users managing its content. A greater repository of semantic behaviours will be created and grow as semantic behaviours are defined.

To support our multiagent system we choose JADE [1] which is an open source middleware under LPGL license. JADE offers a development and living environment as well as a set of libraries that allows to execute and to debug agents. This set of development libraries let us a fast way to implement operating-system and device independent agents. These will able us, in the near future, migrate some agents to mobile device and increase the semantic managment of contents at mobile platforms. Also some graphic tools are included to monitoring the agents in real time.

A key issue to using JADE platform is that all implemented protocols are FIPA-compliant and an interaction with other systems are simplified.

Focused on the behaviors, the first specialized one, will be a rule-based behaviour. For such systems, the Jess[4] (Java Expert System Shell) rule based engine is advised. Jess is implemented in Java and this will let us to integrate with Jade easily. It was inspired in CLIPS, and it can compile any file build for this. Jess can manage facts, rules and run the inference engine in the same way, too. Also java objects can be handled and this provide us the chance for creating deliberative behaviours and integrate them into agents.

[3] http://www.symfony-project.org/

[4] http://herzberg.ca.sandia.gov/jess/

Integration is made following the tutorial written at Jade website[5]. It starts creating a rule-inference engine and loading all facts and rules from a file. Finally, executing the engine to get all the new facts that can be deduced from original facts and rules.

4.5 ConExp. Making Rules

One of the most important elements in the semantic content managmement is ConExp [6]. This system is responsible for extracting the rules of deduction, by means of Stem basis, which subsequently become a production system and it will be loaded into a Jade agent.

5 Semantic Tasks

The semantic tasks to be realized in this experiment will be executed by a multiagent system, with a multi-role society where some different kind of agents can be found. Our aim is suggesting new knowledge(tags) to mobile-network users, which makes tagging process easier and improve the audiovisual contents organization. This task will be a *post-tagging task* by the WWW interface and it will be assisted by the multiagent system , in order to improve the features of MW2.0 explained before(see section 2), like managing personal information, collecting and to sharing digital objects.

First of all, we built a concept lattice, by means of attribute exploration, focused on "events" and trying to describe all multimedia content that could be generated by assistants through mobile phone's cameras. Processing this concept lattice with the FCA tool, Conexp, we can obtain the Stem basis associated to this lattice and transforms it into a production system and written in Jess syntax to be inserted into agents. The interpretation of an Stem Basis as a production system is sound, as a Tagging exploration method. It is based on the prior interpretation of Attribute Exploration method, (used as refinement method on Mowento's Ontology) . Roughly speaking, the modified version of attribute exploration changes original attribute exploration questions. Thus, "Every object that has the attribute $X_1, \ldots, X_n$ has also the attribute Y?" is replaced by "Every photo/video that has the tags $X_1, \ldots, X_n$ must to have the tag Y?". Also we can obtain an new ontology which represents the evolved knowledge, after post-tagging, which is included in the new lattice. (See fig 4).

The multiagent system has a specialized agent, which is the supervisor, for all contents and it plans all semantic tasks (see fig 5). Currently, the task in which we are interested for is "suggesting tags" by means of a deliberetive (rule-based) behaviour. Therefore, when a new content arrives to Mowento's platform, the supervisor creates a new agent to execute the proper semantic behaviour.

[5] http://jade.tilab.com/doc/tutorials/jade-jess/jade_jess.html
[6] http://conexp.sourceforge.net/

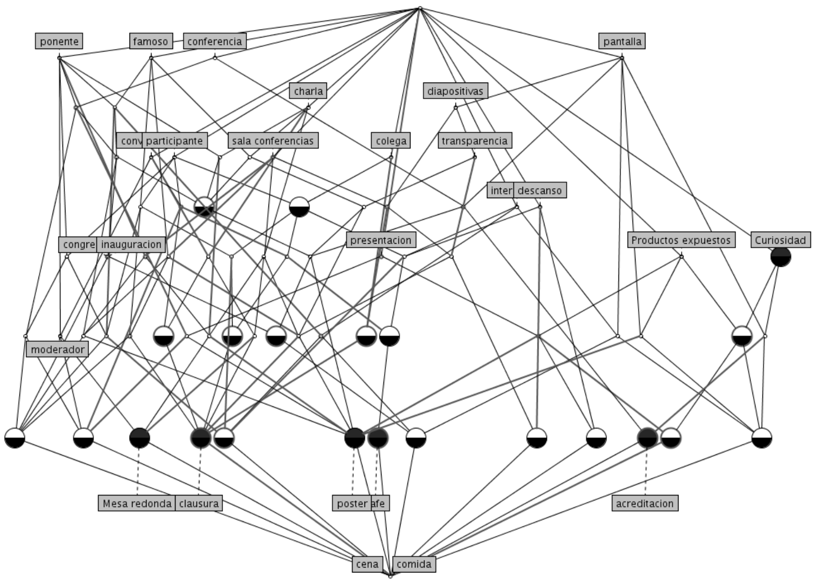


Fig. 4. FCA tool translation: Concept lattice

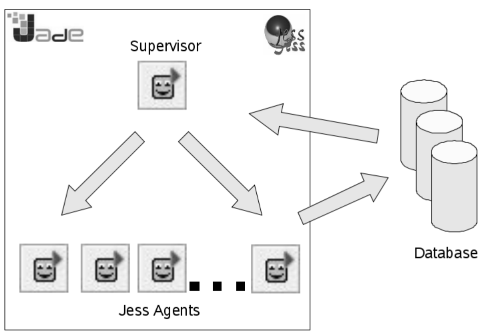


Fig. 5. Agents Architecture

This new agent, read the content from database, load the rules from the Jess file created above and initializes the rule's engine with facts (tags) written by the user. The engine is executed to obtain a new set of facts (tags) which will be inserted into database to be suggested to the user, through the web interface.

Formal Concept Analisys[5] has emerged as an interesting tool for the extraction of formal concepts from labeled audiovisual content. This is our first approach for building a repository of semantic behaviours and get totally semantic mobile social network.

6 Conclusions and Future Work

In this paper we present an application of a Multiagent System to Mobile Web 2.0. The aim of the MAS is the semantic management of user generated content, suggesting and adding semantic metadata through an specific interpretation of STEM basis.

The biggest handicap to use Semantic Web tools on mobile devices is related to usability of these as generator of metadata. This handicap is (partially) solved here, for some extent, using an agent platform monitoring the proper content managment generated by every user. We found that the use of Formal Concept Analysis tools and techniques can be very useful for generation of more sophisticated metadata that the user generates from mobile devices.

Future work is aimed to satisfying other semantic/cognitive requirements mentioned:

- Visual debugging of personal ontologies.
- Expanding, through the FOAF network of the user, the knowledge generated (with the labels suggested).

References

1. Bellifemine, F., Poggi, A., Rimassa, G.: Jade - a fipa-compliant agent framework. Technical report, CSELT (1999)
2. Borrego-Díaz, J., Chávez-González, A.M.: Visual ontology cleaning: Cognitive principles and applicability. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 317–331. Springer, Heidelberg (2006)
3. Ganter, B., Wille, R.: Formal Concept Analysis - Mathematical Foundations. Springer, Heidelberg (1999)
4. Jaokar, A., Fish, T.: Mobile Web 2.0: The Innovator's Guide to Developing and Marketing Next Generation Wireless/Mobile Applications. Future Text Pub. (2006)
5. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Discovering shared conceptualizations in folksonomies. Journal of Web Semantics 6(1), 38–53 (2008)
6. Martignoni, R., Stanoevska-Slabeva, K.: Mobile web 2.0. In: Conference Proceedings of the 20th Bled eConference, pp. 1–18 (2007)
7. Odlyzko, A.: Content is not king. First Monday 6 (2001)
8. Onnela, J.P., Saramaki, J., Hyvonen, J., Szabo, G., Lazer, D., Kaski, K., Kertesz, J., Barabasi, A.L.: Structure and tie strengths in mobile communication networks. Proceedings of the National Academy of Sciences 104(18), 7332–7336 (2007)
9. Smith, G.: Tagging: People-powered Metadata for the Social Web (Voices That Matter). New Riders Press (December 2007)
10. Spivack, N.: New version of my "metaweb" graph – the future of the net. Minding the Planet (2004)

Architecture of Multiagent Internet Measurement System MWING Release 2*

Leszek Borzemski, Łukasz Cichocki, and Marta Kliber

Institute of Computer Science,
Wroclaw University of Technology, Wroclaw, Poland
Leszek.Borzemski@pwr.wroc.pl

Abstract. Internet performance measurements have motivated the development of a measurement system called MWING. Recently the existing system has evolved according to new requirements and now also distributed measurements are handled in Release 2. MWING system Release 2 is a multi-agent system that is used to perform tests, measurements and diagnosis of Internet and Web performance from the end user perspective. Multiple agents freely scattered in the Internet can, in synchronous or asynchronous manner, probe collectively given set of Web sites, and collect mutual performance data. This data can be used in the analysis of Internet performance e.g. for the needs of Content Delivery Networks and Grids. This kind of measurements is also desired and helpful for Internet Service Providers since it enables monitoring and analysis how ISP's domain is perceived by end-users from various Internet locations.

Keywords: Internet measurements, performance evaluation, multiagent system.

1 Introduction

Network protocol measurement tools are utilized for better understanding of computer network. They help network administrators or end users to evaluate and analyze network and Web sites, mostly in the area of performance and reliability. Now, such tools are designed typically for the TCP/IP protocol suite, as well as for the main Web protocol, that is HTTP protocol. A good example is *wget* utility. Usually, they are used *ad hoc* to measure and observe same client-to-Web server interaction.

Several active and passive measurement projects have been built on the Internet, e.g. [2, 4, 6, 9 and 10]. Mostly they are aimed to deal with the performance problem related to whole or a significant part of Internet. These projects obtain large amounts of data measured among several node pairs over a few hours, days or months regarding, for instance, round trip delay, and using specific measurements and data analysis infrastructures where large amounts of measured data regarding, for instance, round trip delay among several node pairs over a few hours, days or months, and using specific measurements and data analysis infrastructures are obtained. Most of them usually have rigidly fixed measurements that do not change over time. The MWING

* This work was supported by the Polish Ministry of Science and Higher Education under Grant No. N516 032 31/3359 (2006-2009).

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 410–419, 2009.

platform is open; its agents can be exchanged over time. The MWING serves to all kind of agents that gather any kind of measurement data. This means it does not have to change with evolving information demands, its agents are changing, and the platform stays.

In case of Web measuring the Keynote [6] is the most advanced benchmarking service that measures Web site's performance and availability from a world-wide network of measurement agents. Unfortunately, now it is a commercial service. In previous years there was a demo version where a user could make some measurements of target URL and obtain valuable information about how the target page was seen by Keynote. However the user could not freely define the localizations of Keynote probing servers. To address this issue of complex systems development in distributed environment, we propose to employ a multiagent system (MAS) architectural approach [3, 5, 8] in the development of the system for Internet measurements.

In the paper, we present the Internet measurement platform for distributed measurements called MWING. The MWING has been developed with intention to be used for the purpose of Web page downloading processes analysis as seen from the user's perspective. Architecture of the MWING V2 release is an example of modern distributed programming solution. Because the system has been developed based on multiagent oriented architecture we can deploy (program) any needed measurement and processing functionality concerning Web and Internet. Agents usually are designed to meet special needs, thus making difficult to communicate with them. Some tools like gateway agents try to solve this problem using, for instance, intermediate XML-based messages [3]. Our system has this functionality built-in.

Real-life experimentation with the previous version of system the MWING gave us the basic understanding of how to actively collect and analyze Internet measurements. Gathered information gave us detailed view of how data transmission occurs in various conditions. Unfortunately, in the previous version of the system MWING [1] we could setup only the *starlike* experiments and collect measurements performed from a single Internet location to a multiple Web sites. Now we can setup the *distributed starlike* experiment where a number of Internet locations may have system agent installed that can collectively gather performance data from given sets of other Internet locations. In particular, these agents may probe, either in a synchronous or asynchronous manner, the same set of Web sites to collect performance and reliability information as perceived from various Internet locations against the same target list of Internet locations.

This paper is organized in following way. Section 2 presents main architectural features. We also show how the system evolved. Section 3 presents agents of the system and how they are defined and registered. Section 4 describes a communication mechanism and central data collection. Section 5 concludes our work and discusses future plans.

2 General System Architecture

General system architecture is presented in Fig. 1. This architecture includes two types of system nodes *OpHost* and *Agent Set*.

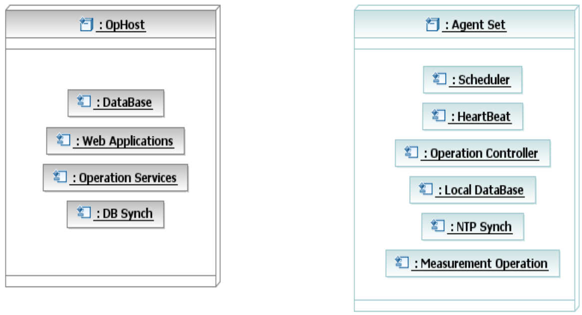


Fig. 1. MWING Rel. 2 - general architecture

OpHost (abbreviation from Operational Host) – this is the controller component of the system that executes management tasks. This is a unique system node that is central to the whole system. It consists of following system components:

- *Database* – central DBMS configured for the needs of the measurement system.
- *Web Applications* – this system component enables communication between system and a user. It features all of the system functionality to enable management of the system.
- *Operation Services* – this component runs procedures that are responsible of managing agents' connectivity and operational parameters, data synchronization between central and agents' storage, processing of heartbeat and reporting gathered results as well as heartbeat data.
- *DB Synch (Database Synchronizer)* – has only one particular function that is to gather measurement results in a central database.

The *Web Application* component provides the user interface to the system and enables managing of operations related to all aspects of the system starting from agents' registration, modification of agents' parameters, addition of new schedules, as well as management of heartbeat and synchronization settings. Gathered data can be displayed thanks to *Result Data Browser* subcomponent while the other one called: *HeartBeat Browser* presents heartbeat data.

Operation Services consists of following subsystems:

- *Operations* – it executes all the actions transferred from the system by user via Measurement Management. Here is placed an engine orchestrating all components activity and ensures that deputed operation is performed as intended and its result is returned.
- *Agent Communication Interface* – is a subsystem that ensures connectivity with an agent to perform management operations.
- *Database Access* – this is a subsystem to put and get data from central database where all agents' results are stored.

Agent Set – this node is an agent node which is responsible for commissioned jobs execution according to given schedule. Agent Set consists of six main components:

- *Measurement Operation* – is responsible for measurement algorithm execution, it performs intended activity and gather raw measurement results. In this example measurement operation component is a WING measurement algorithm implementation. However here can be used any other measurement algorithm.
- *Scheduler* - gives the ability to perform measurements according to some defined schedules that are freely programmable.
- *Local DataBase* - system component intended to support local data storage.
- *Operation Controller* – is the main system component which orchestrates all measurement actions.
- *Heartbeat* – it is an additional component that provides data, based on which it is possible to verify the continuous operation of an agent. Heartbeat data are information sent from an agent to the central node. This information can contain system health indicators such as: free disk space, CPU load and physical memory in use. Heartbeat messages as such indicate connectivity to the agent. As long as central unit receive expected heartbeat messages it assumes that the agent is available. Heartbeat function is crucial for reliable operation as every problem is reported to the central node where a user can trace the Agent Sets health. Since these messages play only supportive role and are not adding any value to measurements they should be kept short.

Comparing the MWING V2 architecture with that developed earlier (Table. 1), one can notice not only the reorganization of components but also some deeper changes. The main difference is that the system V2 is oriented on agents' autonomy. Now the central node has knowledge only: (i) how to start an agent, (ii) what are agent's results, and (iii) what is agent's performance/reliability state. Feature (i) is only about

Table 1. MWING V1 versus MWING V2

What has changed?	MWING V1	MWING V2
Measurement management responsibility	Central component controls agents activity (each measurement execution)	Oriented on agent's autonomy; Central component only monitors agents' states with heartbeats
Scheduler location	In central component which controls agents activity	In agents which decide on their own when to operate
Architecture	Starlike	Distributed starlike
Communication	Sends and receives configuration and operation control data for each measurement performed by an agent Transfer result data after every measurement execution	Configuration data are send to an agent only on its initial setup or reconfiguration Result data transferred in defined time frame Agent's feedback sent with heartbeats

parameters that are required to run an agent according to user needs. Central storage of results requires knowledge of aspects (ii). (iii) is to monitor agent's state and ensure that agent is running as planned and there are no issues critical to agent's activity.

3 Sample Agent

Sample agent is an example of possible agent construction. It consists of two parts:

– Algorithm of the measurements that is performed with *Measurement Operation*,
– Reusable set of components that in our Agent Set node that consists of: *Scheduler*, *Local Database*, *Operational Controller* and *Heartbeat*.

Measurement Operation is the crucial part of an agent, it defines what kind of activity is performed by the agent, what information can be achieved and how. Our sample agent is the WING agent [1]. This agent gathers the Web transaction information measured from the end user perspective. Obtained data correspond to following values:

- MOR – complete download time,
- DNS – resolving IP address for a Web Server period,
- D2S – time between end of DNS communication and beginning of the TCP connection to the web server,
- CON – TCP connection time,
- A2G – time from sending package with ACK flag for TCP connection to sending GET request package,
- FIRST – time from sending GET by agent till the first package with HTTP response comes from the Web server,
- LAST – downloading the rest of requested object period,
- SIZE – size of downloaded object.

In its *Measurement Operation* component the WING's algorithm is placed. The agent is right now multi threaded application that orchestrates its components, inter alia browsing module and sniffer, to achieve desired goal i.e. gather time markers for web transaction of downloading fixed URL resource. In this version of the WING agent we used *wget* tool as browsing engine. This tool has one important advantage over the Internet Explorer solution used previously; it can run in UNIX environment with minimal CPU usage. The load is important since during the activity of WING agent the packet sniffing process is run and high load could influence probing and distort gathered results.

Apart from *Measurement Operation Agent Set* consists of reusable components that enable it to operate in an environment and communicate with the central node. For our sample WING agent as well as all the other ones these components are responsible for the following functions: local data storage, scheduling, synchronization, heartbeat functionality, and orchestration of measurement execution. There among others the communication protocol for connectivity with the central node is defined. These components provide all functionality that enables each single agent to be the Agent Set of the MWING V2 system.

In Fig 1 we can find also *NTP Synch* component of the *Agent Set*. This one enables the agent to work in a synchronized mode. The synchronization with a given NTP server or servers is handled by the operating system where the agent resides. Synchronization is a feature that enables execution of compound measurement series at once on one selected target. In this manner the user can be sure that at the same moment the system probes selected Web server with every working agent.

When we have our measurement algorithm implemented we can add to it all reusable components and provide it as the Agent Set node. The next step is to let the MWING system know about it. This is done with the registration. In the registration process we provide information about the agent's:

– input parameters required for its configuration purposes, both for measurements and heartbeat,
– output data set, what values it returns and what are their data types.

After the agent is registered in the MWING it can be activated by a user. Activation of the agent's single instance compounds of providing following information:

– where the agent is located,
– authentication information, i.e.: username, key and port number, for ssh connection,
– set of values to initiate its configuration parameters.

Since this moment the agent is up and running, gathering measurement data.

Usually the agents are parameterized with some server targets to measure. These server targets that are of interest to the agent are placed in the list of servers or when agent runs in *ad hoc* mode are given directly. For scheduled measurement tasks a list of targets is usually provided and each of its inputs is probed once at the time. Single processing of the whole list composes the measurement series. Items in the list can be freely changed by agent according to the environment conditions, e.g. non-operational targets can be removed or exchanged. They are read each time a series starts. Changes in the list can result from changed interest of targets, connectivity problems to items and other analysis results. If synchronized mode of agent is enabled, all instances of the same type of agent running with the same schedule can probe the same target at one time.

After measurement data are gathered they must be returned. There are three possibilities to return gathered data. In *ad hoc* mode just a single measurement is performed. This enables user to run agent against selected target WWW server just for some simple once-run only test. In this situation the results are simply displayed in the output window along with logging information. This option enables to use the agent as an on-line Web benchmarking service. For regular measurements that are planned and scheduled for a longer period of time it is proposed to store data in a local database or a CSV file. Since local database storage is the most efficient. This option is fully supported by the Agent Set. To use this option an agent do not need to take care about communication with database. It returns data to Agent Set which on its own connects to DBMS and sends data to configured database. Agents are not confined to use local database. They also can use legacy interface which is compatible to one used by the central node when communicating with local DBMS-s. With this legacy system agents store gathered data in CSV files. When this is the case the agent's environment is responsible for sending data to the central node.

The agent has modular architecture and all of its components that are not related to principal agent's activity are components that might be useful for all type of agent. This fact directs further works to creation of a framework that would provide all common functionality so that developer would be concerned only with implementing agent's essential functionality.

4 Communication

Each MWING agent is an autonomous system that operates self-dependently. Our goal is to take advantage of work of all operating agents and build the whole picture from pieces retrieved from a single agent. Merged agents' results are more valuable so there is a central system that gathers data from all agents. It is also preferable to have ability to configure agents so that their activity is managed. To achieve all these tasks the reliable communication mechanism is a must.

The OpHost component is responsible for managing all agent sets which have been registered. It sends new or updated input data to program agent set's behavior. It also initializes the connection with agents for sake of measurements' results data retrieval.

The OpHost uses SSH protocol for communication with agents. Encrypted connection is established each time when new or updated configuration data needs to be uploaded on a given agent or when next portion of result data needs to be downloaded from the agent. Each Agent Set's environment has to have SSH service started and configured according to the setup data provided during registration on the OpHost. Setup data contains information such as agent's IP address, port number, RSA authorization keys. Due to the fact that the SSH based connection is always initialized from the OpHost – there is no need to store any authentication information on the Agent Set's machine.

During first initial configuration, data is uploaded from the OpHost to an agent. Based on it, agent's HeartBeat component can start to send its feedback information (maintenance, performance, detected problems) to the OpHost. By default HearBeat listener service is activated on the OpHost side. It uses UDP protocol for data sending. HeartBeat component installed on Agent Set's environment transmits heartbeat messages in a programmed order. They are received by HeartBeat listening component and stored in database for further processing (e.g. alert generation in case of missing heartbeat reports). It is also possible to target heartbeat reports to any other location.

To present communication flows within the MIWNG system we describe several system operational scenarios. UML sequence diagrams are used for a visualization purposes for a given scenarios. Following scenarios were selected:

- new MWING-type Agent Set registration in the system (agent set enabling for measurements)
- execution of the measurement procedure done by the agent which we previously set up
- collecting measurement result data from remote agents by the OpHost node
- heartbeats propagation from distributed agents to the OpHost node

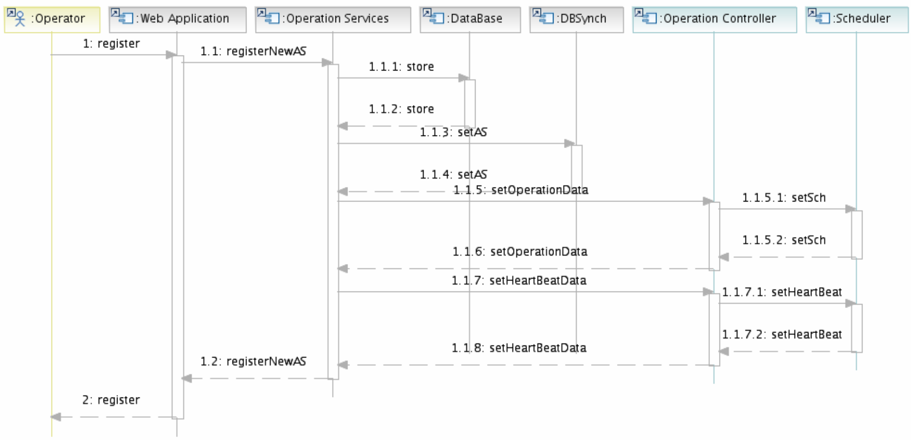


Fig. 2. New agent registration

Now we present the sequence diagrams showing the communication between the central component and the agent. At first, each agent that operates within the system needs to be registered. The sequence diagram presented in Fig. 2 shows the ordered steps of agent registration task. At the top of the diagram the components that take part in activity are listed. The gray ones are located in OpHost node, the blue ones in Agent Set. The registration operation is initiated by Operator, which is any system's user. Such operator fills in the registration form available with Web Applications and sending it lunches the whole sequence of related tasks. OpHost saves received configuration information in its DataBase and also sets DBSynch component to program database synchronization procedure schedule for this Agent Set. Afterwards operational data parameters (scheduler input information and input data for measurements) are sent to the Agent Set. The similar operation is performed for the heartbeat parameter configuration on Agent Set. All this information is then processed at the Agent Set side. Finally, the operation feedback is send to Web Applications component and then to the action requester.

Sample registration for WING agent is fulfilled in with web application form where a user puts information about:

1. configuration parameters, which for WING agent are:
   a. server target list which in particular can be single url address along with corresponding identifiers
   b. agent unique identifier
   c. interface on which agent listens
   d. level of messages to log
   e. time when the series of measurement starts
   f. scheduling for measurements, which simply implies which scheduling algorithm should be used (e.g.: uniform distribution, normal distribution) and with what values of parameters
   g. scheduling for database synchronization which can use the same scheduling as scheduling for measurements

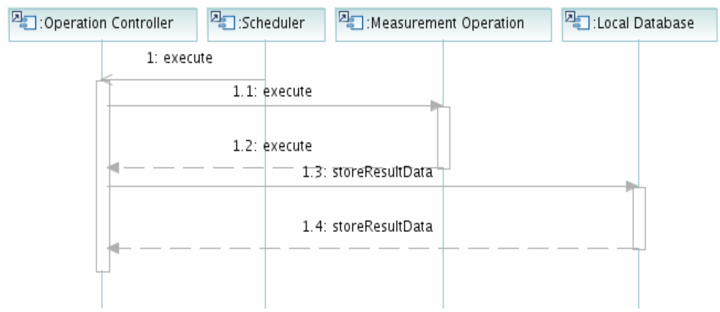


Fig. 3. Measurement Operation Execution sequence diagram

2. HeartBeat configuration which includes:
   a. scheduling which can use the same scheduling as scheduling for measurements or can be triggered with events such as start or end of measurement
   b. Agent Set's information that ensure reliability of its results

Fig. 3 illustrates single measurement execution. This operation is performed entirely at Agent Set. It is initiated with agent's Scheduler which calls Operation Controller that is responsible for two actions:

– calling the Measurement Operation that save collected results into temporary file,
– waiting till Measurement Operation finishes and then loading data from temporary measurement file into prepared database table.

Collecting measurement results from remote agents is action initiated at the OpHost side. In fixed time frames DBSynch component connects to respective agent and downloads file with results. Result file contains data exported from agent's local database from period between two adjacent time frames. The file is identified by a name and identification suffix. If for any reason the operation fails when next time frame comes the DBSynch downloads not only the latest but also these which have not yet been received. In case of problems system can also send alerts.

The heartbeat operation presented in Fig. 4 is both Agent Set and OpHost activity. As it is with measurements also heartbeats are scheduled so the Scheduler initiates

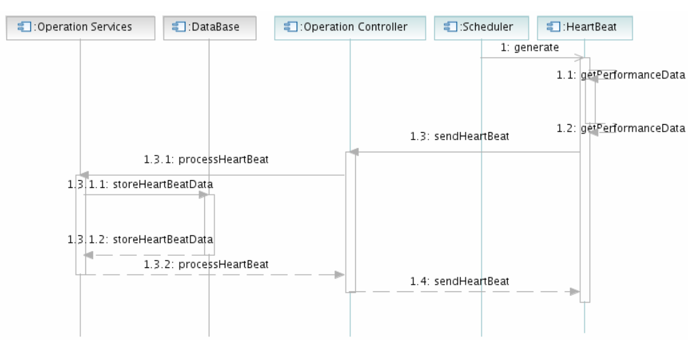


Fig. 4. HeartBeat sequence diagram

also this action. First HeartBeat must gather required Agent's Set information which is then send to Operation Controller. It takes responsibility for sending the message to Operation Services of the central node. If the central node receives the information it is stored in the central database and acknowledgement is sent back to the Agent Set.

5 Conclusion

Further works involves development of an open architecture agent framework and new agent types that would analyze other network parameters. The first planned step is the development of the framework that will structure all agents, so that each of them will have abilities enabled in common shared part. This means that all data storage types, all communication mechanism will be given to each type of agent and also this will ease development of other kind of agents that are supposed to be part of the MWING multiagent system. All the parts that the framework will be build from are these that in Section 2 were listed as agent's reusable components. After completion of such framework the creation of different types of agents will be much faster and easier. The second evolution direction is development of automatic data mining analysis. Current data mining engine that is run against gathered data is to be extended and enriched of new algorithms. This requires deeper investigation of how to implement this solution and how much specialized and from the other end general and common should such solution be. The third way of research focuses of data management that assumes creation of super-database that would be a data mart build at the top of all gathered results from all types of agents.

References

1. Borzemski, L., Cichocki, Ł., Fraś, M., Kliber, M., Nowak, Z.: MWING: A Multiagent System for Web Site Measurements. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS(LNAI), vol. 4496, pp. 278–287. Springer, Heidelberg (2007)
2. CAIDA, `http://www.caida.org`
3. Chen, J.J., Su, S.W.: AgentGateway: A Communication Tool for Multi-Agent Systems. Information Sciences 150, 153–164 (2003)
4. Luckie, M.J., McGregor, A.J., Braun, H.-W.: Towards Improving Packet Probing Techniques. In: ACM SIGCOMM Internet Measurement Workshop, pp. 145–150 (2001)
5. Manvi, S.S., Venkataram, P.: Applications of Agent Technology in Communications - a review. Computer Communications 27, 1493–1508 (2004)
6. Mogul, J.: Clarifying the Fundamentals of HTTP. In: Proc. of WWW 11 Conference, Honolulu, pp. 25–36 (2002)
7. MyKeynote, `http://www.MyKeynote.com`
8. Park, S., Sugumaran, V.: Designing Multi-Agent Systems a Framework and Application. Expert Systems with Applications 28, 259–261 (2005)
9. Paxson, V.: Strategies for Sound Internet Measurement. In: Proc. ACM SIGCOMM Internet Measurement Conference, pp. 263–271. ACM Press, New York (2004)
10. Prasad, R.S., Murray, M., Dovrolis, C., Claffy, K.: Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. IEEE Network 17(6), 27–35 (2003)

Information Flow in a Peer-to-Peer Data Integration System

Grażyna Brzykcy

Poznań University of Technology, Institute of Control and Information Engineering,
Pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Poland
Grazyna.Brzykcy@put.poznan.pl

Abstract. In the paper a problem of semantic integration of XML data in a peer-to-peer agent system is considered. In the system each agent manages its local data, and can communicate and cooperate with other agents by asking and answering queries. Problem of information flow between local data resources is analysed upon a channel theory. The abstract notions of classification and infomorphism are used to clarify the roles of queries and mappings in the system. The query reformulation task is modelled by the composition of an appropriate infomorphisms. In order to specify classifications and infomorphisms ψ-terms are used. Logic programming systems are chosen because of the logical variables and partial data representation, effective techniques of data unification and partial evaluation, suitable built-in terms and procedures, and automated reasoning.

Keywords: P2P system, XML data, schema mapping, query rewriting, declarative specification.

1 Introduction

Nowadays, the problem of data integration is often considered as a task of exchanging and merging data from loosly coupled, and autonomous systems. The notion of P2P (peer-to-peer) architecture is used for the architecture of distributed database [5], and, more generally, for the sets of data. In the P2P architecture there is no global schema, to which queries are directed. Then, it is assumed that a system user may address a query to any local data source. It is worth noticing that P2P architecture is a special architecture of a multi-agent system (e.g., [21]), in which communication between agents (here: partners) is based on asking and answering queries. The agent, by means of relations with the other agents, may insert their information resources into the process of query evaluation. It is progressed to pass the original query to the partner, while, most frequently, the query is reformulated in order to take into account a data schema of the asked agent. Propagation of the query through the system to the next partner's partners induces the final answer to be built of data originated even from the agents, which are unknown to the local one.

We may indicate a few typical problems, which appear in P2P databases. There is necessity to exist a mapping between data schemas of two partners and effective

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 420–429, 2009.

algorithms of query reformulation. One also has to determine a query addressee, and to merge partial answers from divers partners. In the literature there are a lot of works devoted to the problems (e.g., [3, 6, 11, 12, 14, 15, 22]). Studies of the problem of semantic integration of XML data in the multi-agent environment have been also conducted at Poznań University of Technology (e.g., [13, 17, 18, 19]). One of the trends within the studies is connected with the idea of using declarative environments to specify above mentioned problems. We analyze the operations of mapping composition and merging of partial data. Up to this time a few specifications have been presented: the architecture of the system [10], representation of data schemas and data instances, mappings and queries [7, 8], and examples of using unification operation in the task of data merging [9].

The choice of declarative environments seems to be rationally justified, particularly in semantic-oriented systems. Currently, the logical approaches are most often connected with representation and processing of ontological knowledge, however, it is possible to indicate works (e.g., [20]), in which Prolog representation and partial evaluation technique is successfully used to reformulate queries over conceptual (ontological) level into queries to relational databases.

In the paper we present a developed form of the previous papers (e.g., [10, 7]), Particularly, we show that in order to solve a query reformulation problem we only need a notion of query and an operation of query composition. Traditionally queries are distinguished from mappings, but the distinction gives an impression of being unnatural and even mistaken one, because both queries and mappings play a similar operational role in the system, namely, they are used to gain information. The statement has been a starting point in the search for a proper abstraction of a task of information flow between agents. The new abstraction is intended to be aimed at clarifying concepts, simplifying tasks and algorithms and generalizing information processing to inference. In the paper concepts of classification and infomorphism [4] are chosen to interpret the task of information flow. We show that the generalized data structure, ψ-term [1], correctly matches the notion of classification, and Prolog-like computation environments may be effectively used to model and process infomorphisms. Moreover, the same data structures are suitable to model integration of partial and even inconsistent knowledge ([16]). We also present an extended (relatively to [20]) solution of the query reformulation problem for XML data, and, generally, for labelled graph structures. As a consequence of the generalization, we obtain a more homogenous representation of queries and mappings and we can formulate the data integration problem in the more natural way. Moreover, the partial evaluation technique, widely recognized in Prolog systems, can be used to simply compose queries and to define an interface between diverse data nad computation models.

2 Classifications and Infomorphisms

The basic concepts of the theory of information flow are presented by Barwise and Seligman in [4]. The Authors stress the role of regularities that underlie any information flow in the distributed systems. They even form The First Principle, that information flow results from regularities that link parts of the system in a such way that the flow is possible. To model regularities and connections within the system that underwrite the information flow two fundamental concepts are used: the first one is a notion of classification and the second one – a notion of infomorphism.

The term classification, commonly known in diverse areas of interest, is always tightly connected with tokens, which are also called instances or objects, and with types. One can construct a simple example of classification with integer data as tokens and types like a negative number, a non-negative number, a positive number and a zero number with a natural relationship between tokens and types (it is worth to notice that any token may be classified as being of more than one type). If a classification models some device, and the tokens represent all the possible instances of the device, then the types represent all the properties of the device relevant to the model. Below a formal definition of classification is recalled from [4].

A classification **C** consists of three components: a set tok(**C**) of tokens (objects to be classified), a set typ(**C**) of types (objects used to classify tokens), and a binary relation $\models_{C}$ over tok(**C**) × typ(**C**) that defines which tokens are classified as being of which types.

In order to cope with moving information in a distributed system, an infomorphism relationship between classifications is defined. The relationship is the main concept of the theory of information flow. An infomorphism inf: **A** ↔ **B**, from a classification **A** (a domain) to a classification **B** (a codomain), is a pair of functions inf = <inf ˆ, inf ˇ> such that: inf ˆ: typ(**A**) → typ(**B**), and inf ˇ: tok(**B**) → tok(**A**), satisfying the following fundamental property: inf ˇ(b) $\models_{\mathbf{A}}$ α iff b $\models_{\mathbf{B}}$ inf ˆ(α) for each token b from a set tok(**B**) and each type α from a set typ(**A**). One can combine infomorphisms to obtain another infomorphism. This nice and valuable property makes an infomorphism a particularly suitable model for information flow and data integration problems.

Let us assume that there are two agents in a system. The first one, an agent Y, collects data about year of birth (integer numbers) and classifies them by types 1900, 1901, ..., 2009. The second agent, the agent A, holds data about person's age and classifies them by two types: child (any person that is less than 18 years old) and adult. It is obvious that a person's being born 1980 (she is of type 1980) carries the information that she is an adult one (she is of type adult person). The definition of infomorphism is straightforward for these classifications:

$\text{inf}\,\hat{}\,(Y_{type})$ = child, if Y_{type} > the_current_year – 18,
$\text{inf}\,\hat{}\,(Y_{type})$ = adult, if $Y_{type} \leq$ the_current_year – 18
$\text{inf}\,\check{}\,(A_{tok})$ = the_current_year – A_{tok}

The practical sens of the above functions is underpinned by an assumption that we take into account only a year of birth and an age of the same person. The supposition is modelled in [4] by a concept of an information channel. An information channel consists of a family of infomorphisms with a common codomain.

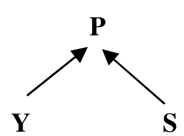


Fig. 1. An information channel with classifications **P**, **Y** and **S**

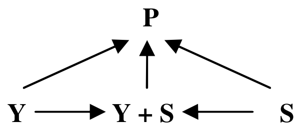


Fig. 2. A commutative diagram of an information channel (with classification **Y** + **S**)

In trying to build more rational example of an information channel, without the redundant information about the year of birth and the age of any person, we introduce a new classification **S** of persons' sex, with two types: female and male. The classification **P** (on the figure 1) denotes a simple classification of persons.

As it is stated in [4], a great construction is allowed for classifications and their infomorphisms – they can be added, or combined into a single classification. For example, it is possible to define a unique classification **Y** + **S** and all the added infomorphisms such that the following diagram commutes (figure 2).

The classification **Y** + **S** forms a basis for comparison of types that classify tokens from both classifications. The tokens of classification **Y** + **S** consist of pairs (y, s) of tokens from **Y** and **S** appropriatelly. We are interested, of course, in those pairs that represent a year of birth (y) and a sex (s) of the same person. One can think of the token of **P** as a "connection" between its components, namely the y and the s. The observation is at the heart of the idea developed in the next section.

3 Representation of XML Data

We assume, that XML data are represented, like in [22], as nested relations. Attributes of the relations are of different types (in the classical sens). Atomic values of attributes are of the simple types (we assume *Int* or *String*), whereas repeatable elements of the same type are treated as elements of a set, and sequences of elements of different types form records. Let us analyse schemas of XML data depicted in figure 3.

We assume that the schemas *source1*, and *source2* correspond to data sources with the very similar domain of application. The schema *answer* reflects the answer data of a query defined by a user and submitted to both of the sources. The * symbol in schemas is used to denote sets (set *xxx** of the elements of the same type *xxx*), and the symbols *Nxx*, *Yxx*, and *Sxx* stand for name, year of birth, and sex, respectively,

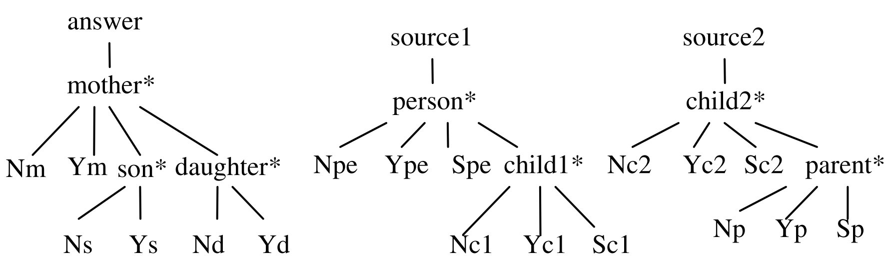


Fig. 3. Examples of schemas (for the answer and the data sources)

attributes of elements from diverse sets (e.g., *Nm* is a name of *mother*, and *Yc2* – a year of birth of *child2*).

Let us interpret the trees in figure 3 as channels with appropriate classifications and infomorphisms. Each *person* from the *source1* consists of four components: a name (*Npe*), a year of birth (*Ype*), a sex (*Spe*) and a set of children (*child1**), with each child (*child1*) built of a child's name (*Nc1*), a child's year of birth (*Yc1*) and a child's sex (*Sc1*). Tokens of the last three components, taken together like in a sum of classifications, form natural description of a child (*child1*). To represent a person (being a parent), one has to indicate the appropriate set of children (*child1**), augmented with a name , a year of birth, and a sex of the person.

In declarative environments, like Life [2], one can find generalized terms, or ψ-terms [1] – data structures that are particularly suitable to represent objects and classes (also called the labelled graph structures), and, of course, schemas and XML data. Each term is of some sort and may have features (symbols on the left side of =>) that describe roles of arguments in the ψ-term and that have values being ψ-terms as well. For instance, any person *X* may be represented as follows:

```
X:person(name=>Npe:String,year=>Ype:Int,sex=>Spe:Sex,
         children=>Ch1s:list_of_ch1)
```

where *X*, *Np*e, *Ype*, *Sp*e and *Ch1s* are variables, which denote, appropriately, terms of the sort *Strin*g, *In*t, *Sex = {male, female*} and *list_of_ch1*, which consists of the terms:

```
child1(name=>Nc1:String,year=>Yc1:Int,sex=>Sc1:Sex).
```

A variable may be used in a given ψ-term many times and each time it denotes the same sharing value.

If we consider only structural aspects of data we can find also other advantages of ψ-term representation. The main sort in any ψ-term may be used to carry information about data type. Therefore, one is able to find not only components in the representation of the composite token *X*, but also a type of the token.

Moreover, one can freely adapt details of a description to a task in hand. For instance, any element of the type *child1* may be represented as *Ch1*, where *Ch1* is a variable name. More precise description of the token is also possible and may be in a form of a set of features. Also, a whole set may be named by a variable, for instance *Ch1s* denotes set in the term of the type *person*. To show that a child *Ch1* belongs to the set *Ch1s* it is possible to use the *member* predicate (built-in predicate in declarative systems) and specify the condition *member(Ch1,Ch1s).* The empty set is represented by the empty list, respectively. The Prolog-like languages and logical variables are even more suitable for data representation than it is seen at the first glance. Logical variables may be used to represent partial data in a natural way. If we know that a child is a girl and has a name *n* but a year of her birth is not known yet, then in the set of children from the source *source1* the incomplete data may be represented as a term *child1(name=>n: String,sex=>f:Sex)*, where *n* and *f* are constants (*f* stands for the value female). Moreover, the same variables denote the same data, so it is rational to consider the term *person(name=>N:String, year=>Y:Int, sex=>S:Sex, children=> h1s:list_of_ch1)* from the set *person** and the term *parent(name=>N:String,year=>Y:Int,sex=>S:Sex)* from the set *parent** to be different representations of the same human beeing, with the same name *N*, the same year of birth *Y*, and the same sex *S*.

With unification operation in logic programming environments we are able to integrate partial data. For instance, taking a term *mother(name=>N:String, year=>Y:Int,sons=>Ss1:list_of_so,daughters=>Ds1:list_of_da)* from a set *mother**, with a term *S1:son(name=>Nc1:String,year=>Yc1:Int)* being a member of the set *Ss1*, and the term *mother(name=>N:String,year=>Y:Int,sons=>Ss2:list_of_so, daughters=>Ds2:list_of_da)* with the term *D2:daughter(name=>Nc2:String, year=>Yc2:Int)* in the set *Ds2*, one can form the appropriate component data of the answer schema, particularly, the term *mother(name=>N:String,year=>Y:Int, sons=>Ss:list_of_so,daughters=>Ds:list_of_da)*, where the collection *Ss* contains the term *S1*, and the collection *Ds* contains the term *D2*.

4 Queries and Mappings as Infomorphisms

With the theoretical foundation of information flow the more general approach to data integration problem is possible. In trying to model the problem of gaining information from a P2P collection of data sources researchers have to focus on notions of classification and infomorphism. Because any classification is defined by tokens and types it is sure to reveal types of data. The postulate related to XML data means explicit representation of XML data stucture (XML schema). An information flow is possible between classifications and involves both types (XML schemas) and tokens (XML data).

Let us investigate more carefully concept of query and process of query evaluation by an agent using the notions from information flow. To simplify the consideration, one can assume that we are dealing with a single schema of data in a database (which does not influence the generality of deliberation). Therefore, the first classification is established. Let us assume the *person* classification from the figure 3 as an example. To obtain any information a query is formed and evaluated against the current state of a database (the tokens of the database classification).

The query consists of a few parts, defined in query languages more or less explicitly. An enquirer has to know the database schema, has to define the schema of answer (what type of data she is interested in) and has to establish correspondences between elements from the database schema, and elements from the answer schema. Therefore, the second classification from the model of information flow, related to a query, is built of the answer schema, which forms a complex type, and has an empty set of tokens at the moment when the query is defined. But during the evaluation process answer data come from the database and fill up the set. For example, a query is assumed about mothers and their children with sons and daughters specified separately (see figure3).

Besides the classifications, in the query there is also a specification of interpretation function on the types. The specification stands for the “upper” function from an infomorphism. The function mapps the elements of an answer type of the query to the elements of the database schema. Also, the process of finding answer data is represented in the infomorphism. It consists of putting forward the instances of answer data, built upon the actual content of the database, and is modelled in the infomorphism by the function on the tokens.

Infomorphisms have a lot in common with the notion of language interpretation. Particularly, the concept of *mother*, with all the component data, has to be defined by means of the concept *person* and its relevant parts. In an infomorphism, the step is modelled by a function on the types and can be interpreted as translation of languages. However, to form an answer, an agent should not use the concept only, but rather the proper instances of the concept (data tuples, tokens).

The way of defining one concept by means of the other one lays at the heart of rules in declarative languages. Therefore, in Prolog-like environments the representation of query is very natural, particularly with ψ-terms as data structures. The request Query about mothers, their sons and daughters is specified below:

```
mother(name=>N:String,year=>Y:Int,
       sons=>Ss:list_of_so,daughters=>Ds:list_of_da) :-
  person(name=>N:String,year=>Y:Int,sex=>f:Sex,
        children=>Ch1s:list_of_ch1),
  set_of(son(name=>Nc1:String,year=>Yc1:Int),
        member(child1(name=>Nc1:String,year=>Yc1:Int,
              sex=>m:Sex),Ch1s), Ss),
  set_of(daughter(name=>Nc2:String,year=>Yc2:Int),
        member(child1(name=>Nc2:String,year=>Yc2:Int,
              sex=>f:Sex),Ch1s), Ds).
```

Any mother is defined as a female person with a set *Ch1s* of children, among which sons form a subset *Ss* and daughters – a subset *Ds*. The built-in Prolog predicate *set_of(T, B, L)* is used to produce the set (list) *L* of those terms *T*, which fulfill the condition *B*. Appropriate correspondences between component elements of the type *mother* and the type *person* are simply specified by the same logical variables.

Traditionally, in a data integration task the second significant problem consists of mappings and query reformulation. It is worth stressing that mapping between schemas of data can be perceived similarly, as infomorphism, since treated operationally, the mapping is a special kind of a query with an answer schema equal to a schema of an asking agent. A mapping M between two data sources from the figure 3 has a following definition as an infomorphism:

```
person(name=>N:String,year=>Y:Int,sex=>S:Sex,
       children=>Ch1s:list_of_ch1) :-
  child2(name=>Nc2:String,year=>Yc2:Int,
        sex=>Sc2:Sex,parents=>Pas),
  member(parent(name=>N:String,year=>Y:Int,sex=>S:Sex),Pas),
  set_of(child1(name=>Nc1:String,year=>Yc1:Int,sex=>Sc1:Sex),
        (child2(name=>Nc1:String,year=>Yc1:Int,
            sex=>Sc1:Sex,parents=>Pa1s),
        member(parent(name=>N:String,year=>Y:Int,
              sex=>S:Sex),Pa1s)), Ch1s).
```

Any person found in the data source *source2* has to be a parent of a child (*child2*), and all such children are taken together to constitute a set of the person's offspring. The built-in predicate *set_of* is used in the definition with the conjunctive condition for children with the same parent.

Infomorphism is then a general and helpful model, which can be used to represent queries as well as mappings in a task of data integration. Also, a query reformulation task can be defined by an appropriate composition of infomorphisms, namely those modelling mappings and queries. Let **Mother** be a classification of answer data for the query Query, **Person** and **Child2** – the classifications of data from the *source1* and the *source2*, respectively, and M – the infomorphism between them.To obtain a reformulated query about mothers, this time directed to the data source *source2*, one has to compose infomorphisms Query and M into the new one from classification **Mother** to classification **Child2**. Formally, the composition of infomorphisms f and g is the infomorphism defined by $(gf)^{\wedge} = g^{\wedge}f^{\wedge}$ and $(gf)^{\vee} = f^{\vee}g^{\vee}$. But in Prolog-like environments the composition of infomorphisms may be performed by unfolding techniques. We partially evaluate the request Query using the interpretation of **Person** in the **Child2** classification (the infomorphism M). The simple unfolding is presented below:

```
mother(name=>N:String, year=>Y:Int,
       sons=>Ss:list_of_so,daughters=>Ds:list_of_da) :-
  child2(name=>Nc2:String,year=>Yc2:Int,
         sex=>Sc2:Sex,parents=>Pas),
  member(parent(name=>N:String,year=>Y:Int,sex=>f:Sex),Pas),
  set_of(child1(name=>Nc1:String,year=>Yc1:Int,sex=>Sc1:Sex),
         (child2(name=>Nc1:String,year=>Yc1:Int,
                 sex=>Sc1:Sex,parents=>Pa1s),
         member(parent(name=>N:String,year=>Y:Int,
                       sex=>f:Sex),Pa1s)), Ch1s),
  set_of(son(name=>Nc3:String,year=>Yc3:Int),
         member(child1(name=>Nc3:String,year=>Yc3:Int,
                sex=>m:Sex),Ch1s), Ss),
  set_of(daughter(name=>Nc4:String, year=>Yc4:Int),
         member(child1(name=>Nc4:String,year=>Yc4:Int,
                sex=>f:Sex),Ch1s), Ds).
```

Because we look for the infomorphism between classifications **Mother** and **Child2**, an appropriate unfolding of the set (list) *Ch1s* have to be performed. Te result takes a following form:

```
mother(name=>N:String,year=>Y:Int,
       sons=>Ss:list_of_so,daughters=>Ds:list_of_da) :-
  child2(name=>Nc2:String,year=>Yc2:Int,
         sex=>Sc2:Sex,parents=>Pas),
  member(parent(name=>N:String,year=>Y:Int,sex=>f:Sex), Pas),
  set_of(son(name=>Nc1:String,year=>Yc1:Int),
     (child2(name=>Nc1:String,year=>Yc1:Int,
             sex=>m:Sex, parents=>Pa1s),
     member(parent(name=>N:String,year=>Y:Int,
            sex=>f:Sex),Pa1s)), Ss),
  set_of(daughter(name=>Nc3:String,year=>Yc3:Int),
     (child2(name=>Nc3:String,year=>Yc3:Int,
             sex=>f:Sex,parents=>Pa1s),
     member(parent(name=>N:String,year=>Y:Int,
            sex=>f:Sex), Pa1s)), Ds).
```

Any mother found in the data source *source2* (with classification **Child2**) has to be a female parent of a child (*child2*), and all such children are devided into two subsets of sons and daughters, respectively. The built-in predicate *set_of* is used twice with the condition for boys and girls of the same female parent.

5 Conclusions

In this paper we consider the semantic data integration problem in an information system consisting of a community of peer-to-peer cooperative agents. We find it justifiable and profitable to analyze the problem as a problem of information flow. The channel theory, and particularly the notions of classification and infomorphism, help us to clarify concepts of schema, schema mapping, query, query evaluation and query reformulation.

In the process of cooperative query evaluation a local query (submitted to the local agent) has to be reformulated into a set of new queries that are addressed to agent's partners. Query rewriting is an important part of data integration tasks in peer-to-peer systems. It depends on the original query directed to an agent, on the agent's partners and on the mappings between agent's local schema and the schemas of the partners. A reformulated query is modelled by composition of the original query and the appriориate mapping. As both of them are infomorphisms, and composition of infomorphisms is also infomorphism, it is possible to model the task of data integration only with a concept of infomorphism.

In this work we also indicate possibility of building a bridge between well founded theory and efficient implementation of infomorphism model. Logic programming systems, particularly those with ψ-terms, can be used to represent complex classifications and infomorphisms between them. Such systems have nice properties originated in logical variables, reasoning tools over partial data and metaprogramming facilities. Particularly, a partial evaluation technique may be applied in order to define composition of infomorphisms. An analogous approach, but limited only to two data sources with ontological and relational knowledge, has been successfully implemented in a new version of QuOnto system [20] for ontology-based access to relational databases. We are working on fully functional XML data integration system with many heterogeneous data sources and sophisticated algorithms of data merging.

The other interesting interpretation of ψ-terms is also possible. In [1] the data models are defined as constraint systems and can be used together with rule-based technology in order to provide an operational base for semantic-oriented computing. Dedicated constraint-solving algorithms may be more efficient then the all-purpose logic reasoning engine.

References

1. Ait-Kaci, H.: Data Model as Constraint Systems: A Key to the Semanic Web, ILOG Inc., Draft (2007)
2. Ait-Kaci, H., Dumant, B., Meyer, R., Podelski, A., Van Roy, P.: The Wild LIFE Handbook, Paris Research Laboratory (1994)
3. Arenas, M., Libkin, L.: XML Data Exchange: Consistency and Query Answering. In: PODS Conference, pp. 13–24 (2005)
4. Barwise, J., Seligman, J.: Information flow. The logic of Distributed Systems. Cambridge University Press, Cambridge (1997)
5. Bonifati, A., Chrysanthis, P., Ouksel, A., Satter, K.-U.: Distributed Databases and Peer-to-Peer Databases: Past and Present. SIGMOD Record 37(1) (2008)

6. Bouquet, P., Serafini, L., Zanobini, S.: Peer-to-peer semantic coordination. Journal of Web Semantics 2(1), 81–87 (2004)
7. Brzykcy, G.: Data Integration In a System with Agents Models. In: Nguyen, N., Jo, G., Jain, L. (eds.) KES-AMSTA 2008. LNCS(LNAI), vol. 4953, pp. 162–171. Springer, Heidelberg (2008)
8. Brzykcy, G., Bartoszek, J.: Context in rules used in P2P semantic data integration system. In: Marchiori, M., Pan, J., de Sainte Marie, C. (eds.) RR 2007. LNCS, vol. 4524, pp. 377–380. Springer, Heidelberg (2007)
9. Brzykcy, G., Bartoszek, J.: Representation of the Partial Knowledge in an Agent System. In: Kubiak, B., Korowicki, A. (eds.) Information Management, pp. 39–46. Gdańsk (2007)
10. Brzykcy, G., Bartoszek, J., Pankowski, T.: Schema Mappings and Agents Actions in P2P Data Integration System. J. of Universal Computer Science 14(7), 1048–1060 (2008)
11. Calvanese, D., Giacomo, G., Lenzerini, M., Rosati, R.: Logical Foundations of Peer-To-Peer Data Integration. In: Proc. of the 23rd ACM SIGMOD Symposium on Principles of Database Systems, PODS 2004, pp. 241–251 (2004)
12. Crespo, A., Garcia-Molina, H.: Semantic overlay networks for P2P systems. In: Moro, G., Bergamaschi, S., Aberer, K. (eds.) AP2PC 2004. LNCS, vol. 3601, pp. 1–13. Springer, Heidelberg (2005)
13. Cybulka, J., Meissner, A., Pankowski, T.: Semantics-driven XML Data Exchange within Web-serviced Business Applications. In: Abramowicz, W., Mayr, H. (eds.) Technologies for Business Imformation Systems, pp. 299–311. Springer, Heidelberg (2007)
14. Fagin, R., Kolaitis, P., Popa, L.: Data exchange: getting to the core. ACM Trans. Database Syst. 30(1), 174–210 (2005)
15. Jung, J.: Query transformation based on semantic centrality in semantic social network. J. of Universal Computer Science 14(7), 1031–1047 (2008)
16. Nguyen, N.T.: Advanced Methods for Inconsistent Knowledge Management. Springer, London (2008)
17. Pankowski, T.: Management of executable schema mappings for XML data exchange. In: Grust, T., Höpfner, H., Illarramendi, A., Jablonski, S., Mesiti, M., Müller, S., Patranjan, P.-L., Sattler, K.-U., Spiliopoulou, M., Wijsen, J. (eds.) EDBT 2006. LNCS, vol. 4254, pp. 264–277. Springer, Heidelberg (2006)
18. Pankowski, T.: XML data integration in SixP2P – a theoretical framework. ACM Data Management in P2P Systems, 11–18 (2008)
19. Pankowski, T., Cybulka, J., Meissner, A.: XML Schema Mappings in the Presence of Key Constraints and Value Dependencies. In: ICDT 2007 Workshop EROW 2007, CEUR Workshop Proceedings. CEUR-WS.org, vol. 229, pp. 1–15 (2007)
20. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking Ontologies to Data. J. on Data Semantics X, 133–173 (2008)
21. Wooldridge, M.: An Introduction to Multiagent Systems. John Wiley & Sons, Chichester (2002)
22. Yu, C., Popa, L.: Constraint-Based XML Query Rewriting for Data Integration. In: SIGMOD Conference, pp. 371–382 (2004)

Adaptive Community Identification on Semantic Social Networks with Contextual Synchronization: An Empirical Study

Jason J. Jung

Department of Computer Engineering
Yeungnam University
Dae-Dong, Gyeongsan, Korea 712-749
j2jung@intelligent.pe.kr

Abstract. To support prompt collaborations, an ontology-based social network platform has been proposed to find the most relevant users by context representation (i.e., personal and group contexts) and matching. Consequently, groups can be dynamically organized with respect to the similarities among the personal contexts by context synchronization. Individual users can engage in complex collaborations related to multiple semantics. In this paper, we want to show and discuss the experimental results collected from a collaborative information searching system with context synchronization. Main empirical issues are i) setting thresholds, ii) searching performance, and iii) scalability testing.

Keywords: Empirical study, Social context, Synchronization.

1 Introduction

A semantic social network is an information space to easily share knowledge between like-minded people. It is built not only by aggregating relationships between people (e.g., co-authoring, affiliation, and so on), but also personal ontologies used by the people [1]. Therefore, topological distance between persons as well as similarities between their personal ontologies can be assumed to indicate how they are contextually relevant. It means that we can discover clusters (referred as a *collaborative* network, CN) for justifying the possibility of ranking search results [2] and cooperating on their tasks and sharing their knowledge [3]. In a case work, we focus on a collaborative system where people through a peer-to-peer network can collaboratively search for relevant information on the web [4,5]. While a user is searching for a certain information, the proposed system tries to be aware of his context (i.e., which topic he is looking for), and to find out who is in a similar context.

We have mentioned two main problems on such contextual collaborations between people in the same CN. First problem is *multidisciplinary* context. Interactions within a CN are limited to efficiently accomplish complicated tasks, i.e., intermixed with multiple contexts. It implies that a multiple context is divided into a set of sub-contexts, so that each sub-context can be compared with people in other CNs [6]. Second problem is *temporal transition*. Up-to-date context of their working state can dynamically change

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 430–439, 2009.

to another over time. In order to deal with these problems, we have investigated semantic representations of i) personal context from his personal ontologies, and ii) group context by integrating a set of user contexts involved in the same group. Efficient coordination was supported to enable collaborators' contexts to be shared [7] and integrated with the so-called organization context [8]. More importantly, at the moment a context transition is recognized, the corresponding user is automatically shifted into more relevant CNs.Thereby, we propose a social mediation based approach to support ad-hoc collaborations between people in different networks by context mapping methods.

2 Contextual Synchronization

Contextual synchronization [3] consists of three steps; i) integrating personal contexts into a group context, ii) detecting contextual transitions, and iii) re-organizing CN's. Firstly, we define a personal context from the personal ontology and working resources.

Definition 1 (Context). *A context $ctx_k^{(t)}$ of user u_k obtained by a mapping function $\mathcal{M}$ at time t is given by*

$$ctx_k^{(t)} = \{c_i | c_i \in \mathcal{M}(\mathbb{O}_k^{\mathcal{P}}, res^{(t)})\} \quad (1)$$

where $res^{(t)}$ is a resource working at the moment.

Given a pair of concepts from two contexts $ctx_i^{(t)}$ and $ctx_j^{(t)}$, the similarity $Sim_{\mathcal{C}}$ is assigned in $[0, 1]$ by context matching finding a maximal matching maximizing the summed similarity between the concepts

Definition 2 (Collaborative network). *Collaborative network CN_i consists of*

$$CN_i = \langle \mathcal{U}_i, \mathcal{V}_i \rangle \quad (2)$$

where $\mathcal{U}_i$ is a set of users and $\mathcal{V}_i \subseteq |\mathcal{U}_i| \times |\mathcal{U}_i|$. A link $v_{\alpha\beta} \in \mathcal{V}_i$ between two users u_α and u_β is attached with $MSim_{\mathcal{C}}(ctx_\alpha, ctx_\beta)$. If we can assume that all of them are contextually cohesive (this testing process is explained subsequently), the group context of CN_i is represented as $ctx_{CN_i}^{\top} = \bigcup_{k=1}^{|\mathcal{U}_i|} ctx_{u_k}$ where $u_k \in CN_i$.

Consequently, we can easily calculate the centrality $CTR(u_k)$ in collaborative network CN_i.Then, the most centralized user (denoted as $u_{CN_i}^{\diamond}$) whose CTR is a maximum is regarded as capable of playing the role of a contextual representative of the corresponding CN.

2.1 Detecting Context Transition

Context transition is based on comparison between a context $ctx^{(t)}$ and previous ones, e.g., $ctx^{(t-1)}$. If the difference is over a threshold τ_{CTX}, we assume that the corresponding user (or users) are researching on distinct resources. For instance, the function for user u_k may be formulated by testing the following step;

$$Sim_{CTX}(ctx_{u_k}^{(t)}, ctx_{u_k}^{(t-1)}) \leq \tau_{CTX}. \quad (3)$$

Instead of the contexts of every individual user, the group contexts of CNs are applied to enable greater efficiency in testing calculation.

More importantly, three semantic factors (e.g., semantic distance matrix $\Delta^\diamond$, semantic distance mean $\mu^\diamond$, and semantic distance deviation $\sigma^\diamond$) are defined to measure temporal patterns indicating relationships between contexts of users in a CN, and group contexts [3]. We mainly focus on a sequence of contexts $[ctx^{(t-T+1)}, ctx^{(t)}]$ by using the sliding windows method, where T is the size of the time interval.

By repeating semantic factor computation in a given time interval over time, statistical distribution of contextual transitions can be established. Based on temporal dynamics of semantic factors, we identify semantically significant transition moments of the contexts during a certain task. Hence, particular special triggering patterns from these signals are regarded as important evidence for contextual synchronization.

For practical computation, activities of users in CN_i are aggregated during a given time interval T, and are represented as a set of concepts from personal ontologies in a form of matrix $\mathcal{W}(CN_i)$. From this, we obtain a sequence of concept sets which are aggregated by both contextual dynamics of; i) a particular user's activities (ctx_{u_k} is indicated by k-th row components in $\mathcal{W}(CN_i)$) and ii) a group context $ctx^{\top}_{CN_i}$ obtained from merging the column components at each moment.

Two-step procedure (i.e., alerting and confirming) for detecting contextual transitions has been implemented [3]. We employ the semantic distance deviation $\sigma^\diamond$ to establish the dispersion of members in CN_i, rather than the group context itself. Subsequently, if some users are detected in this step, the confirming step can establish if their transitions are validated or not, because the semantic distance mean $\mu^\diamond$ is useful for measuring semantic cohesion within a time interval.

2.2 Re-organization

Confirmed users $u_i \in u_{Confirm}$ must be provided with the CN for which the group context is more relevant to ctx_{u_i}. The re-organization process must be conducted by the following objective function

$$\max\left(\sum_{CN_i, CN_j \in \Omega} 1 - Sim_{CTX}(ctx^{\top(t_p+1)}_{CN_i}, ctx^{\top(t_p+1)}_{CN_j})\right) \quad (4)$$

where Ω is a set of all communities from CNs which people can access. This implies that the summation of semantic distances between group contexts of two CNs must be maximized. As a result, topology patterns of these CNs are dynamic. Here, the decomposability property of the context [3] is considered. We want to discover which CNs are most relevant to sub-contexts $ctx^{k\perp}_{u_i} \sqsubseteq ctx_{u_i}$ by using

$$\max \sum_{k=1}^{K} Sim_{CTX}(ctx^{k\perp}_{u_i}, ctx_{CN_j}) \quad (5)$$

where K is the total number of sub-contexts, based on the number of concepts shared with the group contexts of CNs. This implies that user u_i is contextually associated with K CNs. Moreover, as many users are commonly participating in different CNs, the CNs can eventually be merged into a single CN.

3 Empirical Studies

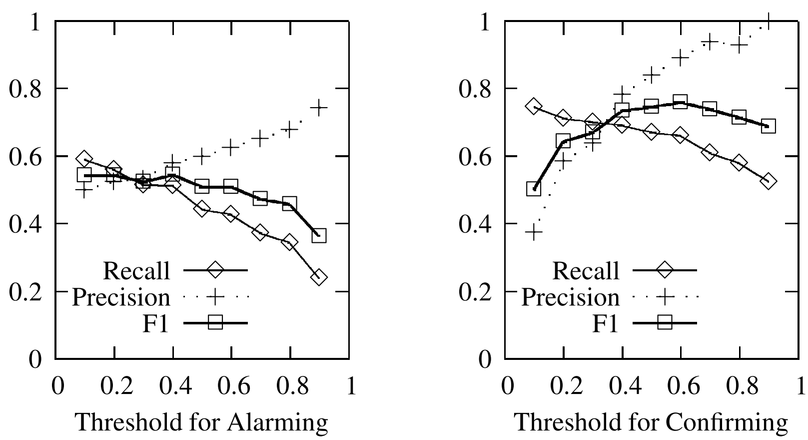
For evaluating the proposed approaches, we have extended the collaborative web browsing system [9] on a peer-to-peer platform. The system specification is mainly separated into two parts; i) peer modules (e.g., user interface module) including a personal ontology editor, and ii) super-peer modules (e.g., peer manager).

Three groups (ten users in each group), G_A (simple co-browsing), G_B (Bayesian synchronization), and G_C (ontology-based synchronization) were organized, and they were asked to build their own personal ontologies with OWL. In addition, these ontologies were enriched by annotating a given set of images[1]. Then, we collected the 30 log sequences by enabling the users to browse the testing bed with their own fixed contexts. After cleansing the collected dataset by the preprocessing scheme proposed in [10], we have prepared the testing dataset, which is composed of 5622 web pages classified according to 28 categories from ODP[2].

3.1 Setting Thresholds by Detecting Contextual Transitions

First issue is to find the best threshold values λ_{Alert} and $\lambda_{Confirm}$. Thereby, we have computed precision of detecting contextual transitions, as changing the threshold values. Due to difficulties on manual annotation,we generated 100 synthesized sequences with 794 contextual transitions by intermixing fragments randomly segmented from the 30 log sequences (i.e., fragments from different log sequences are contextually distinct). We examined the degree to which the transitions in three different groups could be detected with respect to *precision*, *recall*, and $F1$-value calculated by $\frac{2 \times recall \times precision}{recall + precision}$.

As shown in following Fig. 3.1, when $\lambda_{Alert} = 0.4$, we obtained an average $F1 = 0.52$, and the best performance is $F1 = 0.544$. Average performance of the confirming step was $F1 = 0.72$, and when $\lambda_{Confirm} = 0.6$, the best results were exhibited ($F1 = 0.759$). We empirically established the best threshold values $\lambda_{Alert} = 0.4$ and $\lambda_{Confirm} = 0.6$.



[1] This collection is available in http://intelligent.pe.kr/AnnotGrid/

[2] Open Directory Project, http://www.dmoz.org/

3.2 Performance on Dynamic Collaborations

Second issue is to justify whether adaptive community identification has increased performance on social collaborations. We measured the efficiency of online collaboration during browsing, compared with individual browsing or basic co-browsing systems without contextual synchronization. While users in G_A browse with simple collaboration, G_B and G_C perform online collaborations by detecting context transitions dynamically. In fact, users in G_B are supported with recommendation scheme based on Bayesian influence propagation [5]. G_C were the only users provided with the proposed context mapping algorithm. We monitored the performance of information searching tasks in each group over time, by comparing the topics derived from the retrieved information with the topics selected prior to experiments.

As shown in Fig. 1, G_C (contextualized synchronization) has exhibited the best *recall* results (in 9^{th} day, approximately four times higher than G_A, and in 7^{th} day, 79% higher than G_B). This implies that our contextualized synchronization can support

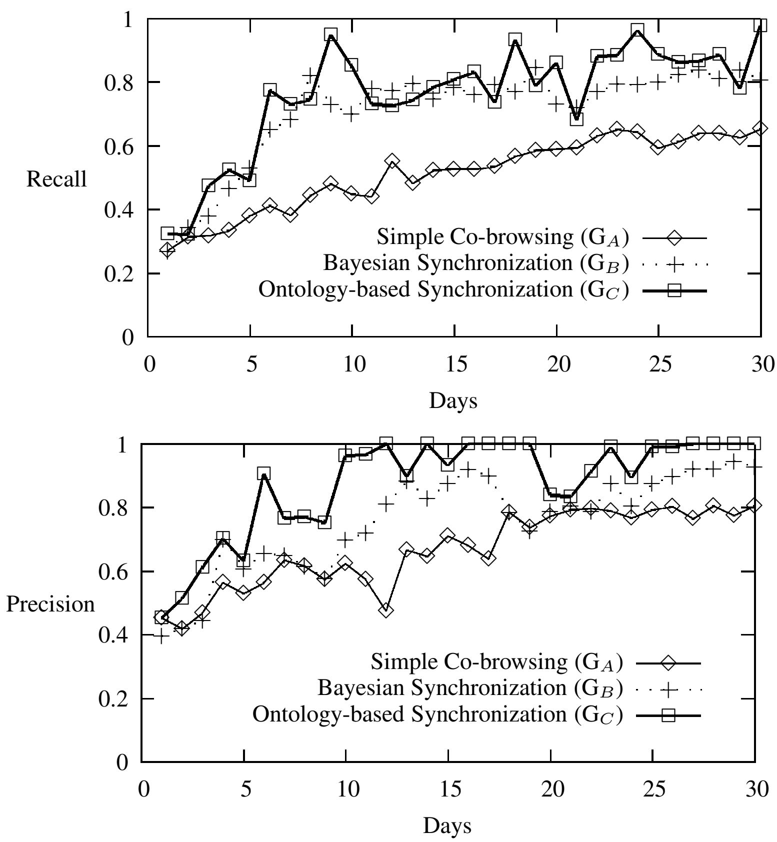


Fig. 1. Performance of information searching from three CNs with *recall* and *precision*

Table 1. Overall performance of contextual collaborations

Communications	G_A	G_B	G_C
Web accesses	6232	3662	3285
Ratio	-	58.76%	52.71%
Group communications	-	13441	12325
Ratio	-	-	91.70%

users, particularly during the early stage. With respect to *precision*, we discovered that collaborative systems finally showed convergence in 80% precision level, even though, during the initial stage, individual browsing exhibited the best performance.

Overall, Table 1 shows the final results of three group members' browsing over four weeks. G_C exhibited only 53% web access with cooperation according to the context. Compared with G_B, the proposed method has slightly improved by 11.5%. Moreover, with respect to the group communications (i.e., the number of interactions within a same CN), compared with G_B, only 91.7% of communication has been occurred in G_C. It means that G_C using adaptive community identification based on contextual synchronization has saved communication resources with less interactions between peers.

3.3 Scalability Testing

Finally, we have tested scalability of the proposed community identification, as expanding the number of peoples (e.g., 20, 40, and 100) and changing the number of CNs (e.g., 3, 5, and 10). Table 2 show the days taken until the evaluation measure (F1-value) reaches 80% level.

We found out that the proposed community identification works well, independently from the large number of users. Moreover, community density (i.e., average number of users in a community) has shown the correlated pattern with the performance. The communities of which density is about 6-10 outperformed.

Table 2. Scalability testing on adaptive community identification

Number of Users	Number of CNs (Community density)	80% Convergence days
20	3 (6.67)	**15.2**
	5 (4.0)	17.5
	10 (2.0)	19.4
40	3 (13.3)	15.3
	5 (8.0)	**13.4**
	10 (4.0)	18.2
100	3 (33.3)	17.7
	5 (20.0)	15.2
	10 (10.0)	**12.4**

4 Discussion and Related Work

Here, we want to discuss several significant issues that we have realized during providing social collaborations between people by considering contextual transitions.

- First issue is how to represent personal contexts by incrementally aggregating semantics from user activities.
- Secondly, we need to compare a single global ontology-based platform with a multiple ontology-based platform.
- Thirdly, more importantly, we need to discuss the dynamics of personal and group contexts on super-peer network.

4.1 Ontology-Based Context Representation

In this work, for representing personal contexts, user actions (i.e., searching patterns) are manually annotated by extracting semantics from their ontologies. At the same time, ontology enrichment can be conducted if users put additional semantics in the ontologies during annotating.

There is a real-world limitation on initial ontologies. Even though there may be a certain way to automate this process, initial ontology should be anyhow manually prepared by either ontology editors or annotation tools. It is similar to the so-called *cold-starting* problem in user modeling-based recommendation systems. We also need to consider annotation patterns taken by users, i.e., how users annotate resources. In other words, we have to find out how the personal ontologies are built. By analyzing the personal ontologies collected during our experimentation, we have realized that there are several types of ontologies. (We want to skip more discussion on this.)

One of the types is the ratio between number of classes and number of properties in an ontology. This ratio possibly indicates how formally the semantics are represented. Through our experimentation, the higher ratio has shown higher precision of ontology matching. Hence, as shown in Table 3, personal ontologies of users (1, 4, 5) has better satisfaction of the recommendations.

Table 3. Information about personal ontologies of G_C

	Class/Property (= Ratio)	Annotated instances	Number of ann. classes (Ratio)
1	64/7 = **9.14**	613	56 (10.95)
2	54/5 = 10.8	442	32 (13.81)
3	75/6 = 12.5	985	64 (15.39)
4	46/7 = **6.57**	550	37 (14.86)
5	47/8 = **5.88**	702	29 (**24.21**)
6	48/4 = 12	537	41 (13.1)
7	49/6 = **8.17**	646	36 (**17.94**)
8	64/4 = 16	632	54 (11.7)
9	51/2 = 25.5	695	37 (**18.78**)
10	52/3 = 17.33	574	43 (13.35)

4.2 Comparison between Single and Multiple Ontologies

In this work, we employ a single centralized ontology, i.e., web directory. This enables the automatic resolution of the semantic heterogeneity problem, because every topic is annotated (or labeled) by referring to the single ontology.

However, we must consider a platform which provides multiple ontologies. Each information source (or system) can build its own ontology. These types of ontologies might be domain-specific and cause semantic heterogeneity problems. In our case, the users' agents can be embedded with their personal interests. Also, they might edit their own personal ontologies. Thereby, ontology alignment (mapping) methods were proposed. Recently, Shvaiko and Euzenat explained the classification method of ontology alignment (matching) [11] and ontology mapping algorithms [12]. Several alignment methodologies were introduced. Since Dieng and Hug proposed an algorithm for matching conceptual graphs using terminological linguistic techniques and comparing superclasses and subclasses [13], Euzenat developed *T-tree* to infer the dependencies between classes (bridges) of different ontologies sharing the same set of instances based only on the "extensions of classes" [14]. Additionally, *FCA-merge* uses formal concept analysis techniques to merge two ontologies sharing the same set of instances while properties of classes are ignored [15]. Meanwhile, *Cupid* is a first approach combining many the other techniques. It aligns acyclic structures taking terminology and data types (internal structure) into account and giving more importance to leaves [16]. Especially, we studied the ontological mediator framework for sharing semantic information between personal crawlers [17].

4.3 Dynamics of Personal Contexts and Group Context

We assumed that the context can be changed during web browsing. In most collaboration systems, the context is coupled and related; i) between people, ii) between groups, and iii) between a person and a group. For supporting online collaborations between users, we realized that deriving and comparing dynamics of contexts is more important than representing and comparing the contexts themselves.

There were two well-known P2P networks; Napster[3] and Gnutella[4]. Especially, we employ a super-peer network scheme similar to Napster. A pure P2P network is a "degenerate" super-peer network where cluster size is one. While this implies that every node is a super-peer with no clients, super-peer networks such as KaZaA[5] use heterogeneity of peers to their advantage. Also, regarding the computational complexity, the overhead of maintaining an index in the super-peer networks is small by comparison to the savings in query cost this centralized index enables [18]. Furthermore, Xiao et al. proposed a dynamic super-peer network based on a dynamic layer management [19].

5 Conclusion and Future Work

In this study, we have investigated contextualized synchronization for online social collaboration system with spatially remote and temporal synchronous characteristics.

[3] Napster. http://www.napster.com/

[4] Gnutella. http://www.gnutella.com/

[5] KaZaA. http://www.kazaa.com/

Contextual transitions of users in a group have been detected, so that they are efficiently shifted into the relevant group communications. As the main contribution of this paper, most importantly, we claim that tracking the contextual dynamics of groups is better to online collaborations than modeling the group context of the groups in a certain moment.

However, there are many problems remaining with this system that must be dealt with by future work. We modified *Levenshtein* edit distance [20] to measure hierarchical path-labeled web pages. In order to support more general types of users, we obviously consider various semantic annotation methods [5] to compare their relationships. Another issue is the topology of the P2P network. As mentioned in [2], a hyperlinked environment has various topological features such as authorities and hubs, thus we must consider the selection process of super-peers. Finally, more globally, by using the grid computing paradigm [21], we also consider the social grid environment, providing k-redundant super-peer networks [18].

Acknowledgement

This work was supported by the Korean Science and Engineering Foundations (KOSEF) grant funded by the Korean government (MEST). (No. R01-2008-000-10196-0).

References

1. Jung, J.J., Euzenat, J.: Towards semantic social networks. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 267–280. Springer, Heidelberg (2007)
2. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM 46(5), 604–632 (1999)
3. Jung, J.J.: Ontology-based context synchronization for ad-hoc social collaborations. Knowledge-Based Systems 21(7), 573–580 (2008)
4. Hassler, V.: Online collaboration products. IEEE Computer 37(11), 106–109 (2004)
5. Jung, J.J.: Collaborative web browsing based on semantic extraction of user interests with bookmarks. Journal of Universal Computer Science 11(2), 213–228 (2005)
6. Terveen, L., McDonald, D.W.: Social matching: A framework and research agenda. ACM Transactions on Computer-Human Interaction 12(3), 401–434 (2005)
7. Gross, T., Prinz, W.: Modelling shared contexts in cooperative environments: Concept, implementation, and evaluation. Computer Supported Cooperative Work 13(3), 283–303 (2004)
8. Simone, C., Divitini, M.: Integrating contexts to support coordination: The chaos project. Computer Supported Cooperative Work 8(3), 239–283 (1999)
9. Jung, J.J.: Semantic co-browsing system based on contextual synchronization on peer-to-peer environment. Computing and Informatics 26(5), 465–467 (2007)
10. Jung, J.J.: Semantic preprocessing of web request streams for web usage mining. Journal of Universal Computer Science 11(8), 1383–1396 (2005)
11. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. Knowledge Engineering Review 18(1), 1–31 (2003)
12. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. In: Spaccapietra, S. (ed.) Journal on Data Semantics IV. LNCS, vol. 3730, pp. 146–171. Springer, Heidelberg (2005)

13. Dieng, R., Hug, S.: Comparison of "personal ontologies" represented through conceptual graphs. In: Proceedings of the 13th European Conference on Artificial Intelligence, pp. 341–345 (1998)
14. Euzenat, J.: Brief overview of t-tree: the tropes taxonomy building tool. In: Proceedings of the 4th ASIS SIG/CR workshop on classification research, Columbus, USA, pp. 69–87 (1994)
15. Stumme, G., Maedche, A.: FCA-merge: Bottom-up merging of ontologies. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001), Seattle, USA, pp. 225–230 (2001)
16. Madhavan, J., Bernstein, P., Rahm, E.: Generic schema matching using Cupid. In: Proceedings of the 27th International Conference on Very Large Data Bases (VLDB 2001), Roma, Italy, pp. 48–58 (2001)
17. Jung, J.J.: Ontological framework based on contextual mediation for collaborative information retrieval. Information Retrieval 10(1), 85–109 (2007)
18. Yang, B., Garcia-Molina, H.: Designing a super-peer network. In: Dayal, U., Ramamritham, K., Vijayaraman, T.M. (eds.) Proceedings of the 19th International Conference on Data Engineering (ICDE 2003), pp. 49–60. IEEE Computer Society, Los Alamitos (2003)
19. Xiao, L., Zhuang, Z., Liu, Y.: Dynamic layer management in superpeer architectures. IEEE Transactions on Parallel and Distributed Systems 16(11), 1078–1091 (2005)
20. Levenshtein, I.: Binary codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory 10(8), 707–710 (1996)
21. Zhuge, H.: Semantics, resource and grid. Future Generation Computer Systems 21(1), 1–5 (2004)

Self-organizing Fuzzy Logic Steering Algorithm in Complex Air Condition System

Michal Lower

Wrocław University of Technology,
Institute of Computer Engineering, Control Robotics
Wybrzeze S. Wyspianskiego 27, 50-370 Wrocław, Poland
michal.lower@pwr.wroc.pl

Abstract. This paper proposes considering the problem of automatic controlling of heating and air-condition system in conference halls with changeable structure and shape. Modern conference halls are often designed in such a way that there are possibilities of making many different arrangements within the limits of one expanse. The great hall can be divided into e.g. three smaller rooms. It is obvious that each possible configuration should require different algorithm of heating and air condition systems controlling.

Keywords: intelligent house, control systems, fuzzy logic.

1 Introduction

The concept of „intelligent house" becomes more and more popular and it is used in the context of modern office buildings, public utilities and detached houses [1],[2]. It means that buildings are equipped in their own control system. The system responds to stimuli from inside and outside of the building, controlling devices of intelligent house in a programmed way. It could help people in doing this and it could minimize the cost of functioning of the house. Generally the house automatic system controls the second-rate systems as ventilation, heating, lighting, audio-visual systems and others.

In the paper the problem of automatic controlling of heating and air condition system in conference halls with changeable structure and shape is considered [3],[4]. Modern conference halls are often designed in such a way that there are possibilities of making many different arrangements within the limits of one expanse. The great hall can be divided into e.g. three smaller rooms. It is obvious that each possible configuration should require different algorithm of heating and air condition systems controlling.

The problem can be presented on the example of one conference hall which can be divided into two or three smaller rooms (Fig. 1 shows an example).

In order to keep the proper comfort in each room separated from one big surface, the heating and air condition systems should be equipped in independent controlling systems. Such solution starts to be questionable when independent rooms become one

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 440–449, 2009.

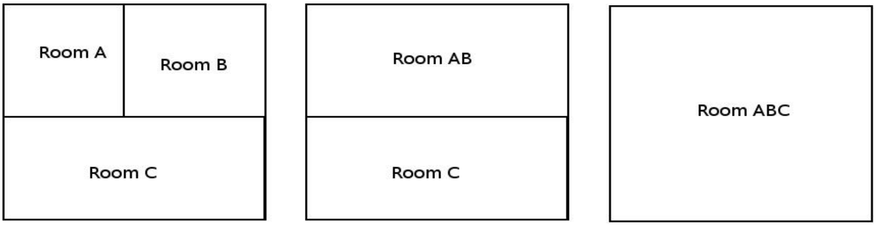


Fig. 1. The examples of conference hall configurations

common hall and when the control algorithm is not changed or auto-adapted, then unwelcome phenomena can appear that are connected with mutual interference and inducing of independent control systems working with the same object. In special conditions unstable oscillation can also appear. In the case of air conditioning the controlling systems could stabilize in extremely unoptimal way. At the same time one system will be cooling the room and the other will be heating.

Changing different controlling algorithms depending on hall configurations is certainly possible in the hand mode but it could be inconvenient for the service. It demands reliability and the trained staff. The idea of intelligent house is aimed on automatic solving of such problem. In order to do it the system of fuzzy diagnosis is designed on the basis of efficiency parameters assessment of system. The diagnosis of actual room configuration could allow for changing the control system automatically.

One can find examples for using fuzzy logic algorithms in literature [5],[6],[7],[8].

2 The Temperature Control in the Conference Hall

Control systems in rooms usually do not need too complicated control algorithms. The temperature comfort provision in a room means the maintenance of the constant temperature value independently of outside factors. In most cases the PI controller is sufficient for such constant value control [9],[10],[11],[12]. From the analytic point of view, the system complicates when coupling between rooms starts to appear.

2.1 The Hall Divided into Separated Parts

There is a diagram of control algorithm for divided rooms shown in Fig. 2. Rooms A and B have two separated control systems. There are independent sensors and air-condition devices in each room. One can set the different points in each control system (Fig. 2 shows an example).

In the system like this the feedbacks are so small that can be passed over.

2.2 The Connected Rooms

The configuration presented above could be changed and rooms A and B could be entirely united. In such situation there are two sensors and two separated air-condition systems in the room. Unless the control algorithm is changed, the block diagram of

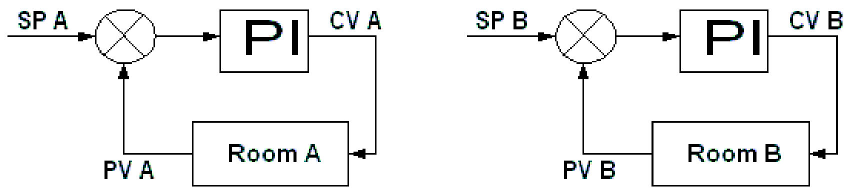


Fig. 2. The block diagram of two separated control systems for rooms A and B

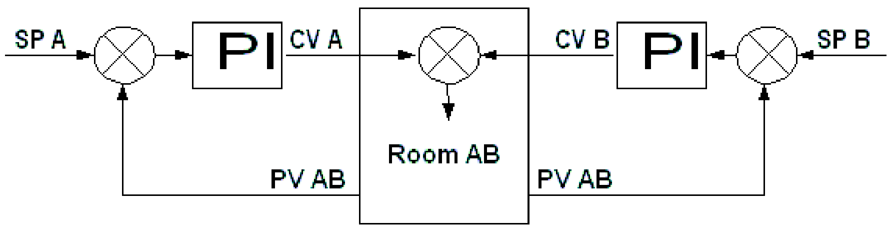


Fig. 3. The block diagram of two separated control systems for united rooms A and B

control system will look as in Fig. 3. The temperature values taken from both sensors will be similar; however, the power of air condition devices will balance each other. In real system the situation can be more complicated because indications of temperature sensors can differ one from another in the range of 1 to 2 degrees for many reasons.

The system like this brings many threats connected with mutual disturbances and with induction of control systems working with the same object. Under special circumstances unstable and undamped oscillations may appear. The control power balancing can cause stabilizing of control systems in extremely unoptimal state. In the situation when the demand for the output power is e.g. 50%, one from the two systems could set on 0% (that means only cooling the room) and the other one could set on 100% (only heating). Similar phenomenon could appear in the case when the set

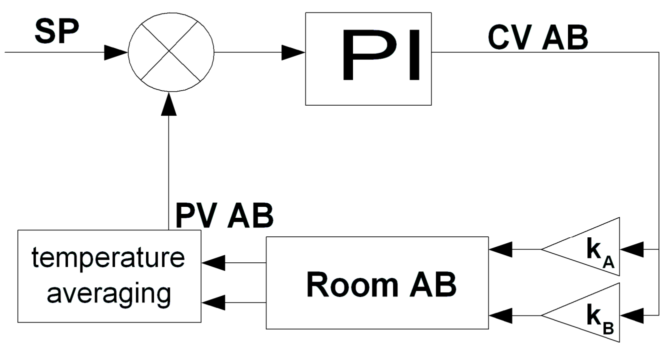


Fig. 4. The block diagram of common control system for united rooms A and B

points SP of both systems differ one from another and the temperature in the room stabilizes between set points of control systems.

To recapitulate, in the situation when rooms are united, the control algorithm should be selected according to Fig.4. Gain coefficients $\mathbf{k_A}$ and $\mathbf{k_B}$ are constant. They determine proportional heating factor division into two systems.

3 The Assumptions of the Model

The heat in a room as a function of time is expressed with the following formula [13]:

$$Q(t) = c_p \rho V T(t) \tag{1}$$

where

c_p – specific heat of the air [J/(kg K)] ,
ρ – air density [kg/m^3] ,
V – volume of heat store (the room) [m^3] ,
T – temperature inside the room [K] ,
Q – the heat giving out of the building [J].

The heat increment in the room is derivative of heat Q after time:

$$Q(t)/dt = P_d + P_o + P_z - P_{zw} = c_p \rho V T(t)/dt \tag{2}$$

where

P_d – the supplied power [W] (power of radiators),
P_o – the exporting power or the power supplied from adjoining rooms [W] (from adjoining rooms which have different temperature),
P_z – the power of interferences [W] (additional heat source, e.g. lightning, projectors, other electric devices producing considerable heat amount)
P_{zw} – the power of loss of heat [W] (the heat left out of the building).

In presented equations one could assume that parameters c_p (specific heat of the air), p (air density) adopt constant values. The values T (air temperature), P_d (the supplied power) result directly from measurement and from parameter CV (control value) as follows:

$$P_d = CV\ p \tag{3}$$

where

p – the coefficient of proportionality (p = const).

Parameter P_z (the power of interferences) is exactly connected with the specific solution and the room equipment. In general, in conference halls the devices which generate great amount of heat have recurrent nature and they are easy to identification but globally the analysis is complicated. This is why in the paper the influence of disturbances will be omitted, thus one assumes that parameter $P_z = 0$. Parameter P_{zw} (the power of loss of heat - the heat left out of the building) results from the building characteristic and is non-linear function of outside temperature and the temperature of the room.

$$P_{zw} = f(T_{zw}, T) \tag{4}$$

where

T_{zw} – the outside temperature

Assuming that the temperature of the room is changing in the small range in comparison with the outside temperature changing, one can assume that, simplifying, P_{zw} is the function of outside temperature.

$$P_{zw} = f(T_{zw}) \text{ where } T = \text{const} \tag{5}$$

Globally, on the base of equation (2) for the entire conference hall, P_{zw} is presented with the equation (6).

$$P_{zw} = P_d \text{ when } T'(t) = 0 \text{ and } P_o = 0 \tag{6}$$

where P_d is a sum of all P_{dn} of each room (n – room number).

$$P_d = P_{dA} + P_{dB} + ... + P_{dn} \tag{7}$$

and

$$P_{zw} = P_{dA} + P_{dB} + ... + P_{dn} \tag{8}$$

Parameters P_{zwn} for particular hall configurations can be determined by dividing total value P_{zw}, in proportion to the outside surface of walls.

$$P_{zwn} = k_n P_{zwn} \tag{9}$$

where

P_{zwn} – the power of loss of heat of room n.
k_n – the proportional coeficient of room n to the outside sufrace of walls.
If the entire conference is characterized by the value P_{zw} and it was divided into three smaller rooms then:
P_{zwA} – the power of loss of heat of room A,
P_{zwB} – the power of loss of heat of room B,
P_{zwC} – the power of loss of heat of room C.

$$P_{zw} = P_{dA} + P_{dB} + P_{dC} \tag{10}$$

$$P_{zwA} = k_A P_{zw}, \quad P_{zwB} = k_B P_{zw}, \quad P_{zwC} = k_C \ P_{zw}, \text{ where } k_A + k_B + k_C = 1 \text{ and}$$
$$k_A = \text{const}, k_B = \text{const}, k_C = \text{const}$$

On the base of equation (2) the parametr P_o is presented with the equation (11).

$$P_o = c_p \rho V T'(t) + P_d - P_{zw} \tag{11}$$

In such objects as a conference hall, the control is usually with constant set value and the constant temperature maintenance in the room is not a problem for PI controller, thus one could assume that T ' (t)=0 and

$$P_o = P_d - P_{zw}. \tag{12}$$

So the following equations result from the equation (12).

$$P_{o\,A} = P_{d\,A} - P_{zw\,A},\quad P_{o\,B} = P_{d\,B} - P_{zw\,B},\quad P_{o\,C} = P_{d\,C} - P_{zw\,C} \tag{13}$$

$P_{d\,A}$, $P_{d\,B}$, $P_{d\,C}$ – the power connected with control value (the power delivered to executive devices).

4 The Fuzzy Model of Recognition of System Which Adapts the Controlling Algorithm to the Hall Configuration

The fuzzy recognition model is based on the observation of control system parameters P_o. According to the chapter 3 (model assumptions), observation of the parameters should be sufficient for the correct assessment of the configuration. Keeping appropriate proportion of parameters P_d, each of possible hall arrangements should be marked by the characteristic set of parameters P_o.

The presented fuzzy model is built on basis of the Mamdani fuzzy model, the membership functions of input parameters are defined as triangular, the membership functions of output parameters are defined as constant. The P_o values, counted out on the basis of P_d values, are input parameters of fuzzy conclusion model. The recognized hall configuration is the output model parameter and it is the base of choosing the proper control algorithm. The type of room configuration is the output parameters of fuzzy module.

Making an assumption that the conference hall can be divided into three rooms, it should have three air-condition process systems and three measurement systems.

The types of room configuration:

AB, C – the parts A and B of the hall are connected, the part C is divided,
A, BC – the parts C and B of the hall are connected, the part A is divided,
AC, B –the parts A and C of the hall are connected, the part B is divided,
ABC – the entire conference hall,
A, B, C – the conference hall divided into three separated rooms.

According to the formulas (10) and (13) one can calculate proportion parameters k_A, k_B, k_C for the building and next one can calculate parameters $P_{o\,n}$ on the base of formulas (14).

$$\begin{aligned} P_{o\,A} &= P_{d\,A}(1-k_A) - k_A(P_{d\,B}+P_{d\,C}) \\ P_{o\,B} &= P_{d\,B}(1-k_B) - k_B(P_{d\,A}+P_{d\,C}) \\ P_{o\,C} &= P_{d\,C}(1-k_C) - k_C(P_{d\,B}+P_{d\,A}) \end{aligned} \tag{14}$$

4.1 Parameters Characteristic for Different Rooms Configurations

According to the former part of this chapter, one can assume that each room configuration can approximately be characterized by the specific set of parameters P_{oA}, P_{oB}, P_{oC}. Relying on output assumptions, the following rules could be observed:

(i) each room separated from others should have parameter P_o closed to 0,
(ii) the connected rooms have possibility of giving warmth to each other so (P_{oA} >0 and P_{oB} <0) or (P_{oA} <0 and P_o b>0). In the independent control system

(Fig. 4) the state (P_{oA} =0 and P_{oB} =0) could exist if SP A = SP B. In the common control system (Fig. 3) the equilibrium state (P_{oA} =0 and P_{oB} =0) exists if parameters k and p guarantee the proportional power distribution onto executive devices,

(iii) the connected rooms have possibility of giving warmth to each other so (P_{oA} >0 and P_{oB} <0) or (P_{oA} <0 and P_{oB} >0). In the independent control system (Fig.4) the situation like this will be guaranteed if SP A ≠ SP B. In the common control system (Fig. 3) such situation will be guarantee if parameters k and p are matched in the way that makes the disturbance of equilibrium possible

(iv) when rooms are united the measured temperature difference should be closed to 0, the growth of measured temperature difference in rooms shows their division.

If the case of stable and optimal working of control system for identification of room configuration during system operation the theorems *(i), (ii)* and *(iv)* can be used. For making possible the recognition of proper room configuration and using that information for other kinds of tasks, unconnected with automation, the *(i), (iii)* and additionally *(iv)* should be respected. That means that in the divided room the ka and kb should have values which cause a little unequilibrium in provided power for executive devices so that the P_o system parameters will be different than zero. Changing of this value to zero will let recognize the moment of rooms uniting and will make possible the system change for the proper algorithm.

5 The Membership Functions of Fuzzy Reasoning Model

The fuzzy reasoning model has input parameters P_o for each room and the output parameter which is a reflection of the room arrangement.

Relying on rules *(i),..., (iv)* the case being a target of analysis should be considered individually. On such base one can define membership functions for each parameter P_o and fuzzy reasoning rules.

The membership functions of parameters P_o for each room are proposed to be defined as triangular sets and the output parameter which is a reflection of the room arrangement, as constant.

For the hypothetical room adapted to division into three parts the following fuzzy sets could be defined:

FOFF (FlowOFF) –the lack of giving warmth out of the room P_o=0,

FMO (FlowMediumOut) –the room gives out medium amount of warmth P_o>0,

FHO (FlowHightOut) –the room gives out a lot of warmth $P_o >> 0$,

FMI (FlowMediumOut) – the room takes in the medium amount of warmth P_o<0,

FHI (FlowHightOut) – the room takes in a lot of warmth $P_o << 0$,

The constant sets of functions could be defined according to the former description AB_C, A_BC, AC_ B, ABC, A_B_C.

The fuzzy rules for presented example could be defined in the following way:

$P_{o\,A}$ == FMO and $P_{o\,B}$ == FMI and $P_{o\,C}$ == FOFF => AB_C

$P_{o\,A}$ == FMI and $P_{o\,B}$ == FMO and $P_{o\,C}$ == FOFF => AB_C

$P_{o\,A}$ == FMO and $P_{o\,B}$ == FOFF and $P_{o\,C}$ == FMI => AC_ B
$P_{o\,A}$ == FMI and $P_{o\,B}$ == FOFF and $P_{o\,C}$ == FMO => AC_ B
$P_{o\,A}$ == FOFF and $P_{o\,B}$ == FMO and $P_{o\,C}$ == FMI => A_BC
$P_{o\,A}$ == FOFF and $P_{o\,B}$ == FMI and $P_{o\,C}$ == FMO => A_BC
$P_{o\,A}$ == FOFF and $P_{o\,B}$ == FOFF and $P_{o\,C}$ == FOFF => A_B_C
$P_{o\,A}$ == FHI and $P_{o\,B}$ == FMO and $P_{o\,C}$ == FMO => ABC
$P_{o\,A}$ == FHO and $P_{o\,B}$ == FMI and $P_{o\,C}$ == FMI => ABC
$P_{o\,A}$ == FMO and $P_{o\,B}$ == FHI and $P_{o\,C}$ == FMO => ABC
$P_{o\,A}$ == FMI and $P_{o\,B}$ == FHO and $P_{o\,C}$ == FMI => ABC
$P_{o\,A}$ == FMI and $P_{o\,B}$ == FMI and $P_{o\,C}$ == FHO => ABC
$P_{o\,A}$ == FMO and $P_{o\,B}$ == FMO and $P_{o\,C}$ == FHI => ABC

6 Conclusions

The presented conception, according to assumptions, protects the system from negative phenomena connected with control systems disturbances (Fig. 5 and 6 shows the example). The presented algorithm can be used for identification of room configuration regardless of its role which is the care of proper control algorithm. In case of the room recognition the critical point could be a situation when rooms are uniting in one space and control systems will not cause interference to each other. The situation like this will not influence negatively the control system operation but will not allow for correct recognition of room configuration. If the additional aim is automatic recognition of rooms configuration, the input of different set points of control systems in the room divided into parts is recommended. The little difference between the set point and the expecting value should not influence the users' comfort and could cause the effective recognition of the room configuration.

The additional aspect that could be brought up in the article is recognition which takes the influence of high power devices that extra heat the room into consideration. According to the author's experience the recognition of such situation could be

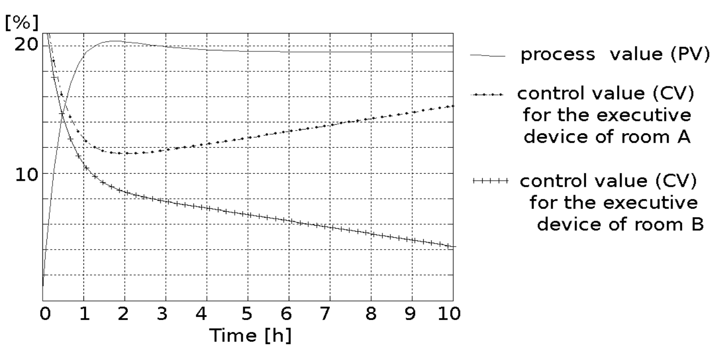


Fig. 5. The results of control system working for united rooms A and B without self-organizing fuzzy logic steering algorithm

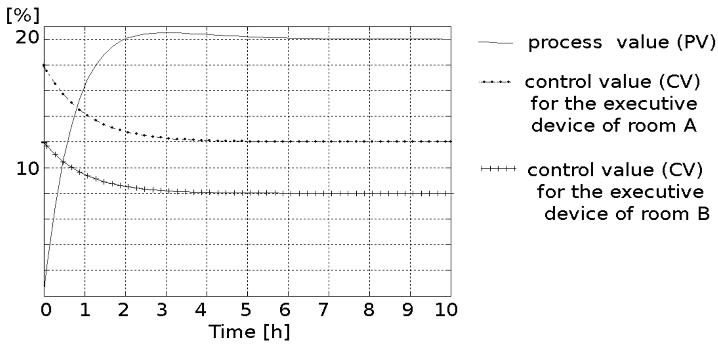


Fig. 6. The results of control system working for united rooms A and B with self-organizing fuzzy logic steering algorithm

possible but phenomena like these need the individual analysis. On the other hand, in the intelligent house turning off high power devices could be controlled directly and such information could be used for the room configuration recognition.

The presented conception was tested on the examples which included the room division for maximum three parts. The division in more than for three parts complicates the general analysis of the problem. The number of possible stances and rules is increasing several times. It is necessary to pay attention to the fact that considering the arrangement individually, the conclusion system could be simplified for the sake of the real physical rooms structure. In the considered arrangement, the division into three rooms with the possibility of joining one with another has been analysed. The system could be simplified in the case when in real structure one room is between two others in such way that the others could not be joint together. The similar simplification can be used in divisions of rooms into more than three parts.

The next aspect which should be presented is using the system in real structure. The realization of presented algorithm based on controllers dedicated especially to intelligent houses is currently rather difficult. Such controllers usually do not have the proper tools of fuzzy control algorithm. One could try to get round the lack of technical possibilities in controller by means of SCADA systems [14]. There can be traditional control algorithms which are switched by the supervision system of SCADA in the controller. That means that the fuzzy conclusion analysis can be executed using the separate program in PC computer.

For the room recognizing, the situation when the room is united and control systems do not cause interference to each other could be a critical point. Such situation will not have bad influence on the control system work but will not let recognize the room configuration correctly. If the additional target is the automatic recognition of rooms configuration one recommends introducing different set points of control systems in the room divided into parts. When the set point differs slightly from the expected value it shouldn't influence the user's comfort and simultaneously it will cause effective recognition of the room configuration.

References

1. Figueiredo, J., Costa, J.M.S.: A Two Level Hierarchical Control Strategy Applied to an Intelligent House. In: Intelligent Signal Processing, Portugal, Faro (2005)
2. Hui, X., Yunshi, X., Jiguang, Y., Cong, C., Chen, Y., Lei, J.: Swarm intelligence based design of intelligent house embedded system. In: Fifth World Congress on Intelligent Control and Automation, WCICA 2004 (2004)
3. Liu, Z., Xie, W., Huang, Y.: The energy saving control of air condition system in intelligent building. In: Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on Publication Date (2002)
4. Hwang, J.C.: Assessment of air condition load management by load survey in Taipower. IEEE Transactions on Power Systems 16(4) (November 2001)
5. Lin, L., Xue-Hai, Y., Zun-Quan, X.: University: Multicriteria fuzzy decision-making methods based on intuitionistic fuzzy setsstar, open. Journal of Computer and System Sciences 73(1) (February 2007)
6. Lower, M.: Simulation model of human individual in quiet standing based on an inverted pendulum with fuzzy controller. In: Proceedings of 2008 International Conference on Machine Learning and Cybernetics, China, Kunming (2008)
7. Król, D., Lower, M.: Fuzzy measurement of the number of persons on the basis of the photographic image. In: Nguyen, N.T., Borzemski, L., Grzech, A., Ali, M. (eds.) IEA/AIE 2008. LNCS(LNAI), vol. 5027, pp. 102–107. Springer, Heidelberg (2008)
8. Król, D., Lower, M., Szlachetko, B.: Helicopter intelligence in hover quality improvement of the fuzzy regulator. In: Sixth International Conference on Intelligent Systems Design and Applications, ISDA 2006, China, Jinan (2006)
9. Halawa, J.: Determining the settings of cascade controllers with the convergent method. Przeglad Elektrotechniczny 84(9) (2008)
10. Halawa, J.: Determining the settings of PI and PID controllers with a convergent method using computer aided design. Control and Cybernetics 34(4) (2005)
11. Halawa, J.: Parameters of controllers in cascade system versus its dynamics. Systems Analysis, Modelling, Simulation 43(1) (2003)
12. Christensen, H.: Domestic Robots. In: IEEE Mediterranean Conf. On Control and Automation, MED 2002, Portugal, Lisbon (2002)
13. Sieniutycz, S.: Thermodynamics, thermokinetics and optimal control. In: International Conference Physics and Control (2003)
14. Daunoras, J., Lukočius, R., Virbalis, J.A.: Pulse Sensor of Environ-Physiological Parameter Monitoring System. Electronics and Electrical Engineering 6(78) (2007)

Errors in Structure Self-organization: Statistical Analysis

Grzegorz Kukla[1] and Dariusz Król[2]

[1] Faculty of Computer Science and Management,
Wrocław University of Technology, Poland
Grzegorz.Kukla@pwr.wroc.pl
[2] Institute of Informatics,
Wrocław University of Technology, Poland
Dariusz.Krol@pwr.wroc.pl

Abstract. This paper contributes to the agent-based community by providing a better understanding of error propagation mechanisms in structure self-organization focusing especially on the statistical analysis of experiments. The research was verified based on the experiments conducted on simulation model. The results provide some ideas of robustness - the knowledge how to design the most error resistant architectures in complex environments. Additionally, well-known statistical tests are used to verify the hypothesis. Finally, open research issues with practical relevance are proposed.

1 Introduction

Self-organization is a process that a system returns to an organized state spontaneously after surroundings change [4,15]. Until now, many self-organization methods have been developed for communication networks in general [3] and multi-agent systems (MAS) in particular [5]. Self-organization addresses the main requirements in such networks [10,14], i.e. scalability, reliability, fault tolerance, error handling and network lifetime and opens novel solution spaces.

This paper will look at self-organization from the viewpoint of system structure [12] and error propagation phenomenon [1,9], i.e., how a system reacts to environment changes (errors) to maintain a stable and organized state. It is commonly agreed on that self-organization is necessary for the evolution of an open system, especially an agent-based system [16]. For example, if the living environment has changed, an organism (an agent system) must be able to adjust itself (through restructure) to adapt to the new environment. Otherwise, the organism will not fit to the new environment and will stop.

2 Structures and Errors in MAS

After self-organization process the structure of the system is often changed. Some of the changes are less or more useful to the efficiency of the distributed system. We should be able to take advantage of changes resulting in robustness increase of the system.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 450–459, 2009.

2.1 Structures

The way the elements of a distributed information system are organized is called a topology. The topology of the system determines the communication patterns utilized within it. It must be emphasized the topology has the influence on agents' interactions within the system. In this regard, the topology can be seen as an abstraction of complicated agents collaboration processes. It also establishes hierarchy between the agents and defines how data and knowledge is shared among them [16]. Therefore, the topology must not be confused with the physical organization of the system, such as its network structure. Further information about topologies and their metrics can be found in [7]. For example, the work [16] presents an overview of four basic multi-agent models: (1) a Web-like topology [2], (2) a Star-like topology, (3) a Grid-like topology, and (4) a Hierarchical Collective topology, assesses their advantages and disadvantages in terms of agent autonomy, adaptation, scalability, and efficiency of cooperation.

The rest of this paper will characterize five basic topologies utilized in multi-agent structures: full topology, star topology, ring topology, tree topology and random topology. In full topology all entities communicate with each other directly. Thus there are no selected entities. The full topology is characterized by the maximal level of redundancy. But the direct communication between entities gives also the maximum reliability of system's operation, because the more entities a message has to cross at its way from sender to receiver, the more probable a failure. In star topology there are two kinds of entities: one central entity and many peripheral entities. Peripheral entities connect only to the central one. Thus the central entity is connected to every other entity in the system. There is no redundancy in star topology. Therefore a failure of any of the links causes the system not to be fully connected. Nevertheless, if one peripheral entity fails, the rest of the system can operate normally. The worst case scenario is when the central entity fails. In this case the system cannot operate at all. In ring topology every entity has exactly two neighbors. Thus, the system is organized in a one cycle containing all elements. Obviously, the redundancy is present in this topology. Between every two entities there are always two possible paths. The advantage of this topology is increased reliability by a minimal cost. The disadvantage of it is the long average distance between entities. It indicates, that a message passed from one entity to another has to cross many indirect entities. Therefore a probability of a failure is significant. In tree topology there are no cycles. The redundancy is also not present as each pair of entities is connected by exactly one path. The advantage of this topology is that the communication overhead is balanced between all entities. The disadvantage is that the entities are equally important in system's operation. This indicates that failure of any entity can seriously damage the whole system, because every entity is indispensable for communication. Random topology is built in a random manner. There is no regularity or hierarchy in connections. The presence of a redundancy is therefore random. Also the distribution of connections is not even. There can be parts of the system which are highly interconnected and parts loosely interconnected. The general properties of such structures are difficult to assess.

2.2 Errors

An error is considered as an improper state of one or more elements of a distributed system. This definition does not say anything about the nature of the error or its genesis. For the needs of study of error propagation phenomenon the classification of the symptoms denoting the presence of the error seems to be more useful than the classification of the errors themselves.

After Santoro [11], one can specify the following types of error symptoms occurring in multi-agent system: contamination, commission, omission, and crash. The contamination error symptom manifests itself by invalid (not consistent with specification) data sent from one processor to another. The source of this symptom might be a fault application installed on the processor sending a contaminated data. The commission error symptom is manifested by the invalid communication behavior consisting in sending a message not meant to be sent according to the protocol. It can be a result of a deliberate attack or a software fault. The omission error symptom manifests itself by not sending a message which should have been sent according to protocol. It can be a result of connection error. The crash error symptom is caused by the failure of the processor and consists in stopping its activity both connected to processing and communication.

3 Experimental Results

The experiments tested multi-agent systems' immunity to error propagation phenomenon depending on their topology using simulation methods. A problem of distributed sum algorithm was introduced to model system's operation. This algorithm originates from well known distributed problems, such as broadcast and distributed agreement problems. Different MAS models operating according to distributed sum algorithm were constructed. Artificial fault sources were introduced into each model. A single model was therefore characterised by the following attributes: topology, number of elements, error probability and the type of an error. The experiment consisted in examining the operation of different models. As the result of each simulation the error percentage was calculated. Further analysis was made using this measure. More detailed description of experiments can be found in [6].

The examination of the error propagation phenomenon by means of quantitative and qualitative analyses [6] leads to the following conclusions. Firstly, the topologies utilized in MAS can be ordered from the most to the least immune to error propagation in the undermentioned way: full topology, random topology, ring topology, star topology, tree topology.

The qualitative analysis permitted explaining the reasons for this order. The full topology is the most immune one, because it contains the most highly connected topology. It is the most redundant structure of the examined. The random topology performs slightly better than the ring topology, probably because of the average distance between entities. Both topologies contain redundancy, but an error propagating in the ring topology affects more entities, because the invalid message traverses the bigger part of the structure before it gets detected and corrected. The weakest immunity to error have the topologies not employing redundancy - the tree and star topology. And again the topology having lower average inter-entities distance (the star topology) performs slightly better.

Another observations concern the error propagation of different error types. The error types can be ordered from the least to the most harmful: omission error, commission error, contamination error, crash error. The omission error can be easily cancelled by the redundancy. The results indicate that the topologies employing redundancy (the full, random and ring topologies) are almost completely immune to this kind of error. The visual analysis pointed out that, the information lost introduced by the omission error is cancelled by the presence of the replicated data in the other entities. On the other hand, when the redundancy is not present (like in the star and tree topologies), the information is ultimately lost. The commission error is slightly less harmful than the contamination error. The nature of these kind of errors seems similar - the damage is propagated along with the modified message. The difference is, that in commission error, the message is created by one of the entities and in contamination error, the existing message is modified. The modification is more harmful, because when the real message is affected, the situation, when its receiver does not have the proper value already, is more probable. The crash error is the most harmful error type, because it often leads to the disconnection of the topology and irreparable information lost.

The examination of the influence of error probability influence on error propagation phenomenon leads to the conclusion, that the sooner error occurs the more damage it causes. This relationship can be explained by using the visual analysis. When the error occurs in the early stage of distributed sum algorithm execution, the structure is in an empty state. Every value emerging within the system is received by entities and passed further. In later stages, the refusal of the improper message is more probable, because the entities mostly contain the right value already.

4 Statistical Analysis

In previous section the results of quantitative and qualitative analyses were shortly presented. Nevertheless the analysis conducted so far is not sufficient to accept or reject the hypothesis with high probability. The quantitative analysis operated purely on the average value of error percentage measure. It did not take into account the distribution of the observations. The reasoning based on the average only is weak and cannot be recognized as sufficient. On the other hand, the qualitative analysis provided some interesting results showing the exact differences in the error propagation process in different topologies. The weakness of this method is the fact, that it operates only on the selected experiments' results. Therefore it also cannot be a base for the research verification.

The verifications of the research requires usage of another tool. The large number of observations qualify the obtained results for statistical analysis. Application of mathematical statistics allows reasoning based on the whole population of results, but taking into account every single observation at the same time. Mathematical statistics is a classical method of verifying scientific hypotheses. Therefore using it to analyse the obtained results is a natural choice. From the point of view of this work the most useful statistical methods are statistical hypothesis testing and a correlation examination. The former method allows accepting or rejecting the hypothesis with a specified certainty degree. The latter examines the influence of one factor on the other. Using these tools it is possible to compare the data gathered from simulation of distributed systems having different topologies. The result of this comparison should be sufficient to verify the hypothesis. Furthermore the statistical analysis can be used to point out the factors having the influence on the error propagation phenomenon.

The hypothesis testing is one of the inferential statistical methods. In this work the classical hypothesis testing model will be utilized. The purpose of this model is to employ sample data collected during experiments to evaluate the hypothesis. According to [13], a hypothesis can be defined as a prediction about a single population or about the relationship between two or more populations. This work is particularly interested in the latter case. The goal of the statistical analysis is to verify if the error propagation in distributed systems having different topologies process differently. The sets of observation of error percentage grouped by the utilized topologies make up the samples. Each sample represent the population of observations of error percentage in specified topology. The hypothesis that need to be verified is, that the populations are different. This hypothesis is called the research hypothesis. It is a general statement of what is predicted. In order to employ the classical hypothesis testing model the research hypothesis needs to be transformed into the two opposing statistical hypotheses - the null hypothesis (H_0) and the alternative hypothesis (H_1). In the next step of the statistical inference procedure, the data is evaluated by using the appropriate inferential statistical test. Choosing the right test is crucial for obtaining reliable results of the procedure. The results of the statistical inference is called the test statistic. This statistic is then interpreted by using tables containing critical values of the expected test distribution. As the result of the interpretation it can be determined if the results of the research are statistically significant. The statistical significance tells if the observed difference in experiment is the result of the presence of experimental effect or is caused by chance.

As mentioned before the choice of the right inferential statistical test is extremely important. The factors determining the choice are: the goal of the test, the distribution of observations inside the samples and the number of observations. The goal of the test determines the property of the population that is considered. Some tests verify hypotheses concerning the mean value of the population, the others concern variation or median. Another factor determining the choice of the inferential statistical test is the distribution of observations inside the samples. Basically tests can be split into two groups - parametric or nonparametric tests. Parametric tests assume that the sample data come from the population of a specified distribution. Contrary, the nonparametric tests do not make any assumptions on the distribution. The third factor influencing the choice of the inferential statistical test is the number of observations inside the sample. Some tests are reliable only in case of large samples while the others do not require much data to provide the proper conclusions.

This paper utilizes the classical hypothesis testing model to compare the populations of error percentage values coming from the results of simulations of different distributed information systems. The goal of the test in such case is to show that the difference between the populations is statistically significant. In order to choose the inferential statistical test the distribution of the data need to be examined. Most of the statistical test assume that samples represent population having the normal distribution. Next paragraphs will determine if the obtained data can be analysed using these tests.

Figure 1 presents the histogram of the error percentage observations from the complete data set. The histogram shows that the distribution of data is not bell-shaped, what can indicate it is not normal. After filtering out the observations, where no error occurred (error type is none) the histogram presented in Figure 2 was obtained. The distribution of data visible on histogram is still not normal. To ultimately decide if the distribution of the data is normal, the Q-Q plot of the data has been made. It is presented in Figure 3. If the distribution was normal, the points in the figure would agree with the line. The discrepancy

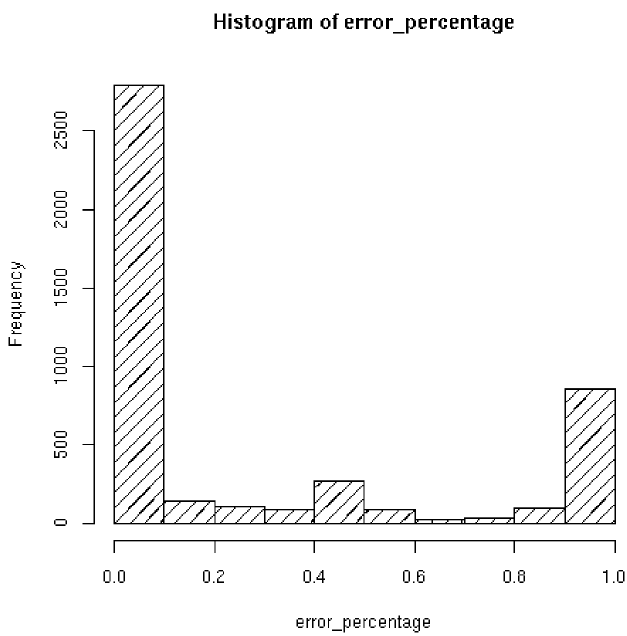


Fig. 1. Histogram of error percentage from complete data set

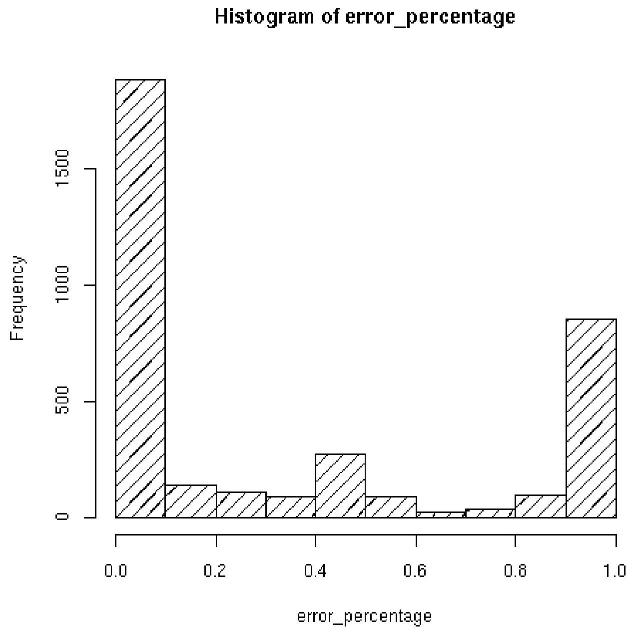


Fig. 2. Histogram of error percentage from filtered data set

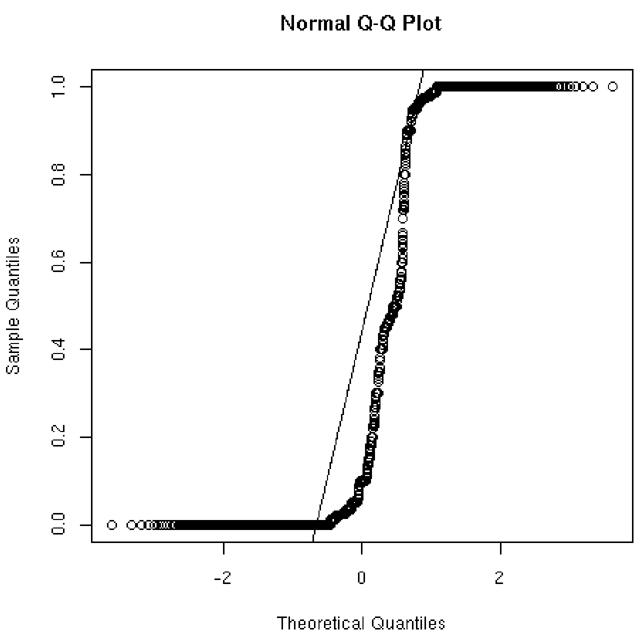


Fig. 3. Q-Q plot of filtered data set

between points and line is significant especially at the both ends of the range. From the conducted analysis it can be deduced, that the observations collected during the experiments do not have the normal distribution.

In a situation, when the observations are not normally distributed, the usage of nonparametric statistical inference test seems to be reasonable. Following the guidelines provided in [13] the Kruskal-Wallis one way analysis of variance on ranks was chosen. This test verifies, if in a set of k independent samples do at least two of the samples represent the population with different median values. In case of this work, the test is utilized to verify if the observed error percentage samples from the simulations of models having different topologies are coming from the populations having different median values. It is the way to compare the distributions of samples and state if the difference between them is statistically significant. The null hypothesis of the test takes the following form: $H_0 : \theta_{tt} = \theta_{ct} = \theta_{rat} = \theta_{rit} = \theta_{ft}$, where $\theta_{tt}, \theta_{ct}, \theta_{rat}, \theta_{rit}, \theta_{ft}$ are the median values of the error percentage samples coming from the results of simulations of models with the tree, star, random, ring and full topology accordingly. The alternative hypothesis takes the following form: $H_1 : \exists_{\theta_1, \theta_2 \in \{\theta_{tt}, \theta_{ct}, \theta_{rat}, \theta_{rit}, \theta_{ft}\}} \bullet \theta_1 \neq \theta_2$. The null hypothesis states that the considered samples are coming from the populations having the same median value. The alternative hypothesis states the contrary. The *p-value* of the test computed using the *R language* is equal $p = 1.267386 * 10^{-86}$. This result indicates that the null hypothesis can be rejected with the 1% significance level. Basing on the result of the Kruskal-Wallis test it can be stated, that there exists at least one pair of samples representing the populations having unequal median values. In context of this work, the result of the test indicates that the samples coming from simulations of models having different topologies are not coming from the same population. It means that the error propagation measured using the error percentage measure processes differently in different topologies.

The results of the Kruskal-Wallis test confirms only that there exists at least a pair of samples representing populations having different median value. But the result of the test would be the same if four of the populations have the same median value and the one population was different. In order to check if the samples represent populations differing from each other, the Mann-Whitney U test was employed. The goal of the Mann-Whitney test is the same as the Kruskal-Wallis test's. The difference is that the former operates on a pair of samples and the latter can operate on any number of samples. The statistical null hypothesis tested by the U test is: $H_0 : \theta_1 = \theta_2$, where $\theta_1, \theta_2 \in \{\theta_{tt}, \theta_{ct}, \theta_{rat}, \theta_{rit}, \theta_{ft}\}$. The alternative hypothesis is $H_1 : \theta_1 \neq \theta_2$. At this stage of analysis, every pair of samples was examined using the Mann-Whitney test. The resulting *p-values* of the tests are presented in Table 1.

The rows and columns of the Table 1 represent different topologies. FT, RaT, RiT, ST, TT represent the full, random, ring, star and tree topology accordingly. Each cell contains a *p-value* of the evaluation of the U test on two samples coming from the simulation of models of certain topologies. All of the obtained *p-values* are below the 5% significance level. The values distinguished by bold

Table 1. *P-values* of Mann-Whitney *U* test

	FT	RaT	RiT	ST	TT
FT	X	$\ll 0.05$	$\ll 0.05$	$\ll 0.05$	$\ll 0.05$
RaT	$\ll 0.05$	X	$\ll 0.05$	**0.01**	$\ll 0.05$
RiT	$\ll 0.05$	$\ll 0.05$	X	**0.02**	$\ll 0.05$
ST	$\ll 0.05$	**0.01**	**0.02**	X	$\ll 0.05$
TT	$\ll 0.05$	$\ll 0.05$	$\ll 0.05$	$\ll 0.05$	X

are much higher than the other ones. They are closer to the significance level, what may indicate the greater similarity of the pair of samples they evaluate. Nevertheless they are still below the 5% level, which is the typical significance level for Kruskal-Wallis test according to NIST [8]. Based on the results of the U tests, it can be stated that all of the samples have different median values. This may indicate, that the samples were not taken from the populations having the same error percentage distributions and therefore the error propagation course is significantly different in different topologies.

5 Conclusions

In this paper, we studied error propagation in self-organization structures like MAS in terms of statistical analysis. We discussed the different errors and topologies of MAS with respect to propagation process. The statistical analysis supported the observations made during the quantitative and qualitative analysis. The hypothesis stated in previous sections was adopted at the significance level of 5%. The Mann-Whitney test results presented in table 1 indicate, that the error propagation phenomenon measured using the error percentage value occurs differently in different topologies. The low $p - values$ of the utilized test emphasize the statistical significance of the conclusion. The only exceptions are the *U-test* results for the random - star and ring - star topologies pairs. It might be caused by the usage of the nonparametric test, which tends to give the higher $p - values$ under some circumstances. Nevertheless the obtained results give the strong support to the stated hypothesis.

Our future research will focus on generalizing the results found in this study. More specifically, (1) we will study the self-organization process in other structures, e.g. social structures, to determine if these systems follow the similar evolution to MAS; (2) we will study the error propagation during self-organization process to understand how the difference in self-organization processes could affect the immune quality; and (3) if additional data such as topology metrics of a network structure is available, we will study self-organization based on the quantitative measures of stability.

References

1. Albert, R., Jeong, H., Barabasl, A.L.: Error and attack tolerance of complex networks. Nature 406, 378–381 (1999)
2. Cami, A., Deo, N.: Techniques for analyzing dynamic random graph models of web-like networks: An overview. Networks 51, 211–255(2008)
3. Dressler, F.: A study of self-organization mechanisms in ad hoc and sensor networks. Computer Communications 31, 3018–3029 (2008)
4. Eigen, M., Schuster, P.: A principle of natural self-organization. Naturwissenschaften 64, 541–565 (1977)
5. Gaston, M.E., des Jardins, M.: The effect of network structure on dynamic team formation in multi-agent systems. Computational Intelligence 24, 122–157 (2008)
6. Król, D., Kukla, G.: Analysis of the error propagation phenomenon in social structures. Computing and Informatics 28 (to appear, 2009)
7. Król, D., Zelmozer, M.: Structural performance evaluation of multi-agent systems. Journal of Universal Computer Science 14, 1154–1178 (2008)
8. Kruskal-Wallis Test (2009), http://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/kruskwal.htm
9. Lin, J., Kuo, S.: Resolving error propagation in distributed systems. Information Processing Letters 74, 257–262 (2000)
10. Newman, M.: The structure and function of complex networks. SIAM Review 45, 167–256 (2003)
11. Santoro, N.: Design and Analysis of Distributed Algorithms. John Wiley & Sons, Chichester (2007)
12. Schuster, P.: Nonlinear Dynamics from Physics to Biology. Complexity 12, 9–11 (2007)
13. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures. Chapman & Hall, Boca Raton (2003)
14. Strogatz, S.H.: Exploring complex networks. Nature 410, 268–276 (2001)
15. Yu, L.: Self-organization process in open-source software: An empirical study. Information and Software Technology 50, 361–374 (2008)
16. Zhu, Q.: Topologies of agents interactions in knowledge intensive multi-agent systems for networked information services. Advanced Engineering Informatics 20, 31–45 (2006)

Hierarchical Organization of Agents Based on Galois Sub-Hierarchy for Complex Tasks Allocation in Massive MAS

Zaki Brahmi[1] and Mohamed Mohsen Gammoudi[2]

[1] Faculty of Sciences of Tunis, Tunisia
Zaki.Brahmi@isigk.rnu.tn
[2] High School of Statistics and Information Analysis of Tunis, Tunisia
mohamed.gammoudi@fst.rnu.tn

Abstract. A major challenge in the field of Multi-Agent Systems is to enable autonomous agents to allocate tasks efficiently. In previous work, we have developed a decentralized and scalable method for complex tasks allocation for Massive Multi-Agent System (MMAS) based on two steps: 1) hierarchical organization of agent groups using Formal Concepts Analysis approach (FCA), 2) computing the optimal allocation. The first step is computed by one agent named global allocator that computes Galois lattice representing the hierarchical structure of agent groups. Then, it simplifies the completed lattices by pruning unnecessary groups. The second step distributes the tasks allocation process among all agent groups. Nevertheless, the hierarchical organization process is still centralized. Moreover, generation of Galois lattice composed by all concepts ($2^{min(|O|,|A|)}$ concepts in the worst cases. Where 0 and A means, respectively, the set of objects and the set of attributs) and then simplification of the hierarchy of such size are not useful. This paper extends our last approach to distribute the organization process of agent groups among all agents by providing extension to the Pulton algorithm that generates Galois Sub-Hierarchy which is a polynomial size representation of a concept lattice. This decentralized self-organization of agents provides a flexible infrastructure for agents' dynamicity in MMAS.

Keywords: Complex tasks allocation, Galois Sub-Hierarchy, group, concept, FCA.

1 Introduction

A major challenge in the field of Multi-Agent Systems is to enable autonomous agents to allocate tasks efficiently. Tasks allocation is defined, in [18], as the ability of agents to self-organize in groups of agents in order to perform one or more tasks which are impossible to perform individually. In this work, we address the problem of tasks allocation in a cooperative environment. Common goal of agents is to maximize system's overall profit [13] and [20]. This type of allocation finds its applicability in many areas of the real world such as e-commerce [19],

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 460–470, 2009.

distributed delivery vehicles [12], grid computing [16], etc. These applications, generally, involve a large number of interacting agents. Thus, we call this kind of MAS, Massive MAS (MMAS). A MMAS is a very large MAS, containing tens or hundreds of thousands of agents.

In the context of SMAM characterized by a large number of dynamic and heterogeneous agents, traditional approaches based on the negotiation between agents, or a single allocator agent, proved impractical. So, a tasks allocation method for MMAS is judged efficient if it's scalable, decentralized and dynamic. In [2], we have proposed a decentralized and scalable complex tasks allocation method for MMAS. This method is based on the idea of a *group for each sub-task, agents which potentially, able to perform it, and then affect it to the appropriate agent which proposes the minimal cost.* For this finality, agent which initiates the allocation process (named global allocator in [2]) clusters agents according to their abilities to execute sub-tasks, and then, these agents' groups are organized in a hierarchical structure based on FCA approach. Computing Hierarchical Organization (HO) is done on three steps: 1) generates binary relations from all agents and their costs to perform sub-tasks. 2) Builds the complete Galois lattice representing the hierarchical structure of agent groups using a generation concepts lattice algorithm. 3) Simplifies complete Galois lattice by removing unnecessary concepts.

The global allocation is built by the union of a local allocation's set proposed by each group of agents. This method differs from others (e.g. [11]) by the fact that we don't have to generate all possible allocations. However, this method remains centralized at the level of hierarchical organization process. Moreover, generation of Galois lattices composed by all concepts ($2^{min(|O|,|A|)}$ concepts in the worst cases) and then simplification of the hierarchy of such size are not conceivable.

The aim of this paper is to extend our previous approach through the distribution process of HO among all agents by providing extensions to the *Pulton* algorithm [1]. This choice is justified by the fact that it generates a Galois sub-hierarchy which is a polynomial-size representation of a concept lattices. In addition, it offers a good performance compared to its concurrences [1].

This paper is organized as follows: in the next section we present the formalisms on which we base our proposal. In Section 3, we remind our tasks allocation method for MMAS. In section 4, we introduce our distributed hierarchical organization method. In Section 5, we detail the various communications primitives between agents. In section 6 we explain how to handle agents' dynamicity. Finally, we conclude our work and we give some perspectives.

2 Basic Concepts of Formal Concept Analysis Approach

In this section,we remind the mathematical foundations of Formal Concepts Analysis approach (FCA) as they are necessary to understanding this paper. We give the following definitions, for more details, reader is invited to see [7].

A formal context is a triplet $K = (O, A, R)$ for which O is a set of objects (or entities), A is a set of attributes (or properties) and $I(\subseteq O \times A)$ a binary relation between O and A. R associates an object to a property: $(o, a) \in R$ when *o has the property a* or *the property a is applied to the object o*. In our tasks allocation problem, objects are agents and properties are sub-tasks. The incidence relation indicates the sub-tasks in which agents have the necessary capabilities to perform them. Table 1 a) represents the binary relation of the formal context associated to the agents' set MAS= { Ag1, Ag2, Ag3, Ag4, Ag5, Ag6 } and sub-tasks' set ST = { a,b,c,d,e,f,g,h }.

The couple $C = (O_1, A_1)$ is called a *formal concept* of K with $f(O_1) = A1$ and $g(A_1) = O_1$, $O_1 \subseteq O$ and $A_1 \subseteq A$. O_1 is called *extent* of the concept; A_1 is called the *intent* of the same concept. The application f is which with any element o of O associates $f(o) = \{a \in A/(o, a) \in R\}$, and g is the application which with any element a of A associates $g(a) = \{o \in O/(o, a) \in R\}$. These two applications constitute the *Galois correspondence* of the context K. The set L of all formal concepts, provided with order relation $\leq_l$, is a Galois lattice structure called Hass diagram.

For the proposed method, a formal concept is like an agents' group. It connects a set of agents (extent) to a set of sub-tasks (intent).

2.1 Object Concept (Resp. Attribute Concept)

An object (resp. an attribute) x is said to be introduced by a concept C if x is in the extent (resp. intent) of this concept and no ancestor (resp. descendant) of this concept contains x in it extent (resp. intent). C is called the *object concept* (resp. *attribute concept*) of x. Object (resp. attribute) introduced by a formal concept formed his *reduced extent* (resp. *reduced intent*). To obtain a *reduced extent* (resp. *reduced intent*) of a concept, we will remove from its extent (resp. intent) any formal object (resp. formal attribute) which is present in any extent of its ancestor (resp. descendant). For example concept C = ({Ag2, Ag3},{g,b,a,h}) introduces the attribute g and no object. So, C is the attribute concept of g: C = $(\phi, \{g\})$.

2.2 Galois Sub-Hierarchy

The Galois Sub-Hierarchy (GSH) was introduced by Godin [8] with aims to construct a hierarchy of classes [6], in order to reduce the number of concepts. A *GSH* removes all concepts which are neither objects concept nor attributes concept. This means deleting concepts for which its reduced extension and intension are empty. GSH contains a maximum $|O| + |A|$ concepts [4]. In our approach we need at maximum $|A|$ concepts that represent the number of subtasks. Many algorithms have been proposed to build SHG: *ISGOOD* [15], *ARES* [5], *CERES* [9], *Pluton* [1] and [6]. In this work, we opt for using the *Pluton* algorithm because it offers a good performance towards its competitors [1]. This algorithm is based on TomThumb algorithm [4] which proposes a technical issue graphs to calculate a linear extension of the sub-hierarchy Galois. Pluton algorithm is composed of algorithms: *TomThumb*, *ToLinext* and *ToGSH*.

- *TomThumb* takes as input a formal context and returns an ordered list of the simplified extents and intents. We call this order list a linear extension of the GSH. This linear extension is a total order compatible with the partial order $(C, \leq_c)$.
- *ToLinext* takes as input an extension linear list and assembles consecutive pairs (intension, extension) which belong to the same concept. Only pairs formed by an extent directly followed by intent need to be considered. We obtained a topological storing of $(C, \leq_c)$.
- *ToGSH* builds the GSH from the topological storing by computing the edges of the graph.

In this work we have modified and improved this algorithm by: i) adding a new step to meet our need. ii) Increasing its performance by developing its distributed version based on the sub-formal and extending *ToGSH* to *DToGSH* with a better complexity compared to *ToGSH*.

3 Complex Tasks Allocation Method

As defined in [11], the task allocation problem can be formulated as follow: Consider a set T of n tasks $T = \{T_1, \ldots, T_n\}$, where each task T_i is composed of a set of sub-tasks ST_{ij}. Each task T_i has an execution cost $C(T_i)$. The costs of sub-tasks are not given. Consider a set MAS of m agents $SMA = \{Ag_1, \ldots, Ag_m\}$. Each agent Ag_i, have a set of capacities to carry out one or several sub-tasks. The goal is to find the feasible and optimal allocation which forms a set of agents in order to perform the set or tasks' sub-set.

Tasks allocation method [2], and [3] is based on the definition of an allocation which is the assignment' set of each sub-task ST_{ij} to an agent $Ag_k \in MAS$, as the cost proposed by agent Ag_k to execute ST_{ij} is the minimal. Let this principal, our idea is to find for each sub-task an agents' group that can execute it, and then calculate the optimal allocation without calculating all possible allocations. Indeed, our method distributes the task allocation process by sharing the calculation between all responsible agents. In summary, the method is based on the following steps:

1. Hierarchical organization of agent groups using the FCA approach,
2. Each local allocator proposes a local allocation,
3. The global allocator computes by the union of all local allocations after resolution of eventual conflict situations.

In [2] and [3] it is the global allocator agent which handles the first step (See section Introduction). In the context of massive MAS, this approach has some limitations:

1. The method isn't fully decentralized because it is based on one agent so as groups and organizes agents.

2. It is improved that Galois lattices simplification and its generation is costly. Indeed, we should generate and check in the worst cases $2^{min(|O|,|A|)}$ formal concepts (or group of agents) for an eventually simplification in order to built such organization. In addition, existing algorithms in the literature, for generating lattice concepts, become unusable if the number of columns increase.

To avoid these limitations we propose an extension of our method to distribute organization process among all agents involved in the tasks allocation. Indeed, each agent calculates its GSH which represents a sub-hierarchical organization that contains only the groups where the agent is a member. An improved version of the *Pulton* algorithm [4] is made for this purpose.

4 Hierarchical Organization of Agents

Hierarchical structures have been widely used by human organizations because they provide natural means to delegate tasks in order to reduce the communication flows and control activities within organizations [10]. This has motivated the development of our approach that tends to have such structure. In following, we describe hierarchical organization process of agents based on the FCA approach.

The definition of the order relation $\leq_l$ between two concepts C_1 and C_2 means that the set of sub-tasks of agents' group associated to concept C_1 is included in the set of sub-tasks of agents' group associated to concept C_2. This means that during the tasks allocation process sub-tasks that represent intension of concept C_1 will be treated twice by the responsible of the group C_1 and the responsible of the group C_2.

In order to avoid these redundancies, we proceed to the reduction of initial hierarchical organization. Indeed, in our approach we need, in maximum, a number of agents' groups which not exceed the $|ST|$ value. The simplification of the initial Galois lattice backs to a Galois Sub-Hierarchy (GSH). In our approach we have two hypotheses: i)We favor minimizing the communication cost to the storage agents' group and ii) A concept having an empty intension is linked to an inactive agents group (no sub-tasks to perform it).

To check these two assumptions, at the simplification process we simplified only intent. The GSH should remove, moreover concepts that are neither object concept nor attribute concept. In fact, keeping the extensions (agents) respects our tasks allocation idea which aims to find for each sub-task ST_{ij} the group of agents that can execute it.

In the follow, we mean by HOG Hierarchical Organization of agents' Groups that is a GSH which retains agents. Our approach distributes the *HOG* computing among all agents which compose the MAS with minimal of communication. Indeed, we based on the follow idea: e*ach agent has to know the Sub-graph of the GSH (SGSH) which composed of concepts where it's a member*: $SGSH \subseteq GSH / \forall c_j \in SGSH, Ag_i \in c_j.Ext$; with c_j is a concept. So, each agent should build its *SGSH*. The GSH is the merging of the SGSH's set. In order to compute a SGSH, we have developed an algorithm named *ExtPulton* based on this principal:

1. Calculate the *formal sub-context* associated to agent Ag_i.
2. Build the set L which represents linear extension of the SGSH using respectively TomThumb and ToLinext algorithms,
3. Delete from set L concepts of empty intension: L'
4. Compute the complete extension of each attribute concept.
5. Build the SGSH of agent Ag_i

The first step calculates from the global formal context, the formal sub-context related to the given agent. We define a formal sub-context of an Ag_i agent as follows: *Given the formal context (MAS, ST, R), we define the formal sub-context* K *of an agent* Ag_i*, for all objects (Agents)* $AG \in SMA$*, and for all set of attributes (sub-tasks)* $S \in ST$*, by* $K(AG, S) = (AG, S, R)$*, as* $S = R(Ag_i)$ *and* $A = \{a \in SMA / R(a) \subseteq R(Ag_i)\}$.

Each agent use its formal sub-context in order to reduce the time used to build the SGSH, this prevents the generation of unnecessary concepts. The formal sub-context related to agent Ag_3 is the following (Table 1 b)): Agents Ag_4 and

Table 1. a) Formal context: $R(MAS, ST, I)$- **b)** Ag_3's Formal Sub-context

	a	b	c	d	e	f	g	h
Ag1		×	×	×	×			
Ag2	×	×	×				×	×
Ag3	×	×				×	×	×
Ag4				×	×			
Ag5			×	×				
Ag6	×							×

	a	b	f	g	h
Ag1		×			
Ag2	×	×		×	×
Ag3	×	×	×	×	×
Ag6	×				×

Ag_5 are removed from the Ag_3's formal context because they can never belong to the same group where agent Ag_3 is a member. In fact, Ag_4 and Ag_5 can perform, respectively, sub-tasks' set $R(Ag_4) = \{d, e\} \nsubseteq R(Ag_3)$ and $R(Ag_5) = \{c, d\} \nsubseteq R(Ag_3)$.

After execution of *TomThumb* and *ToLinext*, the set L contains concepts which form the SGSH. According to formal context of Table ,
$L = \{(\{2\}, \phi), (\{3\}, \{f\}), (\phi, \{g\}), (\{6\}, \{a, h\}), (\{1\}, \phi), (\phi, \{b\})\}$.

Object concepts that have empty intension, belonging to the set L are inactive agents' groups. So, the set L is reduced to the set L' composed, only, of attributes concepts ($|L'| \leq |R(Ag_3)| < |L|$). This fulfills our initial idea of finding, for each sub-task, agents which can perform it. In our example, groups $(\{2\}, \ \phi \ )$ and $(\{1\}, \phi)$ will be removed and L' becomes: $L' = \{(\{3\}, \{f\}), (\phi, \{g\}), (\{6\}, \{a, h\}), (\phi, \{b\})\}$.

In order to build the SGSH related to an agent Ag_i, at step 4, we compute complete extent of each concept. The set L' contains only concepts that from the SGSH of agent Ag_i, $|L'| \leq |I(Ag_i)|$. In our example the set L' becomes L'= $\{(\{3\},\{f\}),(\{2,3\},\{g\}),(\{2,3,6\},\{a,h\}),(\{1,2,3\},\{b\}\}$.

Step 5) build the SGSH from topological sort of L'. He plays the same role as that of ToGSH. Indeed, we based on the property P of the list L'[4]: let be tow concepts $C_1 \in L'$ and $C_2 \in L'$; $C_1 \leq C_2$ then C_1 is before C_2 in L'. So, if two concepts are consecutive and have different extensions cardinalities, then necessary they are fitted with the order relation $\leq$, else $(|C_1.Ext| = |C_2.Ext|)$ they are at the same level and they have the same set of descendant and ancestor.

PROOF. proof that if two concepts $C_m \in L'$ and $C_n \in L'$ as C_m and C_n are at the sam level in the lattice $(|C_m.E_m| = |C_n.E_n|)$ then the concept C_p is a ancestor of C_m and, also, of C_n : $C_k \leq C_n$ then $C_p \leq C_m$.

According to algorithm 3 which removes all concepts that their extensions agent Ag_i isn't an element, which: $\forall C_i \in L', C_1 \wedge C_2 \wedge \cdots \wedge C_w = \{Ag_i\}$ then

$$C_1 \leq C_2 \leq \cdots \leq C_w$$

Assume that: $\exists C_p \in L'$ as $C_p \leq C_n$ and $C_p \nleq C_m$, Then $C_m < C_p$ (becaus L' is order),[1]

Means that: $|C_m.E_m| < |C_p.E_p|$ So, $|C_n.E_n| < |C_p.E_p|$ because $|C_n.E_n| = |C_m.E_m|$ which is a contradiction with [1]; thus $C_p \leq C_m$. □

The above Algorithm 1 presents the pseudo-code related to step 5:
Algorithm 1 DToGSH

```
1.  Add top concept to the end of list L'
2.  For i = 1 to  | L'| -1 do
     If  |L'[i].Ext| < |L'[i+1].Ext Then
       Add an edge between L'[i].Ext and L'[i+1].Ext
     Else {|L'[i].Ext| = |L'[i+1].Ext|}
      Add an edge between L'[i+1].Ext
      And all sub-concepts of L'[i]
   End If
  End for.
```

4.1 Organization Method Complexity

In this section, we analyze the theoretical complexity of the organization method. TOMTHUMB's complexity is analyzed as $O(|AG| * |I(Ag_i)|)$, that of TOLINEXT algorithm is described in [4] with a complexity of $O((|AG|+|I(Ag_i)|)^3)$. For the third step, in the worst case, the number of concepts that will be removed is equal to the number of agents which can perform sub-tasks that the agent Ag_i is capable, it is the number of formal sub-context's objects of agent Ag_i, $|I(Ag_i)|$. This step's complexity is evaluated in $O(|R(Ag_i)|)$. So, our organization method's complexity is analyzed as in:

$$O(|AG| * |R(Ag_i)| + (|AG| + |R(Ag_i)|)^3 + 2|R(Ag_i)|).$$

This complexity is polynomial according to the number of agents and the number of sub-tasks. But the number of sub-tasks is insignificant compared to the agents' number in a MMAS.

4.2 Computing Local and Global Allocation

The role of local allocator agent is to find the optimal local allocation and to propose it to global allocator agent. For each sub-task ST_{kj} it chooses an agent Ag_i which has the minimum cost to execute ST_{kj}. The idea of choosing agent Ag_i is to ensure that global allocation found is optimal. The search for a local allocation is similar to finding the minimum value in a table, which is estimated to $O(n)$ where n is the number of agents which compose the given group. This number is $|R(ST_{kj})|$.

The second role played by global allocator is to find the global optimal allocation. Indeed, global allocation is the union of all optimal local allocation (ALOp) before resolution eventual conflict situations. Formally, we define a global allocation GAL by:

$$GAL = \bigcup_{1 \leq i \leq nbrG} ALOp_i \tag{1}$$

Where, $nbrG$ means the number of groups obtained after regrouping agents. Indeed, each local allocator communicates its local allocation to global allocator. The latter, checks eventually conflict situations between these allocations. Thus, the global allocation GAL is called optimal. In follows, we propose a proof of optimality of this allocation:

PROOF. Demonstrate that ALG is optimal, is to show that:
$V(GAL) = Min(\{V(ALp_i)\})$, where, $0 < i \leq p$ (p is the number of possible allocations), and ALp_i is a possible allocation. We can present the local allocation (AL_i), following the possible costs of assignments, as follows:

$$\begin{pmatrix} AL_1 : a_{11}, a_{12}, \ldots, a_{1k} \\ AL_j : a_{j1}, a_{j2}, \ldots, a_{jk} \\ AL_p : a_{p1}, a_{p2}, \ldots a_{pk} \end{pmatrix} \text{ Where } \begin{pmatrix} a_{11} = V(ALOp_1) < a_{1w}, 1 < w \leq k \\ a_{j1} = V(ALOp_j) < a_{jw}, 1 < w \leq k \\ \ldots \\ a_{p1} = V(ALOp_p) < a_{kj}, 1 < j \leq k \end{pmatrix} \tag{2}$$

Let us ALP the set of all possible allocations, defined by:

$ALP = \bigcup_{1 \leq i \leq nbrG} AL_i$ its possible costs are:

$$\begin{pmatrix} a_{11} + a_{21} + \cdots + a_{pi} \\ a_{1j} + a_{2j} + \cdots + a_{pj} \\ \ldots \\ a_{1k} + a_{2k} + \cdots + a_{pk} \end{pmatrix} \text{ According to (2): } \begin{pmatrix} a_{11} \leq a_{1w}, \forall 1 < w \leq k \\ a_{j1} \leq a_{jw}, \forall 1 < w \leq k \\ \ldots \\ a_{p1} \leq a_{pw}, \forall 1 < w \leq k \end{pmatrix}$$

So, $\sum_{i=1}^{p} a_{i1} = \sum_{i=1}^{p} V(ALOpi)$.

Thus, $\sum_{i=1}^{p} V(ALOpi = Min(\{V(AL_i)\})$.

According to (1) and (2): $V(ALG)= \sum_{i=1}^{p} V(ALOpi = Min(\{V(AL_i)\})$. □

In order to find ALG, global allocator uses our algorithm baptized $GROUPALL$ (reader can see [3] for more details). This algorithm, after building the HOG, is based on the above steps to calculate ALG:

For each local allocation Al_i communicated by a local allocator:

1. Check if there is a conflict between Al_i and other allocations.
2. If there is a conflict then choose the allocation AL_M to be modified: $Al_i \leftarrow AL_M$ and repeat 1) and 2) until no conflict.
3. Else insert Al_i into the set *Allocation*.

The complexity of these steps is analyzed as in $O(\sum_{i=1}^{|ST|} |G_i.Ext| * i)$. with $|G_i.Ext|$ designs the number of agents which are member of the group G_i. Indeed, the complexity of steps 1 and 2 is $O(|G_i.Ext|)$(i represents locals allocations' number. These local allocations are the contents of the *Allocation*, at each iteration (or at the reception of a local allocation).

5 Agents' Dynamicity

A large scale MAS in open dynamic environment agents are born and died regularly. These ones must organize themselves effectively to complete tasks, subject of allocation. The process of self-organization, for such system, is crucial for the performance of the system as a whole. An appropriate organization may limit the control and the communication costs by improving significantly the performance of a given system [12]. In follows, we present the handling of agents' dynamicity.

5.1 New Agent Added

If a new agent Ag_k wants to participate in the allocation process, it contacts any agent (Ag_i) of the initial organization to be informed about sub-tasks that can execute it with the possible costs. Two cases are presented:

Case 1: Sub-tasks that agent Ag_k can execute them; are included in the Ag_i's formal sub-context (the set of sub-tasks that agent Ag_i can perform them): $Ag_k.ST \subseteq R(Ag_i)$. In this case the agent Ag_i communicates to the agent Ag_k its formal sub-context so that it builds the SGSH using the algorithm developed in Section 4. Similarly, the agent Ag_i proceeds to update its SGSH using the algorithm ARES [5] which is an incremental algorithm of SHG construction. ARES takes in input the new object to add, in our case the agent Ag_k, and the initial SGSH: $SSHG_{Init}$

Case 2: at the opposite case ($R(Ag_i) \nsubseteq Ag_k$) the contacted agent Ag_i communicates the new object (Ag_k) to the global allocator agent. The latter, broadcasts suggestions of the agent Ag_k to all other agents so as they can update their SSHG by using ARES [5]. For updating a SGSH, an agent Ag_j follows an algorithm called *UASGSH*:

1. Find properties P of the new object Ob to add:$R(Ag_i) \cap Ag_k.ST = P$
2. If $P! = \emptyset$ then $ARES(Ob(Ag_k, P), SGSHG_{Init})$

The function $ARES(Ob(Ag_k, P), SGSH_{Init})$ call the algorithm ARES, with $Ob(Ag_k, P)$ and $SGSH_{Init}$ represent, respectively, the new objet to add and the initial SGSH.

5.2 An Agent Leaves

If an agent Ag_i leaves the system, it informs all agents which are members of its group. Each agent proceeds to update its SGSH based by checking for each concept $c_j \in SGSH$ if $Ag_i \in c_j.Ext$. If it's the case Ag_i will be removed form $c_j.Ext$. Also, c_j will be deleted from SGSH, if $c_j.Ext = \emptyset$. Regrouping agents according to their abilities to perform sub-tasks facilitates the management of the dynamicity and agents' self-organization. It offers a flexible infrastructure for tasks allocation process if the structure of groups changes. For the first case, if a new agent Ag_i is involved in the allocation process, each responsible of a modified group checks whether its allocation is modified. An allocation Al will be modified if its value decreases by replacing Ag_j by Ag_i. The agent Ag_i is assigned to the sub-task ST_{ij} (with $< Ag_j, ST_{ij} > \in Al$ and the cost proposed by Ag_i to perform the sub-task ST_{ij} is less than that proposed by Ag_j.). If so, changes will be communicated to the global allocator. The latter checks and resolves eventual conflicts. For the second case if an agent Ag_i leaves the system there are two possible cases:

1. Agent $Ag_i \in G_j$ is assigned to the sub-task ST_{ij}: the responsible of the group G_j alter the assignment $< Ag_i, ST_{ij} >$ by substituting Ag_i by another agent $Ag_k Gj$ which has the minimum cost to execute ST_{ij}.
2. Agent Ag_i is not assigned to a sub-task: the global allocation still impact.

6 Conclusion

In our previous work we have developed a scalable and distributed tasks allocation method for MMAS in cooperative environment. An extension to this method is developed in this work through a distributed hierarchical organization process. Each agent calculates a part of the organization based on the Sub-Hierarchy Galois using a proved version of the Pulton algorithm.

This organizational structure provides a flexible infrastructure for scalability and dynamicity of agents. In addition, it reduces the communication cost between agents and the control of system. As a future work, we intend to apply our approach on real cases such as the web services discovery.

References

1. Arévalo, G., Berry, A., Huchard, M., Perrot, G., Sigayret, A.: Performances of Galois Sub-hierarchy-building Algorithms. In: Kuznetsov, S.O., Schmidt, S. (eds.) ICFCA 2007. LNCS, vol. 4390, pp. 166–180. Springer, Heidelberg (2007)

2. Brahmi, Z., Gammoudi, M.M.: Decentralized method for complex task allocation in massive MAS. In: Proc. Of 1st Mediterranean Conference on Intelligent Systems and Automation (CISA 2008), vol. 1019, pp. 287–293 (2008)
3. Brahmi, Z., Gammoudi, M.M.: Méthode scalable et décentralisée d'allocation de tâches complexes pour les SMA massifs. Ecol'IA, mars, Hammamet, Tunisie (2008)
4. Brerry, A., Huchard, M., Mcconnell, R.M., Sigayret, A., Spinrad, J.: Efficiently Computing a Linear Extension of the Sub-hierarchy of a Concept Lattice. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS, vol. 3403, pp. 208–222. Springer, Heidelberg (2005)
5. Dicky, H., Dony, C., Huchard, M., Libourel, T.: Un algorithme d'insertion avec restructuration dans les hiérarchies de classes. Actes de Langages et Modèles à Objets, Grenoble (1994)
6. Gammoudi, M.M., Nafkha, I.: A formal method for inheritance graph hierarchy construction. Information Sciences 140(3-4), 295–317 (2002)
7. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematical Foundation. Springer, Berlin (1999)
8. Godin, R., Mili, H.: Building and Maintaining Analysis-Level Class Hierarchies using Galois Lattices. In: Proc. of OOPSLA 1993, vol. 28, pp. 394–410. ACM Press, New York (1993)
9. Leblanc, H.: Sous-hiérarchies de Galois: un modéle pour la construction et l'évolution des hiérarchies d'objets. PhD thesis, Univ. de Montpellier II (2000)
10. Montealegre Vázquez, L.E., López, Y., López, F.: An Agent-Based Model for Hierarchical Organizations. In: Proceedings of AAMAS (May 2006)
11. Manisterski, E., David, E., Kraus, S., Jennungs, N.R.: Forming Efficient Agent Groups for Completing Complex Tasks. In: AAMAS 2006, Hakodate, Hokkaido-Japan (May 2006)
12. Sims, M., Goldman, C.V., Lesser, V.R.: Self-Organization through Bottom-up Coalition Formation. In: AAMAS 2003, Melbourne, Australia, July 14-18 (2003)
13. Rahwan, T., Jennings, N.R.: An Algorithm for Distributing Coalitional Value Calculations among cooperating Agents. Artificial Intelligence Journal 171(8-9), 535–567 (2007)
14. Rahwan, T., Jennings, N.R.: Distributing Coalitional Value Calculations Among Cooperating Agents. In: 25th National Conference on AI (AAAI), Pittsburgh, USA (2005)
15. Godin, R., Mineau, G., Missaoui, R.: Incremental structuring of knowledge bases. In: Ellis, G., Levinson, R.A., Fall, A., Dahl, V. (eds.) Proc. of KRUSE 1995 (1995)
16. Samee, U.K., Ishfaq, A.: Non-cooperative, Semi-cooperative, and Cooperative Games-based Grid Resource Allocation. In: Proceedings of IPDPS 2006, Rhodes Island, Greece (April 2006)
17. Sarne, D., Kraus, S.: Solving the Auction-Based Task Allocation Problem in an Open Environment. In: American Association for Artificial Intelligence (AAAI 2005), pp. 164–169 (2005)
18. Tosic, P.T., Agha, G.A.: Maximal Clique Based Distributed Group Formation For Task Allocation in Large-Scale Multi-Agent Systems. In: Ishida, T., Gasser, L., Nakashima, H. (eds.) MMAS 2005. LNCS, vol. 3446, pp. 104–120. Springer, Heidelberg (2005)
19. Tsvetovat, M., Sycara, K.P., Chen, Y., Ying, J.: Customer coalitions in the electronic marketplace. In: Proceedings of the Fourth International Conference on Autonomous Agents, pp. 263–264 (2000)
20. Viswanathan, V., DesJardins, M.: A Model for Large-Scale Team Formation for a Disaster Rescue Problem. In: Proceedings of AAMAS 2005 (LSMAS) (2005)

Communities of Practice and Its Effects on Firm Performance: A Process-Oriented Study

Chulmo Koo[1], Sungyul Ryoo[2,*], Yulia Wati[3], and Woojong Suh[4]

[1] Department of Business Administration, Chosun University, Gwangju, Korea
helmetgu@gmail.com
[2] Graduate School of Information, Yonsei University, Seoul, Korea
ryusr@yonsei.ac.kr
[3] Department of Business Administration, Chosun University, Gwangju, Korea
yuliawati@gmail.com
[4] Department of Business Administration, Inha University, Incheon, Korea
wjsuh@inha.ac.kr

Abstract. The growth of interests in communities of practice is a sign of a paradigm shift in knowledge management. This paper intends to integrate the motivational factors influencing participation in communities of practice and its impacts on firm performance. We identify three determinants affecting participation in communities of practice, and argue that it leads to knowledge exchange, operational performance, and firm performance. The data required for this field study was collected from 122 firms headquartered in Korea. Additional theoretical and practical implications of these findings are also discussed in the paper.

Keywords: Communities of Practice, Knowledge Exchange, Operational Performance, Firm Performance.

1 Introduction

In the new economy where firms are focusing knowledge management, managing knowledge effectively is important in gaining a competitive advantage. Within knowledge literature, there is consensus that knowledge is a key source of value for firms to achieve competitive advantage. However, there is striking dissentience between two different perspectives [1]. In the one perspective, which is dominant accounts in the knowledge literature, knowledge is regarded as being an "entity" that can be possessed and traded by individuals and organizations [2]. In the other perspective, knowledge is viewed as a "process" of knowing how to use knowledge in practice [3-6].

Until now, most organizations adopted knowledge strategies relying on knowledge as content. This approaches focused on technology such as Knowledge Management Systems (KMS). This networked structure has in fact developed into a platform for individuals to develop and share best practices across organizational unit [19-20].

* Corresponding author.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 471–480, 2009.

According to recent research, however, it has been recognized that KMS-oriented knowledge management to improve productivity could not help [7, 8]. Also, Malhotra [9] asserts that KMS is based on the machine theory model about the information processing and control, and thus it gets a specific procedure for getting the pre-qualified results. Hence, it can't explain the differences between input resources and business performance expectations.

Recently, focusing on knowledge as process, many organizations have recognized Communities of Practice (CoP) as an alternative to the KMS [4]. CoP is recognized as a catalyst for the success of organization's knowledge management[3] and an enabler to facilitate knowledge exchange among the participants in the community.

Few studies have considered the role of information technology from a CoP perspective [1]. It is not easy to build and sustain communities of practice or to integrate them with organizational process [22]. Although many authors declare that community of practice creates organizational value, there have been relatively small researches of the linkage between CoP with firm and operational performance. Thus, this paper aims to investigate the antecedents of CoP and its impacts on operational and firm performance. How organizations should encourage CoP to improve its performance is still important research question. This paper examines and recommends a model of CoP and its relation with operational and firm performance.

2 Theoretical Background

2.1 Communities of Practices

In focusing on the structure of the information system community and its knowledge creation and dissemination practices, it is important to recognize the community structure [30]. In their study, they suggest that IS field is best understood as a network of interacting Community of Practice. The CoP concept was first introduced by Wenger and Lave [40] in their book Situated Learning. They argued that CoP is composed by three critical elements. These are mutual engagement, joint enterprise, and shared repository. Mutual engagement refers to the actual participation and commitment of people. The important key is people's common interest. Joint enterprise could be explained while at the beginning CoP form and, members pursue the direction as CoP, simultaneously creating the common responsibility relations within the group. Shared repository refers to the process of pursuing common visions, members create source, namely knowledge banks, containing know-how, methodology, and methods in community. Wenger and Snyder [22] define a community of practice as one where people share their experiences and knowledge in free-flowing creative ways so as to foster new approaches to problem solving improvement, help drive strategy, transfer best practice, develop professional skills, and help to complain, recruit, and retain staffs [20]. An important part of the activity in communities of practice consists in the disclosure and the evaluation of best practice as well as any piece of information or of knowledge related to the relevant practice [18]. Wenger [19] pointed out three main characteristics shared by communities of practice is domain, interactions, and the development of shared repertoire of resources.

2.2 Business Value for Performance

The process of knowledge transfer between and amongst organizations in general is still not well understood [41, 20]. In some organizations, the communities themselves are becoming recognized as valuable organizational assets [21]. Based on organizational performance and operational mode, Chu and Khosla [17] differentiate CoPs by four business strategies. These four strategies consist of induced innovation learning, promote responsiveness, increase core competence, and enhance working efficiency. From case study conducted by Lesser and Storck [21], they have also identified some of the specific business outcomes that influenced by community in term of social aspect. These are decreased learning curve, increase customer responsiveness, reduce rework and prevent reinvention, and increase innovation. The performance of any organization is eventually affected by its ability to establish and preserve its competitive advantage. A company's success at this task depends on multiple factors. In addition to overall market conditions, management, marketing and sales strategy, operational performance and efficiency, and the like, radical innovation assisted by outstanding technology is one way to outperform competitors. The organizational performance of new ventures largely depends on their ability to innovate and to turn technology into business [33].

3 Research Model and Hypotheses Development

A conceptual model has been developed as shown in Figure 1. The model and several hypotheses are developed from prior research.

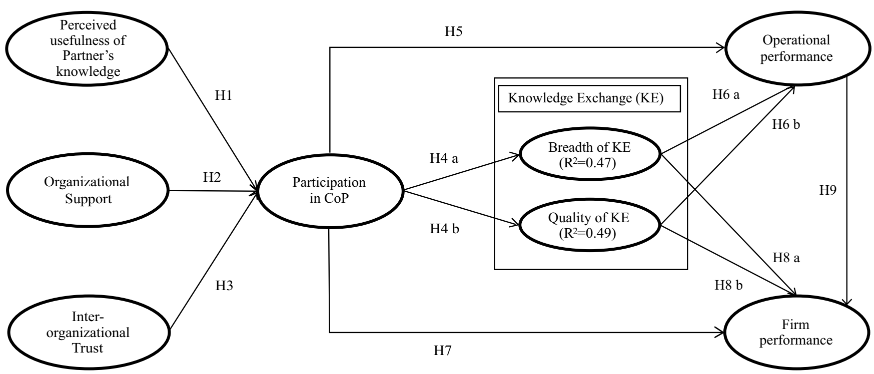


Fig. 1. Research Model

3.1 Communities of Practices (CoP) and Knowledge Exchange (KE)

Knowledge exchange or knowledge sharing refers to the act of exchanging knowledge amongst organization units for current or future benefit [27]. Perceived value, facilitating factor, and social factor have been identified as significant elements that promote participation in CoP. These are very similar to the factors in Triandis model [10], which has been used in several studies of explaining complex human behavior influenced by social, emotional, cognitive, and facilitating conditions. Moreover, member of community of practice know whom to ask for assistance with a problem. It is clear that effective learning depend on the availability of peers and their willingness to act as mentors and coaches [22]. Hence, we hypothesize:

H1: Perceived usefulness of partner's knowledge will positively influence participation in CoP.

Facilitating factor can be viewed as organizational support system for members participating in CoP. Organizational support for CoP formats the external environment that can promote the participation in CoP, not intervene. In this way, when an individual who want to participate in CoP does not be interfered with the external conditions, in fact, this action may occur [10]. Study by Thompson [24] arguing that organizations should sponsor the creation of community of practice. Therefore, we hypothesize:

H2: Organizational support will positively influence participation in CoP.

Trust as social factor is expected to another necessary factor participating in CoP [11, 12, 34]. Interorganizational trust is defined as the extend of trust placed in the partner by the member of focal organization [32, 26]. Trust is known as a good predictor of behavior in previous researches, such as e-commerce [35], business cooperative relationship [36], and individual or group performance [38, 37]. Previous empirical research has also found the causal relationship between trust and knowledge sharing behavior. Tsai and Ghoshal [39] provide empirical evidence to suggest that trust and trustworthy influence resource exchange and combination [28]. Thus we propose the following:

H3: Trust will positively influence participation in CoP.

A community of practice is an ideal forum for sharing and spreading knowledge and best practice across company [22]. Communities of practice shape the distribution of knowledge in organization [23]. For example, study by Graham et al. [25] indicates that the employee's participation in CoP has been patterned to a large extends along the lines of organization. A community of practice is a driver in a learning organization and one way to share intellectual capital [25]. In other word, participation in CoP will affect breadth of knowledge exchange in organization. Therefore, we hypothesize:

H4a: Participation in CoP will positively influence breadth of knowledge exchange.

A community of practice could be also a virtual group of people who share ideas and post best practices on discussion database [25]. Further, sharing practice help pass down idiosyncratic, competency-enhancing knowledge from the organizational to individuals or from one individual to other. Therefore, we hypothesize:

H4b: Participation in CoP will positively influence quality of knowledge exchange.

3.2 CoP, KE, and Operational Performance

Study by Hislop [23] resulted that community of practice affects the innovation process and operational performance of companies. CoPs help to improve job performance by developing individual skills and competencies and by increasing access to expertise to find answers, solve problems, and accomplish tasks. Therefore, we hypothesize:

H5: Participation in CoP will positively influence operational performance.

Social cognitive theory posits that if employee believes they could improve relationships with other employees by offering knowledge, they develop a more positive attitude towards knowledge sharing. Study by Lin and Huang [31] argue that individuals with positive outcome expectations were more likely to share their knowledge. Therefore, it may improve operational performance of task execution. Therefore, we hypothesize:

H6a: Breadth of knowledge exchange will positively influence operational performance.

H6b: Quality of knowledge exchange will positively influence operational performance.

3.3 CoP, KE, and Firm Performance

As organizations grow in size, geographical scope, and complexity, it is increasingly apparent the role of community of practice can improve organizational performance [21]. As firms grow in size, scope, and complexity, CoPs members who regularly engage in sharing and learning based on common interests, could improve firm performance [17, 33]. Thus we propose the following:

H7: Participation in CoP will positively influence firm performance.

Case study of Andersen Consulting Education Group indicates that group of people could collectively address issue of importance to the organization as a whole. Therefore, they could learn something new and valuable that would benefit both the organization and employees [25]. Study by Kaser and Miles [30] also recorded knowledge sharing relationships in some researches result in improve firm performance. Thus, we hypothesized:

H8a: Breadth of knowledge exchange will positively influence firm performance.

H8b: Quality of knowledge exchange will positively influence firm performance.

A company’s success at this task depends on multiple factors. One of these factor is operational effectiveness [33]. Chu and Khosla [17] also argue that one of characteristics of organization performance is working efficiency, which is the result of operational performance. So, it is reasonable to conclude that operational performance will influence firm performance. Therefore, I propose the following:

H9: Operational performance will positively influence firm performance.

4 Research Methodology

We used the survey methodology to collect data for testing the hypotheses. For the measurement items, we adopted existing validated scales and empirical procedures wherever possible and adapted them for this study. A total of 201 responses were received. Among those, cases with missing data were excluded. Hypotheses were tested using the final sample of 122. We used structural equation modeling called Partial Least Squares (PLS) to analysis the research model. PLS is based on an iterative combination of principal components analyses and regression, and it aims to explain the variance of the constructs in the model[13]. Specifically, SmartPLS 2.0 was used for the analysis.

4.1 Measurement Model Testing

We first assessed the measurement model for reliability, convergent validity, and discriminant validity. Table 1 shows the Cronbach's alpha. Overall, the reliability of the measurement scales is good. All αs are greater than 0.9, which is very higher than the recommended cutoff (0.7). Convergent validity was evaluated by using the rule of thumb of accepting items with individual item loadings of 0.50 or above, composite reliability (CR) of 0.70 or above, and average variance extracted (AVE) of 0.50 or above [14, 15]. Based on the results in Table 1, all constructs met the 0.70 CR and 0.50 AVE criterion, supporting convergent validity.

Discriminant validity was assessed in two ways. First, we examined the cross loadings showed that no item loads more highly on another construct than its own construct. Second, we compared the square root of AVEs from each construct with its correlations with the other constructs as a test of discriminant validity (shown on the Table 1 diagonal). Based on the results in Table 1, all constructs passes both discriminant validity tests [14].

4.2 Structural Model Testing

Table 2 reports the path coefficients and t-values for the PLS structural model we tested. The t-values were computed using 300 re-sampling with bootstrapping[13], and their significance level were assessed with a one-tailed distribution with 121 degrees of freedom.

The results show that participation in CoP is influenced by perceived usefulness of partner's knowledge (H1, t=2.77, p<0.01), organizational support (H2, t=3.77, p<0.001), and trust (H3, t=5.04, p<0.001). Its impacts on the breadth (H4a, t=12.48, p<0.001) and quality (H4b, t=13.71, p<0.001) of knowledge exchange are also significant. Also, participation in CoP has a significant direct effect on operational

Table 1. The Convergent validity

	number of items	C.R.	Cronbach's Alpha	AVE
KP	7	0.93	0.91	0.66
Org	3	0.94	0.90	0.83
Trust	4	0.94	0.92	0.81
CoP	4	0.91	0.87	0.72
BIE	5	0.93	0.91	0.73
QIE	4	0.93	0.90	0.76
OP	3	0.95	0.92	0.86
Firm	3	0.94	0.90	0.83

C.R. =composite reliability.

Table 2. Path Analysis Results

		Original Sample (O)	T Statistics (|O/STERR|)	p-value	Supported
H1	KP -> CoP	0.20**	2.77	0.003	yes
H2	Org -> CoP	0.37***	3.77	0.000	yes
H3	Trust -> CoP	0.42***	5.04	0.000	yes
H4a	CoP -> BIE	0.69***	12.48	0.000	yes
H4b	CoP -> QIE	0.70***	13.71	0.000	yes
H5	CoP -> OP	0.51***	4.47	0.000	yes
H6a	BIE -> OP	0.22	1.35	0.089	no
H6b	QIE -> OP	0.11	0.82	0.207	no
H7	CoP -> Firm	0.23*	2.04	0.022	yes
H8a	BIE -> Firm	0.09	0.58	0.282	no
H8b	QIE -> Firm	0.21	1.18	0.121	no
H9	OP -> Firm	0.37***	3.46	0.000	yes

(df=121), (1-tailed test), ***p<0.001, ** p <0.01, * p <0.05.

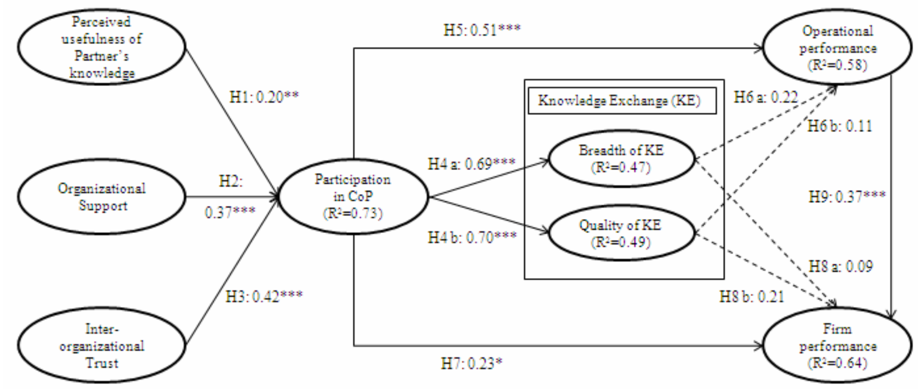


Fig. 2. Result of PLS Analysis

performance (H5, t=4.47, p<0.001). Breadth of knowledge exchange (H6a, t=1.35, *not significant*) and quality of knowledge exchange (H6b, t=0.82, *not significant*) don't influence on operational performance. Further, we found that participation in CoP (H7, t=2.04, p<0.05) have a positive link with firm performance. However, breadth and quality of knowledge exchange (H8a, t=0.58, *not significant*; H8b, t=1.18, *not significant*) have not a significant link with firm performance. Finally, the findings show that firm performance is influenced by operational performance (H9, t=3.46, p<0.001).

5 Discussion, Implication, Limitation

This study focused primarily on the determinants of encouraging the participation in CoP and its direct and indirect impacts on firm performance. The findings shown, indeed, that Community of Practice has a positive impact on operational and firm performance. Accordingly, (1) people are motivated to participate in Community of

Practice have been influence by perceived usefulness of partner knowledge, organizational support, and inter-organizational support. This result supports the study by Jeon and Kim [16] which verified the determinants of knowledge exchange in CoP by the empirical study based on Triandis Model [10]; (2) participation in CoP influences both breadth and quality of knowledge exchange positively; (3) participation in CoP directly affects operational performance as well as firm performance [23]; (4) operational performance influences firm performance positively. It means that antecedents variables used in this research may represent our first purpose.

Contrary to commonly accepted practice associated with knowledge exchange initiative, knowledge exchange does not mediate the impacts of participation in CoP on operational performance and firm performance. It supports study by Haas and Hansen [29] which find that sharing codified knowledge in form of electronic documents saved time during task, but did not improve work quality or signal competence to client. On the other hand, sharing personal advice improved work quality and signaled competence, but did not save time. Process of knowledge exchange negatively affected task performance and influenced firm performance as well. They recommend that different types of knowledge should be considered in order to measure capability of knowledge exchange in affecting performance. We identified that why our result didn't support our proposed hypotheses.

Our study was one of the first to provide empirical evidence that Community of Practice influences both operational and firm performance. In addition, this practice also affects knowledge exchange performance directly. It offers insight to academia and practitioners on the value of Community of Practice. Moreover, this research provides a reasonable answer why organization should encourage their employee to participate in it. We also found that interorganizational trust is the most important antecedents in CoP. Therefore, organization should consider the trust aspect in particular. Organization support and perceived usefulness of partner's knowledge also played significant role in creating CoP as knowledge exchange alternative.

Although this research presents strong evidence regarding the impacts of participation in CoP on operational and firm performance, this study suffers from several limitations. First, all of our samples were located in Korea. Also the sample is a little skewed toward smaller firms. Hence, the interpretation of our results is subject to the constraints of cultural characteristics of smaller firms as well as one country. In order to increase the external validity of the findings of this study, future research incorporating a sample from multiple companies in other countries is needed.

Second, the sample in this study is composed of active participants in the CoP. There is no opportunity to collect opinions from individuals who have ceased to participate in CoP. Moreover, the factors that drove them away from CoP participation may be valuable information for the managers. Future research is needed with a more integrative sample.

References

1. Hayes, N., Walsham, G.: Participation in groupware-mediated communities of practice: a socio-political analysis of knowledge working. Acct. Mgmt. Info. Tec. 11, 263–288 (2001)

2. Nonaka, I.: A Dynamic Theory of Organizational Knowledge Creation. Org. Sci. 5, 14–37 (1994)
3. Brown, J.S., Duguid, P.: Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation. Org. Sci. 2, 40–57 (1991)
4. Brown, J.S., Duguid, P.: Knowledge and Organization: A Social-Practice Perspective. Org. Sci. 12, 198–213 (2001)
5. Thompson, M.P.A., Walsham, G.: Placing Knowledge Management in Context. Journal of Mgmt. Stud. 41, 725–747 (2004)
6. Blackler, F.: Knowledge, Knowledge Work and Organizations: An Overview and Interpretation. Org. Stud. 16, 1021–1046 (1995)
7. McDermott, R.: Why Information Technology Inspired but Cannot Deliver Knowledge Management. Cali. Mgmt. Rev. 41, 103–117 (1999)
8. Malhotra, A., Gosain, S., El Sawy, O.A.: Absorptive capacity configurations in supply chains: Gearing for partner-enabled market knowledge creation. MIS Q. 29, 145–187 (2005)
9. Malhotra, Y.: Why Knowledge Management Systems Fail: Enablers and Constraints of Knowledge Management in Human Enterprises. In: Holsapple, C.W. (ed.) Handbook on Knowledge Management 1: Knowledge Matters, vol. 1, pp. 577–599. Springer, New York (2003)
10. Triandis, H.C.: Values, Attitudes, and Interpersonal Behavior. In: Howe Jr., H.E., Page, M.M. (eds.) Nebraska symposium on motivation, 1979. University of Nebraska Press, Lincoln (1980)
11. Argote, L., McEvily, B., Reagans, R.: Managing Knowledge in Organizations: An Integrative Framework and Review of Emerging Themes. Mgmt. Sci. 49, 571–582 (2003)
12. Lee, H., Choi, B.: Knowledge Management Enablers, Processes, and Organizational Performance: An Integrative View and Empirical Examination. J. Mgmt. Info. Sys. 20, 179–228 (2003)
13. Chin, W.W.: The Partial Least Squares Approach to Structural Equation Modeling. In: Marcoulides, G.A. (ed.) Modern methods for business research, vol. 10, pp. 295–336. Lawrence Erlbaum, Mahwah (1998)
14. Nicolaou, A.I., McKnight, D.H.: Perceived information quality in data exchanges: Effects on risk, trust, and intention to use. Info. Sys. Res. 17, 332–351 (2006)
15. Fornell, C., Larcker, D.: Evaluating Structural Equation Models with Unobservable Variables and Measurement Error. J. Mkt. Res. 18, 39–50 (1981)
16. Jeon, S.H., Kim, J.S.: Determinants for Knowledge Sharing in CoP: Focused on Triandis Model. Korean Mgmt. Rev. 34, 1667–1692 (2005)
17. Chu, Mei-Tai, Khosla, R.: Index Evaluations and business strategies on community of practice. Expt. Sys. App. 36, 1549–1558 (2009)
18. Muller, P.: Reputation, trust, and the dynamics of leadership in community of practice. J. Mgmt Grc. 10, 381–400 (2006)
19. Wenger, E.: Communities of practice and social learning systems. Org. 7, 225–246 (2000)
20. Bate, S.P., Robert, G.: Knowledge Management and Communities of Practice in Private Sector: Lessons for modernizing the National Health Service in England and Wales. Pub. Admin. 80, 643–663 (2002)
21. Lesser, E.L., Storck, J.: Communities of Practice and Organizational Performance. IBM Sys. J. 40, 831–841 (2001)
22. Wenger, E.C., William, M.S.: Communities of Practice: The Organizational Frontier. Harv. Bus. Rev. 78, 139–145 (2000)

23. Hislop, D.: The complex relations between communities of practice and the implementation of technological innovation. Int. J. Inno. Mgmt. 7, 163–188 (2003)
24. Thompson, M.: Structural and Epistemic parameters in communities of practice. Org. Sci. 16, 151–164 (2005)
25. William, G., Osgood, D., Karren, J.: A real life community of practice. Tra. Dev. 52, 34–38 (1998)
26. Wing, S.C., Chan, L.S.: Social Network, social trust, and shared goals in Organizational Knowledge Sharing. Info. Mgmt. 45, 458–465 (2008)
27. Davenport, T.H., Prusak, L.: Working knowledge: How organization manage what they know. Harvard Business School Press (1998)
28. Hsu, M.-H., Ju, T.L., Yen, C.-H., Chang, C.-M.: Knowledge sharing behavior in virtual community: the relationship between trust, self-efficacy, and outcomes expectation. Int. J. Hum.-Com. Stud. 65, 153–169 (2007)
29. Haas, M.R., Hansen, M.T.: Different knowledge, different benefits, toward a productivity perspective on knowledge sharing in organizations. Str. Mgmt. J. 28, 1133–1153 (2007)
30. Klein, H.K., Hirschheim, R.: The structure of IS discipline reconsidered: Implications and reflections from a community of practice perspective. Info. Org. 18, 280–302 (2008)
31. Lin, T.-C., Huang, C.-C.: Understanding knowledge management system usage antecedents: An integration of social cognitive theory and task technology fit. Info. Mgmt. 45, 410–417 (2008)
32. Zaheer, A., McEvily, B., Perrone, V.: Does Trust Matter? Exploring the effects of interorganizational and interpersonal trust on performance. Org. Sci. 9, 141–159 (1998)
33. Weber, B., Weber, C.: Corporate venture capital as a means of radical innovation: Rational fit, social capital, and knowledge transfer. J. Eng. Tech. Mgmt. 24, 11–35 (2007)
34. Kaser, P.A., Miles, R.E.: Knowledge activists: the cultivation of motivation and trust properties of knowledge sharing relationship. In: Acad. Mgmt. Proc., ODC, pp. D1–D6 (2001)
35. Gefen, D., Karahanna, E., Straub, D.W.: Trust and TAM in online shopping: an integrated model. MIS Q 27, 51–90 (2003)
36. Pavlou, P.A., Gefen, D.: Building effective online marketplaces with institution-based trust. Info. Sys. Res. 15, 35–62 (2004)
37. Paul, D.L., McDaniel Jr., R.R.: A field study of the effect of interpersonal trust on virtual collaborative relationship performance. MIS Q 28, 183–227 (2004)
38. Dirks, K.T.: The effects of interpersonal trust on work group performance. J. Appl. Psych. 84, 445–455 (1999)
39. Tsai, W., Ghoshal, S.: Social capital and value creation: the role of intrafirm networks. Acad. Mgmt. J. 41, 464–476 (1998)
40. Lave, J., Wenger, E.: Situated Learning: Legitimate Peripheral Participation. Cambridge University Press, Cambridge (1991)
41. Argote, L.: Organizational Learning: creating, retaining, and transferring knowledge. Kluwer Academic Publishers, New York (1999)

Conceptual Modeling of Semantic Service Chain Management for Building Service Networks: A Preliminary Result

Jason J. Jung and Young-Shin Han

KE@Y.U
Department of Computer Engineering
Yeungnam University, Korea
j2jung@intelligent.pe.kr

Abstract. In this paper, we emphasize that services (more specifically, a service network of a single business sector) should be aligned and matched with each other. This process enables the businesses to be aware of hidden relationships (or dependency) for discovering better service networks. Furthermore, once we have an integrated service network, classical social network analysis methods can be applied to understand structural role of each businesses, so that we can conceptually design a prototype of semantic service chain management.

Keywords: Service chain network, Social network analysis, Ontologies.

1 Introduction

An ad-hoc and automated coalition between businesses is important to a virtual enterprise (VE). It comes together to share experiences (and knowledge) or useful competencies in order to have better business tactics, and whose cooperation is supported through computer networks. A concept of such VEs has been applied to many forms of cooperative business relations, like outsourcing, supply chains, or spontaneous consortium [1]. Especially, with emergence of semantic technologies, semantic VEs have been introduced in [2,3]. The common goal of the semantic VEs is to automate interoperability between heterogeneous businesses which are providing various information by referring to their own knowledge structures (e.g., ontologies).

Services have been regarded as a key factor on business success. These services are provided by not only conventional offline companies and businesses but also online businesses. In terms of service science, more particularly, VEs are regarded as an online service providers, and their services should be aligned with each other. Service chain management enables service organizations to improve customer satisfaction and reduce operational costs through intelligent and optimized forecasting, planning and scheduling of the service chain, and its associated resources such as people, networks and other assets. The area is quite broad, covering field force and workforce automation, network and asset planning

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 481–489, 2009.

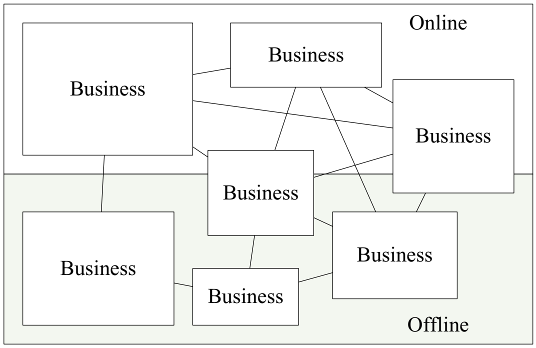


Fig. 1. Traditional business network for a VE including online and offline businesses

and also aspects of customer relationship management, human resources systems and enterprise resource planning. Furthermore, it addresses the key challenge of how all these technologies and systems are integrated into a cohesive blueprint.

According to [4], the notion of service value networks has been introduced. From the value network composed of consumers, service providers, multi-tier and auxiliary enablers, a certain business value can be either created (or increased) or destroyed, as depending on network structures and complexities of the domains.

In this context, we have to consider more general case where a number of different businesses are participating in a VE (e.g., business alliances and networks), as shown in Fig. 1 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 VE has to be automatically compared with other services from different VEs to find out which semantic relationships are involved to them in common. Consequently, once we somehow have a comparison result attached with a certain semantic relationship, a new service can be generated by composing two (or more) of the compared services.

Thereby, in this paper, we claim that a *service network* should be configured for better understandability on service chain management. More importantly, given a service network, we can apply network analysis methodologies which have been introduced in physics and sociology [5]. By using such network analysis methodologies, we can extract meaningful patterns (e.g., distance, centrality, betweenness, and so on) from highly complex networks.

Thus, the main request question of this study are as follows.

- How services can be compared with each other to discover semantic relationships between the services?
- How the relationships between services can be represented as a service network?
- What patterns can be discovered from the service network?
- How the patterns can be employed to service chain management?

Finally, we want to refer to semantic service chain management as a process to manage sharable services annotated by semantic VEs' local ontology.

The outline of this paper is as follows. In the following Sect. 2, we describe semantic interoperability dealing with the problem of semantic heterogeneity between VEs by knowledge matching. Sect. 3 introduces a definition of service network and present network analysis methods for discovering useful structural patterns from service networks. Sect. 4 will give a simple example, and discusses some significant issues and compares our contributions with the previous studies. Finally, Sect. 5 draws our conclusions of this work.

2 Interoperability between Heterogeneous VEs

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 [6] and ontologies [3]). Consequently, the businesses are difficult to be integrated, and more importantly, the VEs are impossible to automatically achieve *strategic* co-operations (e.g., *i*) business rules, e.g., strategies and policies, and *ii*) hierarchical taxonomies for describing the resources) with heterogeneous VEs.

In order to overcome this drawback, we have focused on semantic interoperability between the semantic VEs [3]. A large number of businesses have been inter-related with the others in a same VE or different VEs 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.

Definition 1 (Ontology). *An ontology $\mathcal{T}$ in a VE is built by aggregating a set of faceted taxonomies. Thus, supposing that a set of businesses $\{B_1, \ldots, B_{|\alpha|}\}$ be comprised in VE_α, an ontology $\mathcal{T}_\alpha$ is formulated by*

$$\mathcal{T}_\alpha = \bigcup_{B_k \in VE_\alpha} FT_k \tag{1}$$

where c_{root} in all FT are equivalently aligned. More importantly, domain experts can manually assert alignments $\mathcal{A}_\alpha = \{\langle c_p, rel^\star, c_q\rangle^\star | c_p \in FT_p, c_q \in FT_q\}$. These mappings are expressed with various relations between classes in different faceted taxonomies.

Definition 2 (Faceted taxonomy). *Let $\mathcal{C}_k$ a set of all concepts of a business B_k, participating in a VE_α. A faceted taxonomy FT_k is defined as a set of subclass assertions between classes in the concept set $\mathcal{C}$. Hence, FT_k is given by*

$$FT_k = \{c_{root}, \langle c_i, subc, c_j\rangle | c_i, c_j \in \mathcal{C}_\alpha, c_j = subClass(c_i)\} \tag{2}$$

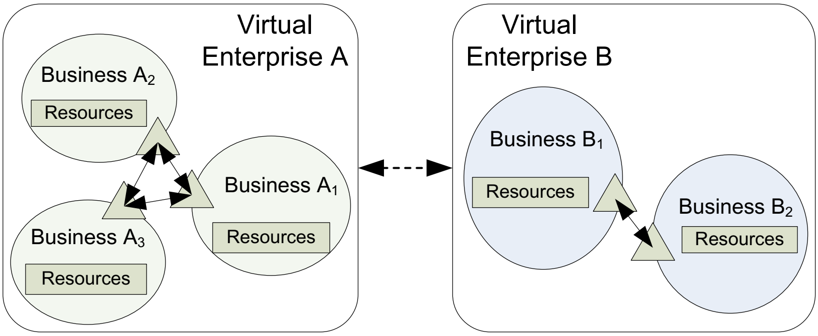


Fig. 2. Interoperability between VEs; Triangles are indicating local ontologies, and the semantic information from each ontology is applied to annotate resources in the corresponding businesses

where c_i *means a superclass of* c_j*. We put* c_{root} *as root class of* FT_k *for convenience.*

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.

As shown in Fig. 2, for the interactions between two virtual enterprises VE_A and VE_B, their organization ontologies T_A and T_B have to be aligned, in advance. Triangles indicate the corresponding local ontologies. Two alignments for i) intra- and ii) inter-organization are shown as solid arrows and a dotted arrow, respectively.

For doing this, many studies have been proposed to provide interoperability by discovering and integrating local knowledge structures between VOs [2]. They can be briefly noted into three issues;

- Incremental discovery of local knowledge [7],
- Knowledge matching (including schema and ontology matching) [8], and
- Interoperability via third-party platforms, e.g., service-oriented architecture (SOA) [9].

We have proposed an efficient method to build a VE by mapping heterogeneous ontologies of businesses, 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.

Additionally, we are focusing on supporting local users (e.g., decision makers) through aligning the ontologies applied to annotate (or classify) the resources on VEs. It means the local users in a certain VE can access to the other VEs 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, so that they can deploy meaningful translation services (e.g., query expansion [10] and transformation).

3 From Business Network to Service Network

Here, we want to assume that in a semantic service chain management, a service should be described with concepts extracted from local ontology of a semantic VE.

Definition 3 (Service). *A description of a service s from B_i is represented as*

$$d_s = \{c_k | c_k \in T_i\} \tag{3}$$

where a set of concepts from concepts in local ontology.

3.1 Building Service Networks

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 process[1], 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 4 (Alignment). *Given two ontologies $\mathcal{T}_i$ and $\mathcal{T}_j$, the alignments between two ontologies are represented as a set of correspondences $CRSP_{ij} = \{\langle c, rel, c' \rangle | c \in \mathcal{T}_i, c' \in \mathcal{T}_j\}$ where rel means the relationship between c and c', by maximizing the summation of class similarities.*

Finally, alignment process makes heterogeneous VEs interoperable (even partially) among them. For example, local users in a VE can easily and transparently access to the other VEs. To do so, VEs have to conduct the ontology matching process in advance. Suppose that a set of VEs $\{L_1, \ldots, L_N\}$ should be interoperable with each other. Alignment process can find out the correspondences between all pairs of ontologies, i.e., L_i obtains $N - 1$ sets of correspondences.

Most importantly, given two services s from B_i and s' from B_j in a semantic service chain management, 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 service chain management.

Definition 5 (Service network). *A service network N is defined as a network among services.*

$$N = \langle S, E, R \rangle \tag{4}$$

where S is a set of services supplied by VEs, and $E \subseteq |S| \times |S|$ means a set of relationships between services. Additionally, $R = \{\equiv, \sqsupseteq, \sqsubseteq\}$ is a set of semantics for describing the semantics of service relationships.

[1] We skip ontology matching processes. Please refer to other literatures [8] for more details.

Table 1. Service relationship discovery by semantic matching process

Scope	Service description	Semantic relationships
In a same business	$d_s = d_{s'}$	$s \equiv s'$
	$d_s \subseteq d_{s'}$	$s \sqsubseteq s'$
	$d_s \cap d_{s'} \neq \phi$	not decidable
In a different business	$d_s = d_{s'}$, $d_s \subseteq d_{s'}$, $d_s \cap d_{s'} \neq \phi$	not decidable
	$\{\langle c, \equiv, c' \rangle \mid c \in d_s, c' \in d_{s'}\} \sqsubseteq CRSP_{ij}$	$s \equiv s'$
	$\{\langle c, \sqsubseteq, c' \rangle \mid c \in d_s, c' \in d_{s'}\} \sqsubseteq CRSP_{ij}$	$s \sqsubseteq s'$

Every VE should be registered in a semantic service chain management.

3.2 Service Network Analysis

Once we have built a service network, a number of network analysis methods can be applied to discover hidden knowledge underlying the service network, as various measures on the networks between people designed from social network analysis [5] and from semantic social network [11]. Note that these measures apply only if the network is connected. These measures are often normalized (between 0 and 1) but we present their simplest form.

Closeness. The inverse of average length of the shortest path between a node s (i.e., service) and any other node in the network:

$$Closeness^i(s) = \frac{|N-1|}{\sum_{s' \in N} spd^i(s,s')} \tag{5}$$

where N is the total number of nodes in a service network, and *spd* is a function measuring the shortest path distance.

Betweenness. [12] The proportion of shortest paths between two nodes which contains a particular node (this measures the power of this node):

$$Betweenness^i(s) = \sum_{s',s'' \in N} \frac{|\{p \in sp^i(s',s), p' \in sp^i(s,s'') | p \cdot p' \in sp^i(s',s'')\}|}{|sp^i(s',s'')|} \tag{6}$$

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 node in interlinked environment. Given initial authority and hub degrees of 1, the degrees are iteratively computed by

$$Hub^i_{t+1}(s) = \sum_{s':\langle s,s' \rangle \in N^i} Auth^i_t(s') \text{ and} \tag{7}$$

$$Auth^i_{t+1}(s) = \sum_{s':\langle s',s \rangle \in N^i} Hub^i_t(s') \tag{8}$$

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 network.

4 Example and Discussion

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 businesses, the businesses 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 businesses until customers), the best one should be selected out of a set of all possible service chains by taking into account the semantic interoperability between the businesses.

As shown in Fig. 3, manufacturing industry sector (e.g., equipment suppliers) can be automatically integrated with medical producer sector (e.g., pharmacy wholesalers). Moreover, if they have semantic-based information systems on open networks, we can obtain semantic relationships between such businesses 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

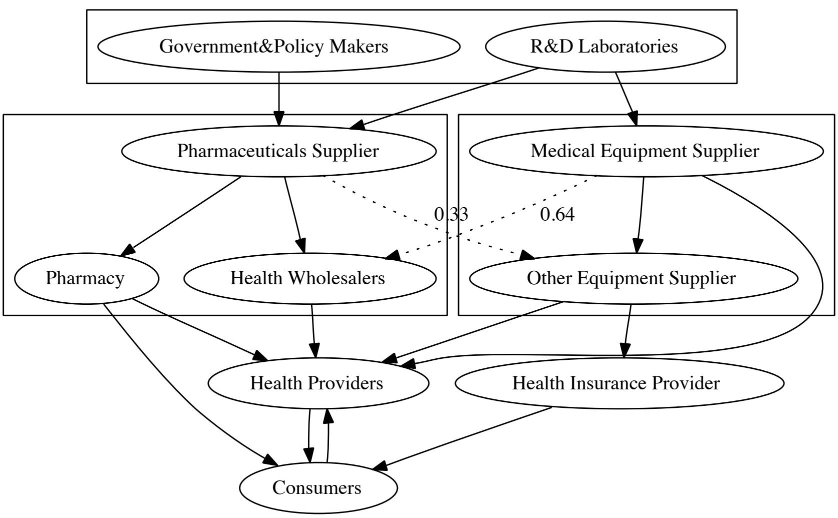


Fig. 3. Health care market model fragmented from [4]

more closely related with each other, compared to “Other Equipment Supplier” and “Pharmaceuticals Supplier.”

As another issue, we can apply some social network analysis methods on the service network. From a given Fig. 3, we can measure various measurements of each business (or each service by a business). As a simple example, with respect to the closeness ($N = 10$) in Eq. 5, the closenesses of “Goverment&Policy Makers”, “R&D Laboratories”, and “Health Providers” are $\frac{9}{1+2+2+2+3+3+3+8+8} = 0.28$, $\frac{9}{1+1+2+2+2+3+3+4+8} = 0.35$, and $\frac{9}{1+1+1+2+3+3+2+1+2} = 0.56$, respectively (Distance from unreachable nodes is assigned with $N-1$). Thus, we can guarantee that “Health Providers” has been located in more important position rather than the others.

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 WSRF[2], SOAP[3], UDDI[4] and WSDL[5]. 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 indexes).

5 Concluding Remarks and Future Work

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 businesses as well as from offline businesses. 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 businesses to put semantics into them. We are expecting some machine learning approached to deal with this issue.

[2] Web Service Resource Framework.
[3] Simple Object Access Protocol.
[4] Universal Description, Discovery and Integration.
[5] Web Service Description Language.

– 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.

References

1. Cardoso, H.L., Oliveira, E.C.: Virtual enterprise normative framework within electronic institutions. In: Gleizes, M.P., Omicini, A., Zambonelli, F. (eds.) ESAW 2004. LNCS, vol. 3451, pp. 14–32. Springer, Heidelberg (2005)
2. Castano, S., Ferrara, A., Montanelli, S.: Matching ontologies in open networked systems: Techniques and applications. Journal of Data Semantics 5, 25–63 (2006)
3. Jung, J.J.: Taxonomy alignment for interoperability between heterogeneous virtual organizations. Expert Systems with Applications 34(4), 2721–2731 (2008)
4. Basole, R.C., Rouse, W.B.: Complexity of service value networks: Conceptualization and empirical investigation. IBM Systems Journal 47(1), 53–70 (2008)
5. Wasserman, S., Faust, K.: Social network analysis: methods and applications. Cambridge University Press, Cambridge (1994)
6. 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 1997), pp. 51–61. ACM Press, New York (1997)
7. Jung, J.J.: Ontological framework based on contextual mediation for collaborative information retrieval. Information Retrieval 10(1), 85–109 (2007)
8. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal of Data Semantics 4, 146–171 (2005)
9. Vetere, G., Lenzerini, M.: Models for semantic interoperability in service-oriented architectures. IBM Systems Journal 44(4), 887–903 (2005)
10. Qiu, Y., Frei, H.P.: Concept based query expansion. In: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1993), pp. 160–169. ACM Press, New York (1993)
11. Jung, J.J., Euzenat, J.: Towards semantic social networks. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 267–280. Springer, Heidelberg (2007)
12. Freeman, L.: Centrality in social networks: Conceptual clarification. Social Networks 1, 215–239 (1979)
13. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of ACM 46(5), 604–632 (1999)

Analysis of the Broadband Internet Penetration in South Korea: Drivers and Challenges

Joon Koh[1], Seonjin Shin[2], Kiyong Om[3], and Inhak Kim[1]

[1] Department of Business Administration, Chonnam National University, Gwangju, Korea
kjoon@chonnam.ac.kr, jesus4000@nate.com
[2] Research Group for Global Korean Business & Culture, Chonnam National University, Gwangju, Korea
sjshin@chonnam.ac.kr
[3] School of Industrial Management, Korea University of Technology and Education, Cheonan-city, Korea
kyom@kut.ac.kr

Abstract. On broadband access to the Internet, Korea is a worldwide leading country in the OECD area. Recently, Korea has moved from DSL to Fiber-to-the home (FTTH) and Fiber-to-the-building (FTTB) subscriptions for the advanced high-speed internet connections. In this paper we provide a brief description of the Korean high-speed Internet market, then analyses key factors influencing the explosive growth in subscribers for the high-speed Internet service in terms of supportive government policy, competition between broadband suppliers, attractive applications and unique Korean culture, and finally try to touch future challenges for the effective use of the superb IT infrastructure. This analysis is expected to give valuable implications not only to Korean policy-makers but also to other countries making an effort to promote the deployment rate of the high-speed Internet service.

Keywords: Broadband, National Innovation System (NIS), Government Support & Coordination, Supply Side, Demand Side.

1 Introduction

South Korea (hereinafter, Korea) continues its commanding lead in the global speed Internet penetration. According to the OECD [16], Korea had in 2007 broadband penetration 29.9 out of 100 were connected to the high-speed always-on Internet (see the Table 1). According this report, operators in several countries continue upgrading subscriber lines to fiber. Fiber-to-the home (FTTH) and Fiber-to-the-building (FTTB) subscriptions now comprise 8% of all broadband connections in the OECD, up from 7% one year ago, and percentage is growing. However, fiber connections account for 36% of all broadband subscriptions in Japan, and 31% in Korea. In Table 1, Korea is ranked the top in the Fiber usage portion of total broadband usage technologies among the top fifteen broadband subscribers per 100 inhabitants. Further, in the criteria of total subscribers, Korea is a global leader in national broadband diffusion.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 490–499, 2009.

Table 1. Broadband subscribers per 100 inhabitants by technology

Rank	Country	DSL	Cable	Fiber/LAN	Other	Total	Total Subscribers
1	Denmark	21.3	9.7	2.9	0.4	34.3	1,866,306
2	Netherlands	20.4	12.7	0.4	0.0	33.5	5,470,000
3	Switzerland	20.5	9.3	0.0	0.9	30.7	2,322,577
4	Korea	10.1	10.6	**9.2**	0.0	29.9	**14,441,687**
5	Norway	22.7	4.5	1.8	0.7	29.8	1,388,047
6	Iceland	29.0	0.0	0.2	0.6	29.8	90,622
7	Finland	24.4	3.7	0.0	0.8	28.8	1,518,900
8	Sweden	17.9	5.6	4.6	0.4	28.6	2,596,000
9	Canada	11.9	12.9	0.0	0.1	25.0	8,142,320
10	Belgium	14.5	9.2	0.0	0.1	23.8	2,512,884

Source: OECD, June, 2007.

These things demonstrate that Korea plays a worldwide leading role in *advanced* high speed broadband penetration.

Korea is moving forward whilst other Asian countries are showing stagnant growth rate. Although there can be various factors for Korea's growth, many researchers have pointed out Korea's strong National Innovation System (NIS) as one of the development and growth factors [21].

Since most of these studies pursue a global perspective, there has been a general lack of empirically-based independent, academic research on issue specific to the Korean context. In this article, first, we review the NIS. Then, based on research on systems of innovation, a framework to analyze the diffusion of broadband internet in a country is attempted to develop. Together, we provide a brief history of Korea's speed Internet deployment, examine factors behind the explosive growth in subscribers for the high-speed Internet service with an innovation system perspective, and find lessons from the case of Korea.

2 Conceptual Background

This section sets the conceptual background for this study. First, innovation and the innovation process are briefly defined in order to delineate the scope of this study. Following that the concept of National Innovation System (NIS) is introduced.

2.1 Concept of Innovation and Innovation Process

Researchers have viewed innovation both as a discrete product or outcome "a new idea, method or device" [4, 9] and as a process "the process of introducing something new" [18]. Also, innovation is defined as adoption of an internally generated or purchased device, system, policy, program, process, product, or service that is new to the adopting organization [4, 26]. Rogers [19] saw innovation as "an idea, practice, product that is perceived to new by and individual or other unit of adoption." While the literature on organizational innovation is very large [18, 24], organizational innovation is generally adopted as “the adoption of an idea or behavior that is new to the organization adopting it” [3]. More narrowly, innovation is defined as “the first or early use of an idea by one of a set of organizations with similar goals” [2]. Innovation activities and all scientific, technological, organizational, financial and commercial steps which actually, or are intended to, lead to the implementation of innovations.

Innovations usually do not take place in a static environment. They are rather a result of a dynamic process involving interplay of several firm-internal and external factors. Innovation process encompasses several systematic steps such as requirement analysis, idea generation, project planning, product development and marketing [25]. The individual steps may overlap each other and in a simplified process be categorized into 3 broad phases; Conception, Implementation, and Marketing [23].

2.2 National Innovation System

National Innovation System (NIS) plays important role in the development of national economy [6]. The outstanding feature of NIS concept is that it deals with system itself rather than individual innovation actors. Innovation systems are country-, region- and/or industry-specific elements which support developing and successfully marketing new products and services.

The concept of national innovation systems depends on the premise that understanding the linkages among the actors involved in innovation is a key to improving technology performance. Innovation and technical progress are the result of a complex set of relationships among actors producing, distributing and applying various kinds of knowledge. The innovative performance of a country depends predominantly on how these actors relate to each other as elements of a collective system of knowledge creation and use as well as the technologies they use. These actors are primarily private enterprises, universities and public research institutes and the people within them.

Freeman [6] defined it as "the network of institutions in the public and private sectors whose activities and interactions initiate, import, modify and diffuse new technologies." According to Lança [11], national innovation system is a social system and dynamic where it is developing a production and reproduction of knowledge of the individual agents and collectives, fundamental resource of the societies contemporaneous. The process of learning, essentially interactive, it makes fundamental in this problematic.

According to the OECD [17], NISs have the following six different functions: (1) technology and innovation policy formulation, (2) performing R&D, (3) financing R&D, (4) promotion of human resource development, (5) technology diffusion, and (6) promotion of technological entrepreneurship. The OECD research [17] also indicates that two main sources are responsible for the NIS diversity. The first source involves the country size and the level of development of that country. The second source relates to the respective roles of the main actors in the innovation system, and the forms, quality and intensity of their interactions.

3 Broadband Internet Deployment in Korea

The history of the Internet in Korea began with the launch of the System Development Network (SDN) in 1982. The Korea Institute of Electronic Technology (KIET), a precursor to the Electronics and Telecommunications Research Institute (ETRI), and Seoul National University, jointly set up the SDN system. With SDN, Korea was only the second country in the world to operate a nationwide Internet system, behind only the United States. In June of 1994, KT, the former state-run telecom giant, launched KORNET, a commercial online service. This debut stimulated the Korean Internet drive, with LG Dacom launching its own online service named Boranet. All of this, however, had not yet reached the level of broadband access. In 1995, the Korean Information Infrastructure (KII) project was started with the aim at establishing a high-speed information infrastructure by 2015. This began with the acceleration of the deployment of broadband Internet.

SK Broadband (Hanaro Telecom), Korea's second largest telecom carrier released the world's first commercial asymmetric digital subscriber lines (ADSL) service, and KT followed suit a year later. ADSL is based on legacy copper telephone lines, transmitting data at an average speed of 2.5 megabits per second, about 50 times faster than conventional dial-up modems. As a result of these efforts, including many high-density apartment complexes, Korea has currently become the world's most accessible country for broadband Internet.

4 Factors behind the Successful Broadband Penetration

Recently, a system of innovation perspective has been receiving much attention from a number of studies [5, 7]. According to the line of research, an individual actor never innovates in isolation, but should interact with other organizational actors to obtain, develop, and exchange knowledge, information, and resources required for innovation. Existing studies on systems of innovation have focused on analyzing the interrelationships among various actors to identify dynamics and complexities in the innovation process at different system levels. The successful penetration process of broadband service innovation in Korea may be depicted more accurately by analyzing interactions of diverse players in the industry.

Based on an innovation perspective as well as the diffusion theory [1, 19, 22], this article explores factors behind the explosive growth in subscribers for the high-speed Internet service, and finds lessons from the success of broadband penetration in Korea in terms of government support/coordination, supply side, and demand side, as depicted in Figure 1.

4.1 Government Support and Coordination

IT Industry analysts attribute the rapid penetration of broadband services largely to the vigorous policy initiatives of the Ministry of Information and Communication (MIC) [15]. In 1999, the Korean government launched "Cyber Korea 21" project to

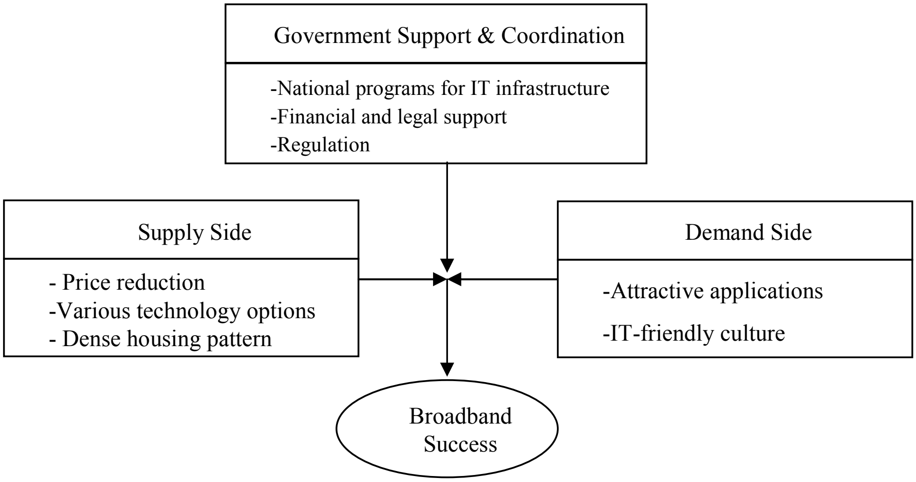


Fig. 1. Factors behind Korea's success in broadband Internet penetration

focus financial and human resources on the technological advancement of information and communication networks and to attract private sector investments. Korea aimed to construct a creative knowledge-based nation and to increase the share of GDP of the knowledge-based industries to that of the OECD by improving the capability of creating, storing, and utilizing information and knowledge. For this purpose, there were four top policy priorities: informatization, promotion of IT industry, deregulation and market liberalization, and encouragement of foreign investment. To strengthen network infrastructure, the KII project was designed to connect 144 core areas via fiber optic cable, enhancing subscriber networks such as ADSL, CATV networks, ISDN, wireless services and satellite networks.

After the success of the Cyber Korea 21 project, the MIC initiated the "Global Leader, e-Korea" project in 2002, aiming at upgrading the country's IT infrastructure to provide the foundation for Korea's vision of becoming a digital hub in Northeast Asia. The ministry emphasized greater cooperation in the Asia-Pacific region in the IT sector so that the information and communication networks of Asian countries can become interlinked and fully utilized. With the IT industry charging ahead on the export front, the ministry expected code division multiple access (CDMA), broadband Internet, system integration, and other key IT sectors to play a pivotal role in exports.

Since 2004, the government developed an IT '839' strategy - a master plan for IT industry. The IT '839' strategy encompasses the promotion of eight new services, including mobile broadband and digital multimedia broadcasting. This would trigger new investment in the building of three types of infrastructure: broadband convergence network, universal sensor network and Ipv6. The establishment of the infrastructure will contribute to creating demand for nine new IT products such as home networks and digital TV.

4.2 Supply Side

One of the most important factors behind the market penetration for Korea's broadband can be credited to low subscription prices resulting from intense domestic competition. Facilities based service providers (FSPs) have freely entered the market without entry and price regulations because broadband Internet service was classified as a value-added service, and have competed with each other head-to-head. In the end, most FSPs entered the market concurrently, setting flat-rate retail prices at a low level to induce long-hour dial-up users. The resulting laissez-faire facilities-based competition has led to the further lowering of prices. FSPs usually have a variety of forms of access for providing broadband Internet services: copper and fiber ADSL, cable modem, Local Area Network (LAN) Ethernet, Broadband-Wireless Local Loop, satellite, wireless Local Area Network (W-LAN) or T1/leased line connections.

One of the major factors supporting market growth was high density in urban areas in Korea. Approximately 70 percent of the country's citizens live in the seven largest cities. Nearly 48 percent of all households live in apartment complexes, where economies of scale work efficiently for FSP market operations. The issue of distance in ADSL does not apply since more than 90 percent of households are located within a radius of 2.2km around the communications switchboards by KT. This fact explains why, for broadband subscriptions, residential Internet use exceeds that of business use in Korea. This contrasts starkly with many sparsely populated Western countries, which suffer from so-called "last mile" problems of bringing Internet lines to every home.

4.3 Demand Side

The popularity of IP telephony may have played a role in the remarkable growth of broadband access in Korea. Serome Technology introduced its 'DialPad' service in Korea in January 2000. One of the major attractions of the DialPad service was that it offered free PC to PSTN phone calls. The IP telephony companies also offer heavily discounted PC to PSTN international calls. DialPad's PC-to-PC international calls were free. In December 2000, DialPad introduced video IP telephony as part of its service.

An increase in the demand for entertainment and network games has also stimulated the growth of broadband Internet services. According to a survey [8], 53.0 percent of Internet users went online for gaming, only second to web surfing (or information searches) at 80.3 percent. The ubiquitous broadband has set the stage for the rapid growth of the multi-user PC games on the web, providing a rich source of income for domestic as well as foreign game makers; NCSoft is an example.

The remarkable growth of Internet banking and e-commerce has positively influenced the diffusion of broadband Internet services in the demand side. As of December 2003, Internet banking services were provided by 21 financial institutions, including 18 domestic banks, Citibank, HSBC and the national post office. The continued boom in Internet business in Korea is also giving fresh hope to online shopping mall web sites. With broadband it is so much faster and convenient to browse through the sites, compare prices, download colorful images and pictures, and find what they want. It saves them a lot of time going from shop to shop.

The rapid increase in e-commerce represents the fact that shopping in cyberspace has become increasingly fashionable in Korea. The high Internet penetration rate coupled with the increasing usage of broadband facilities has likely spurred the trend. In addition, lower costs, coupled with a credible money-back guarantee, have been key ingredients to the success of these cybermalls.

In demand side, another major factor affecting Korea's successful broadband penetration was the advent of an IT friendly Korean culture such as virtual communities, blog booms or IT education enthusiasm. Virtual communities are a new form of communication whereby community members share information and knowledge for mutual learning or problem solving. When offline activities are not feasible because the members are geographically dispersed, adoption of multimedia support, such as video conferencing or PC camera chatting may provide the community with the effects of offline, face-to-face meetings.

However, high speeds in broadband services are needed for efficient usage of multimedia. In fact, there are several virtual community service providers in Korea such as Daum.net and Cyworld.com. These providers have ten million members each and include about one million virtual communities in total. This represents the stimulation of virtual communities in Korea. Virtual community members frequently interact with other members in cyberspace via broadband Internet services and even launch a task-oriented project team as a form of ad-hoc organization.

Furthermore, Korean parents are obsessed with the education of their children, investing competitively large portions of their income into private education. These days, IT training has been a matter of primary concern in Korea. The Ministry of Education and Human Resources Development has spent $1.3 billion on 30-inch televisions, computers and Internet access in 220,000 classrooms. The ministry also started a computer education program for teachers to learn how to use the computers more efficiently in their classes. Now, the computer and Internet have become indispensable for students to do their homework properly. Thus, most families with students subscribe to broadband Internet services.

5 Future Challenges

Recently, Korea has moved from DSL to Fiber-to-the home (FTTH) and Fiber-to-the-building (FTTB) subscriptions for the advanced nationwide high speed internet connections. In particular, KT is actively promoting the very high speed Fiber-to-the home (FTTH) service for early adopters of Korea citizens. This new technology's successful diffusion will be one of emerging challenges in a broadband leading country, Korea.

Furthermore, such a rapid new broadband penetration in Korea will provide new business opportunities for major telecommunication operators such as KT, LG Dacom, and SK Broadband. The continuing demands may provide the momentum for the Korean IT industry to move forward. Today, Korea is witnessing the emergence and proliferation of various Internet-based businesses like online games, animation, virtual communities, and online trading services to name a few. Furthermore, the next-generation network, called broadband convergence network (BcN), is now under a run. It integrates traditionally separated telecom, Internet and broadcasting lines into a single network, promising a speed of 100 Mbps. With a host of futuristic features like voice over Internet protocol, t-commerce, IP-TV, and video on demands (VOD), the MIC expects the BcN will become a mainstream connection technology to the Internet by 2010. Global IT companies view Korea as a "test bed" and R&D lab for global trends because the current broadband Internet trends in Korea will eventually spread across the globe, when broadband Internet access is more widely adopted.

Along with the wider spread of broadband Internet, adverse side effects such as hacking, computer viruses and worms, spam mails, privacy invasion, online crime, Internet addiction and other harmful web sites including pornographic and gambling sites are yet to be confronted. For example, virus reports stood at 38,677 cases in 2002, but the figure more than doubled to 85,023 in 2004 [10]. In addition, the accumulated reports on phishing attempts, or sending fraudulent e-mail aimed at confiscating recipients' financial information, stood at 352 instances from Jan to May of 2005. This already surpassed the 2004 total of 220. According to Sophos [20], 13.43 percent of all junk messages sent this year originated from Korea to earn the unenviable position of second worst, trailing the United States with 42.11 percent. This indicates that Korea's various anti-spam measures are not working. A survey also found up to 41.4 percent of the nation's middle and high school students showed various symptoms of Internet addiction in 2003 with 2.9 percent of them being seriously affected [8]. The study insisted that approximately 10 to 20 percent of high school students could be categorized as web junkies who need treatment even from a conservative perspective. This data tells us that, although Korea has the most advanced Internet network in the world, its cyber security is still fraught with problems. If these serious by-products of Korea's state-of-the-art broadband infrastructure remain unsolved, they could some day pose a real threat to the economy as well as society.

6 Conclusion, Suggestion, and Implication

In this paper, we investigated some key factors that influenced successful broadband penetration in Korea; (1) government support, (2) demand side, and (3) supply side. We also discussed future challenges related to the issues. In order to deal with given challenges, for the future directions, Korea should now prepare and provide well-organized IT education including cyber ethics for old netizens as well as young ages. Also, technical supports for safe online transactions as well as relevant policies for internet security are essential to promote the usage of high-speed broadband Internet service. Moreover social consensus on internet culture should be made. Such mature internet culture may play a role of self-purification against adverse side effects presented above.

The case of Korea's rapid broadband deployment provides several implications for policy makers in other countries. First, to accelerate the initial investment when rolling out broadband networks, government led policies for building an IT infrastructure are needed. Second, unrestricted competition between service providers, as well as between technologies for Internet access, is effective for improving the quality and lowering prices, paving the way for early adoption [14]. Finally, a diverse range of attractive application services should be developed to generate enough traffic for both carriers and content providers. An IT friendly culture also seems to be beneficial in the early adoption and rapid diffusion of new services and technologies.

Korea is indisputably an Internet powerhouse thanks to the nation's forefront infrastructure and technologically savvy citizens. Indeed, many Koreans spend many hours everyday on the Web, playing games, web surfing or chatting. Some critics even say they feel insecure when they are not connected. They are willing to spend real money in the virtual world and quickly accept new technologies, making Korea the global "test bed" for new Internet businesses. The success of Korea's broadband Internet industry holds great significance by encouraging the traditional voice service industry to take a leap forward, transforming itself into a new promising Internet and data-based business.

There could be other factors behind Korea's achievement in high-speed Internet, although they have not been mentioned here. We, however, selected and grouped the most significant factors into three categories: government support and coordination, supply side, and demand side. This gives a coherent and integrated perspective on Korea's broadband penetration. This analysis is expected to act as a blueprint for other countries making an effort to accelerate the deployment of high-speed Internet networks.

References

1. Baliamoune-Lutz, M.: An Analysis of the Determinants and Effects of ICT Diffusion in Developing Countries. Information Technology for Development 10(3), 151–169 (2003)
2. Becker, S.W., Whisler, T.L.: The Innovative Organisation: A Selective View of Current Theory and Research. Journal of Business 40(4), 462–469 (1967)
3. Daft, R.L.: A Dual-core Model of Organizational Innovation. Academy of Management Journal 21(2), 193–210 (1978)
4. Damanpour, F., Evan, W.M.: Organizational Innovation and Performance: The Problem of Organizational Lag. Administrative Science Quarterly 29, 392–409 (1984)
5. Edquist, C.: Systems of Innovation, Technologies, Institutions and Organizations, Pinter, London (1997)
6. Freeman, C.: Technology and Economic Performance: Lessons from Japan, Pinter, London (1987)
7. Freeman, C.: Continental, National and Sub-national Innovation Systems: Complementarity and Economic Growth. Research Policy 31(2), 192–211 (2002)
8. KADO, 2004 Field Survey on the Internet Use, Korea Agency for Digital Opportunity and Promotion (2004)
9. Kimberly, J.R., Evanisko, M.J.: Organizational Innovation: The Influence of Individual and Contextual Factors on Hospital Adoption of Technological and Administrative Innovations. Academy of Management Journal 24(4), 689–711 (1981)
10. KISA (2005), Monthly Trend and Analysis of Internet Incidents, Korea Information Security Agency (June 2005)

11. Lança, I.S.: Mudança Tecnológica e Economia – Crescimento, Competitividade e Indústria em Portugal, First Published, Celta Editora, Oeiras, Portugal (2001)
12. MIC, White Paper: Internet Korea 2004, Ministry of Information and Communication (2004)
13. MIC, Broadband Statistics: Ministry of Information and Communication (March 2005)
14. Moore, G.A.: Crossing the Chasm: Marketing and Selling Disruptive Products to Mainstream Customers. HarperCollins Publishers, Harper Business Essentials (2002)
15. OECD, Broadband Infrastructure Deployment: The Role of Government Assistance, Organization for Economic Cooperation and Development (2002)
16. OECD, OECD Broadband Statistics to June 2007, Organization for Economic Cooperation and Development (2007), http://www.oecd.org/document/60/0,3343,en_2649_34225_39574076_1_1_1_1,00.html
17. OECD, Managing, National Innovation Systems, Organization for Economic Cooperation and Development (1999)
18. Rogers, E.M.: Diffusion of Innovations, 3rd edn. The Free Press, New York (1983)
19. Rogers, E.M.: Diffusion of Innovations, 4th edn. The Free Press, New York (1995)
20. SOPHOS, Sophos Reveals Latest "dirty dozen" Spam Producing Countries (2005), http://www.sophos.com/pressoffice/pressrel/us/20050407dirtydozen.html
21. Suh, J.H.: Korea's Innovation System: Challenges and New Policy Agenda, Discussion Paper Maastricht: Institute for New Technologies (2004)
22. Thirtle, C.G., Ruttan, V.W.: The Role of Demand and Supply in the Generation and Diffusion of technical Change. Harwood Academic, Switzerland (1987)
23. Tiwari, R., Buse, S., Herstatt, C.: Innovation via Global Route: Proposing a Reference Model for Global Innovation Activities. In: Proceedings of the 2nd International Conference Management of Globally Distributed Work, Bangalore, India (2000)
24. Tornatzky, L.G., Fleischer, M.: The Process of Technological Innovation. Lexington Books, Lexington (1990)
25. Verworn, B., Herstatt, C.: Modelle des Innovationprozesses, Working Paper no. 6, Hamburg University of Technology (2000)
26. Zaltman, G., Duncan, R., Holbek, J.: Innovations and Organizations. Wiley and Sons, New York (1973)

Construction of Optimized Knowledge and New Knowledge Creation Support Tool

Akira Otsuki[1] and Kenichi Okada[2]

[1] Science and Engineering, Keio University Tokyo, Japan, and
Ochanomizu University Tokyo, Japan
cecil@ion.ocn.ne.jp
[2] Science and Engineering, Keio University Tokyo, Japan
okada@ics.keio.ac.jp

Abstract. In this study, a tool was constructed that supports the process of trying to organize knowledge newly created after tacit knowledge has been optimized by applying knowledge management strategies, 3C, a marketing mix, and various enumeration methods. The tool was verified by quantitative methods, user feedback studies, and evaluation through comparison with similar tools. Until now, though some theories regarding the organization of newly created knowledge have been advocated based on user feedback studies, the method of concretely applying such theories to real-world business circumstances has not been presented. In the current study, the tool was tested through use in an actual administrative project and proved to be more effective than an already-existing tool used for the organization of newly-created knowledge.

Keywords: Knowledge management, New knowledge creation support, SECI Model.

1 Introduction

A great deal of research has been conducted so far in the knowledge creation support field, but the amount of well-organized, scientifically-conducted, practically-applicable research has so far been insufficient. Reflecting these deficiencies and a desire for improvement, research in the knowledge creation support field is currently in a state of transition.

Nonaka [1] proposed a new knowledge creation support model organized by a knowledge conversion cycle between tacit knowledge and explicit knowledge. In essence, Nonaka has developed and applied the concept of the dimension of tacit knowledge from Polanyi's theory of tacit knowing. Nonaka's model has had a worldwide impact and advanced experimental studies. As such, it has relevance to efforts to improve the organization of newly created knowledge. For example, the concept of knowledge management is found in the knowledge conversion process of the "SECI Model." In this model, there are tacit and explicit types of knowledge, and new knowledge is created by converting it in ways that are mutually beneficial for the individual and the organization according to the systematic knowledge creation theory of Nonaka. The SECI Model process is illustrated in Figure 1.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 500–513, 2009.

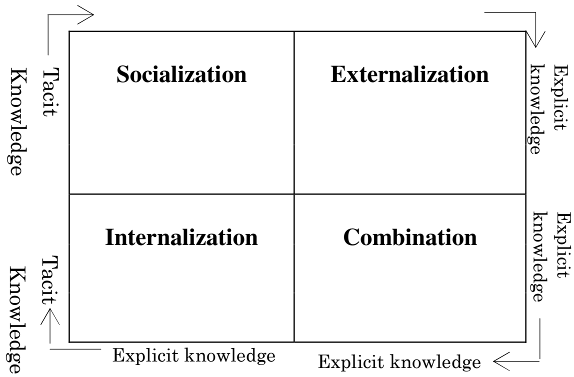


Fig. 1. SECI Model Source: The Knowledge Creating Company

This study proposes a new knowledge creation support tool developed through application of knowledge management and using the 3C's framework, a marketing mix (the 4P's), and the enumeration method. In addition, this tool was tested and proven effective through use in an actual online administrative procedural project, unlike previous efforts to apply support tools based on knowledge creation theory to business contexts. In addition, to verify effectiveness, a comparison with a similar tool woas undertaken. This paper's presentation of this tool is organized as follows.

Chapter 1	Introduction
Chapter 2	Related studies regarding knowledge management and new knowledge creation
Chapter 3	The purpose of this study
Chapter 4	The proposal technique
Chapter 5	The construction and functional overview of this tool
Chapter 6	Fixed quantity verification, effectiveness verification, and comparison verification with a similar tool
Chapter 7	Conclusions and future work

2 Related Studies

The concept known as "Memex," for connecting associated fragments of huge quantities of information, was advocated by the Los Alamos National Laboratory as a precursor tool for knowledge creation support provided by the present-day computer [2]. The concept of artificial intelligence (AI) – creating machines capable of processing information in a fashion similar to humans – was advocated at a conference held at Dartmouth College in 1956. Present day knowledge creation support tools derive from these two concepts. A number of related tools have been proposed so far.

The groupware that Takaya Yuizono [10-11] proposes is a system that supports the group concept of the KJ method on the computer. This system supports the accumulation, organization, and use of individual items of a group of data by means of "wadaman," a card-type multimedia database. On the other hand, the tool proposed in this paper aims at a further improvement of methods for coming up with fresh new ideas by offering an original means for "function relocation," "combining solution ~~pairs~~ alternatives," and achieving concrete results.

"TRIZ" is a methodology that organizes scientific and technological data from a technological perspective, and the "USIT" method, proposed by Nakagawa [9], breaks everything down and rearranges a variety of methods offered by TRIZ in a systematic, integrative fashion that expands its overall capabilities. "USIT" presents five solution generation methods: "making two or more objects from one," "attribute dimension changes,"~~,~~ "function relocation," "combining of solutions," and "solution generalization." However, the solution generation stages of the USIT method involve four kinds of techniques: the "attribute dimension method" (which focuses on the attributes of the object)," the "method of making two or more objects from one" (one object becomes or is divided into two or more)," the "function rearrangement method" (in which various functions are relocated between objects in the system), and the "function connection method" (in which functions are continuously connected). In addition, USIT has difficulty identifying ~~the~~ relationships among information spread over multiple paragraphs satisfactorily. Moreover, it isn't even able to define the nature of the relationships it identifies. Finally, "IMindMap"[16] is a technique (and tool) advocated by Tony Buzan for expressing knowledge by means of illustrations. This tool furthers conceptualization through the use of key words and images by putting the key words and the images most central to the concept to be expressed at the center of the figure and connecting related words and images radially. A complex concept can be compactly expressed and quickly understood using this method. However, this tool only shows the basic fact of a relationship between key words and images without indicating the nature of the relationship or providing detailed information regarding the key words. In addition, it is difficult to illustrate correspondences among multiple relationships satisfactorily, basically because of the layered nature of this tool's illustration expression technique.

3 Concept and Purpose of Study

Through our investigations, we attempted to form an overview of research conducted until now relating to creativity and systematic knowledge creation respectively. It is not possible to express here the multitude of relationships occurring among such a large amount of data. However, Table 1 illustrates an example of 3C and marketing mix used during the initial planning stage. In this example, the tacit knowledge of an organization's individuals is optimized.

In addition, a multifaceted relationship exists among a large amount of data (Figure 2). And the relation forms a network (Figure 3)

It is insufficient to only arrange a large amount of data as the basic material for creating new knowledge. It is important to be able to find meaning in the relationships among the data. Discovery of such meaning can be achieved with the tool developed in this study.

Table 1. Example of 3C and the marketing mix

	Extraction item	Extraction Example
	Proceedings theme	
	Meeting information	Location Date and Attendees
3C	**Customer**	Market size, Market growth, Needs, Purchasing process, Purchasing decision-makers
	Competitor	Number of competitors, Barriers to entry, Financial and other resources (employees, productive capacity), Performance (sales, amount a market share, profit, the number of the customers)
	Company	Sales, Market share, Profitability, Brand image, Technology, Organizational skill, Human resources
4P	**Product**	Products, Services, Quality, Design
	Pricing	Basic pricing, Discounts, Payment terms, Margin trading
	Place	Distribution & sales channels, Transportation, Market range, Sales locations, Quality maintenance – all locations, inventory
	Promotion	Sales promotions, advertising, direct marketing

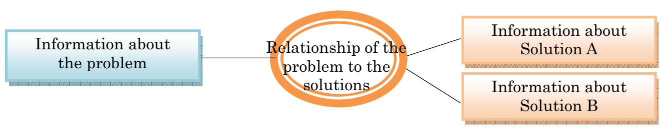


Fig. 2. Example of a multi-faceted relationship

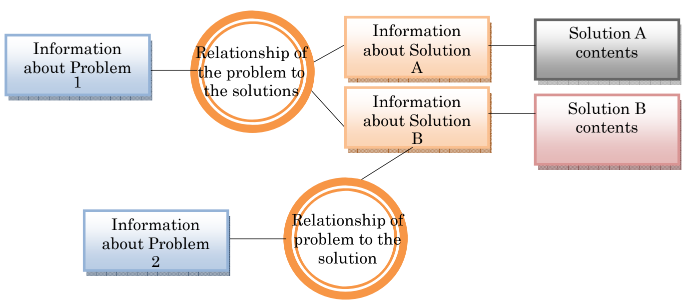


Fig. 3. The multifaceted relationships forms a network

- ○Expression of multi-faceted relationships showing meaningful connections among data in a data group
- ○Expression of relationships forming multi-path networks
- ○Expression of group and network dynamics while paying attention to particular relationships

Tacit knowledge is extracted by applying practical knowledge management theory involving 3C, the marketing mix, and the enumeration method.

4 Proposal Technique

4.1 Technique for Supporting Knowledge Expression

Knowledge that becomes the basis for new knowledge creation is expressed by using 3C analysis and the marketing mix, and the enumeration method for the target problem. In concrete terms, the theme of a new knowledge creation project is first set, and then tacit knowledge is optimized by using 3C analysis and the marketing mix (Table 1, Figure 4).

In addition, knowledge is created by applying the "fault enumeration method" or the "characteristic enumeration method" as indicated at (1) and (2), below, based on information on relevant items extracted by 3C analysis and the marketing mix (Figure5). Such knowledge is used as material when new knowledge is created by finding connections among the elements of the knowledge and displaying these connections in a list. The flow with respect to the problems of "Online administrative procedure" (described later) and each respective enumeration method is shown as follows.

(1) Problem characteristics are enumerated by the characteristic enumeration method

① Noun characteristics

EX） Something expressible by a noun related to the composition of the target

② Adjective characteristics

EX） Size, color, shape, and weight of the target

③ Functional characteristics

EX） Working functions or role in the target

(2) Target problems are enumerated by the fault enumeration method.

EX ○Enumeration of points of dissatisfaction
○Enumeration of problem factors

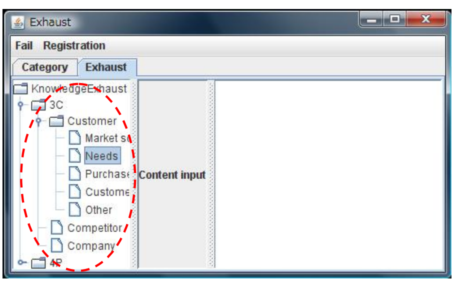


Fig. 4. 3C analysis and the marketing mix

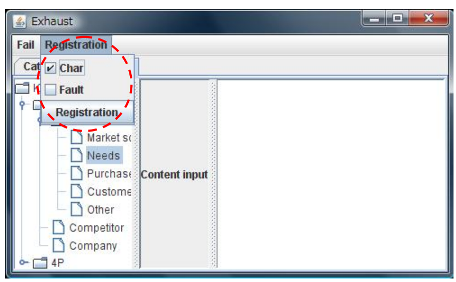


Fig. 5. Fault enumeration method and characteristic enumeration method

4.2 New Knowledge Creation

The knowledge expressed by 4.1 is displayed by using a figure showing multifaceted relationships among different components of knowledge (Figure 6). Following this, additional new knowledge is created based on this figure.

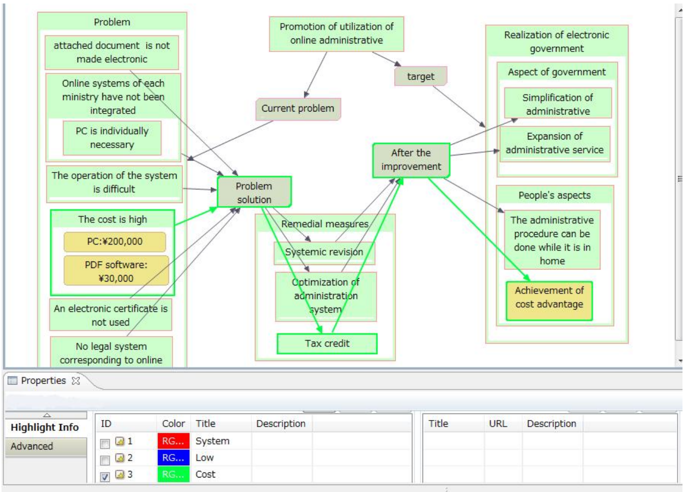


Fig. 6. Figure image

1) The Knowledge Is Organized by Nests

First, the expressed knowledge is organized by nests that group similar content. For example, figure 6 shows the problem of "online administrative procedure" in Japan. Specifically, Figure 6 has 3 main nodes: "problem to target", "remedial measures" and "target". These nodes play the role of parents of their respective nests, within which similar knowledge is grouped. Table 2 is the nest of "problem to target" in Figure 6. There are two child node hierarchies within the parent node of "Problem". This tool can be used to illustrate however many sub-nests are needed.

Table 2. Nest details

Parent	Child1	Child 2
Problem	Essential documents of the administrative procedure are not made electronically available.	
	Online systems of various ministries are not compatible.	Used of private PCs is necessary.
	Operation of the online software is difficult.	
	Costs are high.	・PC →About 200000 yen
		・Acrobat PDF→About 30000 Yen
	Electronic certificates are not widespread.	
	The legal system doesn't accommodate online administrative procedure.	

2) Relationships Are Expressed- in a Meaningful Way

Next, arbitrarily arranged node groups are related in a meaningful way. "Target" and "Problem" are connected by the relational node "Current problem" in <Figure 6>. In addition, "Problem" and "Remedial measures" are connected by the relational node "Problem solutions". Finally, "Remedial measures" and "Realization of electronic government” are connected by relational node “After the improvement”. Moreover, relational nodes can also be used to show relationships among child nodes. In this way, this tool can be used to express multifaceted relationships among knowledge components.

3) Expression of Relationships Forming Multi-path Networks

Next, relationships forming multi-path networks are expressed. In iMindMap (2. Related studies), the target is put in the center, then, knowledge is added to the target in radial fashion. In essence, knowledge is added like the branches of a tree in a unidirectional flow. On the other hand, this same tool can express the multi-path networks as well (Figure 6).

4) Combined Means of Expression that Pays Attention to Particular Relationships and New Knowledge Creation

Finally, new knowledge is created by reference to figure 6. Figure 6 is an example of a current problem in which remedial measures undertaken to move closer to the target and interrelationships among these measures are illustrated. Following this, additional new knowledge is generated based on this figure. In addition, this tool can highlight particular relationships. Figure 6, for example, highlights the node concerning “cost”.

5 Outline of Systems

There are two systems: the first is the knowledge optimization system; the second is the new knowledge creation support tool. The outline of the knowledge optimization system is described as follows. Knowledge is expressed by 3C using the marketing

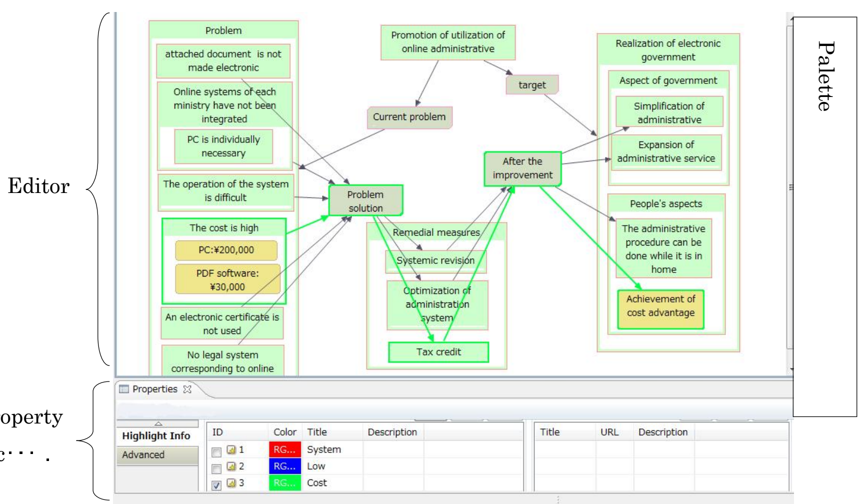
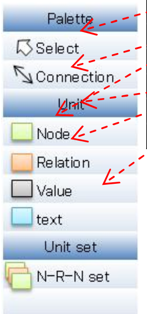


Fig. 7. New knowledge creation support tool image



Select	**The node is selected**
Connection	**The line is drawn between nodes**
Node	**Types of information**
Relation	**Relational node expressing relationships between units**
Value	**Node expressing concrete value and information**
Unit set	**Set of "Nodes", "Relationships", and "Nodes"**

Fig. 8. Palette Image

mix and the enumeration method. This is accomplished by using Java 6.0 and MySQL 5.0. Then the knowledge accumulated using this system is displayed. New knowledge is then created with this tool by reference to the already-expressed knowledge (Figure4, 5). Next, new knowledge creation support tool is constructed by enhancing Shapes of Eclipse GEF using a Windows-based platform. The screen of this tool is composed of the editor, the property, and the palette (Figures 7, 8). Next, this tool supports new knowledge creation by allowing the user to freely draw from the palette to edit the contents of units and unit sets, etc. In addition, this tool has a function allowing emphatic expression of particular relationships. Figure7 shows an example of emphatic expression of relationships concerning cost.

6 Evaluation Experiment

6.1 Outline of Evaluation Experiment

This tool was proven by the method described at 6.1.1 and 6.1.2, below, and by a comparison verification conducted with a similar tool (Free Mind). The verification results are described from6.2. The purpose of the evaluation experiment was to:

○ Determine the degree of improvement in knowledge optimization
○ Determine the degree of quantitative improvement in knowledge produced by new knowledge creation
○ Examine the effectiveness of this tool through a user questionnaire
○ Examine the degree to which relationships that enhance meaning can be expressed
○ Examine the degree to which relationships that extend multi-path networks can be expressed
○ Examine the degree to which particular relationships can be expressed.

6.1.1 Experiment Participants

In this study, experiment participants consisted of a group composed of a composition of knowledge practitioner (KP), a knowledge officer (KO) and a knowledge engineer

(KE). The KP acquires experiential knowledge, the KE promotes the knowledge conversion, and the KO decides the directionality of the organization.

6.1.2 Setting of Proof Problem

The problem that proved this tool was set as follows, 【Problem solutions】, 【New projects】 and 【New system】. To pursue a proof, a conference was held for several days by the participants mentioned at 6.1.1 above.

【Problem solutions】 Examination of use promotion plan of online administrative procedure

Implementation of online administrative procedures is one of the top priorities of the "E-Japan Priority Policy Program "with a goal of making administrative applications and written reports previously submitted using paper documents submit table over the Internet. According to the "IT new reform strategy" promulgated by the IT strategy headquarters on January 19, 2006, online availability of procedures involving documents such as applications and written reports to national and local authorities is targeted for 50% availability by 2010. The solution for this problem was proven using this tool.

【New projects】 Proposal of new projects

The problem of new project proposals for improving market flexibility was set as a theme. The solution for this problem was proven using this tool.

【New system】 Proposal of new system

It was assumed that a certain financial company introduced a new system as an original lending requirement countermeasure. The solution for this problem was proven using this tool.

6.2 Quantity Verification

6.2.1 Comparison Verifications of Degree of Knowledge Optimization and of New Knowledge Creations

Figures 9-11 graphically illustrate the results of quantitatively comparing this tool and FreeMind for the new knowledge creation "Problem solutions", "New projects", and the "Introductions of a new system", respectively. FreeMind is a free tool similar to "iMindMap", mentioned in the " Related studies" above, at section 2. First, knowledge optimization numbers were compared. The numbers of "Problem solutions", "New projects", and "Introductions of a new system" extractions in this tool were 58 (Figure9), 47 (Figure10), and 66 (Figure11), respectively. On the other hand, the numbers for FreeMind were 26 (Figure9), 18 (Figure10), and 32 (Figure11). Next, the numbers of new ideas created based on optimized knowledge were compared. The number of "Problem solutions", "New projects", and "Introductions of new system" extractions in this tool were 35 (Figure9), 50 (Figure10), and 60 (Figure11) respectively. On the other hand, the numbers for FreeMind were 13 (Figure9), 20 (Figure10), and 26 (Figure11). As mentioned above, the numbers for this tool were superior to those of Freemind.

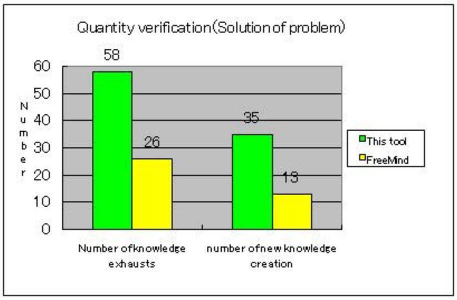


Fig. 9. Quantity graph (Problem solutions)

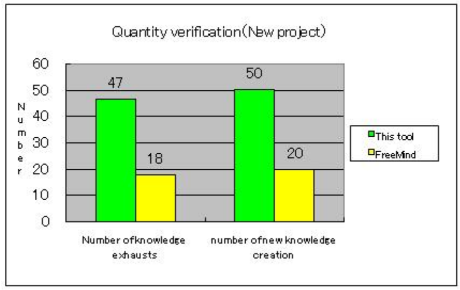


Fig. 10. Quantity graph (New projects)

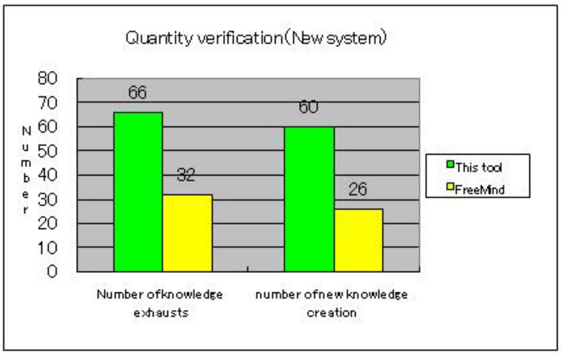


Fig. 11. Quantity graph (Introductions of a new system)

6.2.2 Correlational Analysis of Expressed Knowledge Information and New Knowledge Creation

Next, the analysis of expressed knowledge information and new knowledge creation are described. Figure12 illustrates a correlation analysis of this tool. Figure13 illustrates a correlation analysis of Free Mind. The X axis is the number of expressions of already-accumulated newly created of the "Problem solutions", "New projects", and "Introductions of a new system". The Y axis is the number of expressions of new knowledge creation of the "Problem solutions", "New projects", and "Introductions of a new system".

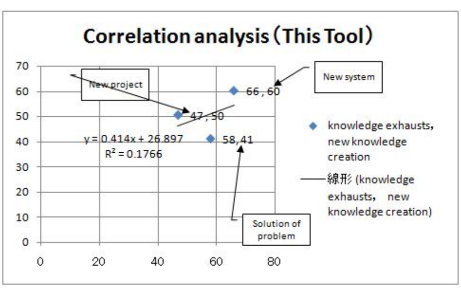


Fig. 12. Correlation analysis of this tool

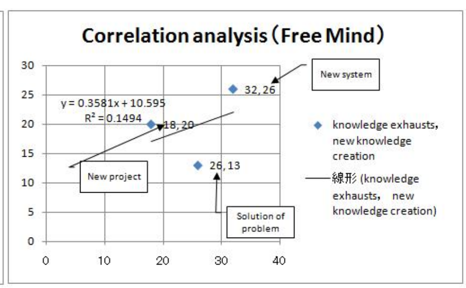


Fig. 13. Correlation analysis of Free Mind

Using available data, an XY plot (scatter chart) was generated. Following this, a linear approximation was expressed as an approximation curve, and the R-2 power value was also expressed. An R-2 value close to one indicates a high correlation. Comparing figures13 and 14, the R-2 value of this tool (R-2 = 0.1766) is higher than that of Free Mind (R-2 = 0.1494). Therefore, this tool produces a higher correlation of already-developed knowledge and new ideas than Free Mind. It can be said that this tool more effectively utilizes implicit knowledge to produce new knowledge than Free Mind.

6.2.3 Result of the Questionnaire

Table3 expresses the results of a user questionnaire (involving eight respondents) comparing the advantages and faults of this tool with Free Mind. Answers to the questionnaire confirmed that clarifying the target (EX: 3C or Marketing mix) contributed to substantially greater knowledge optimization than the degree of knowledge optimization achieved without clarification. In addition, this tool's ease and effectiveness at creating new knowledge through reference to the entire image used to illustrate a problem was also confirmed.

Table 3. Results of the questionnaire

Number of answers	Answers
【Advantages of this tool】	
6	Ease of knowledge optimization due to target clarification (EX: 3C or Marketing mix).
1	Ease of new knowledge creation due to ability to extract details optimized by the enumeration method.
1	No responses.
【Faults of this tool】	
2	It was not bound by 3C, P and 4 enumeration methods items, and it was possible to conceive it freely.
1	3C and marketing mix did not correspond satisfactorily in this project because it was the one chiefly using it when planning.
5	No responses.

6.3 Comparison Verification with Similar Tool

The comparison verification was conducted using the knowledge optimized by 3C, 4P and the enumeration method with this tool and Free Mind. The comparison results are described in (1) - (3), below.

(1) Comparison of Expressions with Multifaceted Relationships Where Meaning was Given

Table 4 shows an extraction of a single relationship (law-related group) from figure 14. It is a 3:1 relationship to which the solution and the problem are connected by the relationship of "Problem solving". Therefore, this tool is flexibly expressible of the multi paragraph relation. Free Mind can do similar (Fig17). But Free Mind is not expressible the relation that gives the meaning. As for this tool, it is expressible. By this, Searching out the relation that pays attention to a certain specific relation even when a lot of relations exist together becomes possible.

Table 4. Multifaceted relationship of Figure 14

The left side （Problem）	Relationship	The right side (Problem Solution）
Essential documents of the administrative procedure are not made electronically available.	Problem solution	Simplification of administrative procedures
Electronic certificates are not widespread.		
The legal system does not accommodate online applications.		

(2) The Flow of Each Target Is Managed

Figure 14 highlights legal system relationships in an online application. A highlight of the group that similarly focuses on "System relationships" is shown in Figure 15. The

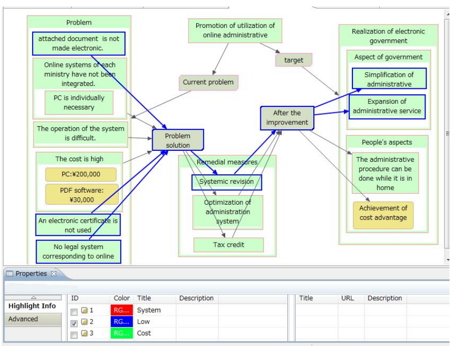


Fig. 14. Highlight example (Law)

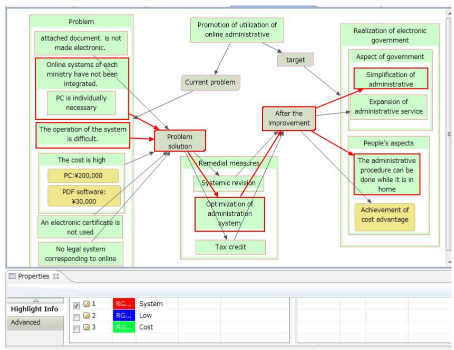


Fig. 15. Highlight example (System)

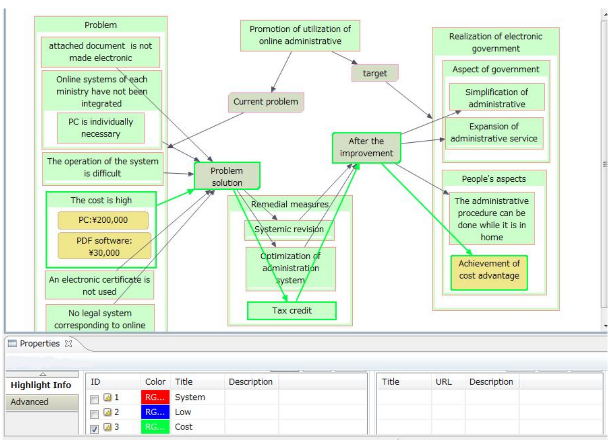


Fig. 16. Free Mind Example (Cost)

group that focuses on "Cost relationships" is shown in Figure 16. These figures illustrate this tool's capability of expressing process flows appropriate for individual targets through the construction of highlighted groups that pays attention to individual relationships. On the other hand, Figure 18 illustrates "System relationships", "Legal system relationships", and "Cost relationships" as expressed using Free Mind. As can be seen, with Free Mind inter-nodal connections are not easily understood. In addition, Free Mind cannot readily manage relationship-focused flows.

(3) Point of Superiority of Free Mind to This Tool

The target problem is clearly show because Free Mind positions the problem at the center of the figure and Knowledge is optimized in relationship to the target in radial fashion (Figures 17, 18).

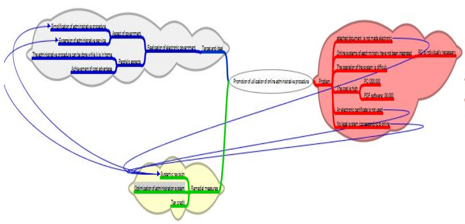


Fig. 17. Free Mind Example (Law)

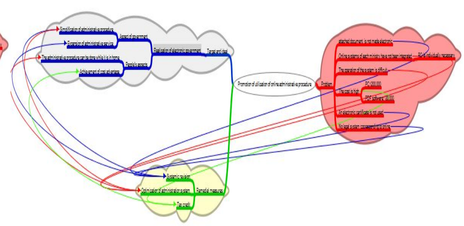


Fig. 18. Free Mind Example (Law, System and Cost)

7 Conclusion

In this study, a tool was constructed that supports organization of newly-created knowledge after tacit knowledge has been optimized by knowledge management, with application of 3C and a marketing mix and various enumeration methods. The tool was verified through quantitative analysis, a user study, and evaluative comparison with a similar tool. In the past, though some theories were advocated based on user studies of knowledge creation, a method of concretely applying theory to real-world business practice was not presented. In this study, therefore, this tool was proven by application to an administrative project. As a result, it was proven to be more effective, overall, than an existing tool and to offer a superior conceptual framework for supporting new knowledge creation.

Future research, including applications of this tool to other real-world projects, can provide additional verification of this tool's effectiveness, solve any problems, and further improve its usefulness.

References

[1] Nonaka, I., Takeuchi, H.: The Knowledge Creating Company. Japanese companies remain an enigma to most Westerners (1996)

[2] Bush, V.: As We May Think. The Atlantic Monthly (1945)

[3] Takahiko, N.: Knowledge Management and CSCW. IPSJ SIG Notes 2000(63) (20000711), 7–12, 2000-IM-37-2
[4] Hideo, U.: Trino: A Knowledge Work Support System. IPSJ SIG Notes 2005(30) (20050317), 1–6, 2005-GN-55-(1)
[5] Noguchi: Conception method, Kodansha (2000)
[6] Lifelong Human Resources Development Center Website, http://www.ab-garden.ehdo.go.jp/index.html
[7] Otsuki, A.: A study about an online administrative procedure. Spread promotion model, GITI Research Bulletin 2006-2007, pp. 198–208
[8] Hirose, M.: The study on accelerations of knowledge creation, IPSJ SIG Notes 2001(33) (20010323), 25-31, 2001-IM-39-5
[9] Nakagawa, T.: Problem Solving Methodology for Innovation: TRIZ/USIT. JCSP thesis magazine 8, 49–66 (2004)
[10] Takaya, Y.: GUNGEN: Groupware for new idea generation support system. IPSJ SIG Notes 94(87) (19941013), 37–42
[11] Takaya, Y.: Implementation of GUNGEN and Its Estimation. IPSJ SIG Notes 97(35) (19970424), 45–50
[12] Kitada, J.: Conception method (1967), succession, inside public new book (1970)
[13] Masanobu, K.: Implementation and Evaluation of KJ-Editor: A Card-handling Tool. Computer software 9(5) (19920917), 416–431
[14] Omi: Card operation tool for multimedia information. Electronic telecommunication society thesis magazine, D-II 79(4), 577–584 (1996)
[15] Ohiwa, H., Takeda, N., Kawai, K., Shimomi, A.: KJ editor: a card-handling tool for creative work support. Knowledge-Based Systems 10, 43–50 (1997)
[16] Buzan, T.: iMindMap, http://www.imindmap.com/
[17] Otsuki, A.: Knowledge Information Extraction from Discussion Report when Plan of Organization Strategy. IPSJ-GN 2008(31), 85–90
[18] Drucker, P.F.: Age of discontinuity — —Plan of coming knowledge society. Diamond Co. (1969)

Agent Verification Design of Short Text Messaging System Using Formal Method

Ali Selamat[1], Siti Dianah Abdul Bujang[2], and Md. Hafiz Selamat[3]

[1,2,3] Intelligent Software System Research Laboratory, Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia
aselamat@utm.my, anarel_03@yahoo.com, mhafiz@utm.my

Abstract. Today, in the evolution of hardware and software system in mobile-driven applications, failure is unacceptable to many entrepreneurs. The use of Short Message Service (SMS) for internet-based business is acknowledged to be one of the applications that have a potential to give unreliable results to the information systems. Once the workload is optimized, the network traffic becomes congested, which minimizes the performance of the system. Therefore, to develop a dependable SMS system, we need a reliable verification method. Besides the agent-based verification method, formal method verification on the design is also needed. This is to ensure that the system design is secured from the unseen errors and bugs in the early stage of development. In order to make this successful, this paper will discuss more on the verification design of short-text messaging system with the implementation of an agent-based system known as model checking. The aim of this paper is to provide a reliable and verifiable system designs, using Spin as the model checker.

1 Introduction

The rapid developments in mobile-driven applications almost offer considerable economic potential for the industry, but there are also accompanied by a broad range of vulnerability risk. The prominent services in the current market for instant, Short Message Service (SMS) has increased the performance of the conventional application. Thus, there are many entrepreneurs interested in converting the conventional application into mobile SMS application for better performance [1],[2] and [3]. The SMS or short-text messaging system has alerted businesses of its convenience for business communication, because of its cost-saving and user-friendly feature. In fact, latest statistic shows that the number of users can increase up to billions of users per month [4]. This tells us that the SMS technology is supporting users' needs. However, there are several cases of mobile systems that have faced the vulnerability risk in the operations especially on computing processes, which reduce user satisfaction. The problems occurred when many entrepreneurs became victims of the operational processes due to poor security, poor data integrity control and inadequate management control once they used

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 514–522, 2009.

the text messaging system. Moreover, Chaki et al. claimed that in the information society, the current critical issue is finding the correct of computer software [5]. The errors in complex software systems have caused large-scale economic losses in the past. It is difficult to detect and fix the software bug in multi-threaded systems. Problems can occur in systems when many components are not able to interact with each other. This problem reduces user satisfaction and increases the maintenance cost. In order to prove the system is developed in a right way, it has to have a back-end design and specifications. Formal analysis is needed in order to overcome the issues that arose. In this paper, we show the verification design of short text messaging system using formal method. We have designed the process of verification handle by our model checking agent and we test it using Spin model checker approach. The remainder of this paper is organized as follows: We present the related works in Section 2. Sections 3 showed the modeling design using model-checking agent. Section 4 shows model checker Spin verifies the Promela model. We conclude our paper in Section 5.

2 Related Works

2.1 Formal Method

Formal methods are mathematical based technique that is used to represent the specification of a software complex system. This technique has become one of the significant ways, which are capable of enhancing the correctness and complexity in identifying errors or predicting the incidence during the whole system design process [6]. Formal methods cover a wide range of mathematically-derived system, designed for verification and validation. Currently, the evolving technology of formal methods in the market is model checking. This is one of the verification methods, which are based on the static analysis that is able to analyze systematically and exhaustively [7]. The interested areas in which most researchers are exploring formal methods include fault tolerance, response time, space efficiency, reliability, human factors, and software structure dependencies.

2.2 Software Model Checking

Model checking is a promising mathematical method for automated correctness verification of safety-critical reactive system [8]. This method makes it possible to prove the correctness of systems using temporal logics to specify system properties. It has better performance, and not just a process of simulations, testings and deductive reasonings. It has been successfully applied in verifying complex systems and communication protocols [9]. Furthermore, model checking provides an effective and efficient evaluation with counter-example to verify the correctness of complex system software [10]. This method effectively represents systematic, exhaustive, mathematical and infinite models for tracing errors. In order to conduct the model checking verification, it requires an automata model to demonstrate the sequence of system processes from one state to another under the action of transition [11]. Fig. 1 shows the methodology of verification model checking [12].

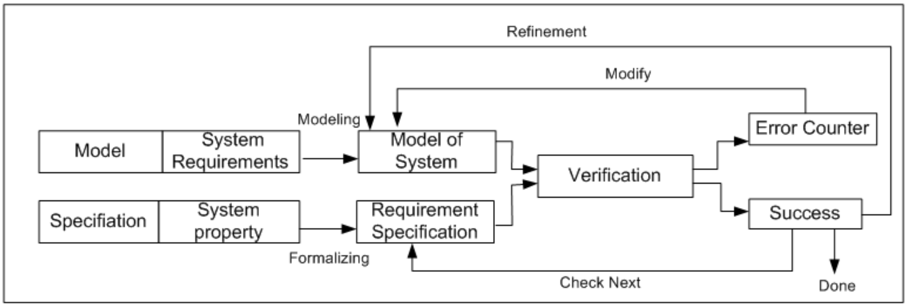


Fig. 1. The methodology of model checking

In verification using model checking, there are several model checkers that can be used for testing the model. For instance, Spin (Simple Promela Intepreter) (http://spinroot.com/spin/whatispin.html) can be written using Promela language and is representative of LTL model checker for communication protocol or concurrent software. Spin provides safety and aliveness properties such as deadlock, invalid end state and non-progress cycle [13]. In fact, Spin is highly recommended for automization especially in many fields such as security protocol verification, control system verification, software verification and optimization schedule [14]. Others, [10] have identified that Verisoft (http://cm.bell-labs.com/who/god/verisoft) written in C and C++ can test hundreds of thousands of input combinations in a short of time and analyze a distributed system.

3 Modeling Agent

Our case study is an SMS Management System, enhanced with an agent-based platform in order to handle the verification issues. We propose a Model Checking Agent (MCa) with other supporting agents including Input Agent (Ia), Output Agent (Oa) and Refine Agent (Ra). The MCa acts as a manager that verifies the incoming message. The message is verified based on the text format and syntax. The verification design of our MCa will then be tested using model checker SPIN. The verification process of text messaging system is showed in Fig. 2. The roles of the agents are explained as follows:

- **Input Agent (Ia)**
  Input Agent is a user interface used in mobile phones as a medium for the interactions between users and the SMS management system. The Ia will receive the request from users and assist the MCa to get the input from them.
- **Model Checking Agent (MCa)**
  The Model Checking Agent's role is as a manager of the whole system. It checks and verifies that the SMS request satisfies the system requirement.

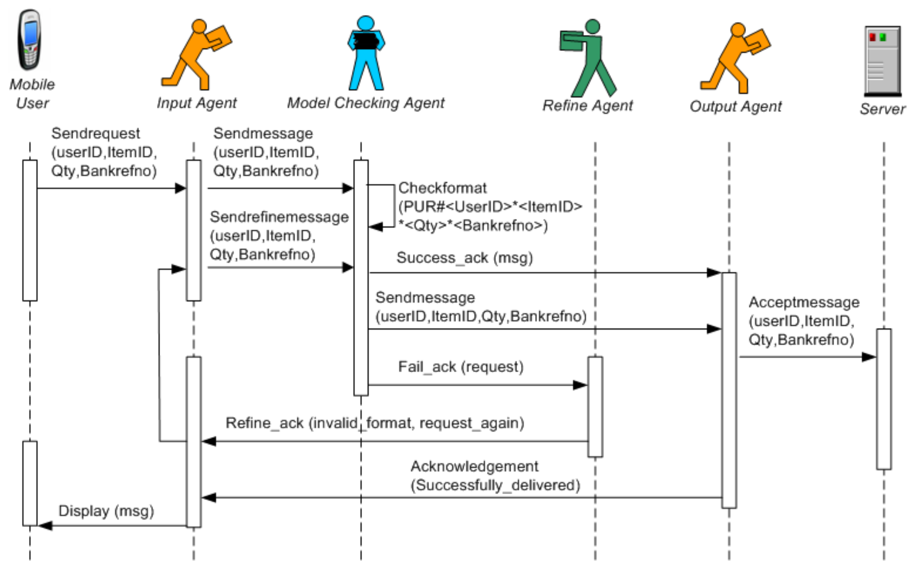


Fig. 2. The verification process of text messaging system

The requirement is verified based on the properties that we mentioned before. This process is done in order to ensure that the system gets a correct and valid message.

– **Output Agent (Oa)**
  Output Agent will be reacted if the MCa successfully verifies the message. Then the Oa will accept the message and send it to the system server. The status of the user's request will automatically be sent in back propagation to the user for acknowledgment.
– **Refine Agent (Ra)**
  However, if the message is identified as failure, the refine agent or Ra will send a 'fail message'. It means that the user has to send back their request again for new process. Otherwise, the message will be considered as violated and will be terminated automatically.

Based on Fig. 2, the protocol for the verification process is as follows:

1. User starts sending request to Ia. Ia represented as an interface that will send message from the user to the MCa.The process continues until it receives an acknowledgement from Oa if the message is identified as a success.
2. Before the message is receive by Server, the MCa will verify every message that come into a system. The verification is done based on the text format and syntax checking.
3. The MCa checks the message and makes the comparison between the message and the system requirement.

4. Server does nothing until it receives the valid message. If the message is identified as an error in syntax and format, the Ra will handle the message and notify Ia to request again. Otherwise, the process is automatically terminated.

In the verification design, the MCA is responsible for checking and verifying the text format as explained details in [15]. We want the user request (User ID, Item ID, Quantity, Bank Reference Number) to satisfy the given specification of the desired system. The rejected or accepted message of the user will follow the rules in Fig. 3:

*:: **Begin** {*
If transaction trace fail to text format = Stop;
THEN
Refine agent sends acknowledgement to Ia;
Ia ask User to send message again;
Request rejected;
STOP;
OR
If transaction trace success to text format = Stop;
THEN
MCA sends message to Oa;
Oa sends acknowledgement to Ia;
Server accept the request;
STOP;
Requirement is triggered;
*}:: **End***

Fig. 3. The rules of verification model

For verification, we have designed the agents to communicate based on the protocol. The properties of the linear temporal logic (LTL) expression are as follows:

1. $\forall(\neg(\Box sendmsg(msg) \rightarrow \Diamond sendmsg(u, i, qty, b)))$
For all, if there is no user request then there is no message sending. The path is not accepted.
2. $\Box(verifytextmsg(u, i, qty, b) \wedge sucessmsg(u, i, qty, b)) \rightarrow (\Diamond sendmsg(u, i, qty, b) \wedge ack())$
If the message is successfully verified, then eventually the sending message is accepted in the system and the acknowledgement is sent.
3. $\Box((sendmsg(u, i, qty, b) U refinemsg()))$ The message will not be accepted for the system until the refinement of the message process is done.

4 Verifying Modeling Process Using SPIN

In this section, we showed the result of verifying the model using Spin as showed in Fig. 4.

```
Process Statement          MCa       Oa
0 User  20  timeout        [msg1]    [ack1]
0 User  24  1              [msg1]    [ack1]
timeout
0 User  20  timeout        [msg1]    [ack1]
0 User  24  1              [msg1]    [ack1]
timeout
0 User  20  timeout        [msg1]    [ack1]
0 User  24  1              [msg1]    [ack1]
timeout
0 User  20  timeout        [msg1]    [ack1]
0 User  24  1              [msg1]    [ack1]
timeout
0 User  20  timeout        [msg1]    [ack1]
0 User  24  1              [msg1]    [ack1]
-------------
depth-limit (-u250 steps) reached
#processes: 2
1 Serve                    [msg1]    [ack1]
0 User                     [msg1]    [ack1]
2 processes created
```

Fig. 4. The verification output of verification model

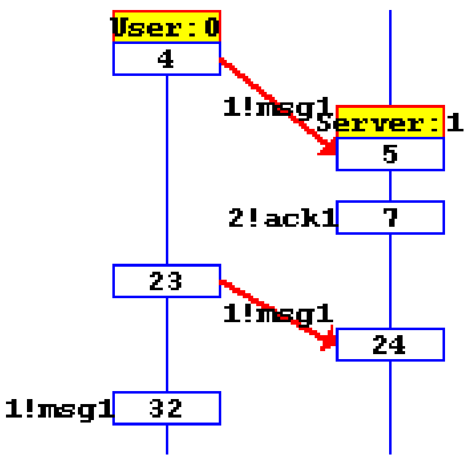


Fig. 5. Message sequence chart for verification model from XSpin simulation

```
Full statespace search for:

        never claim          - (none specified)
        assertion violations        +
        cycle checks         - (disabled by - DSAFETY)
        invalid end states  +
State-vector 36 byte, depth reached 5,  errors: 1
      6 states, stored
      0 states, matched
      6 transitions (= stored+matched)
      0 atomic steps
hash conflicts: 0 (resolved)
2.302            memory usage (Mbyte)
```

Fig. 6. Result obtained from the Spin tool

When the search depth reached state 32, the model checker Spin has successfully found an attack to the verification model. The attack needs to verify the three properties as follows: never-claim, assertion violations and acceptance cycles. The result is simulated by XSpin and has been exported in Fig. 5.

This paper has showed the verification design for the case study by using Promela. The verification involved the SMS sending protocol and the specification of the properties based on the case study. From the result in Fig. 6, we can determine that the text format checking for the properties used 2.302 Mbyte of the computer memory. Although the result is identified as positive for assertion and violation in the state of the system, Spin has showed there is an error in some state for refinement in the counter-example. The success of the modeling that we designed will be based on the formal analysis from Spin which can improve user satisfaction and is cost-effective for maintenance.

5 Conclusions

This paper presents a way to identify the problems that can cause an unreliable system performance by using the model checking agent approach. It is important for the system to be designed accurately for system safety and not affect to the quality of service in telecommunication industry (QoS). It presents a mechanism to specify and verify the SMS properties of syntax and format. The use of model checker Spin shows the accurateness of the design, which provided a counter-example that define sequences of processes that may lead to failure. However, the process is limited due to the errors found in the Spin. We try to improve the process of experiment by making some adjustment of the specification. The performance analysis using Spin has given an effective testing tool to identify software bugs and errors at the early stage of the software development. Also, the proposed agent-based model checking approach is able to be one of the new solutions in overcoming the verification issues especially in SMS. As in the future works, this paper will enhance the results with two properties for SMS quality checking.

Acknowledgement

This work is supported by the Ministry of Science & Technology and Innovation (MOSTI), Malaysia and Research Management Center, Universiti Teknologi Malaysia (UTM) under the Vot 79200 and 79267.

References

1. Valentino, C.A.: Progress Monitoring for the Field Operations of the 2007 Philipine Cencus of Population, National Statistics Office, Philipines, December 10-12 (2007)
2. Mee, V.N., Selamat, A.: SMS Management System for Direct Sales and Network Marketing. In: Fifth International Conference on Information Technology in Asia (CITA 2007) (2007)
3. Shengguang, M., Wei, C., Gang, L., Shitong, W., Liu, W.: An Asset management System based on RFID, WebGIS and SMS. In: 2nd International Conference on Ubiquitous Information Management & Communication (ICUIMC 2008) (2008)
4. Messaging (2009), http://www.gsmworld.com/gsmdata
5. Chaki, S., Clarke, E., Sharygina, N., Sinha, N.: Verification of Evolving Software via Component, Substitutability Analysis. Springer Science+Business Media, LLC (2008)
6. Bollig, B.: Formal Models of Communicating Systems. Springer Science+Business Media (2006)
7. Eugenio, D.S., Francesco, M.D., Marina, M., Giacomo, P.: Web applications design and maintenance using Symbolic Model Checking. In: Proceedings of the Seventh European Conference on Software Maintenance and Reeingineering (CSMR 2003). IEEE, Los Alamitos (2003)
8. Yu, Y., Li, X.: A workflow model with temporal logic constrains and its automated verification. In: The Sixth International Conference on Grid and Cooperative Computing (GCC 2007). IEEE, Los Alamitos (2007)
9. Clarke Jr., E.M., Grumberg, O., Peled, A.D.: Model Checking. MIT Press, Cambridge (1999)
10. Anderson, B.B., Hansen, J.V., Lowry, P.B., Summers, S.L.: Model Checking for design and assurance of e-Business processes. Decision Support Systems 39, 333–344 (2005)
11. Berard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L.: Systems and Software Verification Model Checking Techniques and Tools. Springer, Heidelberg (2001)
12. Katoen, J.P.: Concept, Algorithms and Tools for Model Checking. Lecture Notes of Course, vol. 32, pp. 31–36 (1999)

13. Bang Ki, S., Jin Hyun, W., Yoo, C., Choi Jin, Y.: System Resource Utilization Analysis based on Model Checking Method. An International Journal of Computing and Informatics (Informatica) (2005)
14. Xiao, M., Li, J.: The Modeling Analysis of Cryptographic Protocols using Promela. In: Proceeding of the 6th World Congress on Intelligent Control and Automation. IEEE, Los Alamitos (2006)
15. Bujang, A.S.D., Selamat, A.: Verification of Mobile SMS Application with Model Checking Agent. In: Eighth International Conference on Intelligent Systems Design and Applications(ISDA 2008). IEEE, Los Alamitos (2008)

A Study on the Effects of Online Brand Community Identity on the Characteristics of Community Activity and Behavioral Responses

Hyung-Shik Jung[1], Youngshim Kim[2], and Yu-Rim Kook[3]

[1] Department of Business Administration, Chosun University, Gwangju, Korea
hsjung@chosun.ac.kr
[2] Department of Business Administration, Chosun University, Gwangju, Korea
kys4779@chosun.ac.kr
[3] Graduate School, Chosun University, Gwangju, Korea
cuterimi@hanmail.net

Abstract. The present study investigated the effects of online brand community identity on the characteristics of community activity and behavioral responses. Based on empirical research using questionnaires with 117 consumers who were participating in an online brand community, the study found the following results: First, online brand community identity affected normative community pressure and community bond, and the normative community pressure had a negative effect on the user's intention to maintain participation in the community. Second, online brand community bond did not have a statically significant effect on normative community pressure. Lastly, online brand community bond affected community commitment, which, in turn, affected the user's intention to maintain participation, collaborative brand production, and brand word-of-mouth activities. The results of the study shed light on the importance of exerting efforts to manage consumer relationships through online brand community, particularly for the industry where online brand community leads to brand purchasing behavioral responses of their consumer.

Keywords: Online Brand Community, Brand Community Identification, Community Characteristics, Behavioral Responses, Normative Community Pressure, Community Engagement, Community Commitment, Community Participation Continuance Intentions, Brand Co-production, Brand Word-of-Mouth.

1 Introduction

Unlike off-line communities, on-line communities play a variety of roles for the members, such as providing easy access to information exchange and community participation unrestricted by time or space, as well as acquisition of useful information without extra pay[8,13,14,17]. Aside from these characteristics, since the virtual social structures of the online communities bear close similarities to those of the real world, their importance has received a serious attention from a number of researchers.

In addition, the related industries, also recognizing the significance of online communities, have actively utilized them for brand marketing activities as a means to

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 523–533, 2009.

maintaining close relationships with consumers. Advanced from the Industry-initiated brand websites whose major functions are brand descriptions and product sales, online communities exert efforts to meet the consumers' interests by offering product - experiencing opportunities, sponsoring community activities, performances, and visits to foreign countries, and by forming and developing bond among the community members who use the company's products[11]. In addition, they offer the opportunities to increase the consumer influence by incorporating their feedback on the existing products and ideas for a new product into their product development [1,9].

Brand community is becoming a space where the members can enjoy themselves, create new relationships, and exchange feedback on products by pursuing two-way information exchange between the consumers and the industry, rather than the one-sided relationship of simply acquiring product information[6]. However, despite its growing significance until recently, most research on online brand community has been mostly on the influence of the characteristics of online community on brand royalty or brand asset[4,11,18]. The use behavior of online brand community presses for strong sense of belonging through community identity, which is different off line. For this reason, depending on different online community identities that are perceived by the users, certain relational characteristics are expected to appear[4]. In other words, due to the normative factors that have been formed for themselves out of the sense of belonging and or the bond in the specific virtual space of community that prefers certain brand of product their subsequent behavioral responses are expected to be different. Furthermore, it is hypothesized that the more involved the members are with community activities, the stronger they intend to maintain online community participation, to take part in collaborative brand production, and to be involved in word-of-mouth activity. The present study attempted to explore the effects of strong involvement with community on these three behavior responses and compare the online consumer purchase behavior.

Based on the previous studies of online brand community, the present study looked into consumer responses to the characteristics of brand community activities in the course of their experience with the community, to empirically investigate the influence of brand community and consumer involvement with the brand, and to suggest practical implications that can be used to help accomplish industry goals.

2 Theoretical Background

2.1 Brand Community Activity Characteristics

A brand community is defined as "a specialized, non-geographically bound community, based on a structured set of social relations among admirers of a brand" [14]. It is a space where its members share their interests in a particular brand, showing social identity in their behavior[4,13]. A brand community has its identity and bond as its activity characteristics. Algesheumer, Dholak and Herrmann[4] argue that a brand community identity is that the individual members themselves want to be a member of the brand community, which is the sense of belonging to the community. Therefore, a brand community identity is what sets the goals of the community under the consensus among the community members and develop relationships among them.

On the other hand, the term bond is defined as "the quality of connection as one piece." In other words, it refers to the idea that individual members with an intention to act and interests in common issues set values to the other members of the community[4].

When community identity is set up, a promise or regulation is generated between the operator and the users of the community to run the community effectively, and it is referred to as community norm. That is, it operates as pressure to observe norms that have been generated by the community members' recognition of and agreements on its necessity. Strong online community identity means a brand community has distinct characteristics and goals. It can be speculated that the pressure from the community norms perceived by the community users will be reduced if the user is satisfied with the community characteristics and goals.

H1: Online brand community identity will have a positive effect on community bond.

H2: Online brand community identity will have a negative effect on normative community pressure.

H3: Online brand community bond will have a positive effect on normative community pressure.

2.2 Normative Community Pressure and Intention to Maintain Participation

Intention to maintain community participation refers to the situation that after experiencing a brand community, the users of the community have an intention to maintain the use of the brand community since they acquire their wanted information and relationships with the other members[4]. The difference between community commitment and intention to maintain participation is that while community commitment refers to intention to simply continue the relationships, intention to maintain participation bears the characteristics of involvement combined with active community activities.

It is predicted that online brand community will come up with a variety of social norms such as proper manners for other members possibly because although it is a virtual place made by individuals, it shares very similar characteristics of the real society. It is speculated that the stronger the pressure to observe community norms becomes, which can control the behaviors of community members, the less their intention to maintain community activities. Therefore, the present study generated a hypothesis regarding the community pressure as follows:

H4: Normative community pressure will have a negative effect on intention to maintain participation in online brand community activities.

2.3 Community Bond and Community Commitment

Commitment refers to an explicit promise to maintain the relationships between exchanging members, which can be defined as a desire to continue conducive relationships[7,12]. In other words, commitment can be considered the members' sense of belonging and affective attachment to the organization and their loyalty[9]. However, recent studies have extended the object of commitment beyond organizations, including all to which an individual can feel the sense of belonging. Interests in and attachment to the community through the brand community activities are likely to lead to more intensive interactions by heightening the relationships of the members and the sense of community. It is assumed that the community bond among the members will lead the community users to an intention to maintain their community activities. For this reason, the present study hypothesized the relationship between the two variables as follows:

H5: Online brand community bond will have a positive effect on community commitment.

2.4 Community Commitment, Intention to Maintain Community Participation, Involvement in Brand Production, and Brand Word-of-Mouth Activity

Intention to maintain community participation refers to maintaining active participation in the community. Compared to commitment which can be considered an intention to maintain the relationship in the community, the intention refers to the activeness in the community. Therefore, it is hypothesized that the community users who intend to maintain the relationships with the community will behave more actively in the community.

Involvement in brand production refers to an involvement where consumers convey their opinions about a product after using it, and the company incorporates their input into product development[1, 9]. Thus, it is hypothesized that greater commitment will have the community users want to see their needed items produced, which, in turn, will lead to more active involvement in brand production.

Brand word-of-mouth activity refers to promoting a brand by sharing with people around regarding product information learned from the brand community or one's positive opinions about the brand. The present study attempted to examine whether there are relative differences in behavioral responses of the consumers affected by sustained participation in a brand community that is perceived as useful. The following hypotheses were generated for the investigation.

H6-1: Online community commitment will have a positive effect on intention to maintain community participation.

H6-2: Online community commitment will have a positive effect on participation in collaborative brand production.

H6-3: Online community commitment will have a positive effect on brand word-of-mouth activity.

3 Research Methodology

3.1 Research Design

The purpose of the present study was to empirically investigate the effects of the unique characteristic of online brand community identity on community characteristics, such as normative community pressure and community bond, and on consumers' behavioral responses. A research model was designed as in Figure 1 to test the hypotheses generated for the study.

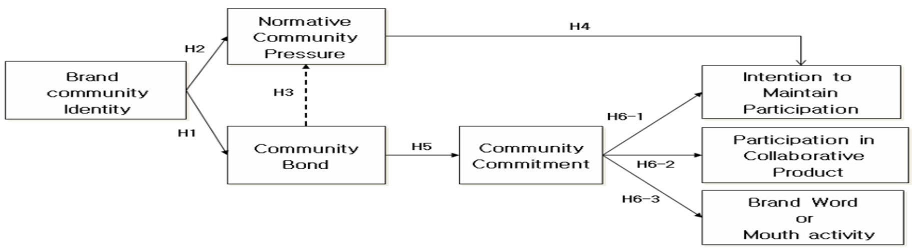


Fig. 1. Research Model

3.2 Sample and Data Collection

Questionnaires were administrated on general consumers who were participating in brand community as a member. The survey was conducted online and offline. As for the online survey, respondents participated using the questionnaire posted on the community website. A total of 150 questionnaires were distributed, and 124 questionnaires were collected. 117 questions were used in the final analysis, excluding 7 questionnaires for incomplete responses.

3.3 Sample Characteristics

The respondents who participated in the study consisted of 40 males (34.2%) and 78 females (65.8%). Of the entire participants, 20 respondents (86.3%) were in their 20s, 14 respondents (12%) in their 30s, and 2 respondents (1.8%) in their 40s. By occupation, the composition was 56 students (47.5%), 30 office workers (25.6%), 12 civil servants (10.3%), 8 professionals (6.8%), 4 self-employed (3.4%), 4 teachers and professors (3.4%), 2 in the art field (1.7%), and 1 housewife (0.9%).

3.4 Measurement of Variables

The variables used in the present study were adopted from previous community related studies, and the validity and reliability of the variables have already been established. The variables were properly modified for the study. A 5-point Likert-type scale was used to measure the variables, 5 being "Strongly Agree" and 1 being "Strongly Disagree."

Using the question items in Churchill[5] and Algesheumer, Dholakia, and Herrmann[4], online brand community identity was measured by 5 items, including the items concerning psychological attachment, identifying oneself as a part of the brand community. Online community bond was measured by 4 items, including the reason for online participation, the degree of emotional interaction with other users in the community, and the degree of getting help from others. Normative community pressure was measured by 4 items, including susceptibility from other and norm perception, using the question items in Algesheumer, Dholakia, and Herrmann [4]. To measure community commitment 6 items were used, including the degrees of the user's closeness to the online community, satisfaction, providing information, and sense of belonging as a member, adopting the question items in Algesheumer, Dholakia, and Herrmann[4], Aaker, Fournier, and Brasle[2]. Using the question items in Algesheumer, Dholakia, and Herrmann[4], 3 question items were developed to measure the user's intention to maintain community participation, which were about the degree of intention to maintain community activities, relationships with other users, and active participation. Participation in collaborative brand production was measured by the 3 items in Hong[1], including participation in product development and suggesting ideas for sales promotion. Word-of-mouth activity was measure by 6 items, including the item about favorable word-of-mouth for the brand, using the question items in Oliver[16] and Hong [1].

4 Data Analysis

4.1 Validity and Reliability Test

Confirmatory factory analysis was conducted to guarantee the validity and reliability of variable. The result showed the follow indexes χ^2=198.67, df=131(p<.00013), GFI=.92, AGFI=.90, NFI=.94, CFI=.97, RMR=.061, implying that all variable had convergent and discriminant validity[3]. Testing reliability with Cronbach's coefficients, all the coefficients were within the range of .756~.906, Showing the reliability of the construct variables. In addition, representativeness of the constructs was

Table 1. Result of Confirmatory Factor Analyses

Variables	Items	Standard loading	Measurement error	t-value	Cronbach's α	Composite Reliability
Community identity	Identity1	.83	.08	10.58	.867	.721
	Identity2	.86	.08	11.30		
	Identity3	.80	.08	10.10		
Community bond	bond1	.76	.09	8.68	.756	.928
	bond2	.76	.09	8.77		
Normative community pressure	normative1	.52	.11	4.55	.681	.810
	normative2	.45	.11	4.08		
Community commitment	commitment1	.31	.10	3.25	.856	.758
	commitment2	.81	.08	10.20		
	commitment3	.84	.08	10.74		
	commitment4	.84	.08	10.78		
Intention to maintain participation	participation1	.89	.08	11.76	.906	.949
	participation2	.83	.08	10.65		
Participation in collaborative production	product1	.95	.10	9.14	.782	.937
	product2	.78	.10	7.74		
Brand Word-or-mouth activity	WOM1	.69	.08	8.24	.861	.972
	WOM2	.79	.08	10.05		
	WOM3	.92	.07	12.64		
	WOM4	.88	.07	11.69		
Goodness of Fit	X^2=198.67 df=131(p<.00013) GFI=.92 AGFI=.90 NFI=.94 NNFI=.96 CFI=.97 RMR=.061					

* Estimated values are statistically significant at the level of 0.001.

Table 2. Correlation Matrix of Research Constructs(Φ matrix)

Construct	Community identity	Normative community pressure	Community bond	Community commitment	Intention to maintain participation	Participation in collabora-tive produc-tion	Brand Word-of-mouth activity
Community identity	**.896**	.62	.73	.64	.65	.08	.32
Normative community pressure	.79 (.06)	**.865**	1.27	.28	.40	.02	.21
Community bond	.86 (.15)	1.13 (.18)	**.894**	.79	.72	.04	.64
Community commitment	.80 (.05)	.53 (.09)	.89 (.16)	**.864**	.73	.08	.39
Intention to maintain participation	.81 (.05)	.64 (.08)	.85 (.16)	.86 (.04)	**.902**	.09	.39
Participation in collaborative production	.29 (.10)	.16 (.11)	.21 (.16)	.29 (.10)	.30 (.10)	**.883**	.18
Brand Word-or-mouth activity	.57 (.07)	.46 (.09)	.80 (.15)	.63 (.07)	.63 (.07)	.43 (.09)	**.901**

1. The numbers along the diagonal line represent Average Variance Extract(AVE). Below the diagonal line are correlations between construct definitions. The numbers in parentheses represent the standard errors. Above the diagonal line are squared correlations between construct definitions.
2. ** p<.01, *** p<.001.

verified from the composite reliabilities ranging .721~.972 which were higher than the commonly accepted level of . 7[10].

4.2 Results of Discriminant Validity Analysis

Discriminant validity was assessed with the analysis of correlation matrix(Φ matrix) that checks the measured difference as among the theoretical different constructs. The results are shown in <figure 2> value of "1" [calculated with correlation ± (2×standard error)], expect a few, out of range correlative coefficient between community bond, community identity, and normative community pressure. Other method of testing discriminant validity was employed with average variance extract(AVE)[19]. The AVE's values were much higher than the squared correlations across the variables.

4.3 Results of Hypothesis Testing and Discussion

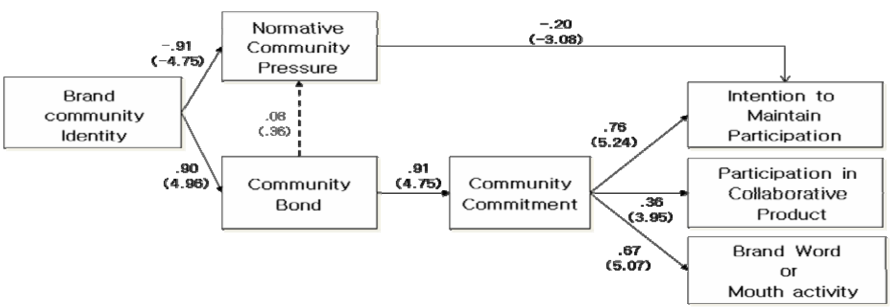
The results of the analyses on the effects of online community identity on behavioral responses of the community users through relationship marketing approach to the characteristics of community activities are shown in <Table 3>. The results showed that the goodness of fit indexes of the research model were x^2=261.43, df=145(p=.00), GFI=.89, AGFI=.87, NFI=.93, NNFI=.92, CFI=.96, RMR=.07 RMSEA=.00, suggesting the superiority o the research model.

Table 3. Results of Analyses on the Research Model

Hy-pothe-sis	Path	Path Coeffic-ient	Standard error	t-value	Hypothesis Acceptance
H1	Community identity→Community bond	.90	.18	4.96***	supported
H2	Community identity → Normative community pressure	-.91	.19	-4.75***	supported
H3	Community bond→Normative community pressure	.08	.22	.36[a]	rejected
H4	Normative community pressure → Intention to maintain participation	-.20	.06	-3.08**	supported
H5	Community bond → Community commitment	.91	.19	4.75***	supported
H6-1	Community commitment → Intention to maintain participation	.76	.15	5.24***	supported
H6-2	Community commitment → Participation in collaborative production	.36	.09	3.95***	supported
H6-3	Community commitment → Brand Word-or-mouth activity	.67	.13	5.07***	supported
Goodness of fit		x^2=261.43, df=145(p=.00), GFI=.89, AGFI=.87, NFI=.92, NNFI=.96, CFI=.96, RMR=.07 RMSEA=.004			

* 1. * p<.05 ** p<.01 *** p<.001. 2. a: not significant.

Hypothesis 1, which is, online brand community identity will have a positive effect on community bond, was supported by the data. It means that the more firmly the community identity is established, the stronger the community bond is. Hypothesis 2, online brand community identity will have a negative effect on normative community pressure, was supported. This means that that the more firmly the community identity is established, the clearer the community norms become. However, the results showed that community bond did not affect normative community pressure at the statistically significant level. Therefore, Hypothesis 3 was not supported. The results seem to indicate that attachment to the community does not necessarily impose the norms or regulation on others. Hypothesis 4, normative community pressure will have a negative effect on intention to maintain to participate in online brand community activities, was also supported. The results also show that when the normative community pressure is firmly established, it sets the atmosphere and motivates the members to



*Outside ( ) indicates standardized β value, inside ( ) indicates t value.

Fig. 2. Results of Analyses on the Research Model

participate negatively. Hypothesis 5, online brand community bond will have a positive effect on normative involvement, was supported. The results indicate that stronger community bond leads to desire to maintain the relationship with the community. Hypothesis 6 on intention to maintain community participation, participation in collaborative brand production, and brand word-of-mouth activity was all confirmed. In other words, the more strongly the community user desires to maintain the relationship with the community, the more actively he or she participates in activities, brand collaborative production, and the brand word-of-mouth activity. Furthermore, the results revealed that the participation was in the simple form of behavioral responses rather than collaborative activities such as suggesting ideas.

The findings of the study indicate that establishing online brand community identify increases consumer bond, and consumer's use behavior of online brand community can lead to greater interests in and attachment to the community and motivate the consumer's activeness. These findings show that online brand community plays the role of improving and maintaining the quality of relationships between consumers and the industry, shedding light on the importance of managing online brand community.

5 Discussion, Implications, and Limitation

5.1 Summary of Study Results and Implications

The present study investigated the effects of community identity on the characteristics of community activities and changes in community user's behavior related to the community's activity characteristics. Based on the analyses of the questionnaires, the study found the following results: First, firmly established community identity was found to have a positive effect on community bond. Establishing clear identity seems to increase the user's satisfaction, which, in turn, strengthens the bond. Second, establishing identify affects establishing order in the community. The development of this normative community pressure appears to reduce the user's intention to maintain participation. However, norms are set by the operator with the consensus of the consumers. Therefore, if the community has an established identity, the user tries to accept the norms unless they poses great inconvenience because they facilitate the interaction between the information the community want and other users, differentiating the community from other brand communities and confirming the identity. Third, community bond was not shown to affect normative community pressure. The results may be because friendly relationship among the community users strengthens the unity and embraces the members' mistakes, rather than increasing the normative community pressure. Fourth, normative community pressure is shown to negative affect the user's intention to maintain his or her participation. Based on the results, it can be inferred that if certain community users observe the norms, that means they intend to maintain participation, but if they do not, that means they do not intend to use the community. Fifth, community bond affects community commitment, indicating the sense of belonging and close bond with the community influence the maintenance of the relationship with the community. Sixth, community commitment affects the user's intention to maintain community participation, participation in collaborative brand production, and brand word-of-mouth activity. This means that maintaining the relationship with the community has an effect of the consumer's active behavior toward community activities.

The findings of the study suggest that establishing community identity is a solid means to maintaining the user's interests in and attachment to the community. The establishment of community identity also means increased community commitment. It is particularly important for the industry because by the characteristics of brand community, if any users maintain the relationship with the community with an attachment to the brand, it is the same as they maintain the relationship with the product. In addition, sustained relationship with the brand community leads to the consumer's active participation behavior, which all cause positive feelings toward the.

5.2 Limitation and Implications for Future Studies

The present study has certain limitations, which are discussed below along with research implications for future studies. First, the present study examined the user behavior with online community commitment. In future studies it is necessary to look into the changes in the user's feelings toward the brand after the active participation. Second, it would be also interesting to conduct a difference analysis by community types since differences are expected in the characteristics of community activity and changes in the user behavior based on the type of the community. Third, in this study the researcher's lack of experiential knowledge about online community posed certain limitations on practical measurement of the community. In future studies more reliable and valid measurement should be developed for in-depth investigation.

References

1. Hong, H.: Transactions: Psychological and Behavioral Outcomes of Commitment to Apparel Brand Online Community. Journal of the Korean Society of Clothing and Textiles 30(6), 916–927 (2006)
2. Aaker, J., Fournier, S., Brasel, D.A.: When Good Brands Do Bad. Journal of Consumer Research 31, 1–16 (2004)
3. Anderson, J.C., Gerbing, D.W.: Structural Equation Modeling in Practice: A Review and Recommended Two-Step Approach. Psychological Bulletin 103(3), 411–423 (1988)
4. Algesheimer, R., Dholakia, U.M., Herrmann, A.: The Social Influence of Brand Community: Evidence from European Car Clubs. Journal of Marketing 69, 19–34 (2005)
5. Churchill Jr., G.A.: A Paradigm for Developing Better Measures in Marketing. Journal of Marketing Research 15, 591–621 (1979)
6. Dellarocas, C.: The Digitization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms. Management Science 49(10), 1407–1424, 18 p., 3 charts (2003)
7. Dwyer, R.F., Schurr, P.H., Sejo: Developing Buyer-Seller Relationship. Journal of Marketing 55, 11–27 (1987)
8. Susan, F.: Consumer and Their Brands: Developing Relationship Theory in Consumer Research. Journal of Consumer Research 24, 343–373 (1998)
9. Gruence, T.W., Summers, J.O., Acito, F.: Relationship Marketing Activities, Commitment, and Membership Behaviors in Professional Association. Journal of Marketing 64, 34–49 (2000)

10. Hair, J.F., Anderson, R.E., Tahtam, R.L., Black, W.C.: Multivariate Data Analysis With Readings, 4th edn. Prentice Hall, Englewood Cliffs (1995)
11. Jang, H., Olfman, L., Ko, I., Koh, J., Kim, K.: The Influence of On-line Brand Community Characteristics on Community Commitment and Brand Loyalty. International Journal of Electronic Commerce 12(3), 57–80 (2008)
12. Moorman, C., Zaltman, G., Deshpande, R.: Relationships between Providers and Market Research; The Dynamics of Trust within and between Organizations. Journal of Marketing Research 29, 314–329 (1992)
13. McAlexnder, J.H., Koenig, H.F.: Building Brand Community. Journal of Marketing 66, 38–54 (2002)
14. Muniz Jr., A.M., O'Guinn, T.C.: Brand Community. Journal of Consumer Research 31(4), 737–747 (2001)
15. Nunnally, J.C.: Psychometric Theory, 2nd edn. McGraw-Hill, New York (1978)
16. Oliver, R.L.: A Cognitive Model of the Atecedents and Consequences of Satisfaction Decision. Journal of Marketing Research 17, 460–499 (1980)
17. Price, L.L., Arnourld, E.J.: Commercial Friendships: Service Provider Client Relationship in Context. Journal of Marketing 63, 38–56 (1999)
18. Thompson, S.A., Sinha, R.K.: Brand Communities and New Product Adoption: The Influence and Limits of Oppositional Loyalty. Journal of Marketing 72, 65–80 (2008)
19. Fornell, C., Lacker, D.F.: Evaluating Structural Equation Models with Unobservable Variables and Measurement Error. Journal of Marketing Research 18(1), 39–50 (1981)

Key Factors Affecting Continuous Usage Intention in Web Analytics Service

JaeSung Park[1], JaeJon Kim[2], and Joon Koh[2]

[1] Business Incubating Center, Chonnam National University, Korea
pamto@hanmail.com
[2] Department of Business Administration, Chonnam National University, Korea
jaejon@jnu.ac.kr, kjoon@jnu.ac.kr

Abstract. Web Analytics Service (WAS) has become a rapidly growing business in the internet transactions between firms. It is, from the client firm's perspective, considered to be a form of IT outsourcing. In this paper, we identify factors which can influence the continuous usage intention of a firm that has utilized WAS, and empirically validate the relationships between the identified factors. In the research model developed and described herein, only information quality among the several quality factors was found to be significantly associated with the satisfaction of the client firm. Relative value, switching cost, service usage period, and satisfaction were all significantly associated with dependence.

Keywords: Web Analytics, Continuous Usage Intention.

1 Introduction

The Internet revolution has resulted in the widespread proliferation of e-business. The advent and ensuing spread of e-business has rapidly and completely altered the mode of transaction in B2C and B2B transactions, specifically from off-line to on-line (Lau, 2007). E-business has replaced existing business paradigms largely due to the ease and efficiency with which one can access and search through huge amounts of knowledge and resources on the internet.

Recently, firms have begun to conduct analyses of enormous amounts of log data in order to formulate business strategies that respond more faithfully to their customers' needs. In order to survive in an era of fierce competition, many firms have attempted to obtain web logs of their clients' activities, and subsequently to use the results of those web log analyses in their decision-making processes. This trend has boosted the Web Analytics industry, which provides efficient web log analytic services for firm-level customers (REUTERS, 2003).

According to the forecast of the IDC, the importance of Web Analytics is demonstrated by the market's healthy growth of 9.3% in 2008, with total revenues of $388.7 million (IDC, 2004). In particular, the ASP sector referred to as the Web Analytics

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 534–541, 2009.

Service (WAS), is set for rapid growth, and is enjoying increased popularity as a B2B business service model (REUTERS, 2003). Some of the key suppliers in the worldwide WAS industry are webSideStory, Coremetrics, MyComputer.com, and IBM Global Services in the USA, in addition to the Korean firms AceCounter, Logger, and SkyLog.

More than ten firms have jumped into the WAS industry in Korea, and some major companies, including Google and Daum communications, have decided to launch WAS businesses. Thus, owing to the fierce competition among service-providing firms in the existing WAS industry, clients are increasingly likely to switch their service providers.

With increases in the risk of clients switching away from their existing partners, service providers must attempt to prevent their clients from discontinuing their existing transactional relationships. This business environment also motivates us to attempt to understand the clients' switching intentions with regard to the WAS industry. Thus, the principal objective of this study is to identify the main factors that influence the continuous usage intentions of firms that use WAS, and to characterize the relationships among the identified factors.

2 Conceptual Background

2.1 Exit and Dependence

Basically, WAS involves a transaction between a service-providing firm and a client firm. From the client's perspective, the client's satisfaction with the WAS firm should determine whether they continue their existing transactions or switch to another provider.

According to Hirschman's theory (1970), when the performance of either individuals in an organization or between organizations decreases, "exit" and "voice" strategies are considered, depending on the costs and benefits of those decisions: that is, one partner may (1) exit (switch to a different partner) or (2) voice his concerns to the partner and attempt to persuade the partner to provide better products or services. Hirschman (1970) suggested that the basic conditions for the exit decision were twofold: (1) the existence of competitors superior to the existing partner, and (2) low cost for discontinuing the relationship with the existing transaction partner.

Laver (1976) argued that the exit decision is characterized by a three-stage process: (1) the existence of an available alternative, (2) the calculations of the cost to transfer, and (3) the evaluation of the expected benefits that might ensue from voicing complaints to the existing partner.

Although the benefits may smaller than expected, thus leading to dissatisfaction, it would be inadvisable to discontinue a relationship with an existing transaction partner if no clearly better alternative exists. This can manifest in several ways; (1) it may be difficult to determine the level of benefit that would be provided by a new channel, (2) the level of expected benefit provided by a new channel might be lower than the benefits of the existing channel, or (3) the switching cost may be higher than the benefit gained from switching to a marginally superior channel.

Further, Aldrich (1979) classified the factors that influence the exit decision as follows: (1) the probability of making a contract with a new alternative, (2) loyalty toward the existing partner, (3) opportunity to effectively voice complaints, and (4) the possibility of retaliation.

Based on the aforementioned studies (Hirschman, 1970; Laver, 1976; Aldrich, 1979), the exit conditions are determined by the power enforced by a partner. If a firm is controlled by its partner on the basis of the relative power between the two firms, then a firm can be considered to depend absolutely on its partner (Emerson, 1962). Such dependence is determined by the degree of importance to the partner in terms of business value (Ganesan, 1994; Heide & John, 1988; Emerson, 1962). Dependence upon the existing channel is known to increase in the following cases: (1) alternative attractiveness is low or there are no options apart from the existing partner or channel (Hirschman, 1970; Aldrich, 1979; Jones, 1998), (2) relative value from the existing channel is regarded as high (Kumar et al., 1998), (3) the switching cost is prohibitively high (Lim et al., 1995), (4) the competitive advantage is high as compared with that of the existing partner (Reed & DeFillippi, 1990), or (5) sunk cost, such as transaction-specific investments, is high (Heid & John 1988; Ganesan, 1994).

2.2 Model of Information System Success

A variety of ways for a firm to pay for and use outperforming software have been developed with the growth of network technologies such as the internet (Zviran & Erlich, 2003). These have been moved to the software service industry using the internet, as one of the primary business models of application service providers (ASP) (Raisinghani & Kwiakowski, 2001). WAS, a type of ASP, can help client firms to operate or analyze their websites in a rental format, without the need for the firms to directly purchase the solution. Accordingly, from the client's perpective, WAS functions as a type of information systems outsourcing.

Some issues associated with the continuous usage of IS are approached on the basis of DeLone & McLean's (1992) theory, which is generally referred to as the Model of Information System Success. They identified the six major categories that explain IS Success, namely system quality, information quality, usage, user satisfaction, individual impact, and organizational impact. According to this theory, information quality and system quality are the antecedents of satisfaction that can enhance usage intention, thereby triggering an individual impact as well as an organizational impact.

Pitt et al. (1995), however, previously pointed out that the quality of IS in DeLone & McLean's (1992) theory is limited to the user's product-oriented evaluation. They added the concept of service quality to DeLone & McLean's (1992) model, and then empirically validated their model. DeLone & McLean (2003) also presented their updated IS success model, analyzing 150 papers that had assessed their original IS success model. This updated model held that information quality, system quality, and service quality affected IS usage and user satisfaction, which would yield certain net benefits.

Based on these theories, we can adapt the IS success model to the context of WAS. Client's decisions regarding continuous usage appear to be influenced by satisfaction, which in turn is affected primarily by the qualities of WAS: that is, information quality, system quality, and service quality.

3 Research Model and Hypotheses

Based on the above discussions, the research model for continuous usage intention in WAS can be shown in Figure 1.

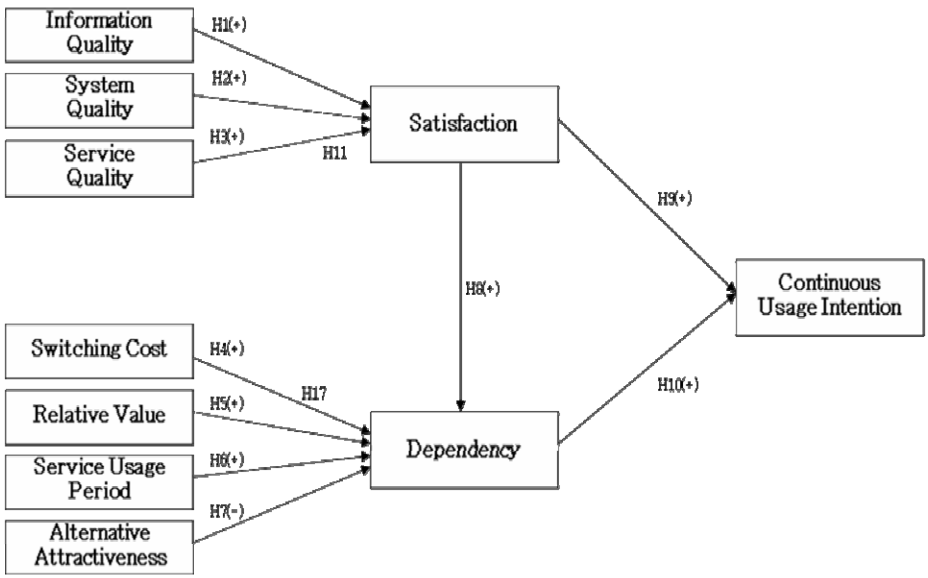


Fig. 1. Research Model

4 Analysis of Research Results

4.1 Test of Validity and Reliability

We tested our model by surveying actual client firms regarding a current provider of the WAS via the internet. The instruments of the study were modified, in part, on the basis of previous relevant studies. The majority of the factor loadings for the items appeared above 0.5. The items were collected and corresponded well to each singular factor, demonstrating a high degree of convergent validity (Hair et al., 1998). As the factor loadings for a variable (or factor) were higher than the factor loadings for the other variables, the instrument's discriminant validity was demonstrated. A principal-component analysis was also conducted in an effort to assess the differences between factors. Each factor was extracted as expected, which explained 75.87, 77.32, and 64.86 percent, respectively, of the total variance--each with eigen values greater than 1.000.

The internal consistency reliability of the variables was evaluated by computing Cronbach's alphas. The Cronbach's alpha values of the variables, ranging from 0.725 to 0.943, were all securely over 0.700, which is satisfactory (Nunally, 1978). The reliability and validity of the research variables were all, therefore, considered acceptable.

4.2 Tests of Hypotheses

The Pearson correlations were calculated for the variables measured by interval or ratio scales. Potential multicollinearity among the antecedents was assessed prior to the multiple regression analysis, because some of the variables were significantly correlated with others (e.g., information quality and empathy). Although several variables showed significant correlations, their tolerance values ranged from 0.355 to 0.998, thereby indicating that multicollinearity is not a likely threat to the parameter estimates (Hair et al., 1998).

Table 1 shows the results of the multiple regression analyses. Our results indicate that all of the three regression models are significant at p<0.001 (F=30.914, 47.150, and 106.72, respectively), and that the predictors of each model explain 44.2 percent, 60.4 percent, and 58.2 percent, respectively, of the total variance. Hypotheses 1, 2, 3b, and 3c address the relationships between the WAS qualities and satisfaction. Information quality was related significantly with satisfaction (beta=0.640, p<0.001), thus supporting Hypothesis 1. However, no relationship between system quality and satisfaction, between responsibility and satisfaction, and between empathy and satisfaction were determined to be significant, thereby rejecting the Hypotheses 2, 3b, 3c.

Table 1. The Results of Multiple Regression Analyses

Model	R^2	adj. R^2	F	β	Tolerance	Results
[1] Satisfaction (SAT)						
SAT=IQ+SQ+RES+EMP+errors	0.457	0.442	30.914***			
IQ (Information Quality)				0.640***	0.355	H-1 Accepted
SQ (System Quality)				0.046	0.539	H-2 Not accepted
RES (Responsibility)				-0.029	0.547	H-3[b] Not accepted
EMP (Empathy)				0.032	0.449	H-3[c] Not accepted
[2] Dependence (DP)						
DP=SW+RV+SUP+AA+SAT+errors	0.618	0.604	47.150***			
SW (Switching cost)				0.247***	0.697	H-4 Accepted
RV (Relative Value)				0.487***	0.773	H-5 Accepted
SUP (Service Usage Period)				0.162**	0.998	H-6 Accepted
AA (Alternative Attractiveness)				-0.066	0.896	H-7 Not accepted
SAT (Satisfaction)				0.248***	0.668	H-8 Accepted
[3] Continuous Usage Intention (CUI)						
CUI = SAT+DEP+errors	0.588	0.582	106.72***			
SAT (Satisfaction)				0.453***	0.672	H-9 Accepted
DEP (Dependence)				0.411**	0.672	H-10 Accepted

*p<0.05, **p<0.01, ***p<0.001.

Hypotheses 4, 5, 6, 7, and 8 assess the effects of the five affecting factors on the variable of dependence. Switching cost, relative value, service usage period, and satisfaction were all significantly related to dependence (beta=0.247, $p<0.001$; 0.487, $p<0.001$; 0.162, $p<0.01$; and 0.248, $p<0.001$). However, the relationship between alternative attractiveness and dependence was not significant, and thus Hypothesis 7 is not supported.

Hypotheses 9 and 10 examine the relationships between satisfaction and continuous usage intention and between dependence and continuous usage intention. Satisfaction and dependence all significantly influenced continuous usage intention (beta=0.453, $p<0.001$; beta=0.411, $p<0.001$). Therefore, the results support Hypotheses 9 and 10, and we interpret this to mean that a positive relationship exists between satisfaction/dependence and continuous usage intention.

5 Discussion and Implications

The WAS industry is rapidly growing, along with the growth and proliferation of e-business. A couple of major firms (e.g., Daum Communications, Google, etc.) have recently branched out into the WAS industry. Thus, competition in the WAS industry is fierce. Currently, existing WAS providers have new business opportunities, in that they can obtain more sales from the expanded market opened up by an expansion of B2B e-business, but these markets are also threatened by the market entry of several major companies. Thus, it is necessary for existing WAS providers to understand the key factors in winning over these major competitors. That is, the major factors that trigger continuous usage intention in WAS customers should be explored by providers. This study provides several strategies and specific action calls for WAS providers on the basis of our empirical findings.

First, a service provider needs to strengthen the client firm's dependence on the provider, in order to reduce the probability of exit. For instance, service providing firms may provide some advanced statistical analysis tools and on-line lectures/content regarding advanced statistical methods which would optimize the clients' use of WAS services. This policy would serve to increase the client firm's perception of relative value and would help them to accumulate transaction-specific assets such as specialized analytical know-hows about the service, thereby augmenting the opportunity cost that would be incurred when switching away from the existing channel.

Second, with regard to the qualities of WAS, service providers must prioritize such quality-related factors as information quality, system quality, and service quality, due to limited resources. If service providing firms experience a scarcity of resources, they may consider prioritizing all of these quality factors simultaneously. However, if not, they should focus on improving only one or two WAS quality factors at a time. The results of the study also imply that information quality is the only factor that significantly explains satisfaction among the qualities of WAS examined herein. Consequently, service providing firms should clearly expend effort to increase the quality of their information.

6 Conclusion and Limitations

This paper examined the key factors affecting the continuous usage intention of client firms of WAS providers. In developing our research model, we attempted to integrate the theories of the Exit/Voice, dependence, and IS success model.

One of the major findings of this study was that a client firm's continuous usage intention is determined by both a motivational factor (i.e., satisfaction) and a hindering factor (i.e., dependence). This interpretation supports some previous studies in which dependence was implicated as a major factor that determines the exit decision when the exit is being seriously considered, due to a decrease in the service provider's performance. Thus, when other competitors become actual options for a client in the WAS, service providing firms must consider both the client's satisfaction factor and dependence factor. Second, the quality of information is the most important factor that a service providing firm can improve to increase the client's satisfaction. The result that only information quality was significantly related to satisfaction implies that the quality of information in the WAS is the most important factor to be considered by service providers, which after all, leads to clients' continued use of WAS providers. Third, relative value, switching cost, service usage period, and satisfaction all significantly affected the dependence. Relative value was shown to be one of the most crucial factors with regard to the dependence of the client firm on the existing provider.

Despite the number of academic contributions and practical implications of this study, it is encumbered by a few limitations, which also suggests possible future research directions. First, with regard to the assessment of continuous WAS service usage intention among clients, we were able to assess this using only one service provider, namely ACECOUNTER. Even though this provider is one of the leading companies in the WAS industry, the company and its data are not representative of the entirety of the WAS industry. Further investigation into other representative service companies will be required for the generalization of our findings. Second, the cross-sectional data is limited to our findings; for example, satisfaction is known to be accumulated over time. Thus, future research will be necessary in the form of a longitudinal study. Last but not least, it would be valuable to find other significant variables that affect continuous usage intention, in addition to the satisfaction and dependence variables examined in this study. More relevant and powerful models for explaining the continuous WAS use of a client might be achievable with further study.

References

1. Aldrich, H.E.: Organizations and Environments. Prentice-Hall, Englewood Cliffs (1979)
2. DeLone, W.H., McLean, E.R.: Information System Success: The Quest for the Dependent Variable. Information Systems Research 3(1), 60–95 (1992)
3. DeLone, W.H., McLean, E.R.: The DeLone and McLean Model of Information Systems Success: A Ten-year Update. Journal of Management Information Systems 19(3), 9–30 (2003)

4. Emerson, R.: Power-Dependence Relations. American Sociological Review 27(1), 31–41 (1962)
5. Ganesan, S.: Determinants of Long-Term Orientation in Buyer-Seller Relationships. Journal of Marketing 58(2), 1–19 (1994)
6. Hair, J.F., Aderson, R.E., Tatham, R.L., Black, W.C.: Multivariate Data Analysis, 5th edn. Prentice-Hall, Englewood Cliffs (1998)
7. Heide, J.B., John, G.: The Role of Dependence Balancing in Safeguarding Transaction-Specific Assets in Conventional Channels. Journal of Marketing 52(1), 20–35 (1988)
8. Hirschman, A.O.: Exit, Voice, and Loyalty: Responses to Decline in Firms, Organizations, and States. Harvard University Press (1970)
9. IDC. An Integrated Approach to Digital Marketing through Web Analytics: IDC (2005)
10. Jones, M.A.: Satisfaction and Repurchase Intentions in the Service Industry: The Moderating Influence of Switching Barriers. University of Alabama (1998)
11. Jones, M.A., Mothersbaugh, D.L., Beatty, S.E.: Switching Barriers and Repurchase Intentions in Services. Journal of Retailing 76(2), 259–275 (2000)
12. Jones, M.A., Mothersbaugh, D.L., Beatty, S.E.: Why Customers Stay: Measuring the Underlying Dimensions of Services Switching Costs and Managing Their Differential Strategic Outcomes. Journal of Business Research 55(6), 441–445 (2002)
13. Kumar, N., Scheer, L.K., Steenkamp, E.M.: Interdependence, Punitive Capability, and the Reciprocation of Punitive Actions in Channel Relationships. Journal of Marketing Research 35(2), 11 (1998)
14. Lau, R.Y.K.: Towards a Web Services and Intelligent Agents-based Negotiation System for B2B eCommerce. Electronic Commerce Research and Applications 6(3), 260–273 (2007)
15. Laver, M.: Exit, Voice, and Loyalty'Revisited: The Strategic Production and Consumption of Public and Private Goods. British Journal of Political Science 6(4), 463–482 (1976)
16. Lim, Y.K., Lee, C.L., Park, T.H.: Effects of Perceived Transactional Characteristics on Channel Members' With-drawal Intentions. Korean Management Review 24(3), 113–144 (1995)
17. Nunnally, J.: Psychometric Theory, 2nd edn. McGraw-Hill, New York (1978)
18. Pitt, L.F., Watson, R.T., Kavan, C.B.: Service Quality: A Measure of Information Systems Effectiveness. MIS Quarterly 19(2), 173–187 (1995)
19. Pitt, L.F., Watson, R.T., Kavan, C.B.: Measuring Information Systems Service Quality: Concerns for a Complete Canvas. MIS Quarterly 21(2), 209–221 (1997)
20. Pitt, L.F., Watson, R.T., Kavan, C.B.: Measuring Information Systems Service Quality: Lessons From Two Longitudinal Case Studies. MIS Quarterly 22(1), 61–80 (1998)
21. Raisinghani, M., Kwiatkowski, M.: The Future of Application Service Providers: Information Strategy (Summer 2001)
22. Reed, R., DeFillippi, R.J.: Causal Ambiguity, Barriers to Imitation, and Sustainable Competitive Advantage. The Academy of Management Review 15(1), 88–102 (1990)
23. REUTERS. The Web Analytics Outlook: Increasing Online Profit: REUTERS (2003)
24. Sujan, M.: Consumer Knowledge: Effects on Evaluation Strategies Mediating Consumer Judgments. The Journal of Consumer Research 12(1), 31–46 (1985)
25. Zviran, M., Erlich, Z.: Measuring IS User Satisfaction: Review and Implications. Communications of the Association for Information Systems 12, 81–103 (2003)

Enterprise Financial Status Synthetic Evaluation Based on Fuzzy Rough Set Theory

Ming-Chang Lee and Jui-Fang Chang

Department of Business Administration, National Kaohsiung University of Applied Sciences

Abstract. Enterprise financial status synthetic evaluation is an important issue. The weight of synthesis evaluation is determined by expert, lending to subjectivity and without considering the redundancy of attributes exists in traditional synthetic evaluation. Recently, an attempt of integration between the theories of fuzzy set and rough sets has resulted in providing a roughness measure for fuzzy sets. Therefore, in this study, we firstly define the attribute reduction based on rough set theory. Secondly, we create membership function of each attributes. Using membership function, we can easily create the judgment matrix. Thirdly, we discuss the weight of attribute and measure of information. Finally, the methods of the synthesis evaluation are present with an example.

Keywords: Enterprise financial status, fuzzy rough set theory, Synthetic evaluation.

1 Introduction

Enterprise financial status analysis is an important issue. A synthetic evaluation method is $R_\circ$ W, where R is called judgment matrix and W is called weight value. In ordering to simplify the calculation, a new approach of synthetic evaluation is discussed.

Recently, the theory of rough sets has emerged as another method for dealing with uncertainty using from inexact or incomplete information (Pawlak, 1991; Pawlak, 1982). Incorporating fuzziness and uncertainty into decision marking problems can generally generate promising alternatives (Yu, 1984), such as a fuzzy technique has been applied vague data in synthetic evaluation (Zimmerman, 1991). Theories of fuzzy sets (Zadeh, 1965) and rough sets (Pawlak, 1982: Pawlak, 1985) are generalizations of classical set theory for modeling vagueness and uncertainty. Therefore, in this research, we use fuzzy set theory to create membership function (Wei et al. 2003) for obtain judgment matrix. We use the definition of weigh of attributes on rough sets theory, and create the set of weight value.

2 Preliminaries

The basic concepts, notations and results of rough sets as well as their extensions are briefly reviewed.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 542–549, 2009.

2.1 Attribute Reduction Algorithm Based on Rough Set Theory

Given an information system (a data set), S = {U, A, V, f}, where U and A are finite and nonempty sets called the universe, and the sets of attributes, respectively. In information system, there exists a function, such that f: U×A⟶V. A is the union of C and D, the intersection of C and D is empty. C is called as conditional attributes, and D is called as decision attributes. The information system is also called decision system, or knowledge system (Zhang and Wu, 2001).

Definition 1. Let P be attribute set and P $\subseteq C$, with any P $\subseteq C$ there is an associated equivalence relation INP (P).

Definition 2. Let R be equivalence relation in U and X is subset of U, then X is called positive domain in R. it write as PosR (X) and

$$\mathrm{Pos}_R(X) = \{ x \in U \mid [u]_R \subseteq x \}.$$

Definition 3. Let{U, A, V, f} be decision system, with A =C∩D =ψ and A = C∪D. Where, C is the set of conditional attribute s and D is the set of decision attributes. If B $\subseteq A$ and IND (B, D) = IND (A, D) then B is called the minimum attributes set.

Definition 4. Let{U, A, V, f} be decision system, with A = C∩D =ψ and A = C∪D. Where, C is the set of conditional attributes and D is the set of decision attributes. If a $\in A$, a is conditional attribute and IND (A- {a}) = IND(A), then a can be reduced from set of A.

Definition 5. The interaction of all reduce sets (red (A)) in A is called the core, it write as Core (A), we have Core (A) =∩ red (A).

2.2 Create Membership Function of Each Attributes

The membership function of a fuzzy set is a generalization of the indicator function in classical sets. In fuzzy logic, it represents the degree of truth as an extension of valuation. Degrees of truth are often confused with probabilities, although they are conceptually distinct, because fuzzy truth represents membership in vaguely defined sets, not likelihood of some event or condition. Membership functions were introduced by Zadeh in the first paper on fuzzy sets (1965). We use the threshold value (b), low bound (a), and upper bound (c) and create the triangular membership function.

Triangular membership function is:

$$\mu(x) = \begin{cases} 0 & x \le a \\ \dfrac{x-a}{b-a} & a \le x \le b \\ \dfrac{x-c}{b-c} & b \le x \le c \\ 1 & x \ge c \end{cases} \tag{1}$$

2.3 The Weight of Attributes Acquisition Method Based on Measure of Information

Given an information system, S = {U, A, V, f}, where U and A ($A = C \cup D$) are finite and nonempty sets called the universe, and the sets of attributes, respectively. Let A be $\{a_1, a_2, \ldots, a_m\}$.

Definition 6. The weight w_i of a_i, is:

$$w_i = \left| \frac{I(A) - I(A - \{a_i\})}{mI(A) - \sum_{j=1}^{m} I(A - \{a_j\})} \right| \tag{2}$$

Definition 7. Let U/ IND (p) be $\{X_1, X_2, \ldots, X_n\}$. The measure information of p definite as

$$I(p) = \sum_{i=1}^{n} \frac{card(X_i)}{card(U)}\left(1 - \frac{card(X_i)}{card(U)}\right)$$

$$= 1 - \frac{1}{[card(U)]^2} \sum_{i=1}^{n} [card(X_i)]^2 \tag{3}$$

Where Card (X_i) denotes the cardinality of X_i

2.4 Synthetic Evaluation

Firstly, we obtain the judgment matrix R by membership function. Second, use fuzzy operator M (• , +), we calculate the R.

$$B = W \bullet R \tag{4}$$

3 The Proposed of Evaluation Process Are

Step 1: Determine financial evaluation indexes.

In financial reporting analysis, it has fifty factors for enterprise failure or distress (Ahn et al., 2000; Hawley et al., 2000). In this study, we selected 12 financial indexes. The evaluation process based on rough set theory can only handle discrete data, so it need to transform the sign data into discrete data, the continuous data into discrete data. The simplest method is to establish a table with one-to-one correspondence between the sign data and the discrete numerical data. In this study, we used threshold value, and the continuous data transform into discrete data.

Step 2: Attribute Reduction Algorithm based on Rough Set Theory.

Using section 2.1, we calculate IND (C), IND (C-{r}), where $r \in C$. From the definition 4, we can find the minimum attribute set.

Step 3: Create membership function of each attributes.

After finding the minimum attribute sets, we create membership function of each attributes. Using threshold value (b), maximum value (c), and minimum value (a) in each attribute, we create the triangular membership function. (See (1)).

Step 4: Calculate the weight of attributes.

Using section 2.3, from measure information, we calculate the weight of attribute.

Step 5: Synthesize evaluation.

4 Illustration- Enterprise Financial Distress

In this section, an example is given to illustrate the proposed synthesize evaluation algorithm.

Step 1: Enterprise financial distress index.

Enterprise "A" has financial data set in 1998 to 2006 (Denoted as Table 1). Therefore, U= {1998,…, 2006}={1,2,…, 10}. In this study, we use 12 financial evaluation indexes are: Return on total assets (x_1), profit ratio of sales (x_2), Profit ratio of total capital (x_3), Working capital to sales ratio (x_4), Inventory turnover ratio (x_5), Account

Table 1. Enterprise "A" financial index in 1998 to 2006

U	X_1 (%)	X_2 (%)	X_3 (%)	X_4 (%)	X_5 (%)	X_6 (%)	X_7 (%)
1	22.35	35..12	25.30	3.27	120.00	73.6	160.0
2	18.52	44.12.	75.45	6.82	645.01	82.7	132.4
3	14.60	19.20	15.63	2.54	685.54	54.12	112.6
4	22.15	38.34	251.3	6.37	102.02	67.96	148.5
5	40.28	25.30	150.5	2.58	420.05	89.54	88.54
6	20.15	39.65	354.2	7.92	255.08	75.51	137.9
7	48.36	45.80	752.9	21.30	421.79	60.24	156.3
8	35.50	34.85	425.8	10.55	621.02	68.56	135.9
9	42.72	30.12	560.7	8.12	435.1.2	84.20	120.42
○	25.0	34.0	330	7.5	635.0	78.0	130.0

○ : Threshold value.

U	X_8 (%)	X_9 (%)	X_{10} (%)	X_{11} (%)	X_{12} $(\%)_2$
1	21.5	105.9	21.9	178.2	189.5
2	12.38	153.4	19.8	498.5	335.2
3	16.72	110.2	18.54	214.6	19.58
4	14.32	190.5	11.24	225.1	210.5
5	22.17	88.5	20.19	345.8	124.7
6	32.54	154.2	41.80	238.7	188.55
7	32.45	161.6	28.58	262.1	190.84
8	25.87	115.1	34.20	214.5	250.85
9	45.12	120.3	25.95	259.8	155.85
○	24.0	128.0	24.80	256.0	186.0

○ : Threshold value.

Table 2. The enterprise "A" financial distress discrete decision table

U	X_1 (%)	X_2 (%)	X_3 (%)	X_4 (%)	X_5 (%)	X_6 (%)	X_7 (%)	X_8 (%)	X_9 (%)	X_{10} (%)	X_{11} (%)	X_{12} (%)
1	0	1	0	0	1	0	1	0	0	0	0	1
2	0	1	0	0	1	1	1	0	1	0	1	1
3	0	0	0	0	1	0	0	0	0	0	0	0
4	0	1	0	0	1	0	1	0	0	0	0	1
5	1	0	0	0	0	1	0	0	0	0	1	0
6	0	1	1	1	0	0	1	0	1	0	0	1
7	1	1	1	1	0	0	1	1	1	1	0	1
8	1	1	1	1	1	0	1	0	0	1	0	1
9	1	0	1	1	0	1	0	0	0	1	1	0

receivable turnover ratio (x_6), Current ratio (x_7), Asset-liability current ratio (x_8), Current Liabilities ratio (x_9), Asset-liability ratio (x_{10}), Equity ratio (x_{11}), pretax profit current debit ratio (x_{12}). Therefore, $C= \{x_1, x_2,\ldots, x_{12}\}$.

Set each financial index threshold value, which is the average value of 20 enterprises on lately 10 years financial index. According with the threshold value, we get the enterprise financial discrete decision table (denoted as Table2). For example, the threshold value of attribute "Return on total assets" is 25% (See Table 1). When (x_1) > 25%, we set (x_1) equal to 1 otherwise set (x_1) equal to 0.

Step 2: To obtain a reducible decision table

We find out that the attributes X_3, X_4 , X_8 , X_{10} have the same value in Table 2, and therefore we remove attributes X_4 , X_8 , X_{10}; the attributes X_2, X_7 , X_{12} have the same value in Table 2, and therefore we remove attributes X_7 , X_{12}; the attributes X_6, X_{11} have the same value in Table 2, and therefore, we remove attributes X_6. Based algorithm of reduction, the reduce attributes are X_1, X_2 , X_3 , X_5 , X_9, X_{11}.

Using definition 3 and 4 in section 2.1, we calculate the minimum attributes set.

IND (R) = {1, 2, 3, 4, 5, 6, 7, 8, 9}
IND (R-(x_1)) = {1, 2, 3, 4, 5, (6, 8), 7, 9}≠ IND (R)
IND (R-(x_2)) = {2, (1, 3, 4), 5, 6, 7, (8, 9) } ≠IND (R)
IND (R-(x_3)) = {1, 2, 3, 4, (5, 9}, 6, 7, 8 } ≠IND (R)
IND (R-(x_5)) = {1, 2, 3, 4, 5, 6, 7, 8, 9} = IND (R)
IND (R-(x_9)) = {1, 2, 3, 4, 5, 6,(7, 8), 9}≠IND (R)
IND (R-(x_{11})) = {1, (2, 4), 3, 5, 6, 7, 8, 9} ≠IND (R)

Since IND (R-(x_5)) equal to IND (R). By definition 4, we can remove the attribute X_5. Since IND (R-(x_i)) not equal to IND (R), ($i \in$ (1,2 ,3, 9,11)). The attributes X_1, X_2 , X_3, X_9, and X_{11} can not be removed and the minimum attributes set is $\{X_1, X_2 , X_3, X_9, X_{11}\}$.

The enterprise financial status reduction decision table is showed as Table 3.

Step 3: Create membership function of each attributes

The final reduce attribute set is $\{X_1, X_2, X_3, X_9, X_{11}\}$. From the Table 1, the threshold value, minimum value, and maximum value in $\{ X_1, X_2, X_3, X_9, X_{11}\}$ are denoted as Table 4, and use the threshold value, low bound, and upper bound to create the triangular membership function.

Table 3. Enterprise financial distress reduction decision table

U	X_1 (%)	X_2 (%)	X_3 (%)	X_9 (%)	X_{11} (%)
1	0	1	0	0	0
2	0	1	0	1	1
3	0	0	0	0	0
4	0	1	0	0	0
5	1	0	0	0	1
6	0	1	1	1	0
7	1	1	1	1	0
8	1	1	1	0	0
9	1	0	1	0	1

Table 4. The threshold value, minimum value, and maximum value

U	X_1 (%)	X_2 (%)	X_3 (%)	X_9 (%)	X_{11} (%)
Threshold value (b)	25	34	330	128	256
Minimum value (a)	12	18	12	86	175
Maximum Value (c)	50	48	760	195	500

The membership function of X_1, X_2, X_3, X_9, X_{11} are:

$$\mu(x_1) = \begin{cases} 0 & x_1 \le 12 \\ (x_1 - 1)/13 & 12 \le x_1 \le 25 \\ (50 - x_1)/25 & 25 \le x_1 \le 50 \\ 1 & x_1 \ge 50 \end{cases}$$

$$\mu(x_2) = \begin{cases} 0 & x_2 \le 18 \\ (x_2 - 18)/16 & 18 \le x_2 \le 34 \\ (50 - x_2)/24 & 34 \le x_2 \le 48 \\ 1 & x_2 \ge 48 \end{cases}$$

$$\mu(x_3) = \begin{cases} 0 & x_1 \le 12 \\ (x_3 - 12)/318 & 12 \le x_3 \le 330 \\ (760 - x_3)/430 & 330 \le x_3 \le 760 \\ 1 & x_3 \ge 760 \end{cases}$$

$$\mu(x_9)=\begin{cases}0 & x_9\le 86\\ (x_9-86)/42 & 86\le x_9\le 128\\ (195-x_9)/67 & 128\le x_9\le 195\\ 1 & x_1\ge 195\end{cases}$$

$$\mu(x_{11})=\begin{cases}0 & x_{11}\le 175\\ (x_{11}-175)/81 & 175\le x_{11}\le 256\\ (500-x_{11})/244 & 256\le x_{11}\le 500\\ 1 & x_{11}\ge 500\end{cases}$$

Step 4: Calculate the weight of attributes and measure information

Therefore, U = {1998, 1999,…., 2006} and V = {X_1, X_2, X_3, X_9, X_{11}}. From (2) and (3) in section 2.3, we obtained w_1= 0.11, w_2 = 0.25, w_3 = 0.28, w_9 = 0.15, w_{11} = 0.21. We obtain W = {0.11, 0.25, 0.28, 0.15, 0.21}.

Step 5: Synthetic evaluation

The value of X_1, X_2, X_3, X_9 and X_{11} in Table 1 substitute to (4), and we obtain the judgment matrix R, where R is (R1, R2, R3, R4, R5).

R1 = (0.796, 0.501, 0.20, 0.78, 0.388, 0.627, 0.006, 0.58, 0.291).
R2 = (0.626, 0.245, 0.075, 0.488, 0.456, 0.431, 0.175, 0.631, 0.757)
R3 = (0.042, 0.199, 0.011, 0.752, 0.435, 0.943, 0.016, 0.777, 0.463)
R4 = (0.473, 0.620, 0.567, 0.067, 0.059, 0.608, 0.498, 0.692, 0.816)
R5 = (0.039, 0.006, 0.488, 0.618, 0.631, 0.786, 0.975, 0.487, 0.984)

Use fuzzy operator M (• , +), we obtain the result: B = W•R = (0.3346, 0.2223, 0.2313, 0.5574, 0.4198, 0.697, 0.3349, 0.645, 0.6799). The result is express as follows when the indices are ranked:

Table 5. The result: B = W•R

1998	1999	2000	2001	2002	2003	2004	2005	2006
0.334	0.223	0.2313	0.5574	0.4198	0.697	0.3349	0.645	0.6779

2003 >2006 >2005>2001>2002 > 2004 >1998 > 2000>1999

4 Conclusion

This research constructed on enterprise synthetic evaluation model based on rough theory. Firstly, we determine financial evaluation index. Secondly, the paper proposes to utilize rough set theory to reduce attributes. Thirdly, we create membership function in each attribute. Through numeric attributes can be represented by judgment matrix.

Fourth, we use measure information to calculate the weight of attribute. Finally, we apply fuzzy operator M (•, +), to obtain the result of synthetic evaluation. Our future will find more effective method for weight generation..

References

[1] Pawlak, A.: Rough Sets. International journal Computer and Information Sciences 11, 341–356 (1982)
[2] Pawlak, A.: Rough Classification. International Journal of Man-Machine Studies 20, 469–483 (1984)
[3] Pawlak, A.: Rough set and fuzzy sets. Fuzzy Sets and Systems 17, 99–102 (1985)
[4] Pawlak, A.: Rough sets, theoretical Aspects of reasoning about data. Kluwer Acadmic, Dordrecht (1991)
[5] Wei, L.L., Zhang, W.X.: Probabilistic rough characterized by Fuzzy Sets. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS(LNAI), vol. 2639, pp. 173–180. Springer, Heidelberg (2003)
[6] Yu, P.R.: Dissolution of Fuzziness for Better Decision Perspective and Techniques. Management Sciences 20, 171–207 (1984)
[7] Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
[8] Zhang, W.X., Wu, W.Z.: Rough sets Theory and methodology. The Press of Science (2001)
[9] Zimmermann, H.J.: Fuzzy Set Theory and Its Applications, 2nd edn. Kluwer Academic Publishers, Boston (1991)
[10] Ahn, B.S., Cho, S., Smkim, C.Y.: The integrated methodology of rough sets and artificial neural network for business failure prediction. Expert system with application 18(4), 65–74 (2000)
[11] Hawley, D.D., Johnson, J.D., Rsina, D.: Artificial Neural Network Systems: A Network tool for financial Decision Making. Financial Analysis Journal, 63–72 (November/December 2000)

An Agent-Based Model for the Adaptation of Processing Efficiency for Prioritized Traffic

Sanja Illes, Dragan Jevtic, and Marijan Kunstic

University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3,
10000 Zagreb, Croatia
{sanja.illes,dragan.jevtic,marijan.kunstic}@fer.hr

Abstract. The increasing number and variety of telecommunication services being offered by networks have emphasized the demand for optimized load management strategies. Obtainable processing efficiency is one of the key elements for all services and has to be dominantly supervised. Different priority levels and classifications can co-exist to support services, such fine regulation of precedence mixed with hard and destructive priority for critical service classes. Intelligent processing efficiency management with adaptive protection of predefined processing power can support such requests. This paper deals with the regulation of packets moving towards service-processing resources. Regulation is based on service priorities and processing load in the agent-controlled part of the network.

Keywords: Intelligent agent, reinforcement learning, Q-learning.

1 Introduction

Resource management is crucial to optimally reducing service time and efficiently utilizing resources in communication networks. The agent paradigm has initiated a trend of inserting intelligent methods into communication components. Self-adaptability will become an imperative feature in the network management in future. Nevertheless, whether an agent transports information (i.e., a mobile agent) or controls that transport (i.e., an intelligent agent) there potentially exist various methods, models and architectures, which could be applied inside the paradigm of agent environments. Agent position is one of the architectural characteristics presented in [1, 2, 3, 4, 5] where a single agent and multi-agent systems managed information transport. For the case where the agent is positioned outside of network nodes, existence of a centrally controlled routing mechanism was assumed in [1]. If the agents are distributed among network nodes, they operate as a team and the routing mechanism is distributed [2]. Resource protection was recognized as another imperative mechanism due to the coexistence of conflicting requests, which can unexpectedly degrade service characteristics or destroy the service [4, 5].

This paper considers a system arranged in the following way. We assume packets represent information-transporting units. Further, we suppose three different types of packets. All packets have to be accepted by the server but based on their allowed

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 550–559, 2009.

admission delay three classes were established. An Intelligent Agent (IA) controls two packet classes, while the third class represents bursty traffic, which we assume has instant access to the processing equipment. In real networks, some user requests, such as link breaking or alternative routing, can create such bursts. The task of the IA is to preserve the allocated capacity of processing power and control packet transportation for the two classes, manipulating according to class priorities.

The paper is organized as follows: first, the reinforcement learning concept is described in section 2, with a general description of the Q–learning algorithm and the internal representation of an intelligent agent. Section 3 describes the communication environment of the intelligent agent for packet flow control, together with variants of reward function settings. Experimental results are described in Section 4, followed by the conclusion.

2 Reinforcement Learning

Reinforcement learning (RL) is a type of learning, which uses feedback signals from the environment as a guideline in the learning process. Typically, RL processes are intensity interleaved with action processes. Feedback signals represent information regarding environmental change in response to an action performed by an agent. Agents, also called RL agents, learn by means of interaction with their environment, evaluating effects of their actions. This way, agents mimic the learning processes of human beings – an agent has its goal, which is accomplished through trial and error. The RL agent perceives its environment through a finite number of states. In each state, the RL agent chooses an action from a finite set of actions. In each discrete time step, the agent perceives the state of the environment s_t, chooses and executes action a_t, receives feedback information, calculates a reward $r_t=r(s_t, a_t)$, and finally goes to the next state $s_{t+1} = \delta(s_t, a_t)$.

Key words in reinforcement learning are *reinforcement function*, *value function*, *policy* and *state value*. The reinforcement function defines agent behavior, since agents try to maximize (or minimize) the sum of rewards received from the initial state to the final state [6, 7]. A policy determines which action to execute in a specified state – i.e., it maps states to actions. The state value is defined as the sum of rewards received from a specified state to the final state, following the given policy. Basically, the value function maps states to state values. The simplest way to present value functions is with a look-up table. A reward function has an emphasis on the immediate effect – it shows how efficient it was to choose a certain action in a given state. On the other hand, the value function shows how good it is for an agent to be in a given state long-term.

2.1 Components of an RL Agent

Internal representation of an intelligent agent consists of actions, states and monitoring devices. The agent receives a reward (scalar) for every executed action. The biggest challenge was in defining the appropriate contributions of different parameters and expressing them as a unique scalar value assigned to the agent.

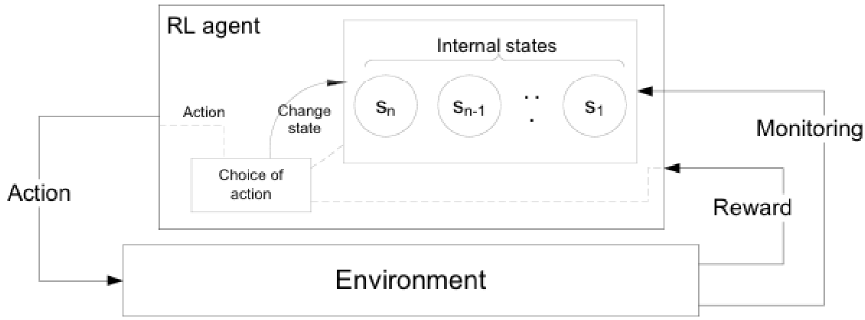


Fig. 1. Components of a RL agent

2.2 A Short Review of the Q-Learning Technique

Q-learning is a form of reinforcement learning which does not associate states with state values, but uses (state, action) pairs and value (Q-value) mapping. After an agent executes chosen action *a* in state *s* and receives reward *r*, the Q-learning algorithm updates the Q-value Q (s, a) as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right), \tag{1}$$

where α is the learning rate and γ is the discount factor (used to incorporate the notion that future rewards are less relevant than immediate rewards). The agent then moves to the next state *s'* and the whole process is repeated. The learning process is complete when the Q-value converges to a certain value. The Q-function uses a two dimensional array of state – action pairs (i.e., a lookup table or Q-table). In our system, states in the Q-table represented the sum of the current server load and the queue occupancy. Therefore, the number of states was equal to the number of all possible sums of server load and queue occupancy.

3 Agent-Based Processing Control

Our system consisted of one intelligent agent (IA), a server module (SM) and packet generator modules (PGM). We assumed the existence of two packet classes (class *A* and class *B*) with different priorities. Two unlimited input buffers were placed to gather input packets from PGM1 and PGM2 related to each packet class (Fig. 2). One buffer was used for class *A* and the other for class *B* packets. Packet classification was used to realize packet prioritization. The task of the IA was to capture packets from their input buffers and transmit them towards the server. Packets sent from the IA reach the server processor passing through the server buffer. If processing power of server went over a predefined threshold, packets were queued. The buffer size was limited. Consequently, if the number of queued packets reached the buffer limit, new packets were rejected. The sequence of passing packets was determined by the IA's actions. We assumed that the input packet rate was always higher than the IA's

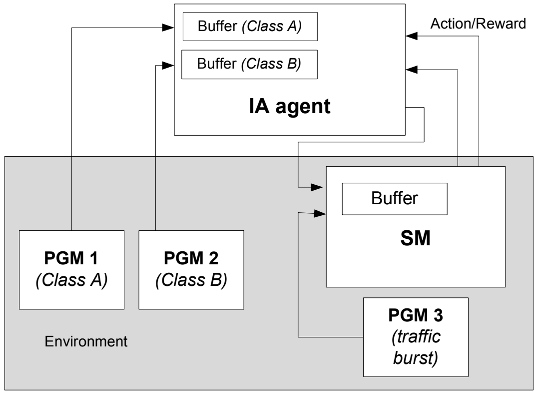


Fig. 2. The modeled environment with three traffic generators and an IA used for packet flow control towards the server processor

decision rate, so packets of both classes were available at the agent's inputs at any moment.

Applying continuous reduction of processing strength on the server side was aimed at freeing the processor for bursty traffic arriving from other sources. In case of a burst, the agent's mission was to reduce transmission towards the server to prevent packet rejection and to preserve processing for the burst, or other external sources with absolute priority in processing.

The IA agent selects an action among the three following actions based on the *Q*-value:

- *action*_1 - transfer class *A* packet,
- *action*_2 - transfer class *B* packet,
- *action*_0 – do not transfer a packet.

The server module stores the received packet into its joint (input) buffer, processes it in FIFO (first in first out) mode and sends feedback information to the agent. In case the given load limit is exceeded, the server holds the packet in the queue, or rejects it if the queue is completely occupied. After transferring a packet, the agent waits for the server response. Based on the feedback information received from the server module, the reward is calculated and the *Q*-table updated.

The probability of agent exploration action was set to 0,05. Initial priority values for packet classes *A* and *B* were set to 1, however precedence in packet admission was controlled by setting this value for each class independently with fine tuning. For example, if the priority of class *A* is set to 1 and for class *B* to 1.2, transport precedence for class B packets will be 20% higher than for class A; if the priority of class *B* is set to 2, transport precedence for class B packets will be 100% higher than for class A.

3.1 Reward Settings

The effectiveness of agent activity was entirely dependent on the reward function. Therefore, we tested the effects of different reward functions on agent operation. The reward sent from the server to the IA contained information about the current processing load, the maximum allowed load and the queue length. The reward signal was calclulated in relation to the previously performed action, as follows,

$$reward0_{lin} = \frac{currentLoad}{maxLoad} + queueOccupancy\,, \tag{2}$$

when the previously executed action was *action*_0, and

$$reward1,2_{lin} = maxReward - \left(\frac{currentLoad}{maxLoad} + queueOccupancy \right). \tag{3}$$

when the previously executed action was *action*_1 or *action*_2. Furthermore, the reward was multiplied with the priority of the packet sent. According to the reward policy, for a small server load and low queue length (zero), a greater reward was received for sending a packet than for not sending a packet to the server. For instance, suppose the server was unloaded and the queue was empty. In such a case, the IA agent would receive a reward value of zero for *action*_0, while the other two actions would be rewarded with the maximum reward. Conversely, if the current load was equal to the maximum allowed load and the queue was full up, the reward would be 200 for choosing *action*_0, and – 200 for the other two actions.

3.2 Impact of the Reward Dynamic

A server buffer was used to reduce packet rejection for cases where instant transfer to the processing side cannot be realized. Additionally, we introduced some modifications in expressions (2) and (3), in which reward functions were described as a linear contribution of addends. The reward signal calculation was composed of two elements: the current processing load and queuing. Additionally, the reward function varied in strength and form, i. e. these addends were used as differ stimulated parameter to the agent:

$$reward0_{\ln} = \ln\left(\frac{currentLoad}{maxLoad} + 0.01 \right) + \ln\left(queueOccupancy + 0.01 \right), \tag{4}$$

$$reward1,2_{\ln} = \ln(maxReward) - reward0_{\ln}\,, \tag{5}$$

where a logarithmic scale of the components was used for both elements in the sum i.e. the reward dynamic was minor, and

$$reward0_{\exp} = e^{\frac{currentLoad}{maxLoad}} + e^{queueOccupancy}\,, \tag{6}$$

$$reward1,2_{\exp} = e^{maxReward} - reward0_{\exp}\,, \tag{7}$$

where an exponential function was applied. Here, we emphasized our assumption that the agent always has packets to choose from in its input buffers, i.e. that the IA's buffers are continuously being filled with packets. If an action was chosen which could not be executed due to an empty input queue, it was tried again. If still unsuccessful, it changed state and chose a new action. This way, the agent was not punished for choosing a “bad” action when input buffer was empty since the intention was not to train the agent to learn as a function of input packet arrival. The primary goal of the agent was to take care of the forwarding of packets to the processing server.

4 Experimental Results

Simulation experiments were performed on the system described above [Fig. 2].

The packet generator modules were designed as a discrete packet generator, which generated packets according to the following expression:

$$timeBetweenPackets = \frac{1}{\lambda} \times \ln(1-u) \times factor\ , \tag{8}$$

where u is random variable in interval [0, 1], generated by a uniform distribution. Parameter *factor* is variable used for additional regulation of the packet generation rate. The PGM1 and PGM2 packet generators continuously sent packets towards the IA input buffers, while PGM3 sent its packets directly to the server by manual control.

As mentioned earlier, the server stored received packets in its input buffer. In case of buffer overload, additional packets were rejected and lost permanently.

A feedback signal was only sent if a packet was received from the IA, and not from PGM3. IA operation efficiency was tested for the following three scenarios:

- the processing rate was faster than IA actions (queuing was minor),
- the processing rate was slow and the IA was required to reduce its rate, and
- the traffic burst generated directly toward the server buffer.

PGM3 was used to simulate an external traffic source for which packets sent to the server were treated as highest priority and, if possible, accepted by the server immediately. In other words, PGM3 generated packets, which ware serviced immediately, if processing was able to accept it. IA actions consisted of making a choice between input buffers (or reducing the rate through action exception) based on the current level of processing load and buffer occupancy, as well as acting in accordance with the given priorities.

Simulation parameters for each scenario are shown in table 1. We analyzed the effects of changing the contribution of the queue value in the reward signal as stated in (4), (5), (6) and (7), and the effects of changing parameter γ.

Table 1. Simulation parameters for each scenario

Server rate	*factor*	λ	Buffer size	Processor load limit	Processing time per packet [ms]		Load step up per packet [%]	
					Class *A*	Class *B*	Class *A*	Class *B*
High	50	1	100	50%	10	50	2	1
Low	50	1	20	50%	500	300	5	3
Burst (Class *B*)	35	1	100	50%	500	300	5	3

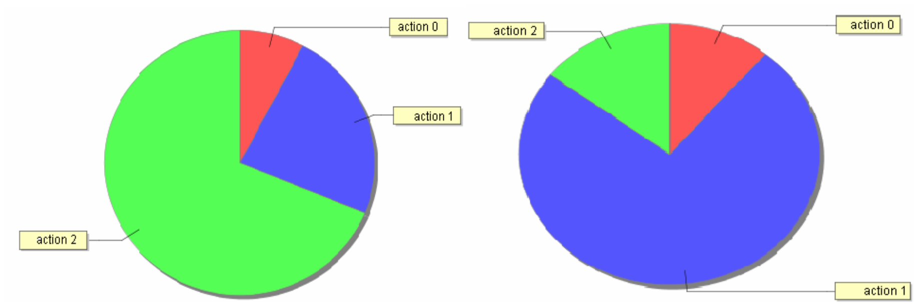


Fig. 3. Left: IA actions for higher processing rate scenario, γ=0, reward computed by (4) and (5), ε =0.05, priority of class *B* was set -1. Right: IA actions for the higher processing rate scenario, γ=0, reward computed by (4) and (5), ε =0.05, priority of class A was set to 2.

4.1 Higher Packet Processing Rate

In case of high packet processing rates, i.e., when processor speed was higher than the agent rate, the main task of the IA was to minimize its *action*_0 choices. Accordingly, IA needed to maximize its *action*_1 and *action*_2 choices, depending class priority. The best results were obtained when using expressions (4) and (5) to compute the reward signal and with the value for γ set close to 0. Using these expressions, the agent chose *action*_0 the fewest times, which is an unavoidable feature of agent exploration.

Conversely, when a more dynamic reward function was used (expressions (2) and (3), (6 and (7)), the number of times the IA decided to avoid sending packets towards the server was larger.

In this case, it was important to ensure that the IA agent minimized the number of its *action*_0 choices, while respecting the prioritization levels associated with the packet classes. Depending on the given priorities, the agent selected the actions as depicted in Fig. 3.

4.2 Lower Packet Processing Rate

In cases where the processing rate was lower than the agent's transmission rate, the main task of the IA agent was to reduce the rate of packet transmission toward the server in accordance with packet priority. According to simulations, agent actions were significantly reduced in such cases, while the predefined priority was still respected.

As expected, the IA agent recognized that it should reduce the number of chosen *action1* and *action2* decisions to keep the processing rate under the established limit, while still adapting action choices to the given priorities. At the start of the simulation, the processing load continually increased until processing load reached a critical value. Queuing functionality accumulated surplus packets during the period in which the agent had to reduce its transfer actions toward the server. This scenario is illustrated in Fig. 4 and Fig. 5.

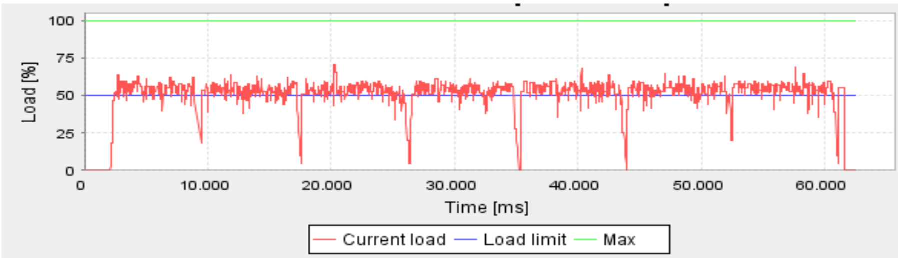


Fig. 4. Server load, α = 1, γ = 0.5, reward found by (2) and (3), ε = 0.05. Class *A* priority = 2

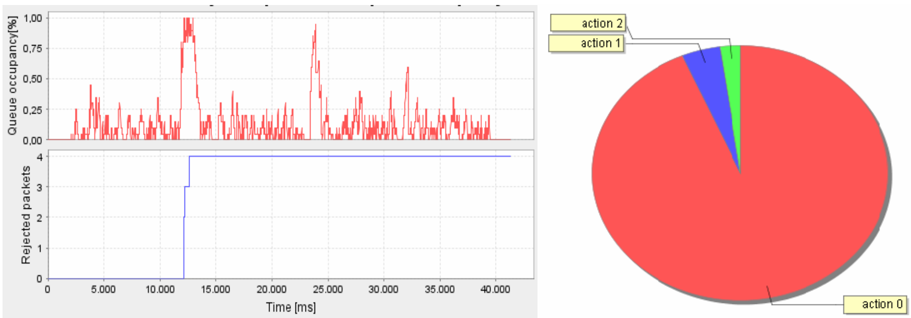


Fig. 5. Queue state and rejection level for high input rate; Left: Packet rejection and queue occupancy graphs. Right: Relation of chosen actions.

Overall, our simulation tests indicate that superior results were achieved when using expressions 4 and 5 to evaluate the reward function. Poorer results were obtained by using expressions 2 and 3, and considerable fluctuations were observed when rewards were evaluated by 6 and 7.

The learning factor α was set to 1, and ε to 0.05 in all simulations. The value for discount factor was tested in interval between 0 and 0.5. However, slowing down the processing rate more than i.e. 300 and 500 ms per packet (see table 1) in relation to the IA transport rate, gives inadequate results and agent was unable to operate.

4.3 Impact of External Bursty Traffic

As mentioned earlier, PGM3 was used to simulate bursty traffic, which impacted server processing power and was not under IA control. This was used to evaluate IA behavior when the processing load was significantly increased by external traffic, i.e. out of IA competences. Running of IA actions was based on server load and queue monitoring, without any knowledge about other traffic sources.

Approximately 15000 ms into the simulation, additional traffic from the PGM3 source was introduced and loaded the processor with the intensity given in table 1. The number of processed packets increased drastically. We could see that the LM agent responded to the new situation from Fig. 6. The number of rejected packets rapidly increased at first, but as the IA adapted its actions, the speed of packets

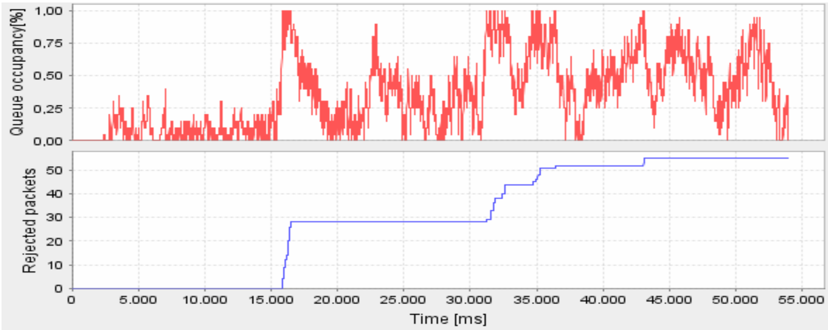


Fig. 6. Impact of external burst – distribution of packet rejection and queued packets

rejection decreased, staying at a constant level throughout most of the simulation. However, some fluctuation and additional packet rejection occurred after 32000 ms, 35000 ms and 43000 ms decreasing in intensity (Fig. 6.). Queue occupancy was also reduced from its maximum, fluctuating around an average of approximately 70%. The agent spent approximately 40000 ms reducing queue occupancy after the beginning of the burst (approximately between 15000ms and 55000ms).

5 Conclusion and Future Work

This paper presents a model for flow control of classified packets, with an emphasis on class priorities. There were two specified classes; however the model is designed to support three classes of packets. Packets were considered to transport elements of information. Each packet charged the processor a fixed amount of processing power. Packet classes were specified by a scalar enabling fine regulation of the priorities between the two specified classes. An intelligent agent held a crucial role in the model, controlling packet flow and managing the priorities. The agent utilized a continuous learning method, called Q – learning.

Simulation results show that our agent responded adequately in different scenarios, including scenarios with bursty traffic applied from an “external” source. The external source was used to simulate real architectures, in which a certain level of processing power should be reserved for instant admittance for special packets with superior priority. The main task of the agent was to limit the charging of processing power over a predefined threshold to achieve efficiency and high-system operation in case of such high-priority bursts.

References

1. Jevtic, D., Kunstic, M., Cunko, K.: Agent-based Connection Control for Digital Content Service. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS(LNAI), vol. 4496, pp. 1035–1042. Springer, Heidelberg (2007)
2. Jevtic, D., Kunstic, M., Ouzecki, D.: The Effect of Alteration in Service Environments with Distributed Intelligent Agents. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS(LNAI), vol. 3683, pp. 16–22. Springer, Heidelberg (2005)

3. Kusek, M., Lovrek, I., Sinkovic, V.: Agent Team Coordination in the Mobile Agent Network. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS(LNAI), vol. 3681, pp. 240–246. Springer, Heidelberg (2005)
4. Jevtic, D., Kunstic, M., Jerkovic, N.: The Intelligent Agent-Based Control of Service Processing Capacity. In: Palade, V., Howlett, R.J., Jain, L. (eds.) KES 2003. LNCS(LNAI), vol. 2774, pp. 668–674. Springer, Heidelberg (2003)
5. Jevtić, D., Kunštić, M., Matijašević, S.: Load Protection Model based on Intelligent Agent Regulation. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS(LNAI), vol. 4251, pp. 284–291. Springer, Heidelberg (2006)
6. Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-Dynamic Programming. Athena Scientific. MIT, Belmont (1996)
7. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning 8, 55–68 (1992)

Multi-agent Service Deployment in Telecommunication Networks

Kresimir Jurasovic, Mario Kusek, and Gordan Jezic

University of Zagreb
Faculty of Electrical Engineering and Computing
Unska 3, HR-10000, Zagreb, Croatia
{kresimir.jurasovic,mario.kusek,gordan.jezic}@fer.hr

Abstract. In this paper an agent based framework for service deployment in telecommunication networks is presented. The framework is intended to improve the process of service deployment in next generation telecommunication networks. This process has been complicated with the introduction of new actors and the migration of service infrastructure from the telecommunication provider to the third-party value added providers. The paper describes the issues related to service deployment, model of a service that is to be deployed on the infrastructure of a provider and agents the framework is comprised of.

1 Introduction

The telecommunication market in the last several years has been changing due to change in trends and expectation of its users. Users are no longer satisfied with basic telecommunication services (e.g. calls, SMS, MMS) due to increase in the capability of mobile terminals. Those terminals now allow users to use different applications that improve their quality of life, listen to music or watch videos. This change follows the changes that are happening on the Internet with the introduction of Web 2.0 also called the "the Social Web" [1]. The idea of Web 2.0 is to connect people though different social networks and allow them to share information (e.g. Wikipedia, YouTube, Flickr) or use different services on the Internet (e.g. Gmail, Google maps, automated weather information, Skype, RSS).

Paper [2] has presented a new Telco 2.0 model that describes the role of telecommunication operators in this new age. He claims that telecommunication operators will have to change by providing new innovative services to their users, integrating their business area with other non-telecommunication areas, providing new technological innovations and cooperating with other companies. This model also causes the change in the value chain [3] since it introduces new roles on the telecommunication market. These roles are filled by third party providers like content and software provider's which create services and content for users. Equipment vendors manufacture user terminals [4].

The ideas from the Telco 2.0 model are already being implemented by introducing the IP Multimedia Subsystem (IMS) and Service Delivery Platforms

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 560–569, 2009.

(SDP). The IMS provides network convergence and ubiquitous access to multimedia services as well as ensures quality of service [5]. SDP introduces two new concepts *multimedia marketplace* [6] and *manage and sell* concept [7]. The basic idea of the SDP is to allow third party value added service providers (VASP) to offer services to subscribers of network operators. Instead of offering services themselves, network operator wraps services provided by his network (e.g. billing, authorization, authorization, rating, different IMS enablers, mobile and broadband enablers) and offers them to the VASP thought the use of standard interfaces. VASP can then make different types of services, offer them to the subscribers and upon request use those interfaces to deliver the service.

Since the majority of services that are offered to the users will be hosted outside operator environment an opportunity for a new role on the market arises shared infrastructure provider or hoster [8]. When creating a new service most, if not all, of the content and service providers will have to create the same subcomponents for their services. In order to reduce the time needed to implement those features, VASPs will be looking for another party that could host their services as well as offer such features. The idea is that a single party (shared infrastructure provider) will offer the necessary network resources, hosting environment as well as the additional features mentioned above.

This paper is organised as follows: Sec. 2 describes the problem of service deployment in telecommunication networks. The architecture of the multi–agent solution to service deployment is described in Sec. 3. Sec. 4 concludes the paper.

1.1 Related Work

There are several frameworks that deal with problem of service deployment on a variety of remote nodes. One of them is the IBM Tivoli framework [9]. The component that deals with deployment is called Tivoli Provisioning Manager. The framework has several components whose functionality include gathering information about all the monitored nodes, automation of the deployment process, security, reporting etc. Tivoli framework encompasses tools and methods for deploying standalone software components. The process of software deployment can be automatized using scripts while update can be performed periodically.

Author in [10] have designed a framework that allows distributed R-OSGi applications to be deployed and managed from within the Eclipse framework. User can use Eclipse to generate a deployment graph for distributed applications as well as for load balancing, replicating modules, redundancy and reporting. This framework provides an efficient method for deploying OSGI based applications. Its disadvantage is that it is limited to one type of services. It also requires constant monitoring of an administrator during the deployment.

TACOMA [11] is an agent based framework used to update multiple remote locations from a single repository. The TACOMA system is divided into two components: the repository and the remote location. The repository contains all the software packages handled by the system which can be used during the update process. The remote location is the system that has to be updated. Periodically, the repository initiates the update process for all the remote locations.

Two agents are used during the process: State.Call agent gathers information while the Pack.Install agent installs packages. The TACOMA framework currently supports only update of rpm packages. Installation has to be do done manually.

Problem with both of these frameworks is that the administrator has to know and understand the domain where the software is deployed. In the agent framework described in this paper agents will be used to automatize the entire process and shield the user from the complexity of deployment. Administrator involvement should be reduced only when an unexpected and unsolvable errors occur during agent deployment. The other problem that the agent framework will try to solve is error handling. If an error occurs during installation an agent can use its knowledge of the domain to try to solve the problem on it own. The described frameworks require administrator to find and solve the problem.

2 Service Deployment in Telecommunication Networks

As the number of actors in the value-chain of the telecommunication market increases a problem with software deployment rises. An example of such problem can be found in the following scenario: network provider is interested in providing a framework for analysing usage of services for other third party providers in his network. So the provider obtains a framework from a software supplier. When installing this framework in the provider's network, the framework has to be customised for the network since it has to collect data from other nodes. Each time a software supplier introduces a new feature or adds support for additional nodes, the software has to be deployed and reconfigured. This process must occur for every provider that bought this framework. The same applies for cases when a network provider offers a service that requires software to be installed on network nodes of third-party suppliers.

One way of solving this problem is to send a human technician to install new software on every node. The easier way would be to have an intelligent mobile agent that knows how to install and configure this type of software on remote nodes. This approach has two benefits. First is that it hides the complexity of the deployment process to the user. The second is that this agent can learn what to do when an unexpected error occurs while installing services or software components in the provider's network. If it knows how to handle such errors it can try to amend the problem by obtaining new components or by changing the configuration. If the problem is to large for solving by an agent, it can try to locate some other agent who knows how to solve it or as a last resort contact a human administrator. This paper will describe an agent based framework capable of deploying new services with minimal interaction with the administrator and that is capable of handling unexpected situations.

2.1 Composition of a Service

The composition of a service is show in Fig. 1. Each service is composed out of software components and additional information about the service. Each

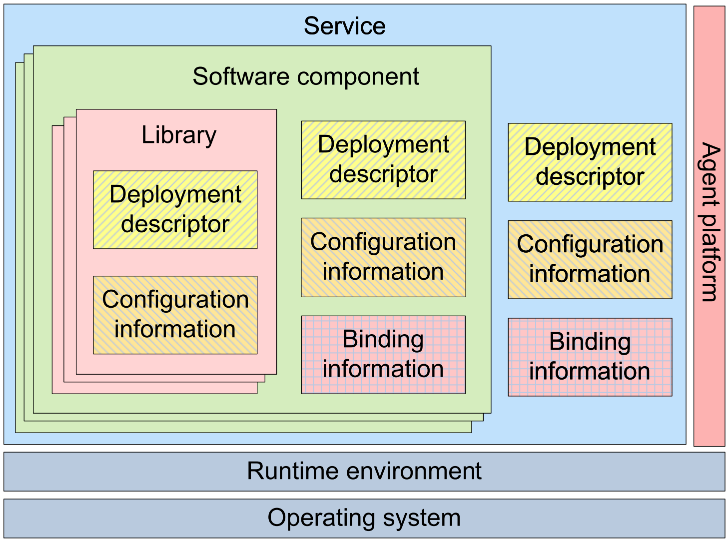


Fig. 1. Service composition

software component is composed out of libraries and additional information about the software component.

A library is a component that can be reused by different software components as well as other libraries. In terms of a Java programming language a library can be seen as a .jar file that contains an API with some functionality. This .jar file can be used in any application that requires it. The same applies for .dll in .NET based applications. Every library can have configuration data that must be defined before it can be used and it also must have a deployment descriptor that defines other libraries it depends on if any.

A software component consists of: software deployment descriptor, software configuration data, libraries and binding information. Software deployment descriptor defines dependencies this software component has on other libraries and other software components. Let's take our framework for analysing service usage as an example. This framework must have access to network nodes in the provider's network that provide different features (e.g. billing, resource allocation, authorization). Thought these nodes it can receive user's service usage information. It can also have an option to send the processed data to a web portal used to represent data. Deployment descriptor also contains information about the environment that is required to start this software in terms of the operating system used, memory allocation, network bandwidth etc. Binding information defines how software components connect to other software. Some binding types might require binding code to be generated dynamically during deployment. An example of this case is when web services are used for communication.

A service consists of a deployment descriptor, service configuration, binding information and software components. Deployment descriptor defines from which software components this service consists of, the location from where it can be obtained as well as access information to the repository if any. Service configuration is used to define how the service is going to be configured. This configuration also influences how each software component is going to be configured. Binding data defines how software components are going to be binding together. It also influences how the binding data for each software component is going to be performed.

Services are run by a certain runtime environment (e.q. Apache server, Java runtime environment, net framework runtime etc.) running on an operating system. Agent platform is used to deploy and manage individual services. This process will be explained in detail in the following sections.

3 Multi-agent Service Deployment

The architecture shown in Fig. 2 shows all the entities from the Telco 2.0 model [2] and the agents or groups of agents that are provided by them. Those agents are:

- *Team Leader Agent (TLA)* - gathers information about the service, software components or the libraries that have to be deployed. It also finds a team of agents that will perform the deployment and performs an initial allocation of tasks;
- *Task Execution Agent (TEA)* - used to perform a task. It has the knowledge to perform a certain type of a task and to handle any exceptions that may occur;
- *Capability Agent (CA)* - This agent has information about the agents that can perform tasks. That information includes agents owner information, information about the tasks agent is capable of performing and agent performance factor;

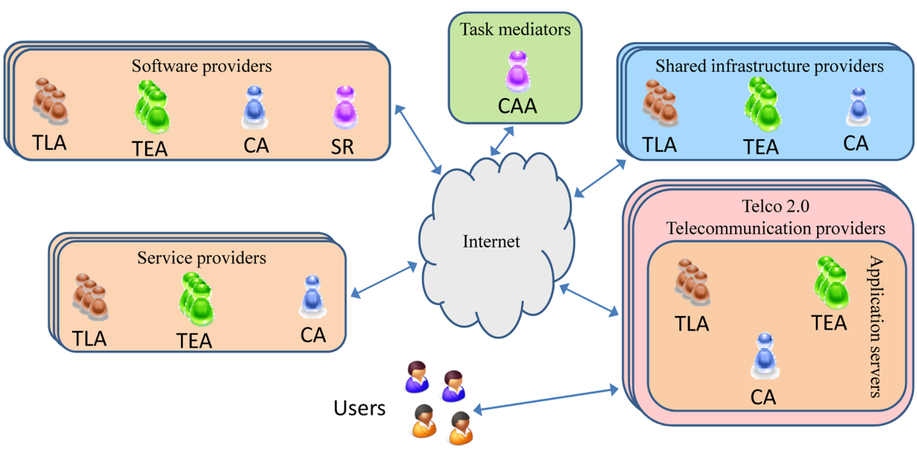


Fig. 2. Agents in telecommunication environment

– *Software Repositories (SR)* - is used to store all software components and libraries;
– *Capability Advertisement Agent (CAA)* - can be considered as a job market. The TLA agent advertises the tasks that needs to perform. Other TEA can then search through all the advertised tasks and find tasks that match their capability. This agent is used only when TLA agent cannot find any matching agents in its CA. Since the CAA agent is provided by third-party agent's performance factor has very little influence while searching for agents.

The process of deploying services by using software agents is divided into two phases: *Information gathering and task allocation* and *Deployment* as shown in Fig. 3. These phases will be explained in detail in the following sections.

3.1 Information Gathering and Task Allocation Phase

The information gathering and task allocation phase consists of the following steps: deployment task definition, information gathering, deployment plan generation, search for suitable agents and initial task allocation.

Deployment Task Definition is the first step in the process of deploying services. In this step the administrator has to define which services should be deployed and where to deploy them. He is provided by a GUI that allows him to create all the deployment parameters of a service: a deployment descriptor, service configuration and binding information.

Information Gathering step involves going through the deployment descriptor of a service and creating a dependency tree for all the software components and libraries the service requires. TLA agent has to download all the deployment descriptors of all the software components. If software component has further

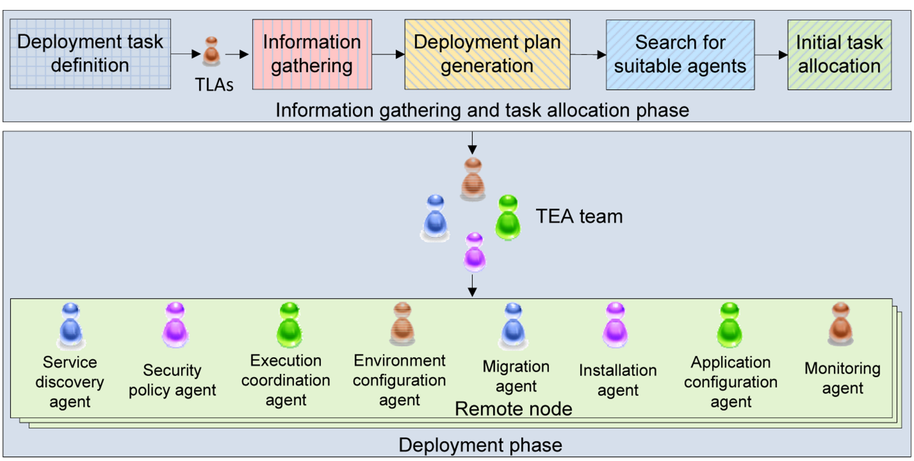


Fig. 3. Software deployment process

dependencies then these steps have to be repeated until all the dependencies are gathered and a dependency tree is generated. After creating a dependency tree the TLA creates a team of information gathering agents (IGA). These agents migrate on to the remote nodes and gather information about them (e.g. software components that are already installed on them, binding information, hardware capabilities, operating system).

Deployment Plan Generation is used to generate a plan that defines which software or software components have to be deployed on each of the target remote nodes. This is accomplished by matching dependency tree created in the previous step with the information the TLA gathered on the remote nodes. The goal of this matching process is to exclude software components that are already present on the remote nodes from the final deployment plan.

Search for Suitable Agents it finds agents that are going to perform the tasks from the deployment plan. This step is important since software agents can learn to deploy specific technologies. Agents can be specialized for one particular deployment type, one step of its procedure or even several types depending on the complexity. The goal of each TEA is to adapt its capability to maximize its performance as well as its workload.

TLA can locate suitable agents by using three methods. The first method is search in his internal database of agents. This database contains a list of agents and agent capabilities TLA has worked before in the past. Each agent is assigned a performance factor. This factor is used by the TLA to determine the quality of the agent. The factor increases every time an agent performs a task successfully or decreases if it fails.

The next method for the TLA is to search in the CA. If the TLA belongs to a larger organization that has his own CA then it will first perform this search locally. Local CA contains information about TEAs provided by this organization as well as agents from other organizations that have partner contracts. Local CA also has a performance factor record for every agent registered in it.

The last step is to search in the CAA. It is a third party agent that allows TLA to advertise its need for TLAs with a particular capability. The TLA advertises its need for agents in it. Every agent can then search through all the advertised tasks and find one they are capable of performing. The CAA can be used by an organization to find new partners that could perform deployment tasks.

The agents found by the TLA form a team of agents that are going to deploy the service. The next step for the TLA is to perform an *initial task allocation*. This step involves distributing all the tasks in a way that maximizes the deployment time as well as reduces the probability of deployment failure. This is accomplished by assigning difficult deployment tasks to agents with high performance factors. After initial task allocation each agent has to analyze its task, decompose it into several subtasks if any. It also has to analyze the entire deployment plan and notify the agents that are executing the task its tasks depend on. This is required when a deployment of one software component depends on some other. The agent deploying the software component has to wait until the deployment of the dependent software component is complete.

Among all the described agents only the TEA and the IGAs migrate. The TEAs migrate on remote nodes where they execute tasks service deployment is decomposed to. The IGAs migrate on to remote nodes to gather information about the hardware and software on the nodes.

3.2 Deployment Phase

In the deployment phase agents that form a team must coordinate their actions and perform all the tasks assigned to them. Except the agents described in the previous sections remote nodes must support the execution of agent operations. There are eight operation support agents (Fig. 3) where each is given only a subset of functionality. Each agent is self dependent and does not require other agents to perform its functions. Together these agents provide all information and functions required during deployment. These agents are:

- *Service Discovery Agent* contains a list of all the services on the node. Except information about installed services on the nodes the agents also has information about different nodes in the network as well as hardware capabilities of the node. One example of such nodes are the billing system, authorization, authentication nodes etc. During the *information gathering* step the IGA used information provided by this agent;
- *Security Policy Agent* restricts access to the functionality within the network to all the agents. All the messages sent to the agents located on the remote nodes must first pass through a security check. The agent analyses the sender, the receiver of the message, owner of the sending agent and which operation this agent is trying to perform and checks its security policies. If the agent does not have permission to perform this operation then the message is rejected otherwise it is forwarded to the corresponding remote agent.
- Agents specialized for certain types of tasks increase the possibility of parallel execution of those tasks. Problem with parallel execution is that all tasks must be synchronized. We need to prevent two agents from performing tasks that are in conflict with one another. An example is when one agent tries to update a software component on one node while the other tries to remove it. The task of preventing such conflicts is handled by the *Execution Coordination Agent*. To prevent conflicts the TLA must notify this agent which tasks its team will do. The notification is sent during the information gathering step. This will only prevent the removal of the services or their reconfiguration that effect TLA's tasks (at least without notifying the TLA agent). In case something changes between information gathering and the execution the TLA will have to change the deployment plan;
- *Environment Configuration Agent* has the task of monitoring application servers, configuration and which applications are running on them. It is also responsible for configuring any parameters required for applications that are to be installed on them. This information is used later during deployment to check which applications are already installed and to configure the application servers if required by the application being deployed;

– The next four agents are used to directly support the execution of tasks by the agents. Those agents are *Migration Agent* which handles any operations dealing with temporary storing software for other tasks, *Installation Agent* which handles installation tasks, configuration tasks are assigned to *Application Configuration Agent* and *Monitoring Agent* which is used for logging and handling states of the software.

The described agents will be used by TEA to execute the tasks. The exact process how the deployment is performed depends on the tasks that the agent is trying to perform but each agent must follow certain guidelines. When the TEA agent arrives at the node first it has to notify the Environment Configuration Agent about the operation it is trying to perform. Then the TEA contacts one of the agents that directly support the execution of tasks. When it reaches the configuration phase is uses the Service Discovery Agent to retrieve the binding information for services in the network it needs. Note that the TEA does not perform the tasks directly on the nodes. Each execution support agent has a certain set of operations that can be performed on it. When the TEA wants to execute a task it sends information about what has to be done to the corresponding agent. It then performs a task and notifies the agent about the result. Once all the agents perform all the tasks they notify the TLA which then notify the administrator.

The learning agents are the TEA agents. If an unexpected error occurs during software deployment this agent has to find the cause of the error. The first step is to try to find some other agent that knows how to resolve this error. If such an agent can be found the agent will learn from him how to resolve it otherwise it will inquire the administrator how to resolve it. After the plan it will remember how this error can be resolved and will apply this same plan next time the same error occurs.

4 Conclusion

The introduction of Telco 2.0 model into the telecommunication domain has changed the role of the telecommunication provider. The change affects not only the services that are offered to the users but the organization of its infrastructure as well. This combined with the increased number of services and actors on the telecommunication market has complicated the process of service deployment. In this paper we have presented an agent based framework that can be used to reduce the complexity of service deployment. Agents in this case can reduce the complexity first by hiding the implementation of a service from the user. When a user wants to deploy a new service it only has to know where the service has to be deployed. It does not have to now how the service is implemented or how it will be hosted on the application servers. This process is performed by agents that know how to deploy each type of service. The second advantage of using agents is that they can learn what to do when an unexpected error occurs during service deployment. The first time an error occurs the agents can find some other agent or the administrator to help him it solve the problem. The next time agent will know how to solve it by itself. This allows the agents to increase to efficiency of the deployment process.

Acknowledgments

This work was carried out within research project 036-0362027-1639 "Content Delivery and Mobility of Users and Services in New Generation Networks", supported by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

1. Podobnik, V., Lovrek, I.: Multi-agent system for automation of B2C processes in the future internet. In: Proceedings of IEEE Conference on Computer Communications (INFOCOM) Workshops, pp. 1–4 (2008)
2. Yoon, J.L.: Telco 2.0: a new role and business model. IEEE Communications Magazine 45(1), 10–12 (2007)
3. Benali, O., El-Khazen, K., Garrec, D., Guiraudou, M., Martinez, G.: A framework for an evolutionary path toward 4G by means of cooperation of networks. IEEE Communications Magazine 42(5), 82–89 (2004)
4. Fischer, Lorenz: Internet and the future policy framework for telecommunications. Technical report, Fischer & Lorenz (2000)
5. Kapov, L.S., Mosmodor, M., Dobrijevic, O., Matijasevic, M.: Application-level qos negotiation and signaling for advanced multimedia services in the IMS. IEEE Communications Magazine 45(7), 108–116 (2007)
6. Carlander, M.L., Olander, J.: Service delivery platforms for the multimedia marketplace. Ericsson Review 1(2), 70–72 (2008)
7. Johnston, A., Gabrielsson, J., Huysmans, C.C.M., Olsson, U.: Evolution of the service delivery platforms. Ericsson Review (1), 19–25 (2007)
8. Carraro, G., Chong, F., Pace, E.: Efficient software delivery through service delivery platforms. The Architecture Journal (12), 7–13 (2007)
9. Gucer, V., Olivieri, A., Campbell, D., Shah, F.S.G., Raeymaeckers, J., Remes, M., Choilawala, M., Hamelin, P., Berens, S., Fernandez, I.: IBM redbooks, deployment guide series: IBM tivoli provisioning manager version 5.1, http://www.redbooks.ibm.com/abstracts/sg247261.html?Open (07.10.2008)
10. Rellermeyer, J.S., Alonso, G., Roscoe, T.: Building, deploying, and monitoring distributed applications with eclipse and r-osgi. In: eclipse 2007: Proceedings of the 2007 OOPSLA workshop on eclipse technology eXchange, pp. 50–54. ACM, New York (2007)
11. Sudmann, N.P., Johansen, D.: Software deployment using mobile agents. In: Bishop, J.M. (ed.) CD 2002. LNCS, vol. 2370, pp. 97–107. Springer, Heidelberg (2002)

Collision Detection for Ubiquitous Parking Management Based on Multi-agent System*

Romeo Mark A. Mateo, Young-seok Lee, and Jaewan Lee

School of Electronic and Information Engineering, Kunsan National University
68 Miryong-dong, Kunsan, Chonbuk 573-701, South Korea
{rmmateo,leeys,jwlee}@kunsan.ac.kr

Abstract. Most of the researchers and designers of WSN-based parking management system used wireless sensors to detect the car presence in automating the parking system. However, the problem of car-to-car collisions in the parking system was not discussed by these researches where the position details of cars are important to analyze the collision event. This paper presents the ubiquitous parking management system (UPMS) where the sensors are used to detect car presence and designed in multi-agent system to disseminate efficiently the information through the system. The proposed system uses a location technique based on RSS which is processed in multi-lateration method to know the position details of the cars. Also, the car active RFID (CARFID) is equipped with vibration sensors to detect the car collision event. The system verifies the collision event in the proposed oriented bounded box (OBB) intersection test. Experiment shows that the collision detection is more accurate if more wireless sensors are used in our proposed technique.

1 Introduction

The current research studies in parking management system have realized the usefulness of ubiquitous technologies to automate car monitoring and provide smart processing of information in wireless environment. Parking management system using wireless sensor networks (WSN) are popular studies of research. Most researches implement wireless sensor devices in sensing presence of car and monitoring location of cars [1, 2]. This method implements WSN to sense the presence of cars and monitors the location slots. Relevant services like car locator system, parking negotiator and other applications in parking system rely on the sensing methods to provide the basic information for parking system management.

Another issue in parking management is there are no means of determining automatically the collision events inside the parking area. An example of a situation is when a moving car, on its way out of the parking area, accidentally hit a parked car. The owner of the parked car which is not in the scenario cannot know that his or her car was hit by another car. In the case of moving cars, the car caused the collision is

* This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 570–578, 2009.

determined by the detailed position, speed and direction of each car. To solve these problems, the system should use a collision detector like vibration sensor, embedded in the car. The collision is detected by the vibration sensor and sends message to the system or directly to the car owner in real-time. Our paper studies the collision detection in parking environment using wireless sensors. There are three cases of collisions with the involvement of cars in a parking area which as follows:

- A moving car collides to another non-moving car
- A moving car collides to another moving car
- An unknown object (non-car) collides to a non-moving or moving car

In the first case, we know that the driver of the moving car has the fault of collision. The only problem is that the car caused the collision can avoid the responsibility because there is no system to identify the car and to detect the event. In the second case, we can know its either or both cars caused the collision. This is proved by the given details of the event like current position, speed, direction and others. To determine the details, each car must have identification and there must be a location technique to estimate the position of cars in the system. In the last case, it is impossible to identify the cause of the collision except if there are cameras monitoring the parking area. In our study, our concern is to tackle the first and second cases.

In this paper, we present the design of ubiquitous parking management system (UPMS) where wireless sensors are used to detect important events in the parking area and the design is based on multi-agent system to disseminate efficiently the information through the system. This study proposes a car collision detection using a vibration sensor and verification method based on the oriented object box (OBB) intersection test. The proposed collision detection uses the location technique based on radio signal strength [3] which is processed in multi-lateration method to minimized errors in position estimation of cars. The collision verification provides a fast OBB intersection test by using the nearest edge point of each object to separate the objects.

2 Related Works

2.1 WSN-Based Parking Systems

The current contributions of studies in parking system are providing smart monitoring of cars by using WSN. Wireless sensor devices are used for automatic vehicle car parking [4] where wireless sensors and infrareds are used for the positioning to enhance the accuracy of positioning. A parking monitoring system using WSN is proposed [5]. The status of the parking field detected by sensor nodes is reported periodically to a data-base via the deployed wireless sensor network and its gateway. The database can be accessed by the upper layer management system to perform various management functions, such as finding vacant parking lots, auto-toll, security management, and statistic report. In [2], describes a simulation where reports from wireless sensor nodes are passed from car-to-car in order to achieve scalable dissemination of information regarding parking spaces. A comparison of data from different sensors for improvements to WSN in car parking system is presented [6]. Most of these studies do not tackle the collision problems in parking environments. In the collision event, a sensor that can sense the change of position of a non-moving car is needed. However, in identifying the cause of collision, we need to determine the details like position, speed and direction of cars.

2.2 Real-Time Collision Detection

Collision detection is fundamental to variety of applications including video games, virtual-reality modeling, geometric modeling, and robotics for simulation of interactions between moving objects. Collision detection algorithms are basic component of 3D video games to ensure the illusion of a solid world. In robotics, collision detection is used in path planning to guide the robots to steer away from obstacles. To translate the intersection of objects in the graph, bounding volume is needed to encapsulate one or more objects [7]. The efficient algorithm requires a simple volume to have a cheaper test in determining the intersection based on the bounding volumes. The Axis-Aligned Bounding Boxes (AABBs) is one of the most common bounding volumes but is limited to determine the intersection accurately. Another test is the Oriented Bounded Boxes (OBB) which is more accurate compared to AABBs and less complex than other tests. An exact test for OBB-OBB intersection can be implemented in terms of the separating axis test. Two OBBs, represented by A and B, and radii of each objects represented by r, are separated if,

$$|T \cdot L| > r_A + r_B \qquad (1)$$

where L is the translated axis and T is the distance between the centers of the two objects. For OBBs it is possible to show that at most 15 of these separating axes must be tested to correctly determine the OBB overlap status. It is also possible 6 of the test are enough to determine the separation.

3 Ubiquitous Parking Management Based on Multi-agent System

The architecture of the proposed ubiquitous parking management system is consisted of three layers: ubiquitous network, middleware and application services layers. In the ubiquitous network, represents the physical networks of different sensor devices and computers communicating in the wireless environment. The components in the middleware layer are transparently executing for the efficiency on managing data from the ubiquitous network layer. Interaction of clients and application services are also handled by the middleware layer. The application service layer is consisted of services for parking management system. Before the car enters the parking area, a screen for available slot is displayed in an LCD screen. On entering, the RFID reader scans the ID of the car. After choosing the empty slot, the cars will be parked and it is sensed by the car detector sensors. This information are sent to the wireless access points and forwarded to the agents for processing of information and car monitoring.

In Figure 1, agents in the UPMS are shown. The proposed framework uses multi-agent approach to provide an intelligent distribution of task and efficient information dissemination within the system is shown in Figure 1. The pattern recognizer agent

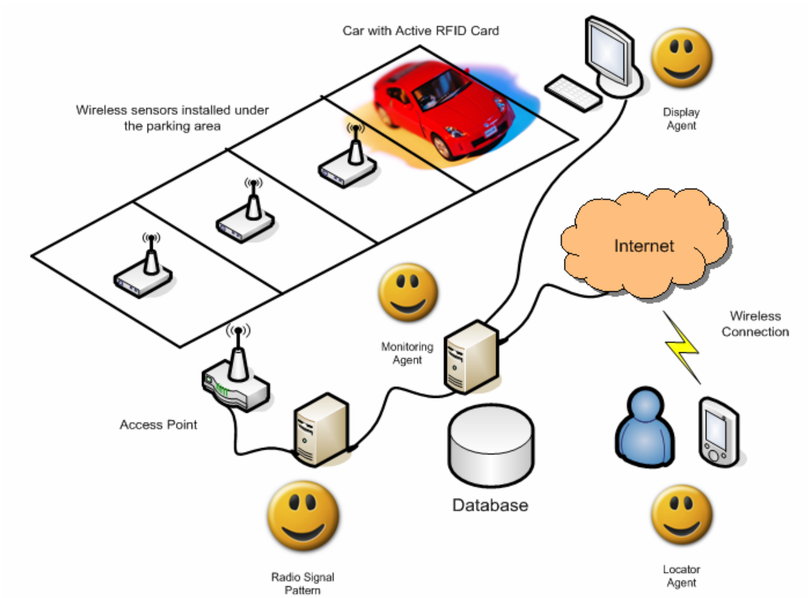


Fig. 1. Application Services and Middleware Layer: Agents in the parking system

(PRA) processes the signal patterns from each parking sensors and determine the location of the car. An access point collects the radio signal strength (RSS) data from the sensors. The PRA sends the information the monitor agent and then it will store the parking event on its database. Real-time display of current parked cars is done by the display agents on each parking area. This communicates with the monitoring agent to obtain the current vacant slots and displays the information in a monitor. Locator agents are agents from a car owner installed in its portable or mobile computers. The multi-agent components are defined as follows:

Monitoring Agent – center communication of the agent system. This stores the information gather from all agents and process the request of information of an agent. The requesting agents are verified by an authentication procedure. If the agent has no privilege of the request then it denies the request.

Locator Agent – installed in a car owner's mobile phone or personal digital assistant (PDA). It requests for the current location slot of the car and informs the owner of the important events inside the parking area. This agent communicates to the monitoring agent where the car is located by using car ID.

Pattern Recognizer Agent – used in location technique for collision detections of cars inside the parking area. This agent processes the pattern classification of signals to determine the location of each car.

Collision Analyzer Agent – detects and verifies the collision inside the parking area. This uses the proposed collision detection. The result from detection and verification processing is sent to the monitoring agent to inform about the event to the affected car owners.

Display Agent – displays the current available slots and other information for the car that entering the parking area. This agent communicates to the monitoring agent for the new information of parking vacancies to be displayed.

4 Collision Detection Agent

The proposed collision detection agent handles the detection and verification of collisions occurs inside the parking area. First, detection of the collision is gathered by the sensors. We used a vibrator sensor module to detect the collisions. Each car inside the parking area is provided with the car active RFID (CARFID) with vibration sensor. The vibration sensor is activated manually by the car owner every time the car is parked in a slot. This sensor is active until the car owner returns to the car and manually deactivates the sensor. The collision event is detected by a sensor based on a threshold value (Φ) of acoustic level (v_i). If the sensor senses a collision by $v_i > \Phi$, the CARFID sends the signal to the parking sensors, S $=\{s_1, s_2,\ldots, s_n\}$, in the parking area to process the collision detection method of the proposed agent. Parking sensors are deployed under the parking slots. These sensors contribute in processing the collision verification of each car inside the parking area. The verification method starts in calculating the distance of each car presented in Equation 1, where P_{tx} is the transmission power, α is the path loss model, P_{rcvd} is the signal strength is used to solve the r or distance.

$$P_{rcd} = c\frac{P_{tx}}{r_i^{\alpha}} \Leftrightarrow r_i = \alpha\sqrt{\frac{cP_{tx}}{P_{rcd}}} \tag{2}$$

The distance value from Equation 2 is used for location estimation procedure in calculating the multi-lateration [7] of the location estimation of each car. The r is the hypotenuse of a triangle between two known coordinates (x and y) of an anchor or node (n) and Equation 3 is a matrix to calculate the multi-lateration method.

$$2\begin{bmatrix} x_n - x_1 & y_n - y_1 \\ & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} \end{bmatrix}\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (r_1^2 - r_n^2) - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) \\ : \\ (r_{n-1}^2 - r_n^2) - (x_{n-1}^2 - x_n^2) - (y_{n-1}^2 - y_n^2) \end{bmatrix} \tag{3}$$

This is a derivation from the Pythagorean Theorem where at least three anchors are needed to solve the unknown coordinate. However, the output from the received signal r can have errors because of the obstructions and interference inside the parking are. To minimize the errors in getting the approximate value from the procedure, the system needs more information by setting additional anchors. These anchors are represented by n in the Equation 3. The function on Equation 3 can be derived in Equation 4.

$$\mathbf{A}^{T}\mathbf{A}\mathbf{x}=\mathbf{A}^{T}\mathbf{b} \tag{4}$$

The minimization of each signal in determining the location estimation of a car is shown in Equation 4. This is also called normal equation for the linear squares problem. This information is gathered by the pattern recognizer agent and passes the location information to the collision detection agent. After sending the known location to the proposed agent, the proposed agent selects the cars that are nearest to the car that triggered the collision. The position of each selected cars are determined by basing from its previous locations. Assume that the CARFID sensor is placed at the center and has a default size. The current position of car is parallel with the line drawn from the previous and current coordinate of the CARFID sensor. After determining the position, the procedure calculates the separation lines of the cars. The *L* is original *y* axis coordinate and translated to another vector based on the coordinates edges and get the value of *x* where *y* is 0 and then adds the distance to original vector *L* to get *L'* and the angle is now calculated by $\cos^{-1}|\mathbf{L}||\mathbf{L'}|$. The *L'* constructs the new coordinate and is the basis of calculating Equation 1. At the start of the separation test, the nearest edges from other cars or objects are determined to draw the separation line shown in Equation 5.

$$edge = \min\{\sum_{j=1}^{I} \left| e_{ij} - c_k \right|\} \tag{5}$$

The Euclidean distance of the edge point *j* of object *i* and the object center *k* is calculated. After choosing the edges from Equation 5, these edges are used to calculate the separating points. The first length is calculated by the scalene triangles formed by the length from the centers of objects (edge_{eA} and edge_{eB}) and the length from the edge chosen to each object center. The proportion value from adding the two lengths in each edge length is used to calculate the r_A. The proportion value of edge_{eA} is multiplied to the *L'*. Also, the same procedure is done to other object. This procedure is done twice from object *A* to *B* and *B* to *A*. If the result of two tests is either true from Equation 1 then it is said there are no collision happened.

Figure 2 illustrates the separation procedure between two cars using their sensor to locate each position placed at the center of cars. These calculations are done by the

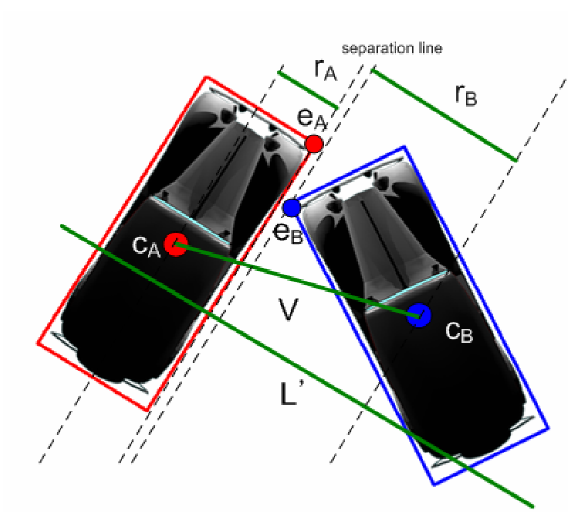


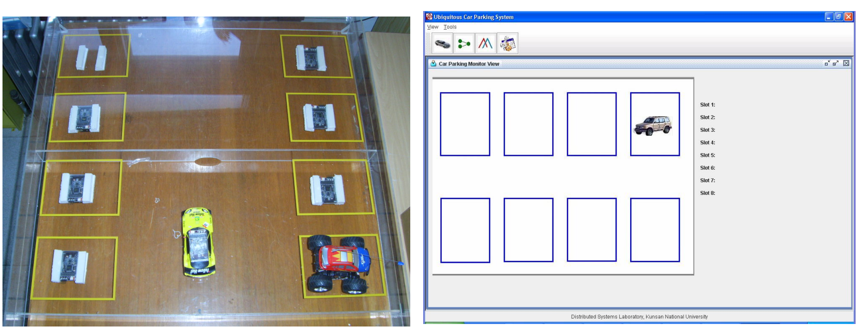
Fig. 2. Separation lines of two cars using the vector line from center point of each car

collision detection agent after the pattern recognizer agent sends the location information of the cars involved in the collision.

5 Simulations

Our design of the ubiquitous parking management system (UPMS) used wireless sensor motes which is a 2.4 GHz IEEE 802.15.4 compliant RF transceiver. The active RFID with the vibration sensor is put on the top of the car model every time we simulate our proposed technique. Every time the wireless sensor detects the presence of the car, it sends a message to the active RFID. The proposed collision detection agent and pattern recognizer agent (PRA) were implemented in the ubiquitous parking management system simulator. The program for the active RFID was coded and designed in nesC, which a programming for sensors and the agents were programmed in Java. Each ID is encoded in a wireless sensor mote to simulate the active RFID. The area of the CARFID is 8.5 cm by 5.5 cm which is equal to the size of an ATM card and the thickness is about 1 cm. We used the same frequency of the wireless sensor motes to the CARFID. CARFID is used for the location technique based on pattern recognition of RSS [3]. The program of the CARFID, to interact with the wireless sensors, is coded and designed in nesC as well as the wireless sensors. PRA which is used to receive and process the RSS is programmed in Java. The IDs of each wireless sensor which is for the slot location and CARFID which is for the car identification were included in the program. Also, the collision event is processed in the CARFID. If collision event is detected then it sends the information to each parking sensor to process the collision verification of the proposed collision detection agent.

The interaction of the parking platform simulator is presented in Figure 3. The area of the parking simulator platform is 2.4 x 0.1 m and placed at inside of an office type room. Each slot area is 18 x 13 cm with a boundary of 3.5 cm between each slot and drive way of 36 cm. The parking platform is consisted of 8 slots embedded with wireless sensors. In Figure 3a, a car parked in a single slot and at the same time in Figure 3b shows the output display from the parking platform simulator. The output



(a) (b)

Fig. 3. The ubiquitous parking management system simulator (a) and parking monitor program displays the output from the platform (b)

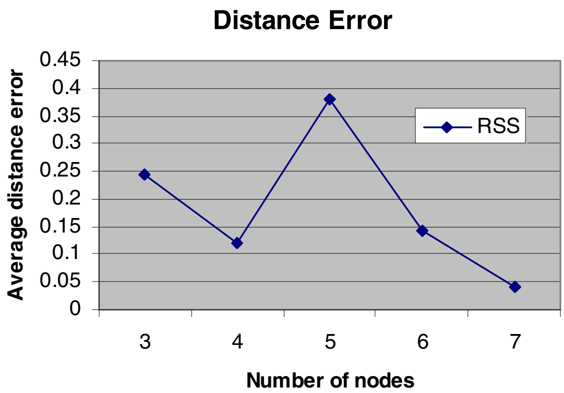


Fig. 4. Distance error in meters using the location estimation based on RSS by increasing the number of nodes

display is updated every second and the values are stored in the database. Also at the same time, the active RFID from the car model sends data to all sensors. Each sensor determines the RSS from the active RFID and sends the values to the sink node with the pattern recognizer agent. The PRA gathers RSS data and associates the slot address where car presence is detected by the wireless sensor mote. After knowing the identification of the car, the information is stored in the database of the system. This procedure is also used for the car collision scheme by determining the location and position of the car using the details on the previous section (Section 4). If there is a collision event detected by the sensor, the system will notify the car owner of the event to their car locator agent.

The accuracy of the location technique in determining the position of cars is critical in the procedure of the collision detection. This includes the information needed to determine the current positions of the car by the previous and current coordinates, and the calculation of the separating lines between cars. An area of 10 meters by 10 meters is used to simulate and evaluate the location scheme. Also an obstruction or interference was introduced in adding the fifth sensor. In Figure 4, the distance error of the multi-lateration by gradually adding sensors in processing the proposed scheme is determined by a line graph. We observed in the graph the peak error is on adding the 5^{th} sensor. The error from the 5^{th} sensor is increased because of the obstruction and interference in the parking area. If we add more sensors, the error is compensated by the additional nodes until it reduces to 0.04.

6 Conclusion and Future Works

This paper shows the design of ubiquitous parking management system (UPMS) where wireless sensors are used to detect important events in the parking area. Also, the design is based on multi-agent system to disseminate efficiently the information through the system. This study presents a car collision detection scheme using vibration sensors and verification method based on the oriented object box (OBB) intersection test that is implemented by the proposed collision detection agent. The proposed collision detection agent interacts with the pattern recognizer agent to locate the position of cars. The collision verification provides a fast OBB intersection test by using the nearest edge point of each object to separate the objects. We observed in the graph that there is a peak error on adding the 5^{th} sensors but if we add more sensors, the error can be compensated by the additional nodes.

This paper contributes on the minimization of calculating the separation lines of collision detection by introducing the nearest edges point. The optimization of determining the position of cars and location estimation is still considered as future work.

References

1. Benson, J.P., O'Donovan, T., O'Sullivan, P., Roedig, U., Sreenan, C.: Car Park Management using Wireless Sensors. In: Proc. of IEEE Conference on Local Computer Networks, pp. 588–595 (2006)
2. Miura, S., Zhan, Y., Kuroda, T.: Evaluation of Parking Search using Sensor Network. In: Proc. of International Symposium on Wireless Pervasive Computing, p. 6 (2006)
3. Mateo, R.M., Yoon, I., Lee, J.: Location Technique based on Pattern Recognition of Radio Signal Strength using Active RFID. In: Proc. of IWRT, pp. 7–18 (2008)
4. Chou, L., Sheu, D.C.C., Chen, H.W.: Design and Prototype Implementation of a Novel Automatic Vehicle Parking System. Proc. of International Journal of Smart Home 1(1), 11–16 (2007)
5. Tang, V.W.S., Zheng, Y., Cao, J.: An Intelligent Car Park Management System based on Wireless Sensor Networks. In: Proc. of International Symposium on Pervasive Computing and Applications, pp. 65–70 (2006)
6. Kumar, R., Chilamkurti, N.K., Soh, B.: A Comparative Study of Different Sensors for Smart Car Park Management. In: Proc. of International Conference on Intelligent Pervasive Computing, pp. 499–502 (2007)
7. Ericson, C.: Real-Time Collision Detection. Morgan Kaufmann Publishers, San Francisco
8. Karl, H., Willig, A.: Protocols and Architectures for Wireless Sensor Networks. John Wiley & Sons, Ltd., Chichester

Compatibility and Conformance of Role-Based Interaction Components in MAS

Nabil Hameurlain

LIUPPA Laboratory, Avenue de l'Université, BP 1155, 64013 Pau, France
nabil.hameurlain@univ-pau.fr
http://www.univ-pau.fr/~hameur

Abstract. This paper deals with compatibility and conformance of role-based interaction components in MAS, namely in which case the first role component is a model (specification) and the second an implementation (or a refined model). First, we provide a semantics for the implementation of roles, which is defined upon alternating simulation relation, and show that the proposed semantics is compositional for role's composition. Then, we prove the independent role's implementability property, expressing that roles can be implemented independently of each other once their specifications are known. Finally, we study the preservation of the compatibility of roles by the proposed role's implementation semantics.

1 Introduction

It is now widely recognized that interaction is one of the most important characteristic of complex software. In Multi-Agent Systems (MAS), software architectures are open and dynamic, where heterogeneous components are naturally represented by interacting autonomous agents, which can enter or leave the system at will. Interactions among these autonomous agents is fundamental since, agents belonging to the same application need to interact and coordinate their activity to carry out their common global goal, whereas, agents belonging to different applications, as in an open scenario, may also need to interact, for instance, to compete for a resource.

Modeling interactions by roles [7] allows a separation of concerns by distinguishing the agent-level and system-level concerns with regard to interactions. In this setting, roles are basic buildings blocks for defining the behavior of agents and the requirements on their interactions.

In our previous work [9], we defined RICO (Role-based Interactions COmponents) model for specifying complex interactions based on the roles. RICO proposes a specific definition of role, which is quite simple and can be exploited in specifications and implementations. In the RICO model, when an agent intends to take a role, it creates a new component (i.e. an instance of the component type corresponding to this role) and this role-component is linked to its base-agent. Then, the role is enacted by the role-component and it interacts with the role-components of the other agents. In [10] a Petri-net based formal specification for RICO is given together with their behavioural compatibility and substitutability semantics depending on the context of use

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 579–588, 2009.

(environment). The proposed approach uses the concept of usability of roles by the environment that is a safe utilizing of the roles by that environment, and then provides a formal framework for modeling usable role-components and their composition.

This paper deals with the conformance of roles in MAS that is in which case the first role component is a model (specification) and the second an implementation (or a refined model), and the contributions are: (1) to provide a semantics for the implementation of roles, which is defined upon alternating simulation relation [1], and show that the proposed semantics is compositional for role's composition (2) to prove that roles can be implemented independently of each other once their specifications are known, (3) to study the existing link between compatibility and implementation of roles, namely the preservation of the compatibility of roles by role's implementation.

The structure of the paper is as follows. Section 2 presents our Role-based Interactions COmponents (RICO) specification model which is based on the Components-nets formalism that combines Petri nets with the component-based approach. In section 3 we provide a compatibility relation between roles defined upon a usability which is related to the proper termination property of their composition with an environment. In section 4, we give a semantics for role's implementation and show that this semantics is compositional for role's composition. In this section, we also show the consistency of this semantics w.r.t the compatibility that is the preservation of the compatibility relation by the role's implementation semantics given in this paper. In section 5, we present conclusion and related approaches.

2 Role-Based Interaction Components Modeling

2.1 The Component-Nets Formalism (C-Nets)

Backgrounds on Labelled Petri nets. A marked Petri net $N = (P, T, W, M_N)$ consists of a finite set P of places, a finite set T of transitions where $P \cap T = \emptyset$, a weighting function $W : P \times T \cup T \times P \rightarrow \mathbb{N}$, and $M_N : P \longrightarrow \mathbb{N}$ is an initial marking. The preset of a node $x \in P \cup T$ is defined as ${}^\bullet x = \{y \in P \cup T, W(y, x) \neq 0\}$, and the postset of $x \in P \cup T$ is defined as $x^\bullet = \{y \in P \cup T, W(x, y) \neq 0\}$. Let $M: P \longrightarrow \mathbb{N}$ be a marking of the Petri net. A transition $t \in T$ is enabled under a marking M, noted M[t >, if $W(p, t) \leq M(p)$, for each place p. In this case t may occur, and its occurrence yields the follower marking M', where $M'(p) = M(p) - W(p, t) + W(t, p)$, noted M[t> M'. The enabling of a sequence of transitions $\sigma \in T^*$ and its occurrence are defined inductively, noted for simplicity M[σ> M'. Let A be a set of methods, that is the alphabet of observable actions, and $\{\lambda\}$ denotes special unobservable actions. The symbol $\{\lambda\}$ plays the usual role of an internal action. We denote as $LN = (P, T, W, M_N, l)$ the (marked, labelled) Petri net in which the events represent actions, which can be observable. It consists of a marked Petri net $N = (P, T, W, M_N)$ with a labelling function $l: T \longrightarrow A \cup \{\lambda\}$. Let ε be the empty sequence of transitions, l is extended to an homomorphism $l^*: T^* \longrightarrow A^* \cup \{\lambda\}$ in the following way: $l(\varepsilon) = \lambda$ where ε is the empty string of T^*, and $l^*(\sigma.t) = l^*(\sigma)$ if $l(t) \in \{\lambda\}$, $l^*(\sigma.t) = l^*(\sigma).l(t)$ if $l(t) \notin \{\lambda\}$. In the following, we denote l^* by l, LN by (N, l), and if $LN = (P, T, W, M_N, l)$ is a Petri net and l' is another labelling function of N, (N, l') denotes the Petri net (P, T, W, M_N, l'), that is N provided with the labelling l'. A sequence of actions $w \in A^* \cup \{\lambda\}$ is enabled under the marking M and its occurrence yields a marking M', noted M[w>> M', iff either M = M' and w = λ or there exists some sequence $\sigma \in T^*$ such that $l(\sigma) = w$ and M[σ> M'. The first condition accounts for the fact that λ is the label image of the empty sequence of transitions. For a marking M, Reach (N, M) = $\{M'; \exists \sigma \in T^*; M[\sigma> M'\}$ is the set of reachable markings of the net N from the marking M.

Components nets (C-nets). The Component-nets formalism [9] combines Petri nets with the component-based approach. Semantically, a Component-net involves two special places: the first one is the input place for instance creation of the component, and the second one is the output place for instance completion of the component. A C-net (as a server) makes some services available to the nets and is capable of rendering these services. Each offered service is associated to one or several transitions, which may be requested by C-nets, and the service is available when one of these transitions, called *accept-transitions*, is enabled. On the other hand it can request (as a client) services from other C-net transitions, called *request-transitions*, and needs these requests to be fulfilled. These requirements allow focusing either upon the server side of a C-net or its client side.

Definition 2.1 (C-net)
Let CN = $(P \cup \{I, O\}, T, W, M_N, l_{Prov}, l_{Req})$ be a labelled Petri net. CN is a *Component-net* (C-net) if and only if:

1. The labelling of transitions consists of two labelling functions l_{Prov} and l_{Req}, such that: $l_{Prov} : T \longrightarrow Prov \cup \{\lambda\}$, where $Prov \subseteq A$ is the set of provided services, and $l_{Req} : T \longrightarrow Req \cup \{\lambda\}$, where $Req \subseteq A$ is the set of required services.
2. The set of places contains a specific *Input* place I, such that $^\bullet I = \varnothing$ (*Instance creation*),
3. The set of places contains a specific *Output* place O, such that $O^\bullet = \varnothing$ (*Instance completion).*

Notation. We denote by [I] and [O], which are considered as bags, the markings of the Input and the Output place of CN, and by Reach (CN, [I]), the set of reachable markings of the component-net CN obtained from its initial marking M_N within one token in its Input place I. Besides, when we deal with the graphical representation of the C-nets, we use ! and ? keywords for the usual sending (required) and receiving (provided) services together with the labeling function l instead of the two labeling functions l_{Prov} and l_{Req}.

Definition 2.2 (soundness)
Let CN = $(P \cup \{I, O\}, T, W, M_N, l)$ be a *Component-net* (C-net). CN is said to be *sound* iff the following conditions are satisfied:

1. Completion option: $\forall M \in$ Reach(CN, [I]), [O] $\in$ Reach(CN, M).
2. Reliability option: $\forall M \in$ Reach(CN, [I]), $M \geq [O]$ implies M = [O].

Completion option states that, if starting from the initial state, i.e. activation of the C-net, it is always possible to reach the marking with one token in the output place O. *Reliability* option states that the moment a token is put in the output place O corresponds to the *termination* of a C-net without leaving dangling references.

We use alternating simulation as commonly used in software specification, namely for interface automata [8], which is slightly less general than the original [2]. The alternating simulation represents the behaviour of two C-nets where the C-nets simulate each other regarding a partitioning of actions into input ("controlled") and output ("observed") ones. The second C-net simulates all controllable actions of the first one, whereas the first C-net simulates all observable actions of the second one.

Definition 2.3 (alternating simulation between C-nets)
Let CN = $(P \cup \{I, O\}, T, W, M_N, l)$ and CN' = $(P' \cup \{I', O'\}, T', W', M_{N'}, l')$ be two *Component-nets*. A binary relation $R \subseteq Reach(CN', [I']) \times Reach(CN, [I])$ is an alternating simulation if whenever M' R M and $a \in Prov \cup Req' \cup \{\lambda\}$, it holds that:

1. If $M[a? > M_1$ and $a \in Prov$, then $\exists M'_1, M'[a? > M'_1$ and $(M'_1, M_1) \in R$,
2. If $M'[a! > M'_1$ and $a \in Req'$, then $\exists M_1, M[a! > M_1$ and $(M'_1, M_1) \in R$,
3. If $M'[a > M'_1$ and $a = \lambda$, then $\exists M_1, M[a > M_1$ and $(M'_1, M_1) \in R$,

Alternating simulation $\leq_{AS}$ is defined as the largest such relation. We say that CN' A-simulates CN, written CN' $\leq_{AS}$ CN, if there exists an alternating simulation containing $(M_{N'}, M_N)$. Alternating simulation $\leq_{AS}$ is a preorder relation (reflexive and transitive).

Asynchronous (parallel) composition of C-nets. The *parallel composition* of C-nets, noted $\oplus$: C-net $\times$ C-net $\longrightarrow$ C-net, is made by communication places allowing interaction through observable services in an *asynchronous* way. Given a client C-net and a server C-net, it consists in connecting, through the communication places, the request and the accept transitions having the same service names: for each service name, we add one *communication-place* for receiving the requests/replies of this service. Then, all the accept-transitions labeled with the same service name are provided with the same *communication-place*, and the client C-net is connected with the server C-net through these communication places by an arc from each request-transition towards the suitable communication-place and an arc from the suitable communication-place towards each accept-transition. In order to achieve a syntactically correct compound C-net C = A $\oplus$ B, it is necessary to add new components for initialization and termination: two new places (an Input and Output place), noted {Ic, Oc}, and two new not observable transitions, noted {ti, to}, for interconnecting the input place {Ic} to the original two input places via the first new transition {ti}, and the two original output places to the output place {Oc} via the second new transition {to}. Thus, the composition of two C-nets is also a C-net, and this composition is commutative and associative.

2.2 Specification of Roles Components and Their Composition

In our RICO model [9], a role component is considered as a component providing a set of interface elements (either attributes or operations, which are provided or required features necessary to accomplish the role's tasks), a behavior (interface elements semantics), and properties (proved to be satisfied by the behavior). In this paper, since we are interested in behavioural compatibility and implementation of roles, we will consider only theirs behavioural interfaces, that is the set of (provided and required) services together with the behaviours.

Definition 2.4 (Role Component)
A Role Component for a role $\Re$, noted RC, is a 2-tuple RC = (Behav, Serv), where,

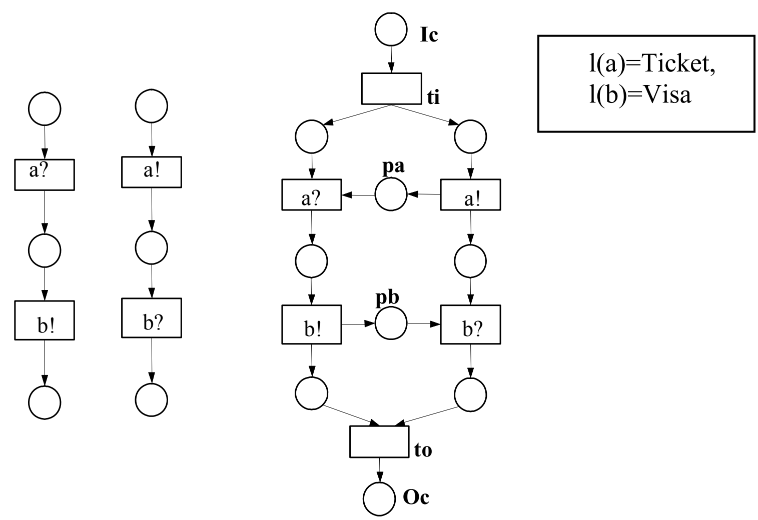
- Behav is a C-net describing the life-cycle of the role $\Re$.
- Serv is an interface, a set of public elements, through which RC interacts with other role components, for instance messaging interface. It is a pair (Req, Prov), where Req is a set of required services, and Prov is the set of provided services by RC, and more precisely by Behav.

Since the life-cycle of roles is specified by C-nets, we say that a role component satisfies the completion (resp. terminates successfully or properly) if and only if its behaviour that is its underlying C-net satisfies the completion (resp. is sound).

Definition 2.5 (Role Components composition)
A Role (Component), RC = (Behav, Serv), can be composed from a set of (primitive) Role-Components, $RC_i = (Behav_i, Serv_i)$, i = 1, …, n, noted $RC = RC_1 \otimes \ldots \otimes RC_n$, as follows:

- $Behav = Behav_1 \oplus \ldots \oplus Behav_n$.
- Serv = (Req, Prov) such that $Req = \cup Req_i$, and $Prov = \cup Prov_i$, i=1, …, n.

The composition of two role-components is also a role-component, and this composition is commutative and associative. Besides, we note that the composition operator $\otimes$ allows composition of many instances of the same role as in interaction protocols [9].



RC_1 = Customer **RC_2 = Ticket-service** **$RC = RC_1 \otimes RC_2$**

Fig. 1. RC1 = (Behav1, Serv1), RC2 = (Behav2, Serv2), RC = RC1 ⊗ RC2 = (Behav, Serv), where l(a)= Ticket, l(b) = Visa[1], Serv1 = ({Visa}, {Ticket}), Serv2 = ({Ticket}, {Visa}), and Serv = ({Visa, Ticket}, {Ticket, Visa})

[1] The names of transitions are drawn into the box.

Example 1. Let's take the example of the ticket service and the customer. Figure 1 shows RC_1 representing the behaviour of the customer, and RC_2 the behaviour of the Ticket-service. The Ticket service initiates the communication by sending one Ticket and waits for the payment (Visa). By receiving the Ticket, the customer makes the payment of the ticket by Visa. The result of the composition of the behaviours of RC_1 and RC_2, $RC = RC_1 \otimes RC_2$, is shown in figure 1. These two roles as well as their composed role terminate successfully.

3 Role-Based Interaction Components Behavioural Compatibility

In component-based software engineering, classical approaches for components compatibility deal with components composition together with their property preservation [11]. In this paper, we consider this approach for role components together with their context of use that is the environment in which they interact and coordinate their behaviour. Namely, we deal with the correctness of the composition of roles with their environment when reasoning about the proper (or successful) termination property. First, let define the notion of the environment of a role.

Definition 3.1 (Environment)
Let RC_1 = ($Behav_1$, $Serv_1$) and RC_2 = ($Behav_2$, $Serv_2$), be two roles such that $Serv_i$ = (Req_i, $Prov_i$), i=1, 2.
RC_2 is called an environment-role (or environment for simplicity) of RC_1, and vice versa, iff $Req_1 = Prov_2$, $Req_2 = Prov_1$.
We let ENV(RC), the set of environments of the role component RC.

Definition 3.1 expresses that for two role components RC_1 = ($Behav_1$, $Serv_1$) and RC_2 = ($Behav_2$, $Serv_2$), such that both sets of interfaces $Serv_1$ and $Serv_2$ completely match, i.e. $Req_1 = Prov_2$ and $Req_2 = Prov_1$, the role component RC_1 is considered *a role-environment* (or *environment* for simplicity) of RC_2 - and vice versa.

Given a role-component and its environment, it is possible to reason about the proper termination of their composition. Based on that, we define the usability relation:

Definition 3.2 (usability)
RC is usable iff $\exists$ Env $\in$ ENV(CR), Env $\otimes$ RC terminates successfully. We say that Env *utilizes* RC.

We are finally ready to give adequate definition for roles behavioral compatibility, which is based on the usability.

Definition 3.3 (compatibility)
Let RC_1 and RC_2 be two usable roles.
RC_1 and RC_2 are Compatible, noted $RC_1 \approx_C RC_2$ iff $RC_1 \otimes RC_2$ is usable.

Example 2. Let take again the example of the ticket service and the customer. Now, in figure 2, the ticket service RC_5 initiates the communication by sending one `Ticket` and wait of the payment (either `Visa` or `eCash`). The role components RC_3 and RC_4 are then two examples of the customer's behaviour. By receiving the `Ticket`, they solve an internal conflict and determine the kind of payment. The roles RC_3 and RC_5 (resp. RC_4 and RC_5) are usable, since for instance RC_3 utilizes RC_5

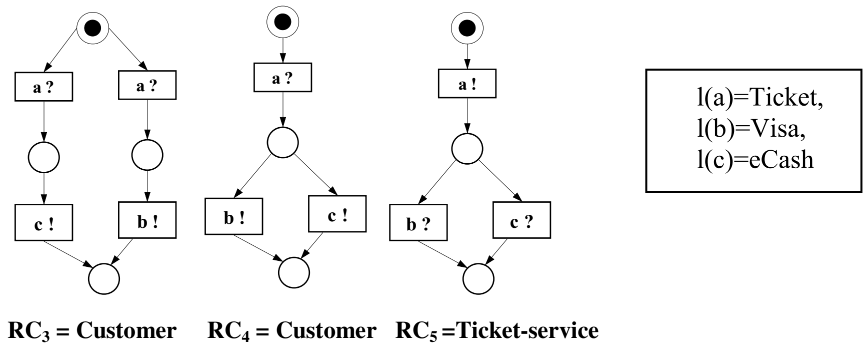


Fig. 2. RC_3 utilizes RC_5, RC_4 utilizes RC_5, where $Serv_3 = Serv_4 = (\{Visa, eCash\}, \{Ticket\})$ and $Serv_5 = (\{Ticket\}, \{Visa, eCash\})$

(resp. RC_4 utilizes RC_5) and vice versa. For instance, the two roles RC_3 and RC_5 shown in figure 2 are compatible that is $RC_3 \approx_C RC_5$ holds since $RC_3 \otimes RC_5$ is usable. Indeed, $RC_3 \otimes RC_5$ terminates successfully.

4 Role-Based Interaction Components Conformance Semantics

Our main interest is to provide a formal semantics for conformance between the specification and implementation of roles that is, in which case the first role component is a model (specification) and the second an implementation (or a refined model). In this setting, we use alternating simulation semantics.

Definition 4.1 (Implementation of roles)
Let $\Re$ be a role and RC = (Behav, Serv) be its role component specification. Let RC' = (Behav', Serv') be a role component.
RC' implements RC, noted RC' $<<_I$ RC, iff Serv' = Serv and Behav' A-simulates Behav that is Behav' $<<_{AS}$ Behav.

The conformance between the specification and implementation of roles is formalised using alternating simulation. The environment drives the specification RC of the role $\Re$. Any implementation RC' of RC must conform to RC as long as it is receiving input from the environment. Besides, the specification RC must conform to the implementation RC' as long it is sending output to the environment. The behaviour of RC' (resp. of RC) on sequences of inputs (resp. output) that cannot be provided (resp. required) by the environment is not considered.

Example 3. As an example, consider the roles RC_3 and RC_4 given in the figure 2; it is easy to check that $RC_3 <<_I RC_4$, since RC_3 simulates all the provided (input) services of RC_4 and RC_4 simulates all the required (output) services of RC_3. Nevertheless RC_4 does not implement RC_3, since once RC_4 simulates the provided (input) service {a?}, RC_3 cannot simulate all the required (output) services {b!, c!} of RC_4.

The following lemma expresses that the relation $<<_I$ is compositional. In order to check if Env ⊗ R $RC'_1 <<_I$ Env ⊗ RC_1, it suffices to check $RC'_1 <<_I RC_1$, since the latter check involves smaller roles and it is more efficient.

Lemma 4.1 (Composability of role's implementation)
Let RC_1 and RC'_1 be two role components such that $RC'_1 <<_I RC_1$.
For any role component Env, $(Env \otimes RC'_1) <<_I (Env \otimes RC_1)$.

A desirable property of role components is that roles can be implemented independently of each other once the specifications are known. The following theorem states that the semantics given for implementation of roles satisfies this property:

Theorem 4.1 (Independent role's implementability)
Let RC_1 and RC_2 be two role components. Let RC'_1 and RC'_2 be their respective implementation, that is $RC'_1 <<_I RC_1$ and $RC'_2 <<_I RC_2$.
We have $(RC'_1 \otimes RC'_2) <<_I (RC_1 \otimes RC_2)$.

As expected, the implementation relation $<<_I$ preserves the usability of roles by the environment. Namely, the compatibility between role components should not be affected by replacing a specification of a role by its implementation. The following core theorem studies the preservation of compatibility by implementation, dealing with the compatibility relation together with the implementation relation given in this paper.

Theorem 4.2 (Consistency: Compatibility preservation by implementation)
Let RC and RC' be two usable roles such that $RC' <<_I RC$.
$\forall\, Env \in ENV(RC), Env \approx_C RC \Rightarrow Env \approx_C RC'$.

5 Conclusion and Related Work

The aim of this paper is to propose a definition of conformance of role's behaviour in MAS, that is in which case the first role component is a model (specification) and the second an implementation (or a refined model). First, we provide a framework for modelling usable role-components together with their composition. Then, we give a formal semantics for the implementation of roles, which is defined upon alternating simulation relation. We show that the proposed semantics is compositional for role's composition. Then, we prove that roles can be implemented independently of each other once their specifications are known. Finally, we study the existing link between compatibility and implementation of roles, and namely the preservation of the compatibility relation proposed in this paper by role's implementation semantics.

The issues of compatibility/interoperability and conformance are widely studied in software engineering by many researchers in different area like web services and multi-agent systems. [4, 5, 8] use automata models to decide compatibility and substitutability which is a refinement relation between services. All these approaches use only synchronous communication whereas we consider asynchronous message passing in our framework. In [4], the compatibility relations are too restrictive since the framework does not support nondeterministic and either these relations demand that the interfaces of both services match or they require the deadlock-freeness of their composition. In our framework, we focus on behavioral compatibility instead of structural compatibility of (non-deterministic behaviour of) roles. In addition, the proposed behavioural compatibility relation between two roles in this paper demands the usability of their composition which is related to the proper termination property. In [8], the compatibility relation between two interface automata is defined as the existence of a legal environment that is an environment that can make them work together without deadlock. According to the alternating approach, the refinement relation between interface automata is defined upon the alternating simulation. Recently, [5] apply the theory of interface automata to web services interfaces and propose algorithms for compatibility and substitutability checking. This work is close to ours, since it uses alternating approach, whereas the most common use of alternating simulation in our framework is for conformance checking of role's behaviour. In [6], they adopt an approach based on process algebra, and conformance is defined as the contract refinement, which can be applied on each of the service inside a composition independently. This work is close to our notion of conformance between the specification and implementation of roles. The difference is that our work is based on alternating approach (simulation) while they use a (theory of should) testing together with a synchronous communication model. The unique work on compatibility and conformance for asynchronous communication that we are aware of has been developed in [1] using Petri nets. In that paper, contracts are specified by open workflow net, and the approach for formalizing conformance, called accordance, between two contracts uses the proper termination property to define the notion of workflow nets strategy for a workflow net. This notion of strategy is the same as our notion of usable environment for a component role. Then, accordance is defined as refinement of strategies, and as our work, they prove that contract refinement can be done independently. This approach is close to ours previous works on substitutability and refinement of roles [10], and the difference is that our work is based on alternating approach (simulation) to give a semantics for conformance checking of role's behaviour. Finally, in the area of MAS, interoperability and conformance are studied in the context of interaction protocols [3]. In that paper, interoperability is the capability of a set of entities of producing a conversation that is legal w.r.t the protocol, and as in our works, the conformance between entities uses alternating simulation. Nevertheless, this approach to the definition of interoperability is very particular since it is related to linear semantics (traces) and does not deal with property preservation such as deadlock freeness or proper termination. In addition, the preservation of interoperability w.r.t. the substitutability of conformant entities is not studied.

References

1. van der Aalst, W.M.P., Lohmann, N., Massuthe, P., Stahl, C., Wolf, K.: From public views to private views – correctness-by-design for services. In: Dumas, M., Heckel, R. (eds.) WS-FM 2007. LNCS, vol. 4937, pp. 139–153. Springer, Heidelberg (2008)
2. Alur, R., Henzinger, T.A., Kupferman, O., Vardi, M.: Alternating refinement relations. In: Sangiorgi, D., de Simone, R. (eds.) CONCUR 1998. LNCS, vol. 1466, pp. 163–178. Springer, Heidelberg (1998)
3. Baldoni, M., Baroglio, C., Martelli, A., Patti, V.: A priori conformance verification for guaranteeing interoperability in open environments. In: Dan, A., Lamersdorf, W. (eds.) ICSOC 2006. LNCS, vol. 4294, pp. 339–351. Springer, Heidelberg (2006)

4. Bordeaux, L., Salaun, G., Berardi, D., Mecella, M.: When are Two Web Services Compatible? In: Shan, M.-C., Dayal, U., Hsu, M. (eds.) TES 2004. LNCS, vol. 3324, pp. 15–28. Springer, Heidelberg (2005)
5. Beyer, D., Chakrabarti, A., Henzinger, T.: Web service interfaces. In: WWW 2005, pp. 148–159. ACM, New York (2005)
6. Bravetti, M., Zavattaro, G.: Contract Based Multi-party Service Composition. In: Arbab, F., Sirjani, M. (eds.) FSEN 2007. LNCS, vol. 4767, pp. 207–222. Springer, Heidelberg (2007)
7. Cabri, G., Leonardi, L., Zambonelli, F.: BRAIN: a Framework for Flexible Role-based Interations in Multi-agent Systems. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) CoopIS 2003, DOA 2003, and ODBASE 2003. LNCS, vol. 2888, pp. 145–161. Springer, Heidelberg (2003)
8. De Alfaro, L., Henzinger, T.A.: Interface automata. In: Proc. of ESEC/FSE. Software Engineering Notes, vol. 26(5), pp. 109–120. ACM, New York (2001)
9. Hameurlain, N., Sibertin-Blanc, C.: Specification of Role-based Interactions Components in MAS. In: Choren, R., Garcia, A., Lucena, C., Romanovsky, A. (eds.) SELMAS 2004. LNCS(LNAI), vol. 3390, pp. 180–197. Springer, Heidelberg (2005)
10. Hameurlain, N.: Contextualizing Behavioural Substitutability and Refinement of Roles in MAS. In: Artikis, A., O'Hare, G.M.P., Stathis, K., Vouros, G. (eds.) ESAW 2007. LNCS, vol. 4995, pp. 193–207. Springer, Heidelberg (2008)
11. Hameurlain, N.: On Compatibility and Behavioural Substitutability of Component Protocols. In: SEFM 2005, pp. 394–403. IEEE Computer Society, Los Alamitos (2005)

Towards a Methodology for Modeling Deontic Protocols Using the Organizational Petri Nets Formalism

Stéphanie Combettes

University of Toulouse – IRIT Lab
118 Route de Narbonne 31062 Toulouse, France
stephanie.combettes@iut-tlse3.fr

Abstract. Interaction protocols are a key mechanism for modeling the coordination of real world organizations and deontic rules. Modeling deontic protocols require resources, roles and their activities, as well as the protocol actions to be described. The Organizational Petri Nets formalism enables both these components and their relationships to be taken into account. After having introduced an intuitive definition of this formalism, we propose a methodology to model deontic protocols using this formalism. This methodology is also applied to a library borrowing protocol.

Keywords: Deontic protocol, Organizational Petri Net, Methodology.

1 Introduction

Multi-Agent Systems are widely adopted to model real world organizations since they provide social abstractions and sophisticated interaction protocols. Interaction protocols are a key mechanism that helps to coordinate distributed agents through explicit rules. Interaction protocols are often governed by deontic rules that determine what is authorized, what is forbidden and what could be repaired. Although dealing with these deontic rules adds complexity, taking them into account helps to represent all the possible behaviors, even the undesired ones. Thus, the protocol description is more realistic and the representation of the actor behaviors is more comprehensive. Moreover, the possibility of penalizing an actor having a non-authorized behavior can lead to this actor consequently behaving in an optimal way. The stages for modeling deontic protocols are first to *identify and underline the ideal and non-ideal (violation and repair) actions*, second to *distinguish the actors from the resources* they handle, third to *penalize the actors* performing non-ideal actions, and fourth to *coordinate the occurrences of the actions* constituting the steps of the protocol. Moreover, protocols require analysis, validation, simulation, or implementation.

The suitability of High Level Petri Nets for specifying, validating, simulating, and even implementing protocols has already been shown [1]. However, no formalism can deal with all the four previous stages. We have developed a new formalism called Organizational Petri Nets (OgPN) [2], which enables deontic protocols to be modeled while encompassing the previous stages. In this paper, we present a methodology to model deontic protocols with the use of Organizational Petri Nets. Following the framework for evaluating agent-oriented methodologies proposed by [3], our

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 589–598, 2009.

methodology provides a life cycle process, a set of concepts and models, a set of techniques, and a modeling language. Among the different components of this methodology, we focus on the process that helps designers to model a deontic protocol. We illustrate this process through an example.

This paper is organized as follows. Section 2 gives a global view of the OgPN formalism, describes the example of the library item-borrowing protocol, and presents the life cycle of a protocol. Section 3 describes the first step of the process to model a protocol and its application to the working example. Section 4 describes the second step of the process and its application to the example. Section 5 describes the process to analyze the modeled protocol and applies it to the library protocol. Finally section 6 concludes and discusses.

2 The Methodology

2.1 Organizational Petri Nets: An Intuitive Account

The Petri Net (PN) formalisms are well-adapted to specify, validate, simulate and implement reliable protocols [4]. Besides, some extensions of PN enable to model and structure the active (actors) and passive (information resources) entities such as the Petri Nets with Object formalism [5] which combines PN technology with the Object-Oriented Approach. Others enable the deontic aspects such as Deontic Petri Nets to be captured [6]. However, none of the Petri Nets extensions integrates the stages addressed in the introduction. We have developed a new formalism called the Organizational Petri Nets (OgPN) [2] which coherently combines and integrates the deontic aspects and the Object-Oriented Approach. As a consequence, OgPN enables, among other things, actors to be distinguished from resources, the ideal actions to be distinguished from the non-ideal (violation and repair) ones, and actors to be penalized. As the OgPN formalism is an extension of Petri Nets formalism, it offers the capability of analyzing and validating the model. In the following paragraph, we give an intuitive account of the OgPN.

The OgPN is composed of places representing states, transitions representing actions and tokens that are objects (complex data structures). Tokens represent two kinds of objects: the active ones (the *actors*) and the passive ones representing the forms, documents etc. These passive objects, called the *resources*, are handled by the actors that can update, modify, read them etc. Each actor has a pre-defined attribute *penalty* that records the amount of the actor penalties. Each transition enables to perform several actions. When a transition occurs, the actors present in the input places handle the resources (also present in the input places) through the execution of actions associated with this transition. Actions are broken down into three categories: ideal, violation and repair. To each action (and so to each transition executing this action) is associated a *cost* depending on the action type: the cost is null for an ideal action, positive for a violation one and negative for a repair one. The value of the cost also depends on the seriousness of the action. It is important to note that each repair action must follow a violation action and repairs only one violation action. Each non-ideal action has an actor responsible for it, and when an action is performed its cost is added to the penalty of the actor responsible. A transition sequence performed by an actor responsible corresponds to one behavior of this actor. Thus, the behavior of each actor can be evaluated and analyzed: the more important the actor penalty is, the more reprehensible its behavior is.

2.2 Illustration: The Lending Policy of a Library

To illustrate our methodology, we describe the *library item-borrowing protocol* by describing the roles involved, the borrowed items, the lending policy, and library sanctions.

First, two *roles* are involved: borrower and librarian. Different categories of borrowers are defined according to their age, and a specific lending policy is associated with each category. Second, three types of *items* can be borrowed: books, CDs and DVDs. There are different categories of books and DVDs according to the age of the public for whom the items are intended. Third, the *lending policy* is described. An adult may borrow a maximum of ten items with a limit of eight books, three CDs and two DVDs. A teenager may borrow a maximum of six items with a limit of five books, two CDs and one DVD. A child under twelve may borrow only books or DVDs intended for the young. The loan duration for books and CDs is three weeks. For DVDs, it is one week. An item may be reserved provided it has not already been reserved. An item loan may be renewed once. Loans of items that were reserved by another borrower cannot be renewed. Finally, the *lending sanctions* are the following: any violation of the lending policy implies the recording of penalty in the borrower file. The fine imposed on borrowers for late return is 0.15 Euros per day per item. In case of damaged materials, the borrower may choose to repair them or to pay the library to replace them. In these two cases, his fine will be reduced.

2.3 Description of the Life Cycle of a Deontic Protocol

The protocol integration requires three dimensions to be taken into account: the informational, organizational and behavioral. The informational model represents the resources. The organizational model refers to the logical actors and the activities they are able to perform, and determines the ideal, violation and repair actions. The behavioral model determines the actions of the protocol and their coordinated sequences. With this in mind, the life cycle of a deontic protocol consists of the six following steps:

1. Definition of the *Informational* and *Organizational Models*
2. Definition of the *Behavioral Model*
3. Analysis and properties of the modeled protocol
4. Simulation
5. Implementation
6. Adaptation.

In this methodology, s*teps 1* and *2* correspond to the production of an OgPN. They are interdependent as they provide information to one another and correspond to the three dimensions of a protocol, each of them giving rise to a model. As this is not a linear process, it requires iterations to refine and make these three models coherent. *Step 3* enables the models obtained in *steps 1* and *2* to be validated. *Steps 4* and *5* will not be developed here as numerous studies have shown the suitability of High Level Petri Nets for simulating and implementing protocols in addition to their specification and validation [1]. *Step 6* consists in modifying the protocol according to the analysis of its previous similar executions. Some studies have shown that adaptation can be performed on some Petri Nets extensions [7],[8]. However, due to lack of space, adaptation will not be studied here.

In the remainder of the paper, the steps 1, 2 and 3 of this methodology are developed.

3 Step 1: The Informational and the Organizational Models

3.1 Description of These Two Models

The **Informational Model** refers to any kind of data, called resources, which can be handled or exchanged by the actors (or roles) of the organization. It structures the resources in object classes (sharing the same features) that are handled, and possibly describes some integrity constraints (e.g. a book cannot be borrowed simultaneously by two borrowers). One also wants to express more flexible, institutional constraints [9] that may be violated because they are stated by the organization and not physical constraints. An example of an institutional constraint is a book that should be returned by the borrower within three weeks. In reality, this constraint may be violated.

The **Organizational Model** defines the contribution of each role with regard to the others and the rules applied to grant these roles. This determines what agents have the permission to do within a protocol. It first structures roles in object classes sharing the same features and defines the capacities they must have. It means that a role has some characteristics (attributes) and may perform a list of actions through methods. Second, the organizational model attributes to each role, permission to perform actions on resources defined in the informational model, and possibly the responsibility for actions that violate a constraint. The permissions are implicitly given by the types (in the role) of method parameters, which reference the object classes of the informational model. An actor is responsible for all the actions implemented by methods it can perform.

OgPN formalism enables complex data structure usually used in normative systems (norms, documentation, etc.) to be described, notably thanks to the object classes OgPN integrates. Moreover, powerful mechanisms (such as classification, polymorphism, etc.) are used, and particularly inheritance that enables permissions between roles to be propagated, as its members share attributes, methods and constraints. So, the deontic constraints are propagated through the hierarchy [10]. It should be noticed that the lower the object classes in the hierarchy are, the more constraints (obligations, permissions or interdictions) they have. These two models give rise to the definition of object classes processed by the OgPN.

3.2 Application to the Item-Borrowing Example

Considering the informational model the Item class is first defined. This class has three subclasses: Book, CD and DVD. They inherit attributes from the Item class, but also have

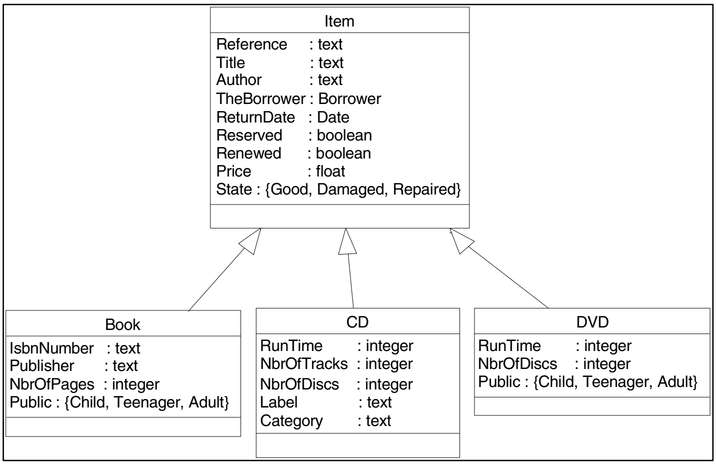


Fig. 1. The Informational Model

additional attributes related to their specificity: the Book class has two specific attributes: IsbnNumber and Publisher. It should also be observed that some attributes (e.g. ReturnDate, TheBorrower) are modified during executions of the protocol (fig.1).

The organizational model (fig.2) consists of two classes (corresponding to the roles), Borrower and Librarian, since there are two roles involved. The Borrower has the permission to handle some items as this type appears in the role methods.

Each role has a Penalty attribute that records the amount of penalty incurred by the actor. In this lending protocol the Librarian does not support deontic rules but this attribute appears as the Librarian is an actor. The librarian may be supposed to belong to another protocol that manages the librarians working in a library and this protocol could be composed with the lending one (of course in a coherent way!) and this will provide a complete view of the item management.

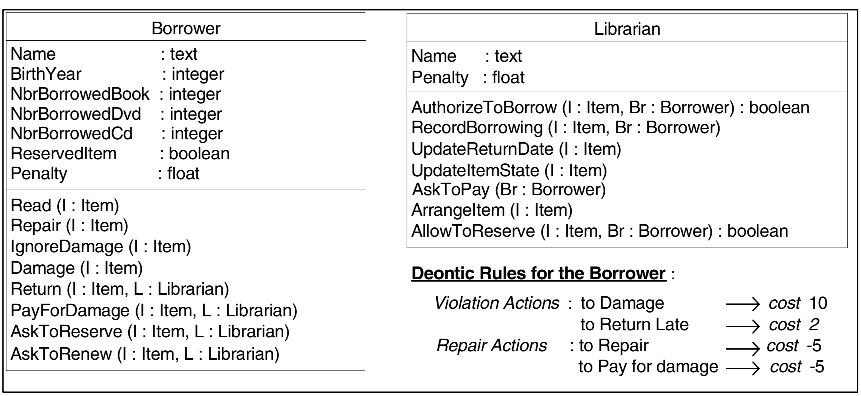


Fig. 2. The Organizational Model

In fig. 2 the deontic rules are also described, particularly the violation and repair actions and their associated cost.

4 Step 2: The Behavioral Model

4.1 Description of the Behavioral Model

The **Behavioral Model** defines the protocol's sequences of actions associated with the transitions, by identifying the places of these transitions and their coordination with control patterns [11]. Each actor and each resource are represented by a token in an input place.

Thanks to the expressive power of Petri Nets, a lot of control structures are represented such as alternative, sequence, loop and concurrency. Thus, they enable deontic constraints between two actions (such as x followed by y) to be represented. The permissions granted to a role are expressed by the nature of the arc between an information place and a transition. Moreover, an additional constraint must be respected: if we consider a repair transition and a violation one, the former must always occur after the latter. This model gives rise to the elaboration of the net.

4.2 Application to the Item-Borrowing Example

The behavioral model is described in the fig. 3. This Petri net includes twelve transitions and twelve places, typed with the previously defined classes. The initial state is defined by the three initial marked places (ReadyToWorkLibrarians, WaitingBorrowers and AvailableItems) and corresponds to a situation where a librarian is free at his desk waiting for a borrower intending to borrow an item. The final state is defined by the three sink places (AvailableLibrarians, ServedBorrowers and ReturnedItems) and represents a situation where the borrower has returned a borrowed item and the librarian becomes available.

Transitions are greyed to distinguish between ideal transitions (white: T1, T2, T4, T6, T7, T9, T10 and T12), violation ones (dark grey: T3 and T8) and repair ones (light grey: T5 and T11). For each non-ideal transition (violation and repair actions), the actor responsible for it and its cost are specified by the notation *<name>:cost* next to that transition. For example, the violation transition T3 (Damaging) is associated with the following notation
:10. It expresses the fact that the Borrower (Br) is responsible for T3 and gets a penalty of 10 if T3 is performed.

Let us explain two typical scenarios. An ideal behavior "to borrow and read an item, and to return it in due time and without damage" is represented by the transition sequence: s1 = T1.T2.T6.T7.T9.T12. The cost of this sequence is 0 and consequently, the borrower incurs no penalty. Let us consider a non-ideal behavior: a borrower chooses an item, reads it, damages it, and finally returns it in time and pays for the damage. This scenario is represented by the sequence:

s2 =T1.T2.T3.T4.T6.T7.T10.T11.T12. The penalty incurred by the borrower agent linked to the Br variable during an execution of this sequence is the sum of the costs of the transitions, that is (10) + (-5) = 5.

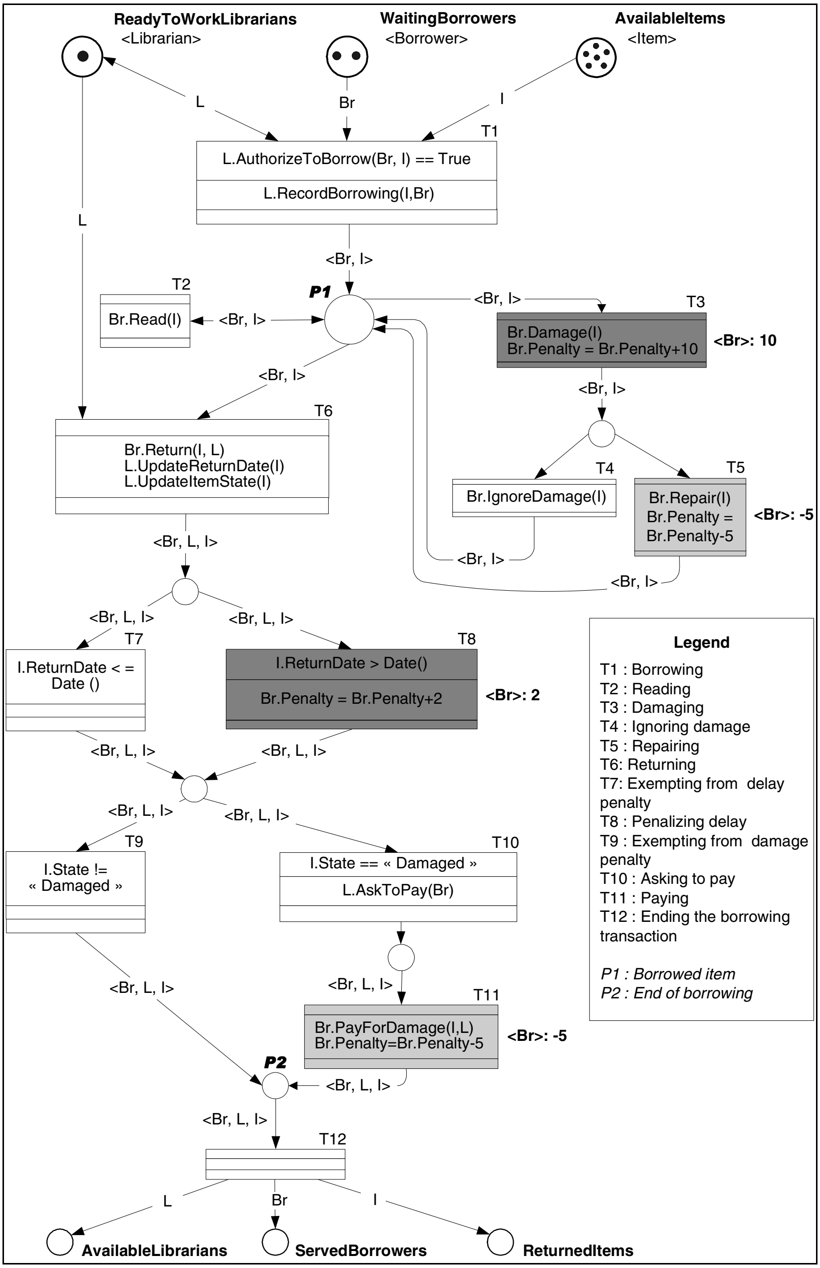


Fig. 3. The Behavioral Model

5 Step 3: Analysis and Properties

As OgPN formalism is a well-founded extension of classical Petri Nets, the properties of a system can be analyzed. We give here the main properties of Organizational Petri Nets, which consist of *standard* and *deontic* properties.

5.1 The Standard Properties

The **standard properties** consist in proving the presence of loops or cycles, deadlocks, the (un-)accessibility of a goal, final or home state, the boundedness (no infinite growing of the number of tokens) or the lost of tokens in a hole place. However, we do not elaborate here how to prove them [4]. Some of the properties are common to any model of any system, such as the absence of deadlocks. Other properties are specific to the considered system and are just the formal statement of its requirements and needs. One also distinguishes properties relative to the behavior of the whole system from properties about a specific object.

Applied to the working example, the net of fig. 3 contains no deadlock, and no cycle from which it is impossible to get out. The requirement saying that "no item is lost" corresponds to a place invariant that guarantees that a token in the place AvailableItems can reach, in any case, the place ReturnedItems.

Besides, merging coherently initial places with final ones enables the protocols to be studied from different points of view. For example if WaitingBorrowers and ServedBorrowers places are merged, it shows the possibilities for a borrower to interleave or sequence the lending of several items.

5.2 The Deontic Properties

The **deontic properties** enable to analyze first the coherence of an OgPN, and second the deontic quality of the behavior of the actors involved in the protocol.

Considering the *coherence* of an OgPN, some rules concerning the non-ideal actions and their occurrences must be respected. Thus,

- each violation can and must be repaired by one repair action
- each repair action must repair exactly one violation.

As a consequence, no violation requires several actions to become repaired and no repair action can repair several violations at once [2].

In the fig. 3, the repair transition T5 (respectively T11) can only be performed after the occurrence of the violation transition T4 (respectively T10).

Concerning the *deontic quality*, the behavior of an actor needs to be characterized by evaluating penalties. With this in mind, we define the set, called Behav(actor), that includes all the sequences where actor is the actor responsible for at least one transition. It is a rational language as soon as the underlying Petri Net is bounded. Comparing the cost of each possible sequence enables the sequences to be classified according to a preferred-sequence order. The lower the cost is, the more preferred the sequence is. The minimum cost is zero and corresponds to an ideal sequence.

Considering the working example, the set of sequences where the borrower is responsible for is:

Behav(Borrower)=T1.(((T2|(T3.T4)|(T3.T5))^n.T6)|T6).(T7|T8).(T9|(T10.T11)).T12. The librarian is not responsible for actions in this protocol, so Behav(Librarian)= ∅.

6 Discussion and Conclusion

In this paper, we have presented a methodology based on the Organizational Petri Nets (OgPN) for modeling deontic protocols. This methodology is illustrated through a library lending protocol. The OgPN formalism provides a formal method to model deontic protocols thanks to the rigorous mathematical definition, which makes analysis, simulation and validation possible with the Petri Nets. Moreover, it integrates all together the three dimensions of these protocols, which are the informational, organizational and behavioral dimensions. So, OgPN helps to take into account the resources, the roles, the ideal and non-ideal actions and their coordination, and offers the possibility to penalize the actors having an undesired behavior. The proposed methodology focuses first on the deontic protocol modeling process, and second on the analysis of the obtained model. The modeling process gives rise to three models: the informational, organizational and behavioral models, corresponding to the three dimensions of a deontic protocol.

This methodology could enrich agent-oriented methodologies like GAIA [12] or MOISE+ [13]. Indeed, in these methodologies, protocols are considered as a key component but they lack formalization. Thanks to the OgPN formalism, our methodology can be used to enrich or extend these methodologies as it offers means to penalize, analyze, or simulate role behavior.

Evaluating our methodology with the framework proposed by [3] brings to the fore its strengths and its weaknesses. These strengths are: a life cycle, a modeling language (OgPN formalism), a set of concepts of multi-agent systems (roles, protocols, tasks…), and a set of techniques (notably the analyzability). However, its main weakness is the absence of a tool to implement real world applications. Providing a tool would improve our methodology and thus, it is one of our main concerns for future research.

A second issue is to study the adaptation of protocols modeled with our methodology. The adaptation of a protocol requires the analysis of executions of protocols, in a similar way to process mining, notably by analyzing log files.

References

1. Hanachi, C., Sibertin-Blanc, C.: Protocol Moderators as Active Middle-Agents in Multi-Agent Systems. Journal of Autonomous Agents and Multi-Agent Systems 8(3), 131–164 (2004)
2. Combettes, S., Hanachi, C., Sibertin-Blanc, C.: Organizational Petri Nets for Protocol Design and Enactment. In: International Joint Conference on Autonomous Agents and Multi agent Systems (AAMAS), Hakodate, Japan (2006)
3. Sturm, A., Shehory, O.: A Framework for Evaluating Agent-Oriented Methodologies. In: Agent Oriented Information System Workshop, pp. 94–109 (2003)
4. Murata, T.: Petri Nets: Properties, Analysis and Applications. IEEE 77(4), 541–580 (1989)

5. Sibertin-Blanc, C.: High Level Petri Nets with Data Structure. In: The 6th European Workshop on Application and Theory of Petri Nets, Espoo, Finland, pp. 141–170 (1985)
6. Raskin, J., Tan, Y., van der Torre, L.: Modeling Deontic States in Petri Nets. In: The 7th Dutch AI Conference (NAIC), Rotterdam, The Netherlands (1995)
7. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering Colored Petri Nets from Event Logs. International Journal on Software Tools for Technology Transfer (STTT) 10(1), 57–74 (2008)
8. Guan, S., Lim, S.: Modeling Adaptable Multimedia and Self-Modifying Protocol Execution. Future Generation Computer Systems 20(1), 123–143 (2004)
9. Meyer, J., Wieringa, R., Dignum, F.: The Role of Deontic Logic in the Specifications of Information Systems. In: Logics for Databases and Information Systems, pp. 71–115 (1998)
10. Cuppens, F., Cuppens-Boulahia, N., Miege, A.: Inheritance Hierarchies in the ORBAC Model and Application on a Network Environment. In: The 2nd Foundations Computer Security Workshop, Turku, Finland (2004)
11. Van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow Patterns. International Journal on Distributed and Parallel Database 34(1), 5–51 (2003)
12. Wooldridge, M., Jennings, N., Kinny, D.: The GAIA Methodology for Agent-Oriented Analysis and Design. Journal of Autonomous Agents and Multi-Agent Systems 3(3), 285–312 (2000)
13. Hübner, J., Sichman, J., Boissier, O.: MOISE+: Towards a Structural, Functional and Deontic Model for MAS Organization. In: Proc. of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Bologna, Italy (2002)

Towards an Adaptive Grid Scheduling: Architecture and Protocols Specification

Inès Thabet[1,2], Chihab Hanachi[2], and Khaled Ghédira[1]

[1] Laboratoire d'Ingénierie Informatique Intelligente,
Campus universitaire de la Manouba 2010, Tunisie
Khaled.ghedira@isg.rnu.tn
[2] Institut de Recherche en Informatique de Toulouse IRIT, UMR 5505,
Université Toulouse 1, 2 rue du Doyen-Gabriel-Marty, 31042 Toulouse Cedex 9
{Ines.Thabet,Chihab.Hanachi}@irit.fr

Abstract. Grid is known to be a heterogeneous, distributed and dynamic environment. In order to take fully advantages from grid power, Grid scheduling must take into consideration the environment's constraints and be adaptive. In this work, Grid architecture is fully rethought in terms of agents in order to implement a cooperative and adaptive scheduling. At a macro level, our architecture enables flexible cooperation among its components using high level interaction protocols. At the micro level, agents in charge of scheduling perform an adaptive behaviour since they are able to perceive their environment and its disturbances, to reason and to deliberate about the actions to undertake in order to adapt. This is made possible by the use of Belief-Desire-Intention mechanisms. For that purpose, we propose a conceptual model useful for the perception function. Also, a typology of adaptive rules useful for the deliberation step is given. Component's behaviour are specified and simulated with Petri-Nets.

Keywords: Grid scheduling, BDI agent, interaction protocols, Petri-Nets.

1 Introduction

This paper deals with Grid Scheduling system (GS) which is one of the main components of a Grid. For each application submitted by a user, the GS selects a set of available and adequate resources for performing the application's tasks. Resources selection and tasks allocation follow an objective function and it is based on information about the application requirements and the resources capabilities.

The design and implementation of a Scheduling System are more complex in a Grid than in traditional environments such as clusters or supercomputers. The complexity is due firstly to the heterogeneity of resources: the Grid nodes have different speeds and the network connections between them are variable. Secondly, the Grid resources are geographically distributed on multiple domains and are most of the time autonomous. Finally Performance and availability of resources vary over time: a compute node may become unavailable because of crashes, overloaded, etc. Grid complexity, dynamicity and instability yield perturbations that can disturb application

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 599–608, 2009.

execution on the Grid. Therefore, in order to take fully advantage of computational Grid, the GS must be able to handle such perturbations while taking into account the dynamicity, instability and heterogeneity of the environment.

Some adaptive scheduling systems have already been proposed in the literature [1, 2, 3, 4, 7], but these systems have some restrictions. Numerous systems support only one type of adaptation namely "rescheduling". When a perturbation occurs on a resource, it consists in migrating the impacted application (or portion of application) to another resource. As described in [5], obviously, other types of adaptation can be investigated such as application process adaptation, Grid resource reorganization, quality of service adaptation, etc. Other adaptive Grid scheduling systems, such as [6] apply an ad-hoc application level scheduling, consisting in generating and applying a new schedule plan which takes into consideration the actual resources states. In this case, the scheduling service is tightly coupled to the application and thus can not be generalized to other applications. Finally, some adaptive scheduling systems [7, 8] are based on contracts which govern the adaptation (migration, requirements modification) and its consequences (compensation of the delay, cost increase, etc). Unfortunately contract management is a very complex activity generating overheads that decrease Grid performance. Moreover it is an application-dependent technique.

The aim of this paper is to propose a Grid architecture enabling a cooperative and adaptive Scheduling. The architecture is designed with the idea to meet the three following requirements. First, its Global Scheduler should be easily applied to different types of application, thus the GS should be uncoupled to applications. The second requirement is to build a robust architecture with software components able to cooperate to support distribution while preserving their autonomy. The third requirement is flexibility, allowing the GS to express several adaptation modes. To meet these requirements, our contribution is threefold:

- *Firstly*, at a macro level, we give a specification of a multi-agent Grid architecture. More precisely, we define the architecture components and the interaction protocols supporting their cooperation. Cooperation is viewed as a mean to facilitate adaptation. We specify accurately the component's behaviours and simulate them with Petri-Nets [9]. The use of agent technology is justified since it takes into consideration the Grid complexity, particularly autonomy, distribution and the dynamic nature of its components.
- *Secondly*, at a micro level, we specify the internal architecture of the scheduling components. These components are modelled according to the BDI (Belief-Desire-Intention) [10] architecture which allows the scheduling components to perceive their environment, to reason, to collaborate and to make adaptation decisions directed by goals.
- *Thirdly*, we provide an original conceptual model of the Grid and a typology of adaptation rules useful for scheduling and adaptation.

This paper is organised as follows. In section 2, we introduce the multi-agents architecture for a cooperative and adaptive Grid scheduling: its components and their interaction protocols. In section 3, we describe the BDI Scheduler in charge of adaptation and its different components. In section 4, we conclude and give some perspectives.

2 Overview of the Multi-agent Grid Architecture

Agents are known to be an appropriate paradigm for modeling complex, open and distributed systems. Indeed, agents provide high level abstraction to build autonomous, reactive and proactive software components with social abilities enabling to communicate with sophisticated languages and protocols. The benefits obtained by applying multi-agents system to the Grid have been discussed in the literature [11]. As a consequence, a great number of systems applying agent paradigm to the Grid have been proposed ([12], [13]). Unfortunately, most of them implement a non adaptive scheduling. The originality of our work is to sweetly combine social and reasoning agent's abilities. More precisely, the first feature allows our grid components to cooperate to find an adequate partner to execute optimally application's tasks even if a perturbation occurs. The second feature allows our Grid Schedulers based on a Belief-Desire-Intention architecture to perceive the environment, to reason and to adapt in the case of disturbance.

In this section we propose a multi-agents architecture providing a cooperative and adaptive scheduling. The adaptiveness is facilitated by three features. At a macro level, the use of high-level protocols, like contract net protocol, eases tasks allocation even if the Grid structure evolves. Moreover, protocols use a performative based language (namely FIPA-ACL [14]) that facilitates the interaction between the heterogeneous Grid components. At a micro level, the Global Scheduler follows a BDI behaviour including explicit adaptive actions in the case of disturbance.

2.1 A Three Layers Architecture

All the Grid components are modelled as agents and the architecture is organized in three layers:

- The top level layer called the *user interface* (UI) is able to interact both with the user and with the middle layer. It submits the user query to the most adapted Global Scheduler. A user query corresponds to an application made of several coordinated tasks.
- The middle level layer called *Global Level* consists in several cooperative Global Schedulers (GS). Each one manages one or several domains, schedules applications, and allocates tasks execution to its Local Schedulers (LS) according to a tasks allocation protocol (section 2.2.1), and/or sub-contracts them to its acquaintances according to a contract Net Protocol (section 2.2.2). The GS is also in charge of Grid adaptation in the case of disturbances. The adaptation is guided by a rule base that describes different actions to execute depending on the perceived event. The GS functioning is detailed in section 3.
- The lower layer is the *local level* made of Local Schedulers (LS) and resources. Each LS manages one or several resources belonging to the same domain. LSs receive tasks from the GS and they are responsible for their scheduling according to a local policy. It consists in determining the order in which tasks are executed in their local domain. Resources provide high computing capabilities, execute tasks and return tasks results and their possible states (idle, breakdown, etc) to the LS. The LS has also to detect failures, to analyze them and to find

the necessary actions to adapt locally. Adaptations performed by LSs can deal with one of the three following aspects: i) the scheduling policy. Adaptation here consists in changing the local policy to take into consideration the perturbation event; ii) the local schedule. Adaptation here corresponds to rescheduling at the local domain level according to resources availabilities; iii) resources organization. It consists in adapting the role of different resources (example changing the type of task a resource is in charge of execution). Thus the LS searches for a feasible adaptation action, if there is no adequate adaptation, the LS sends a notification to the GS to undertake adequate action. The adaptation is also guided by a rule base that describes different adaptation action depending on the detected event (see section 3.4).

2.2 Protocols for Coordinating Grid's Schedulers

Let us now precise the protocols that are deployed on the one hand between Global Schedulers and on the other hand between a Global Scheduler and its Local Schedulers. Regarding the cooperation among GSs, it follows the contract net protocol [15], while between a GS and the LSs under its control, a tasks allocation protocol is used. Since the local tasks execution is less expensive than a remote one (from a communication cost point of view), the GS will first deploy the tasks allocation protocol, and in case this latter fails, the GS will execute the contract net protocol in order to sub-contract a remote tasks execution.

Each protocol transmits, receives and interprets communication acts. We use here FIPA-ACL [14] that supports high level communication between the different Grid components and eases their semantic interoperability. Let us now details each protocol

Tasks Allocation Protocol. This protocol allows the Global Scheduler to allocate tasks to its Local Schedulers. This allocation depends on the Global Scheduler objective (minimizing the time execution, load balancing between domains, etc). The GS selects the best LS according to that objective. The LS has the possibility to accept or reject the transmitted tasks according to its policy, to its objective and to the state of the resources it manages. Once it has accepted the submitted tasks, if some problem occurs, the LS may have to perform some adaptation.

The Contract Net Protocol Ruling Global Schedulers Interactions. The Contract Net [15], a very known protocol in multi-agents area, allows an initiator Global Scheduler to sub-contract tasks to other GSs. We use the Contract Net in our architecture since it allows the GS to define conditions placed upon the execution, to receive several proposals from its acquaintances, to negotiate with them for that execution and to choose the GS that maximizes its objective. The initiator GS requests proposals describing the tasks to be executed and the conditions. It then acts as the manager of the tasks. Conditions include requirements such as time (desirable period to complete the execution) or performance constraints (storage capacity, bandwidth, etc). The GSs receiving the request can either propose or refuse. The Manager receives proposals, evaluates them and chooses the best GS. An acceptance message will be sent to the chosen GS and a refuse message will be sent to the other GSs. Once the Manager sends an acceptance message, the GS has to perform the sub-contracted tasks and to send back a completion message when it performs tasks successfully or a failure message when it fails to execute the tasks.

Protocols between Schedulers. The Petri Net of figure 1 provides a detailed view of interaction protocols between the Grid's Schedulers and their local behaviours in order to perform a plan1. This Petri Net shows how a Global Scheduler (called Manager) interacts with other Local and Global Schedulers. This Petri Net has been specified and simulated with the renew platform [16]. The Petri nets are known to have the adequate expressive power to describe behaviour (control structure, internal actions and communication) of distributed and parallel systems. Moreover, they provide several advantages such as a well-founded semantics, an executable specification, and a mean to enable simulations and proofs of behavioural properties.

This whole behaviour is made of four interconnected sub-nets:

- The Manager is a Global Scheduler in charge of a given user query (Plan). The Manager's part of the net is composed of transitions T1 to T10. Once the Manager accepts a plan (T1), it decomposes it into sub-plans (T2) and allocates them to a Local Scheduler (T3) and/or sub-contracts them to another Global Scheduler (T4). If the sub-plan is allocated to the Local Scheduler, the result could be a success stored by the transition T7 in the "Answer Success" place. On the contrary, if the Local Scheduler fails, a global adaptation is performed (T8). This adaptation could fail, and the final result is put in the "Answer failure" place, or could succeed and then a new loop is started by putting the sub-plan in the "Plan Accepted" place in order to schedule it (T1) again and so on.
- The Local Scheduler and resource's part of the net, made up of the transitions T11 to T21, describes the behaviour of a Local Scheduler (LS) and a resource in a local domain. The LS could accept (T12) or reject (T11) an allocated sub-plan. If the sub-plan is rejected, the Manager will try to allocate it again to another LS (T3) or to sub-contract it to one of its acquaintances (T4). If the LS accepts to treat the sub-plan (T12), the LS schedules it in the local domain (T13). It consists in allocating a resource to each task of the sub-plan and then the execution by the resources could take place. If the execution of all the tasks succeeds, the result is put in the "Local Success" place. In any other case, the tasks execution is interrupted and a local adaptation is performed (T16). If the local adaptation succeeds, a new loop is repeated.
- The right part of the net, made of the transitions T22 to T24, is a view on the behaviour of a Global Scheduler acquaintance (called GS2) playing the role of a sub-contractor for a GS. After the reception of a sub-plan ("sub-plan sub-contracted") place, GS2 will treat it (T22). T22 is an abstract transition which could be expanded to be compliant with any Global Scheduler behaviour (Manager's net). The treatment could succeed and be put in the "Global success place" or could fail and be put in the "Global Failure" place.

It is important to remark that the transitions T3 and T4 are also abstracted transitions corresponding respectively to the tasks allocation and to the contract net protocols as described in section 2.2. The Global Plan (T2) represents the internal behaviour of the Global Scheduler and will be detailed in section 3.1.

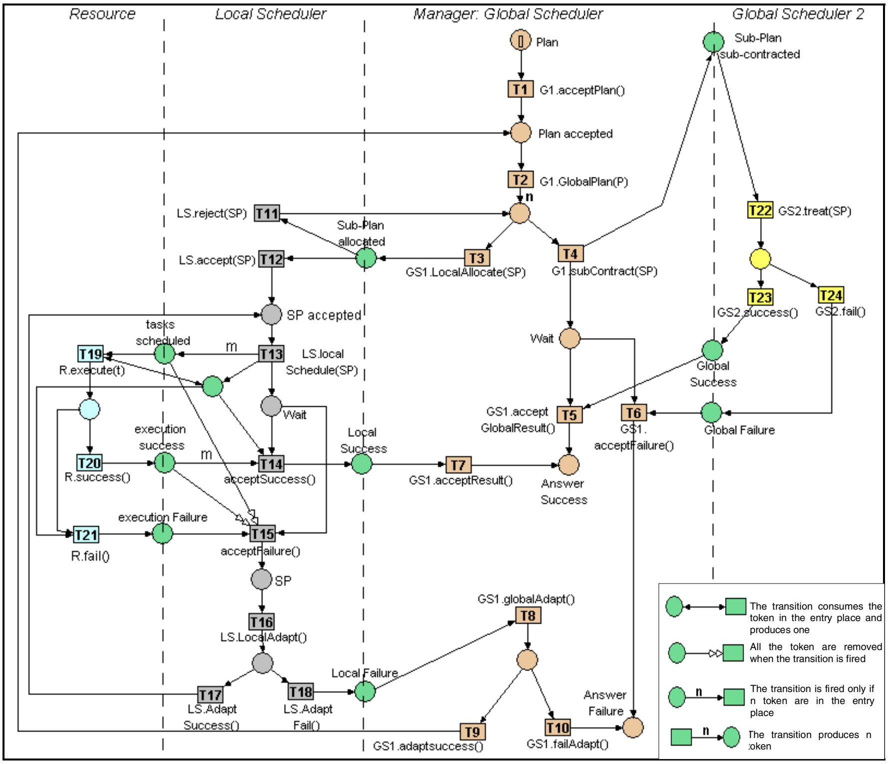


Fig. 1. Petri Net Global Scheduling Functioning

3 The Global Scheduler and Its Adaptive Behaviour

The Global Scheduler is responsible for assigning application's tasks to a specific Local Scheduler and/or for sub-contracting them to its acquaintances. Moreover, as Grid environment is dynamic and complex, the GS has to take into consideration these constraints and to undertake the necessary actions to adapt in the case of disturbances. To allow the GS to reason, to collaborate, to make adaptation decisions directed by goals and to consider the environment state, we choose a BDI (Belief-Desire-Intention) architecture [10]. The aim of this section is to describe such an architecture and its influence on the GS behaviour.

3.1 General Principle

The BDI behaviour is based on three concepts: Beliefs, Desires and Intentions. In our context, *Beliefs* are information about the Grid and its perturbations. The GS beliefs are described through a conceptual model detailed in section 3.2. *Desires* formalize high level grid objectives that the GS aims to reach. Objectives can be expressed by performance metrics. *Intentions* correspond to plans of actions to perform in order to achieve an objective. In our case, we have adaptation plans and tasks plans.

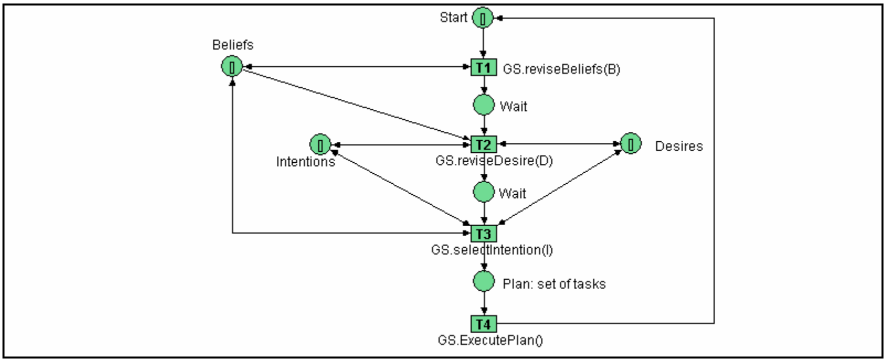


Fig. 2. BDI Scheduler functioning

Figure 2 shows a Petri Net specification of the BDI behaviour of the GS Scheduler. This behaviour details the Global Plan transition (T2) of figure 1. Each BDI concept is represented by a place and the GS performs a loop consisting in:

- Perceiving the environment and revising its beliefs (T1). All the information about the environment and its evolution are stored in the "Beliefs" place.
- Revising the desires (T2). According to the context, some objectives could be given up if impossible, and others introduced.
- Selecting intentions (T3) and therefore plans for a possible execution.

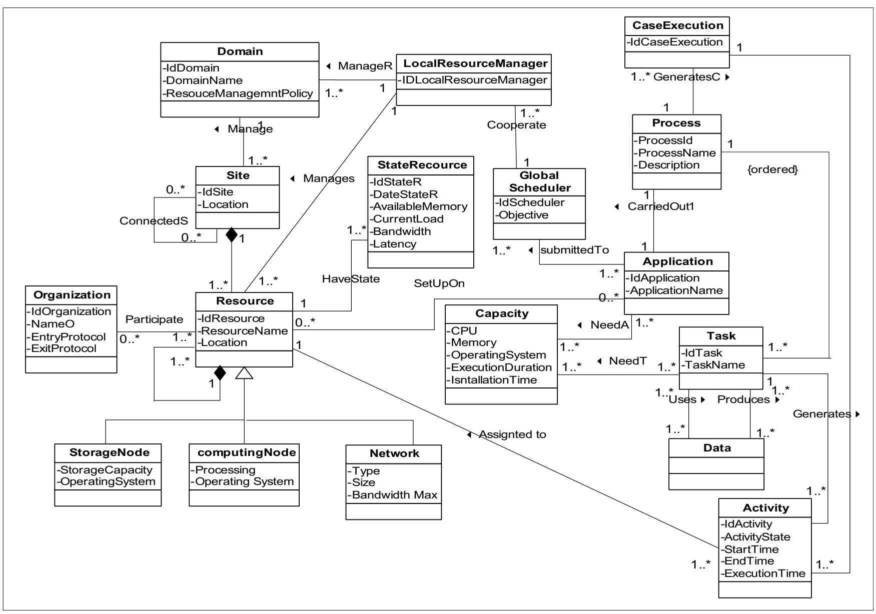


Fig. 3. Conceptual Model of Grid environment

- Executing the selected plan (T4) modifies the environment, then the transition ReviseBeliefs (T1) updates the beliefs and a new loop is started.

The language of the net, which gives all the possible behaviours, is $(T1.T2.T3.\ T4)^n$ where n is the number of loops.

3.2 The Beliefs

We model the GS beliefs with a conceptual model that describes information on the Grid environment that the GS owns or acquires via sensors. In the model suggested, the beliefs represent all the information on resources, their availabilities, their states (allowed, free, waiting...), information on applications and tasks requirements (required memory, operating system, start time, termination time...) and their states (affected, waiting, finished...). Thus, the Beliefs include both static and dynamic information.

3.3 The Desires

Desires are the objectives that the GS aims to achieve. Grid users are basically concerned with their application's performance, (the cost, the maskepan, etc), while the GS is concerned with global Grid performances. Several metrics can be observed: The total flow time [17] that is the time that the completion of all tasks has taken. The maximum flow time [17] is the maximum time a task has spent in the system. The GS aims to minimize these metrics. The metric workload completion [17] is the number of completed applications compared with the total number of applications in the Grid. The task completion [17] is the number of completed tasks compared with the total number of tasks in the Grid. The GS tries to maximize these two metrics.

3.4 The Intentions

Intentions correspond to the selected plans to execute among a set of possible plans via effectors (resources). A plan is a set of actions to execute in order to achieve the Global Scheduler objective while taking into consideration the environment state. The selected plans must be in conformity with the realization of the desires (objective). The selection of plans can be based on a hierarchy of plans according to their degree of effectiveness (estimated a priori or measured) in the achievement of an objective. We distinguish tasks plan from adaptation plan.

Tasks plans: are a set of coordinated tasks according to a process and treated by the GS using its Beliefs, its Desires and its Intentions.

Global Adaptation plan: In the case of disturbance, the GS must reason about the perturbation and if necessary gives up the current plan in execution and computes an adaptation plan. Adaptation plan are actions to undertake in order to adapt in case of disturbing events. The actions are already described in [5] and can correspond to one of the following types: re-scheduling, adaptation of the organisational structure, adaptation of the quality of service, adaptation of the scheduling policy. Re-scheduling consists in computing and executing a new schedule plan. The adaptation of organisational structure concerns resources that are not isolated but belong to an organization in which they hold a role. The adaptation in this case consists in reorganizing these organisational structures (adding or removing resources, roles modification, etc.). The adaptation of scheduling policy allows the GS to dynamically switch from one policy to another. The plans are expressed in the form of active rules: `When < Event > If < Condition > Then < Action >`.

The part < Event > relates to the situation that must prevail in the environment and corresponds to disturbances. The part <Condition > corresponds to properties of tasks, resources, etc., and the part < Action > corresponds to adaptations.

4 Discussion and Conclusion

This paper proposes a multi-agent architecture for a cooperative and adaptive scheduling in Grid Systems. The architecture components interact through high-level protocols, and the Global Scheduler follows a BDI behaviour. This framework presents several advantages, including:

- *Explicit and flexible coordination*. We have followed the separation of concerns principles and modelled the Grid scheduling system as several interacting and coordinated components (Resource, Local Scheduler, Global Scheduler), each component coping with a well defined function. The coordination has been considered as a first class component, isolated and described by interaction protocols. These protocols give means to allocate or to sub-contract tasks in an opportunistic and dynamic way. Moreover, this explicit description of the coordination, through protocols, features numerous software engineering properties: easy design, understandability, maintenability and verifiability.

- *Semantic Interoperability*. The use of FIPA-ACL, that includes references through ontology and that admits language content as a parameter, eases semantic interoperability by allowing Grid components to adopt the same vocabulary. In this way, we also provide a conceptual model of the Grid on top of which a Grid ontology can be built easily.

- *Adaptive behaviour*. We use a BDI behaviour to allow the Global Scheduler to perceive the environment, to reason and to make decisions about what actions to perform, directed by goals. This functioning allows the GS to handle Grid perturbations and to undertake the necessary actions. For that purpose, we provided a typology of active rules. This declarative and modular expression of rules eases the maintainability and the understandability of the Global Scheduler adaptive behaviour.

Regarding future work, on the basis of our Petri Net specifications, we intend to implement a simulator of our architecture and measure the adaptiveness and the efficiency of our solution, in particular, interaction overhead, robustness and scalability in case of disturbances. The JADEX multi-agent platform, which integrates BDI architecture, will be used for that purpose.

References

1. Wrzesinska, G., Maassen, J., Bal, E.S.: Self-Adaptive Applications on the Grid. In: 12th ACM Sigplan symposium on Principles and Practice of Parallel Programming, California, pp. 121–129 (2007)
2. Vadhiyar, S.S., Dongarra, J.J.: Self Adaptivity in Grid Computing. Concurrency and Computation: Practice & Experience 17, 235–257 (2005)
3. Lee, K., Sakellariou, R., Paton, N.W., Fernandes, A.A.A.: Workflow Adaptation as an Autonomic Computing Problem. In: The 2nd Workshop on Workflows in Support of Large-Scale Science (Works 2007), Proceedings of HPDC 2007, California (2007)
4. Chakravarti, A.J., Baumgartner, G., Lauria, M.: The organic grid: self-organizing computation on a peer-to-peer network. IEEE Transactions on Systems, Man, and Cybernetics, Part A 35(3), 373–384 (2005)
5. Thabet, I., Hanachi, C., Ghedira, K.: Vers une architecture de type agent BDI pour un ordonnanceur de grille adaptatif. In: Conference on software architecture, Montreal, pp. 19–33 (2008)
6. Berman, F., Wolski, R., Casanova, H., Cirne, W., Dail, H., Faerman, M., Figueira, S., Hayes, J., Obertelli, G., Schopf, J., Shao, G., Smallen, S., Spring, N., Su, A., Zagorodnov, D.: Adaptive Computing on the Grid using AppLeS. IEEE Transaction on Parallel and Distributed Systems 14(4), 369–382 (2003)
7. Reed, D.A., Mendes, C.L.: Intelligent Monitoring for adaptation in Grid Applications. IEEE 93(2), 426–435 (2005)
8. Vadhiyar, S.S., Dongarra, J.: A Performance Oriented migration Framework for the Grid. In: CCGrid, IEEE Computing Clusters and the Grid. CCGrid, Tokyo, pp. 130–139 (2003)
9. Petri, C.A.: Fundamentals of a theory of asynchronous information flow. In: IFIP Congress, Amsterdam, vol. 62, pp. 386–390 (1963)
10. Rao, A.S., Georgeff, M.P.: BDI Agents: from Theory to Practice. In: The First International Conference on Multi-Agent Systems (ICMAS 1995), California, pp. 312–319 (1995)
11. Foster, I., Jennings, N.R., Kesselman, C.: Brain meets brawn: why Grid and agents need each other. In: The 3rd International Conference on Autonomous Agents and Multi-Agent Systems, New York (2004)
12. Cao, J., Jarvis, S.A., Saini, S., Kerbyson, D.J., Nudd, G.R.: ARMS: an Agent-based Resource Management System for Grid Computing. Scientific Programming 10(2), 135–148 (2002)
13. Shi, Z., Huang, H., Luo, J., Lin, F., Zhang, H.: Agent-based Grid computing. In: 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2007), Rio de Janeiro (2007)
14. Foundation for Intelligent Physicals Agents, `http://www.fipa.org`
15. Foundation for Intelligent Physicals Agents: FIPA Contract Net Interaction Protocol Specification, `http://www.fipa.org/specs/fipa00029/SC00029H.pdf`
16. The reference Net Workshop, `http://www.renew.de/`
17. Iosup, A., Epema, D.H.J., Franke, C., Papaspyrou, A., Schley, A., Song, B., Yahyapour, R.: On Grid Performance Evaluation Using Synthetic Workloads. In: Frachtenberg, E., Schwiegelshohn, U. (eds.) JSSPP 2006. LNCS, vol. 4376, pp. 232–255. Springer, Heidelberg (2007)

Dynamic Execution of Coordination Protocols in Open and Distributed Multi-Agent Systems

Wassim Bouaziz and Eric Andonoff

IRIT / UT1, 2 rue du Doyen Gabriel Marty
31042 Toulouse Cédex, France
{Wassim.Bouaziz,Eric.Andonoff}@univ-tlse1.fr

Abstract. Coordination protocols are widely recognized as an efficient mechanism to support agent interaction involved in modern Multi-Agent System (MAS) applications. This paper proposes a solution for dynamic execution of coordination protocols in such open and distributed MAS applications. More precisely, this paper shows how agents can dynamically take part in conversations, where the role they are intended to hold within a protocol is played without the need of prior knowledge.

Keywords: Coordination Protocols, Protocol Execution, Multi-Agents Systems.

1 Introduction

Modern Multi-Agent Systems (MAS) applications such as electronic commerce, business processes or crisis management involve distributed, heterogeneous and autonomous agents. Moreover, this partnership is generally open and dynamic built: This means that participating agents do not know each other and moreover they can be involved in several applications for which they hold different unknown roles. In such a context, agent coordination issue is important [1].

Coordination based-protocols are widely recognized as an efficient mechanism to share resources and coordinate activities of agents involved in MAS [2]. Coordination protocol can be defined as a set of coordination rules that agents respect during a conversation. These rules define whom, when and how agents may interact in a conversation. The importance of protocols in MAS results from the fact that they not only structure and organize the interaction between agents but also provide means for making coordination rules explicit by defining agent roles and the way they interact in an organizational view.

For these reasons, coordination based-protocols haves became the subject of extensive research, considering protocols as first class entities and addressing engineering issues such as design, validation and implementation of these protocols [2,3,4,5,6,7,8]. Some of them only addressed coordination protocol description issues [9,10,11], but we also found in the literature propositions dealing with coordination protocol execution. Some of these propositions are dedicated to specific application domains, such as electronic commerce [6], web service conversations [7]. Others provide independent application

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 609–618, 2009.

domain solutions [2,12,3,4,5], and propose a distributed protocol execution in which each agent fill a role and execute it.

Unfortunately, even if these solutions deeply investigate protocol execution they lack dynamic role integration. Now, this dynamic acquisition of role is crucial in order to face modern MAS applications. Indeed, agents involved in such applications must perform different roles without being re-initialized and re-started. For instance, an agent involved in a contract-net protocol (application) can later be lead to negotiate using an auction protocol with other agents involved in another application.

The main question addressed in this paper is: how can agents dynamically acquire role behavior in a conversation at run-time in order to coordinate with other agents?

In previous papers, we have already provided a solution to deal with description and dynamic selection of protocols [13], [8]. Thus, this paper addresses the two following research issues: how agents extract a role behavior from a protocol representation and how they execute it dynamically.

The reminder of the paper is organized as follow. Section 2 introduces our solution for role behavior extraction from protocols. Section 3 explains how agents can dynamically execute extracted roles without being restarted or even modified using a specific component called Micro Role engine. Finally, section 4 compares our solution with related works while section 5 concludes the paper.

2 Extracting Roles from Protocol Specification

We have already proposed a coordination protocol ontology. We do not report here this ontology (the reader will find a complete presentation of it in [13] or at the following address http://w3.univ-tlse1.fr/ceriss/soc/perso/bouaziz/ontology). The section rather focuses on role extraction; it presents the model used for describing extracted roles and also proposes an algorithm for extracting roles.

2.1 Role Meta-model

A role is a behavior expressing how an agent can interact when it is involved in a protocol conversation. This behavior is defined as a coherent set of actions. These actions are activated when an event occurs. For example when a message is sent or received or when a temporal event is triggered. The execution of these actions progresses the role local state and consequently the global state of the protocol. In general, each participating agent must assume a well-defined behavior expressed as permitted actions in a conversation. Giving a protocol description, role behavior must be extracted separately, as a specific entity. For this purpose, we precede by defining a meta-model for roles. Thus, for each role of a protocol, we instantiate this model by extracting information related to the considered role from our Protocol ontology. Figure 1 describes the main concepts of roles.

This meta-model depicts role concepts and their relationships. A role is described through its profile and its behavior. The role behavior defines the different allowed actions it enacts. An action has a type, its data field, an operation that implements it, and a set of input and output events.

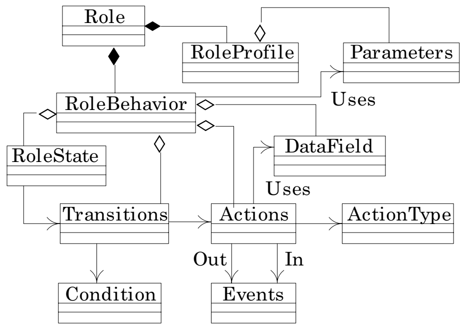


Fig. 1. Role Meta-model

Note that an action is executed only when an event occurs; then, it activates its corresponding operation. An event can be a temporal message or a message sent or received by the role. When executed, an action can produce a set of events. The procedure flow of actions is defined in the transition concept. A transition defines the possible sequence actions for each role behavior. A transition can be conditional or unconditional. Finally, a role behavior has a set of states defining the progress of the interaction. A role state changes only when a set of transitions is activated.

2.2 Extracting Role Behavior from Protocols

We provide here an algorithm for role extraction and instantiation. This algorithm produces an XML document describing the extracted roles from a protocol specification, according to the Role meta-model.

The algorithm named Role Extractor and visualized below, has three inputs: the Protocol ontology's url, the protocol name and the role name. It starts by generating an XML tree with an empty node, in accordance with the XML document. It implements queries which select information from the protocol specification according to the concepts defined in the Role meta-model. It leads to the definition of XML nodes (tags) describing both role profile and behavior. To make these queries possible, we have defined a set of predicates in the Protocol ontology. For instance, the conjunction of the two predicates *IsResponsible*() and *Involes*() returns the actions of a role in a protocol. We also defined a set of functions to manipulate trees. For example, to add a new child to a node, we use the *AppendChild*() function.

2.3 Role Concretization

After having extracted and generated role description with respect to the defined Xml schema, we translate this representation to a Java class. The generated Java class represents only the skeleton of the role, where transitions and actions are implemented as Java methods. In addition, a programmer has to concretize in Java either the code to be executed by the actions described in the role, or the call to services of agents holding the role.

Algorithm 1. Role Extractor

```
Input UrlProtocolOntology PO, ProtocolName PN, RoleName RN
Output RoleBehavior TR
Variables S: State, A: setof(Action)
Begin
    TreeRoleName TR ← Create_New_Empty_Tree(Schema S)
    RootElement.name ← RN
    // instantiate actions
    A ← {Ø} : A'← {Ø}
    A ←  Select (?A) From PO
         Where IsResponsible(?RN,?A) and Involves(?PN,RN)
    For Each action a in A do
      CNode ← Current_Node(TR, Actions )
      Element E ← Create_Action_Node(a,PN,RN)
      If InitialState(?S,?RN) and Activates (?a,?S)Then
         E ← InitialAction(E)
      Endif
      If FinalState(?S,?RN) and Activates (?a,?S) Then
         E ← FinalAction(E)
      Endif
      CNode ← AppendChild(E,CNode)
      A'←  Select (?A) From PO Where Succeeds(?a,?A)
      // ?a is the current action and ?A is a variable
      // representing other actions following ?a
      While A' is not empty Do
         // instantiate transitions
         a' ← Current_Element(A')
         CNode ← Current_Node(TR, Transitions)
         Element E ← Create_Transition_Node(?a,?a')
         CNode ← AppendChild(E,CNode)
      End While
      // instantiate roleStates
      CNode ← Current_Node(TR, RoleStates)
      S ← Select roleState From PO Where Activates(?ai,?S)
      Element E ← Create_State_Node(S)
      CNode ← AppendChild(E,CNode)
    End for
End
```

3 Dynamic Roles Integration

This section introduces our strategy for role integration and execution at run time. It presents the MR engine component we defined for loading and executing role behavior. It also explains how an agent holding a role pilots this MR engine.

3.1 Micro Role Engine Component

To hold a role, an agent needs to load and execute the extracted behavior at run time. Load means instantiate the role behavior according to the internal state of the agent (with it specific settings or parameters), and execute means enact it on the fly.

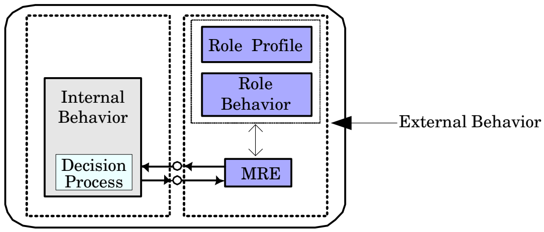


Fig. 2. Agent with Micro Role Engine

In order to support this, we extend the traditional agent behavior embedding a specific component, called Micro-Role (MR) engine, responsible for loading and executing extracted roles. When created, an agent is equipped with this specific component. It will run its MR engine only when it will participate to a conversation ruled by a specific protocol. Thus, we distinguish the internal behavior of the agent from its external behavior, which corresponds to the actions the agent has to execute when it holds a role in a conversation. The agent is itself responsible for the execution of its internal behavior (what it has to do), while the MR engine it integrates is responsible for the execution of its external behavior (what actions it has to execute within the role it holds). The connection between the internal behavior and the role behavior monitored by the MR engine is supported through communication between agent local behavior and the MR engine it integrates. This communication permit to give values to the different variables defined in the role profile. Figure 2 below illustrates the architecture of an agent integrating a MR engine. This latter executes the actions corresponding to the held role. So, when an agent integrates a new role, the MR engine initializes the different variables of the role (role profile), reads the role specification, triggers the actions to be done and waits for incoming messages from others agents involved in the protocol.

The actions triggered by the MR engine correspond to messages sent to the other participants of the conversation. When receiving a message, the MR engine reads the role specification and triggers the actions to be done according to its current state. Then, either it waits for new messages or it ends its participation in the protocol.

To go further into details, the MR engine is a small component, which extends the basic behavior of agents, and which is able to execute role behaviors described by our role meta-model. In fact, for each role behavior, we have a corresponding Java class that exactly implements the behavior defined in the role. Thus, the MR engine is able to load using an ad hoc class loader a compiled Java class and to execute it.

Dynamic role enactment is a nontrivial problem because of the distribution of participants as demonstrated in works addressing the enactability of protocols issue [14,15]. Regarding our proposition, visibility on exchanged messages between participating agents is partial since we adopt, as in [2] a moderator-based approach. It is demonstrated in [2] that the moderator maintains the global state of the protocol, constraints the behavior of participating agents and enforces the protocols rules. Thus, the moderator is the host of a conversation: it ensures that the enactability of the protocol is correct because it maintains the global state of the conversation. The different roles involved in the protocol share this global state. Consequently, enacting protocols using moderators impose to the different agents playing roles to exchange messages in the right order and constraint agents to act correctly (i.e. in consistency with the protocol rules). In addition, the messages exchanged are point-to-point messages, involving only two roles: a participant and a moderator. Conseuquently, in this paper, we address neither the problem of protocol enactability, nor the conformance of the extracted roles to the protocol; we only focus on how agents can dynamically acquire roles.

3.2 How to Communicate with the Micro Role Engine?

It is also important to discuss about driving role execution highlighting the communication between the internal behavior of the agent and its MR engine. We distinguish three objectives for driving role execution: (i) binding values to variables corresponding to those defined in the role profile, (ii) making decision which corresponds to the agent strategy in the way it holds the role, and (iii) supervision of the MR engine execution. The first objective is mandatory for a minimal execution of a role, while the others are required for a more advanced execution of a role. We now discuss about the solution we provide to support a communication ensuring driving role execution in order to reach these three objectives.

Consequently, we propose to define an interface between the internal behavior of the agent and the MR engine. This interface ensures the communication between the agent and its MR engine providing services.

First, two services are provided to support the binding of values introduced before: RequestVariables(parameters) and ProvideValues() where parameters correspond to the values defined in the role profile. The first service will be used by the agent to receive a request from the MR engine. The second service provides the MR engine with values given to the parameters. For instance, an agent holding the Manager role within the Iterative Contract Net [17] protocol decides which criteria to use for bid evaluation and consequently which bid to select. Likewise, an agent holding the Contractor role decides to reply to a CFP (call for proposal) or not.

Second, we propose two other services to support both making decisions and role execution supervision. These two services are the following: RequestDecision() and ProvideDecision(). Regarding making decisions, RequestDecision() corresponds to a request from the MR engine when alternatives occur in role execution, and ProvideDecision() is used to answer to this request. Of course, to be able to answer to a request, we consider that agents have a set of basic capabilities. For instance, an agent holding the Manager role in the Iterative Contract Net protocol is able to evaluate a bid, to compare two bids... This step corresponds to strategic aspects in the way of holding a role for an agent. Regarding the supervision of the MR engine execution, the ProvideDecision() service is used by agents to suspend, follow-up or stop the execution of the role.

Regarding making decisions, agents increase their capabilities by using a Process Decision ontology. Such an ontology defines a set of strategies for each role embedded in coordination protocols. Strategies are represented as sets of rules. An agent that has previously defined as being able to interpret a strategy described in the ontology is then be able to load and execute any interaction protocol that is defined in concordance with this ontology. The presentation of this ontology is out of the scope of the present paper.

Consequently, we do not impose any constraints on the type of agents holding roles but just assume they ensure the four listed services. The type of these agents can differ according the application in which they are involved. For instance, in the context of a cross-organizational business process, participating agents must integrate a workflow engine in order to execute their local processes, which are a part of the inter-organizational process.

4 Related Works

Regarding roles extraction, all proposed solutions in the literature depend on the nature of the used protocol description formalism. Existing works propose a specific operation, named Projection, to extract roles [5,12,14] from protocol. For example if the protocol is described as a Petri net, the projection is realized by replacing all transition in the protocol description (or rules, depending on protocols specification formalism), which have a send or a receive action for which the extracted role is neither a sender nor a receiver, with a virtual transition and then those transitions are removed by determinization. The generated role skeleton is a final state machine in this case. In [5] authors propose an algorithm to tackle the role extraction problem. The protocol is described as a set of rules described using SWRL language. This algorithm transforms and replaces the rules that have the role R as receiver or as sender of messages. [12] proposes an algorithm for generating a final state machine representing the role from protocol specified using a variant of Petri Net. Finally, [14] describes a protocol as a final state machine and applies the same principle as [12] for extracting role behavior.

Our role extraction algorithm is different. It uses the semantic relations defined in the Protocol ontology as predicates to extract role behavior performing general queries supporting both the selection of all the actions of a given role and the reconstitution of the coordination of these actions (input events, content, output events, variables…).

Regarding execution of roles, we distinguish three types of propositions. The first one proposes a static integration of role, i.e. integration of role when designing agents holding the role. For instance, in [3], roles are defined separately as an XML document and then transformed in an executable code, which is manually integrated into agent behavior at design-time. In [5], Desai et al propose OWL-P, based on commitments, for describing business process protocols. In this representation, role behavior is defined as a set of declarative rules. When an agent participates in a cooperative business process (inter-organizational business process), it generates the skeleton of the role it is going to hold in the business process protocol (i.e. it generates the rules of the role behavior), augments this skeleton with its local decisions (policies) and deploys the result as a service. This solution allows reusability of roles since it separates roles and policies but it does not support dynamic integration of role. Finally, in Jade [18], roles are also defined separately from agents in an interaction protocol library. They are implemented as Java classes, and are integrated into agent's whishing holding them when programming agents code, i.e. at design-time.

The ROPE environment [19] is a representative sample of the second type of solutions: pre-defined integration of role. In ROPE, an agent fulfills a role if it is able to offer the required services for role execution. That means that agents have an a-priori knowledge about needed services for role execution. Consequently, this environment does not support dynamic integration of a-priori unknown roles.

We found only very few contributions supporting dynamic integration of roles. These works correspond to the third type of contributions for role execution. [20] proposes a formalization of the needed operations for an agent whishing holding a role. The proposed formalization only considers cognitive agents. This work has the same objectives as ours, for dynamic enactment and deactment of roles, but unfortunately it does not give any information about role specification and does not adopt an engineering perspective to deal with this problem. [12] considers roles as separated entities from agents holding them. Indeed, roles are hold by external agents called Participation agents. Shared variables are used to manage the interaction between agents and their Participation agents. This solution, like ours, does not make any hypothesis on the internal structure of the agent but it increases the number of interaction in the system. Finally, in [21] authors propose a platform for interacting web process using the concept of role described using a Petri net formalism. A process acquires a role at runtime, which starts and executes conversations, when some interaction with published services is required. Although dynamic loading and executing roles is supported, the communication between the process and the loaded role is limited to one-shot interactions provided by the input and output interaction at the starting and termination of role execution. Contrary, in our solution we propose an interface to allow the communication between the internal behavior of the agent and the MR engine, which is responsible to execute the role. This communication is necessary to guide the execution of the MR engine if, for instance, decisions have to be taken.

5 Conclusion

This paper has presented a new approach to deal with dynamic execution of coordination protocols in open and distributed modern multi-agent applications. Agents involved in such applications can dynamically change their roles and adopt others. In order to support such dynamic changes, the paper proposes to generate, from a protocol description, each role behavior embedded in this protocol, and to allow each participating agent to integrate the role behavior it will hold within the protocol at run-time. More precisely, the contribution of the paper is twofold. First, it defines a meta-model for role description and an algorithm to extract role (profile and behavior) from protocol specification. Thus, role behavior skeletons are automatically generated from protocol specifications and then, are augmented with a concrete realization of their actions. Second, the paper extends the internal behavior of agents equipping them with a specific component, called Micro Role (MR) engine, which supports dynamic (i.e. at run-time) loading and execution of roles. The paper also proposes to define an interface allowing the internal behavior of agents to pilot role behavior execution through the MR engine giving values to variables defined in the role profile.

We believe that this contribution is important since supporting dynamic execution of roles permit agents to face new open and distributed modern MAS applications such as e-commerce, e-government, crisis management and web services conversations [1]. It makes the participation to several conversations at the same time possible, since agents can switch their role behavior at run-time without being shutdown, retooled and restarted. So, agents can easily participate in different conversations using a protocol unknown a priori. Consequently, it makes agents freer and increases their autonomy.

Our research efforts are now focused on the implementation of our proposal using the Wade platform [16]. We planed to generate, for each extracted role, a java class describing both role behavior and profile and also to implement our MR engine whose aim will be to execute the behavior specified in this class. Our second objective is to deal with tactical aspects of role execution i.e. how the internal behavior of an agent can interact efficiently with the MR engine in order to participate in a conversation at the very best.

Acknowledgements

This work has been supported by the ANR (French Research Agency) through the IsyCri project (Interoperability of Information Systems in Crisis Management).

References

1. Huhns, M., Singh, M.: Research Direction for Service-Oriented Multi-agents Systems. Journal on Internet Computing, 65–70 (2005)
2. Hanachi, C., Sibertin-Blanc, C.: Protocol Moderators as active Middle-Agents in Multi-Agent Systems. Autonomous Agents and Multi-Agent Systems 8(2), 131–164 (2004)
3. Cabri, G., Leonardi, L., Zambonelli, F.: BRAIN: A Framework for Flexible Role-Based Interactions in Multiagent Systems. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) CoopIS 2003, DOA 2003, and ODBASE 2003. LNCS, vol. 2888, pp. 145–161. Springer, Heidelberg (2003)
4. Carabela, C., Beaune, P.: Engineering a Protocol Server Using Strategy-Agents. In: Mařík, V., Müller, J.P., Pěchouček, M. (eds.) CEEMAS 2003. LNCS, vol. 2691, pp. 413–422. Springer, Heidelberg (2003)
5. Desai, N., Ashok, U., Chopra, A., Singh, M.: Interaction Protocols as Design Abstractions for Business Processes. Journal on Transactions on Software Engineering 31(12), 1015–1027 (2005)
6. Tamma, V., Phelps, S., Dickinson, I., Wooldridge, M.: Ontologies for Supporting Negotiation in E-Commerce. Engineering Applications of Artificial Intelligence 18(2), 223–236 (2005)
7. Benatallah, B., Casati, F., Toumani, F.: Representing, Analyzing and Managing Web Service Protocols. Journal on Data and Knowledge Engineering 58(3), 327–357 (2006)
8. Andonoff, E., Bouaziz, B., Hanachi, C.: Protocol Management Systems as a Middleware for Inter-Organizational Workflow Coordination. Journal of Computer Science and Applications 4(2), 23–41 (2007)
9. Mazouzi, H., el Fallah-Serghrouchni, A., Haddad, S.: Open Protocol Design for Complex Interactions in Multi-Agents Systems. In: Conference on Autonomous Agents and Multi-agent Systems, pp. 517–526. ACM, Italy (2002)
10. Toivonen, S., Helin, H.: Representing Interaction Protocols in DAML. In: van Elst, L., Dignum, V., Abecker, A. (eds.) AMKM 2003. LNCS, vol. 2926, pp. 310–321. Springer, Heidelberg (2004)

11. Paurobally, S., Jennings, N.: Protocol Engineering for Web Services Conversations. Int J. Engineering Applications of Artificial Intelligence, Special Issue on Agent Oriented Software Engineering 18(2), 237–254 (2005)
12. Sibertin-Blanc, C., Hameurlain, N.: Participation Components for Holding Roles in MultiAgents Systems Protocols. In: Gleizes, M.-P., Omicini, A., Zambonelli, F. (eds.) ESAW 2004. LNCS, vol. 3451, pp. 60–73. Springer, Heidelberg (2005)
13. Bouaziz, W.: Une Ontologie de Protocoles pour la Coordination de Systèmes Distribués. Journées Francophones sur les Ontologies, Tunisia (2007)
14. Tevfik, B., Jianwen, S., Xiang, F.: Analyzing Conversations of Web Services. Journal on Internet Computing 10(1), 18–25 (2006)
15. Desai, N., Singh, M.P.: On the Enactability of Business Protocols. In: International Conference on Artificial Intelligence (AAAI), AI & the Web track (2008)
16. Banzi, M., Caire, G., Gotta, D.: WADE: A Software Platform to Develop Mission Critical Applications exploiting Agents and Workflows. In: Industry Track, Conference on Autonomous Agents and Multiagent Systems, Estoril, Portugal (2008)
17. FIPA, Foundation for Intelligent Physical Agent. Fipa Interaction Protocol Specifications, Technical Report (2002)
18. Bellifemine, F., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE. Wiley, Chichester (2007)
19. Becht, M., Gurzki, T., Klarmann, J., Muscholl, M.: ROPE: Role Oriented Programming Environment for Multi-Agents Systems. In: Conference on Cooperative Information Systems, Edinburgh, Scotland, pp. 325–333 (1999)
20. Dastani, M., van Riemsdijk, M.B., Huslstijn, J., Dignum, F., Meyer, J.J.: Enacting and Deacting Roles in Agent Programming. In: Workshop on Agent Oriented Software Engineering, New-York, USA, pp. 189–204 (2004)
21. Fabra, J., Al varez, P., Banares, J.A., Ezpeleta, J.: Runtime Protocol Binding: Flexible Service Integration by Means of Flexible Service Interactions. In: IEEE International Conference on Service Interactions, Services Computing, vol. (1), pp. 291–298. IEEE Computer Society, Hawaii (2008)

An Agent-Based Simulation Model for Analysis on Marketing Strategy Considering Promotion Activities

Hideki Katagiri, Ichiro Nishizaki, Tomohiro Hayashida, and Takahiro Daimaru

Graduate School of Engineering, Hiroshima University,
1-4-1, Kagamiyma, Higashi-Hiroshima, 739-8527, Japan
{katagiri-h, nisizaki, hayashida, takahiro-daimaru}@hiroshima-u.ac.jp
http://www.hil.hiroshima-u.ac.jp/

Abstract. This paper proposes a new agent-based model for simulation analysis on firms' marketing strategy in competitive markets environments. In the previous model proposed by Tay and Lasch, firm agents determine their marketing strategies such as price and product attribute at every period so as to maximize their profit. This paper shows that the firm agents constructed based on the Tay-Lasch model do not learn the preferences of consumer agents under a very simple situation. This article proposes a novel agent-based model based on a learning classifier system so that the firm agents have abilities to learn the preferences of consumer agents. Moreover, in order to construct a more realistic simulation model, promotion activity of firm agents is incorporated in the strategy of firm agents. Some relationship between the strategy the firm and profitable firm are clarified through simulation analysis.

Keywords: Agent-based simulation analysis, promotion activity, learning classifier system.

1 Introduction

In traditional analytical models for studying markets, it is often assumed that demand and supply are homogeneous, buyers and sellers have complete information, and consumers seek to maximize utility and firms seek to maximize profits. However, analytical methods become intractable when one takes away these simplifying assumptions.

In order to introduce a method to study complex and ill-defined business environments, agent-based simulation analysis, in which simulations of real-world phenomena are built from the microscopic actions and interactions of individual agents, has been developed in financial market [1,4] .

Recently, agent-based simulation models has been drawn attention in the field of economics and marketing [3,6,7,10]. Zhang *et al.* [10] present an agent-based model of consumer purchase decision-making to cope with the dynamic changes and complexities in the real-world business environment. They exhibit the emergent decoy effect phenomenon, which is a market dynamic phenomenon

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 619–628, 2009.

originating from the individual behavior of heterogeneous consumers and their interactions in the real-world complex market. Delre *et al*D[3] propose an agent-based model to simulate the efficacy of different promotional strategies that support the launch of a product. In particular, they concentrate on the targeting and the timing of the promotions and show that promotional activities highly affect diffusion dynamics. These models consider the situation where only one firm exists in the market.

Tay and Lusch [8] consider a competitive market and model individual agents as artificial adaptive agents, in which the artificial firm agent have a set of decision rules (if-then statements) that are evolved over time as they learn from their experiences. These if-then statements represent the agents' hypotheses about how the market works and characterize the decision-making process of the agents. The agents improve their performance through a simple process of feedback where their cognitions are adjusted based on the world they encounter. which is accomplished with the aid of a genetic algorithm.

In this paper, we firstly indicate that learning systems in the Tay-Lasch model do not work as they expect and that firm agents in their model do not have capabilities of learning the preference of consumer agents. Then, we propose a new agent-based model by incorporating learning classifier systems, especially eXtended Classifier Systems (XCS) [2,9] with an artificial market. Next, we incorporate firms' promotion activities into the firms' strategy as well as product attribute and price. We execute simulation analysis on which strategies are beneficial for earning much profit.

2 Agent-Based Simulation Model

In this paper, we consider two types of agents; one is a group of consumer agent $N = \{1, 2, \ldots, n\}$, and the other is a group of firm agents $M = \{1, 2, \ldots, m\}$. The aim of each of consumer agents is to buy at most one of the products provided by seller agents so that he/she satisfies he/her preference. The purpose of each of firm agents is to take an action based on his/her marketing mix strategy; firm agents determine their marketing strategy such as price and product at every period, considering other agents' strategy at previous period.

Fig. 1 shows the flowchart of the agent-based simulation model proposed in this paper. At the beginning of each period, firm agents announce their prices and product attributes based on extant information and the decision rules that have recently proven to be the most effective. The consumer agents respond by communicating their desires for each seller's product and, thus, the quantities that they will buy from the sellers. At the end of each period, the firm agents compute their profits and evaluate the effectiveness of the rules that they have used and update the rules track record. Periodically, each firm agent will assess the overall performance of all their rules since the last revision and create new rules to replace the rules that have performed poorly.

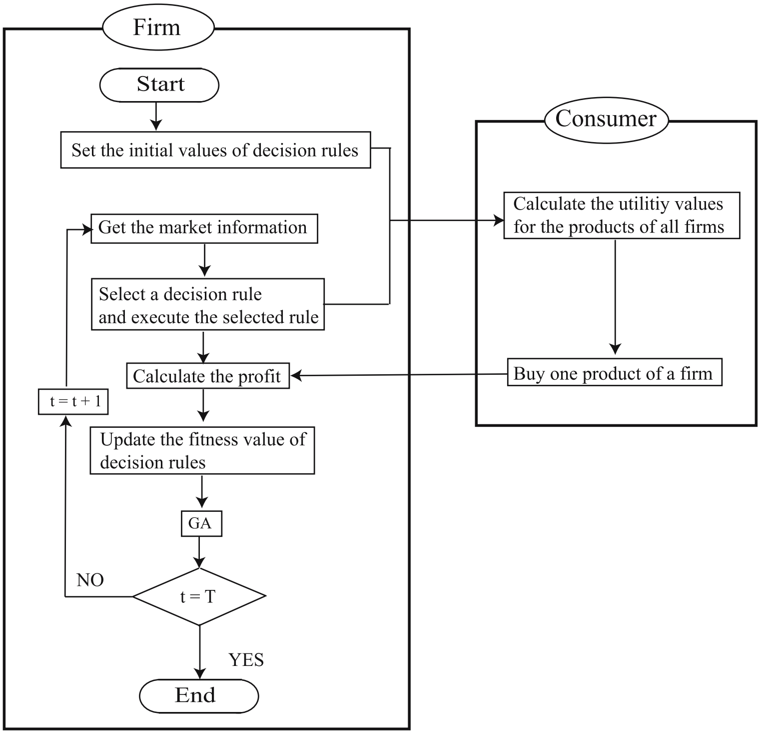


Fig. 1. Flowchart of the agent-based simulation model

2.1 Decision Making of Consumer Agents

In general, consumers desire to buy products with low price and with attributes which match with their preferences. In this point of view, Tay and Lusch [8] assume that the utility function, which they call demand function, is given in the following form:

$$D_{i,j}(t) = \max\{\ \alpha - \beta \times p_j(t) - \gamma \times |a_j(t) - g_i|, 0\} \tag{1}$$

where $p_j(t)$ and $a_j(t)$ are price and attribute of product that firm j make at period t. g_i denotes an ideal product attribute of consumer i. Parameter $\alpha (\geq 0)$ is a constant expressing buying inclination. Also, β and γ are weighting constants with respect to price and product attribute, respectively.

We assume that each consumer buys one of products with probability in proportion to the values of utility functions. To be more precise, the probability that consumer i buys product of firm j at period t is $\frac{D_{i,j}(t)}{\sum_{j=1}^{m} D_{i,j}(t)}$. If it holds that $D_{i,j}(t) = 0$ for all j, then consumer i does not buy any product at that period.

Table 1. Market environments

$s_p(t)$ or $s_a(t)$	Level 1 (low)	Level 2 (m-low)	Level 3 (m-high)	Level 4 (high)
$\hat{p}(t)$ or $\hat{a}(t)$	0–2.5	2.5–5	5–7.5	7.5–10

2.2 Decision Rules of Firm Agents and Their Evolution

In the agent-based simulation model by Tay and Lasch, firm agents have decision rules which take form of if-then statements and evolve their rules using genetic algorithm so as to earn profit as much as they can. However, the evolutionary learning system of firm agents' rule in the Tay-Lusch model has a crucial problem with respect to the learning ability, as shown in the results of simulations later. It should be noted here that the rule whose fitness value is larger may not be always useful for general conditions. In this point of view, the fitness values of rules should be directly associated with the conditions. However, since the if-part of rule set in the Tay-Lasch model corresponds to all possible conditions (environments) in advance, the fitness value of the rule does not directly correspond to each of conditions where the rule is regarded as a useful one. Therefore, more appropriate learning system of agent-based model is necessary, and we propose a novel agent-based model based on learning classifier systems.

Following the Tay-Lusch model, we assume that the values of price and product attribute take real values in the interval $[0, 10]$, respectively. Firms determine the prices and product attributes at every period after they roughly perceive the averages of prices and product attributes at the previous period. Let $\hat{p}(t)$ and $\hat{a}(t)$ denote the mean values of price and attribute of all firm agents at period t, respectively. In our model, we assume that firm agents realize $s_p(t)$ and $s_a(t)$ as shown in Table 1 instead of $\hat{p}(t)$ and $\hat{a}(t)$, where $s_p(t)$ and $s_a(t)$ denote the levels of price and product attribute, respectively.

Firm agents determine the price and attribute based on their rules. We apply learning classifier systems, especially the eXtended Classifier System (XCS) [2,9], for a learning system of firm agents. Unlike the Tay-Lusch model, if-part of the rule in XCS corresponds not to all conditions but to only a specific condition. Table 2 shows that an example of classifiers.

Table 2. Example of rules of firm agents based on XCS

cl_{jk}	$s_p(t)$	$s_a(t)$	$p_j^{out}(t+1)$	$a_j^{out}(t+1)$	$P_{kj}(t)$	$F_{kj}(t)$	$\epsilon_{kj}(t)$
1	3	1	4	5	60	0.5	10
2	2	1	7	2	65	0.6	15
.	.	.	.	.		.	
.	.	.	.	.		.	
.	.	.	.	.		.	
l_j	4	1	6	3	70	0.4	5

In Table 2, cl_{jk} denotes the kth classifier of firm j, and l_j is the number of classifiers that firm j has. The then-part $p_j^{out}(t+1)$ and $a_j^{out}(t+1)$ denote price and product attribute that firm j takes at the next period $t+1$. Following the Tay-Lasch model, we also assume that then-part takes integer values from 1 to 7. $P_{jk}(t)$ denotes a predictive value of classifier cl_{jk} which expresses the predictive profit at period t when cl_{jk} is selected as the rule used. $\epsilon_{jk}(t)$ denotes the error at period t which expresses the difference between the predictive value and the ideal value of cl_{jk}. $F_{jk}(t)$ denotes the fitness value which expresses predictive accuracy of cl_{jk} at period t.

In our agent-based simulation model, the algorithm of the decision rule of firm j is constructed based on learning classifier systems as follows:

Step 1: Set t=1.

Step 2 (Computation of predictive benefit): Calculate $B_{jk}(t)$ which denotes a predictive benefit of classifier cl_{jk}:

$$B_{jk}(t) = \frac{\sum_{cl_{jk}\in A_a} P_{jk}(t) \times F_{jk}(t)}{\sum_{cl_{jk}\in A_a} F_{jk}(t)} \tag{2}$$

where A_a denotes a set of classifier in which if-part matches with the market environments and then-part is a.

Step 3 (ϵ-greedy selection): Take a random action at probability ξ, and take an action $a_{\max} = \mathrm{argmax}_a\{B_{jk}(t)\}$ with probability $1-\xi$. where ξ is a sufficiently small positive constant.

Step 4: If $t > 1$, then go to Step 5. Otherwise, return to Step 2.

Step 5 (Update of predictive value): Update the predictive value of cl_{jq} at period t as follows:

$$P_{jq}(t+1) \leftarrow (1-\lambda)P_{jq}(t) + \lambda \times \{(p_j(t) - c_u) \times sl_j(t) - c_f\} \tag{3}$$

where $sl_j(t)$ denotes the number of products of firm j sold at period t, c_u is a unit product cost, and c_f is a fixed cost. Parameter λ characterizes learning speed, which is given as a positive constant.

Step 6 (Update of error): Update the error of cl_{jq} at period t as follows:

$$\epsilon_{jq}(t+1) \leftarrow (1-\lambda)\epsilon_{jq}(t) + \lambda|\{(p_j(t) - c_u) \times sl_j(t) - c_f\} - P_{jq}(t)|. \tag{4}$$

Step 7 (Update of fitness value): Update the fitness value of cl_{jq} at period t as follows:

$$F_{jq}(t+1) \leftarrow (1-\lambda)F_{jq}(t) + \lambda \times \kappa^{'} \tag{5}$$

where $\kappa^{'}$ denotes the ideal fitness value of classifier selected at period t, which is defined as

$$\kappa^{'} := \frac{\kappa}{\sum_{cl_{jk}\in A_a} \kappa_{jk}} \tag{6}$$

where

$$\kappa := \begin{cases} 1 & \text{if } \epsilon \le \epsilon_0 \\ \vartheta(\frac{\epsilon}{\epsilon_0})^{-v} & \text{otherwise} \end{cases} \tag{7}$$

Here, ϑ is a positive constant smaller than 1, and v is a positive constant more than 0.

Step 8 (Crossover): Select two parents from among a set of classifiers that have selected at the period t through the roulette wheel selection. Apply two point crossover for if-part of the selected two parents at probability $prob_{cro}$ and regard the generated classifiers through the crossover as offspring. Set the predicted values, fitness values and error values of the offspring the mean values of their parents, respectively.

Step 9 (Mutation): Select the gene locus of if-part and then-part of two offspring, and replace the original values by the random integer values from 1 to 10 at mutation probability $prob_{mut}$.

Step 10: Add the generated offspring to a set of classifiers and set $t = t + 1$. If $t = T$, then terminate the algorithm. Otherwise, return to Step 2.

2.3 Simulation Experiment

In this section, we investigate the learning ability of firm agents by comparing our model and the Tay-Lusch model under a very simple environment. We consider the situation where the ideal product attributes of all the consumers are 0, namely, $g_i = 0, \forall i$. Also, we set $\beta = 0$, which means that consumer agents do not pay attention to price at all. Needless to say, if firms agents have abilities to learn, they come to recognize the consumers' preference and regard the rules whose then-part of attribute are 0 as a useful rule set.

We set $n = 40$, $m = 4$, $T = 2000$, $c_f = 10$, $c_u = 1$, $prob_{cro} = 0.8$, $\alpha = 5.0$, $\beta = 0$, $prob_{mut} = 0.001$, $l_j = 10$ for any j. We set the initial value of the number of classifiers that each of firm have and its maximum as 500 and 10000, respectively. We execute simulation 100 runs and calculate its mean value. Tay and Lasch [8] calculate mean values of outputs for 360 periods. Therefore, in order to compare the simulation results of our model with those of the Tay-Lasch model, we follow the output pattern of Tay-Lasch model.

Fig. 2 shows the mean values of product attribute of our model and the Tay-Lasch model. It is observed from Fig. 2 that mean of the attribute value that the firm agents is about 5. This implies that in the Tay-Lasch model, firm agents

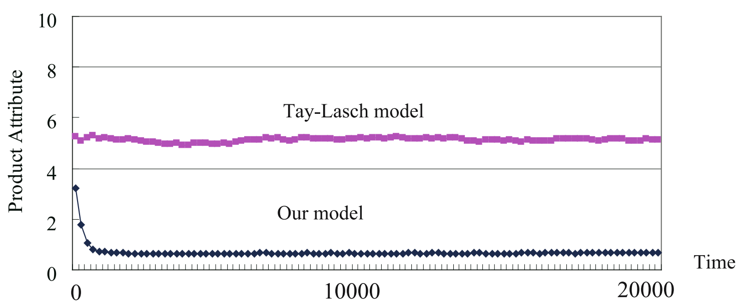


Fig. 2. Simulation results

do not realize the ideal attribute values of consumer agents and that the firm agents take the values of product attribute as random numbers in the interval $[0, 10]$. Therefore, it is highly possible that the learning system provided by Tay and Lasch does not work as well as they expect.

On the other hand, the simulation result of our model shows that firm agents recognize the ideal attribute value of consumer agents through learning process because the mean of the attribute value the firm agent set is close to 0.

From this simulation under a very simple marketing environment, the learning ability of firm agents in our model is superior to that of Tay-Lusch model. This is crucial with respect to the validity of the simulation model because firm agents are modeled with intent to behave adaptively as artificial adaptive agents who have abilities to adapt environments.

3 Agent-Based Simulation Model Considering Firms' Promotion Activities

In the real-world competitive market, firms perform their promotion activities to encourage consumer to buy their commodities. However, in the Tay-Lasch model, the firm agents are not allowed to perform promotion activities. Therefore, in order to make agent-based model in the market more realistic, we incorporate the promotion activity with a set of the rules of firm agents.

3.1 Modeling

By extending our model naturally along the model proposed in the previous section, we assume that firm agents have the set of rules which includes promotion levels (see Table 3). In other words, each of firm agents determine not only levels of price and attribute but also the level of promotion.

On the other hand, we assume that each consumer agent i has the following utility function for commodity of firm j sold at period t:

$$D_{i,j}(t) = \max\{\ \alpha - \beta \times p_j(t) - \gamma \times |a_j(t) - g_i| + \delta \times pr_j(t), 0\} \tag{8}$$

where $pr_j(t)$ is a promotion level of firm j at period t and $\delta(\geq 0)$ is a weighting coefficients with respect to the promotion level.

Table 3. Rules of firm agents considering promotion activity

cl_{jk}	$s_p(t)$	$s_a(t)$	$s_{pm}(t)$	$p_j(t+1)$	$a_j(t+1)$	$pr_j(t+1)$	$P_{jk}(t)$	$F_{jk}(t)$	$\epsilon_{jk}(t)$
1	3	1	4	4	5	1	60	0.5	10
2	2	1	2	7	2	5	65	0.6	15
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
l_j	4	3	2	6	3	5	70	0.4	5

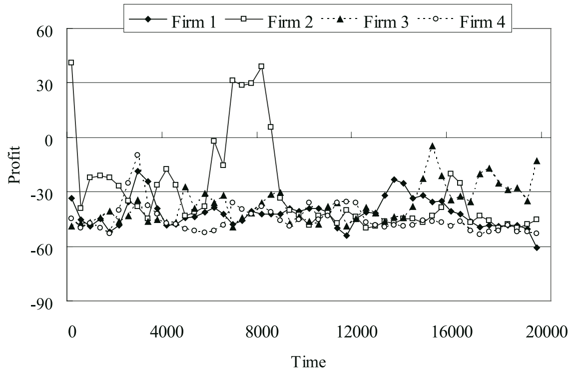


Fig. 3. Profit of firms (Tay-Lasch model)

3.2 Simulation Analysis

Setting the parameters $n = 40$, $m = 4$, $T = 2000$, $c_f = 10$, $c_u = 1$, $\alpha = 5.0$, $\beta = 0.5$, $\gamma = 2.0$, $\delta = 0.5$, $c_{pr} = 20$, we execute simulations and investigate what promotion strategies are profitable.

At first, we examine the validity of the agent-based simulation model. It should be noted here that firm agents are modeled with intent to behave adaptively so as to maximize their profit. Therefore, if the firm agents can learn the consumers preferences, then they evolve their rules appropriately and earn large profit. In this point of view, we compare the profits the firms earn in our model with those in the Tay-Lasch model. Figs. 3 and 4 show the results of Tay-Lusch model and our model, respectively.

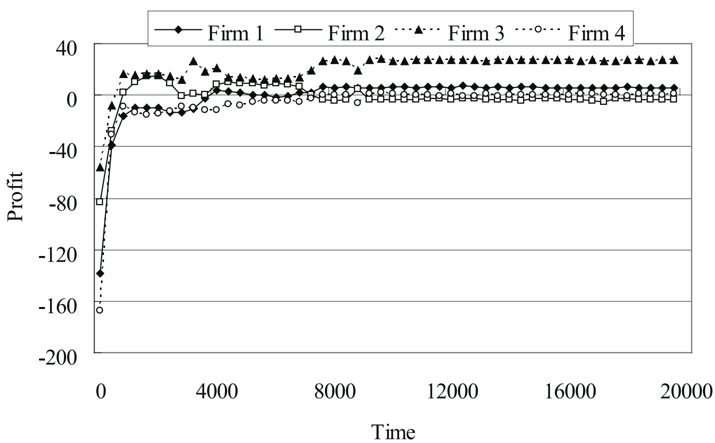


Fig. 4. Profit of firms (Our model)

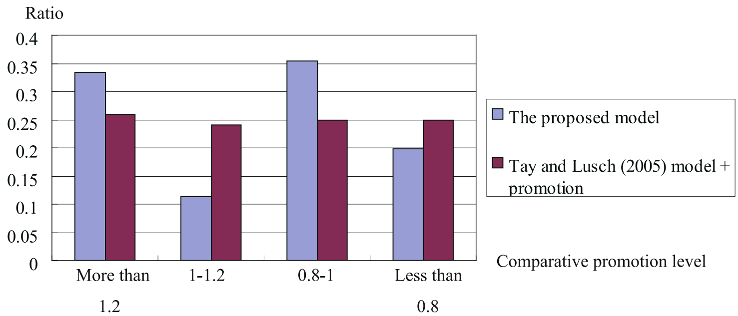


Fig. 5. Relationship between the most profitable firms and their comparative promotion levels

Comparing the results of Figs. 3 and 4, we observe that firm agents in our model earn larger profit than the Tay-Lusch model. This result is consistent with the fact that the learning system of the Tay-Lasch model is not appropriate, as discussed in the previous section.

Now we investigate the profitable strategy with respect to firms' promotion activities. To do this, we examine the relationship between the firm which earns the largest profit and its promotion level set by the firm. Fig. 5 shows the result of simulation.

The abscissa axis denotes the comparative promotion level of the most profitable firm for other firm agents, and the longitudinal axis denotes the ratio. Fig. 5 shows that in the Tay-Lusch model, the ratios for all cases are almost the same. This is probably due to the lack of learning abilities of firm agents in the Tay-Lasch model. On the other hand, in our model, the ratios in the ranges $[0.8, 1]$ and $[1.2, \infty)$ are relatively larger. Thus, even if firm agents take relatively low promotion levels, they may earn the largest profit. Needless to say, it is important to focus not only promotion levels but also price and product attributes, and further simulation analysis is needed to investigate the profitable strategy.

4 Conclusion

In this paper, we have proposed a new agent-based simulation model based on learning classifier systems in which firm agents evolve their marketing strategies so as to maximize their profits by learning the preference of consumer agents. We have shown that the firms' agent in the proposed learning system apparently have better abilities to learn the preference of consumer agents than those in the existing Tay-Lasch model. Furthermore, we have considered promotion activities of firm agents and incorporate their promotion activities into the firms' strategy. We have executed simulation analysis on the relationship between firms' promotion strategy and their profits. Although further simulation analysis is necessary, we have shown that our model has a great potential to analyze the profitable strategy in the real-world market.

References

1. Bangnall, A.J., Smith, G.D.: A multiagent model of the UK market in electricity generation. IEEE Transactions on Evolutionary Computation 9, 522–536 (2005)
2. Butz, M.V., Wilson, S.W.: An algorithmic description of XCS. Soft Computing 6, 144–153 (2002)
3. Delre, S.A., Jager, W., Bijmolt, T.H.A., Janssen, M.A.: Targeting and timing promotional activities: An agent-based model for the takeoff of new products. Journal of Business Research 60, 826–835 (2007)
4. Keown, A.J., Pinkerton, J.M., Young, L., Hansen, R.S.: Recent SEC prosecution and insider trading on forthcoming merger announcements. Journal of Business Research 13, 329–338 (1985)
5. DMcCarthy, E.J.: Basic Marketing: A Managerial Approach. Richard D. Irwin, Homewood (1960)
6. Emmanouilides, C.J., Davies, R.B.: Modelling and estimation of social interaction effects in new product diffusion. European Journal of Operational Research 177, 1253–1274 (2007)
7. Powell, J.H., Wakeley, T.M.: Evolutionary concepts and business economics: towards a normative approach. Journal of Business Research 56, 153–161 (2003)
8. Tay, N.S.P., Lusch, R.F.: A preliminary test of Hunt's General Theory of Competition: using artificial adaptive agents to study complex and ill-defined environments. Journal of Business Research 58, 1155–1168 (2005)
9. Wilson, S.W.: Classifier fitness based on accuracy. Evolutionary Computation 3, 149–175 (1995)
10. Zhang, T., Zhang, D.: Agent-based simulation of consumer purchase decision-making and the decoy effect. Journal of Business Research 60, 912–922 (2007)

Agent-Based Simulation Analysis for Equilibrium Selection and Coordination Failure in Coordination Games Characterized by the Minimum Strategy

Ichiro Nishizaki, Tomohiro Hayashida, Hideki Katagiri, and Noriyuki Hara

Department of Artificial Complex Systems Engineering,
Graduate School of Engineering of Hiroshima University,
1-4-1, Kagamiyama, Higashi-Hiroshima, Hiroshima, 739-8527, Japan
Tel.: +81-82-424-7604; Fax: +81-82-422-7195
{nisizaki,hayashida,katagiri-h,nhara}@hiroshima-u.ac.jp

Abstract. In this paper, to analyze equilibrium selection and coordination failure in coordination games, we develop an agent-based simulation system in which artificial adaptive agents have a mechanism of decision making and learning based on neural networks and genetic algorithms. Using the simulation system, we examine the strategy choices of agents and formation of equilibria in the steady state, and compare the experimental result given by VanHuyck *et al.* (1990) with our simulation result.

Keywords: coordination games, agent-based simulation, adaptive agents.

1 Introduction

In this paper, we deal with coordination games with multiple equilibria which are strictly Pareto ranked. Although we can think about persuasive criteria for equilibrium selection such as payoff dominance, risk dominance and security, it is difficult to predict which equilibrium will be realized because of uncertainty about actions of opponents. Because it is difficult to deductively predict actions of players for a given game, criteria for equilibrium selection have been evaluated by experimental researches [4,5,10,13,14,19,20,17].

Van Huyck *et al.* [19,20] carry out experiments on coordination games with multiple equilibria which are characterized by the minimum strategy and the mean of strategies, and they provide experimental facts about behavior of human subjects in the games. The result from the experiment on the coordination game characterized by the minimum strategy can be summarized as follows. (i) Inductive selection principle seems to be used for equilibrium selection. (ii) Recognizing actions selected by the others brings forward the time needed for converging to an equilibrium. (iii) Actions of subjects converge at a secure equilibrium with lower payoffs as the number of players becomes larger. From the experiment on the coordination game characterized by the mean of strategies, the following conclusion is derived. The result of the first game lays down succeeding

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 629–638, 2009.

behavior of the players; the mean of strategies on and after the second game can be predictable.

In most of standard mathematical models in economics and game theory, it is assumed that players are rational and maximize their payoffs, and therefore they can discriminate between two payoffs with a minute difference. Such optimization approaches are not always appropriate for analyzing human behavior and social phenomena, and models based on adaptive behavior can be alternatives to such optimization models. Recently as complements of conventional mathematical models, a large number of adaptive behavioral models have been proposed [2,3,6,7,8,9,12,15,16,18,21].

As mentioned above, numerous experiments have been accumulated to examine human behavior in coordination games. Although in experimental studies, situations in accordance with mathematical models are formed in laboratories and human subjects are motivated by money, in such experiments with human subjects there exist limitations with respect to the number of trials, the number of subjects, variations of parameter settings and so forth.

It is natural and relatively easy to model actions of human subjects in simulation systems with adaptive artificial agents. Simulations can be a promising approach to modeling situations where it is difficult to assume hyper-rational behavior of decision makers. We suppose that simulation is a complement to experiments with human subjects because an extensive range of treatments can be easily performed by varying values of the parameters characterizing games in simulation systems while there exist the above mentioned limitations in experiments with human subjects. As concerns such approaches based on adaptive behavioral models, Holland and Miller [11] interpret most of economic systems as complex adaptive systems, and point out that simulation using artificial societies with adaptive agents is effective for analysis of such economic systems. Axelrod [1] insists on the need for simulation analysis in social sciences, and states that purposes of the simulation analysis include prediction, performance, training, entertainment, education, proof and discovery.

In this paper, we focus on the coordination games characterized by the minimum strategy used in the experimental research by van Huyck *et al.* [19]. We develop a simulation system with artificial adaptive agents for analyzing behavior of agents in the coordination games, and analyze data from the simulations with several settings. The decision mechanism of the agents in our simulation system is implemented by a neural network. The synaptic weights and thresholds characterizing the neural network are revised so as to obtain larger payoffs through a genetic algorithm.

Using the simulation system with adaptive artificial agents, we compare the result of the experiment by van Huyck *et al.* [19] with that of the simulations, and examine behavior of agents in the coordination games. Varying the degree of security of the game, information given to players, and the number of players as parameters, we arrange three settings of the simulations. In section 2, we briefly review the experimental results of the coordination games. Section 3 is devoted to describing the agent-based simulation system with decision and learning mechanisms based on neural networks and genetic algorithms. In section 4, we examine the results of the simulations, and compare the experimental data with them. Finally in section 5, we give a summary of the simulations and some concluding remarks.

2 Minimum Strategy Coordination Game

In this section, we summarize the results of the experimental investigation by van Huyck *et al.* [19]. Because the coordination games treated in the experiment are characterized by the minimum values of the strategies selected by players, we refer to this game as the minimum strategy coordination game.

Let the set of players be $N = \{1,\ldots,n\}$. All the players have the common set of strategies: $S = \{1,\ldots,\bar{s}\}$. Let $x_i \in S$ denote a strategy of player i. Then, the payoff function of player i is represented by

$$\pi(x_i,\underline{x}_i) = a\min(x_i,\underline{x}_i) - bx_i + c,\ \underline{x}_i = \min(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n),\ a > b > 0,\ c > 0. \tag{1}$$

The payoff of player i decreases with a strategy x_i of self, and increases with the minimum $\underline{x}_i$ among strategies of the others. To guarantee positive payoffs, the constant c is added. A best response to $\underline{x}_i$ is the same as the strategy $\underline{x}_i$. If all the player choose the same strategy $x \in S$, the strategy profile $(x,\ldots,x)$ is a Nash equilibrium; each of the strategies $x \in S$ can potentially compose a Nash equilibrium.

When a player fails to predict the minimum strategy $\underline{x}_i$, it follows that the player chooses another strategy $x_i \neq \underline{x}_i$, and an outcome of the game results in disequilibrium. When multiple equilibria can be Pareto ranked, the Pareto dominant equilibrium may not always be chosen even if players can respond optimally, and then players fail to coordinate their strategies. It is supposed that an occurrence of disequilibrium or coordination failure is the most crucial issue in the coordination games.

In the minimum strategy coordination game, the strategy profile $(\bar{s},\ldots,\bar{s})$, in which all the players choose the maximum strategy $\bar{s}$ and they obtain the maximal payoff $\bar{s}(a-b)+c$, is the payoff dominant equilibrium. Then, the maximum strategy $\bar{s}$ is called the payoff dominant strategy. However, as the number of players n grows large, an incentive to deviate the payoff dominant strategy becomes also large because of strategic uncertainty.

The concept of security contrasts with the efficiency of the payoff dominance in the minimum strategy coordination game, and the conflict between them should be considered. The maximin strategy guarantees a certain constant payoff, and it leads to a Nash equilibrium. However, this Nash equilibrium is not efficient obviously. A player who chooses the secure strategy 1 obtains the payoff $a-b+c$, no matter which strategy the other players select; therefore the strategy profile $(1,\ldots,1)$ is called the secure equilibrium.

Van Huyck *et al.* [19] deal with the minimum strategy coordination games, and carry out an experimental investigation about conflict between the payoff dominance and the security. They provide two games: game A with $a = 0.2$, $b = 0.1$ and $c = 0.6$, and game B with $a = 0.1$, $b = 0.0$ and $c = 0.6$. Game B has no risk of choosing large strategies because the coefficient of the second term which means a penalty for choosing larger strategies is zero. In treatment A, game A is repeatedly played and the number of player is 14 to 16. In treatment B, game B is repeatedly played and the number of player is also 14 to 16. In treatment A', after playing game B, game A is played. Treatment C is the same as treatment A except for the number of players; the number of players in this treatment is two. These treatments are concluded in the following.

Treatment A**:** In the first round, 2 % of players choose the secure strategy, and 31 % choose the payoff dominant strategy; in the final 10th round, the percentage of choosing the secure strategy increases to 72 % and that of the payoff dominant strategy decreases to 7 %. Namely, as rounds go on, the number of players choosing the secure strategy increase. In the case where the distribution of chosen strategies is informed all the players, players begin to choose the secure strategy earlier than in the case where only the minimum strategy is informed.

Treatment B**:** In the first round, 5 % of players choose the secure strategy, and 84 % choose the payoff dominant strategy; in the final 5th round, the percentage of choosing the secure strategy decreases to 2 % and that of the payoff dominant strategy increases to 96 %. Namely, as rounds go on, the number of players choosing the payoff dominant strategy increase.

Treatment A'**:** After playing game B, in the first round of game A, 29 % of players choose the secure strategy, and 36 % choose the payoff dominant strategy; in the final 5th round, the percentage of choosing the secure strategy increases to 84 % and that of the payoff dominant strategy decreases to 5 %. Namely, although the influence of experience of coordination to the payoff dominant strategy can be observed in the early rounds, as rounds go on, such influence has disappeared.

Treatment C**:** In trials where pairs are fixed for all the rounds, in the first round, 29 % of players choose the secure strategy, and 36 % choose the payoff dominant strategy; in the final 7th round, the percentage of choosing the secure strategy decreases to 7 % and that of the payoff dominant strategy increases to 89 %. In trials where pairs are randomly matched for all the rounds, in the first round, 11 % of players choose the secure strategy, and 39 % choose the payoff dominant strategy; in the final 5th round, the percentage of choosing the secure strategy decreases to 6 % and that of the payoff dominant strategy increases to 50 %. In both cases, as the number of players decreases, players become apt to choose the payoff dominant strategy.

The above mentioned results are summarized as follows. As subjects acquire experiences, (i) in case with substantial risk, the secure strategy outnumbers; (ii) in case without risk, the payoff dominant strategy rises to predominance; and (iii) as the number of players decreases, the payoff dominant equilibrium is more likely to realize.

3 Simulation Model

In this paper, we develop a simulation system with adaptive artificial agents repeatedly playing the minimum strategy coordination games, and we analyze data from the simulations in which a variety of runs are performed by varying values of some parameters characterizing the game. An artificial adaptive agent in our agent-based simulation system has a mechanism of decision making and learning based on neural networks and genetic algorithms.

3.1 Decision Making by a Neural Network

Agents repeatedly play the minimum strategy coordination game; agents obtaining larger payoff are likely to reproduce in the next period, and conversely agents obtaining only a little payoff are likely to be weeded out. In our artificial genetic system, the whole population consists of m game groups, and in each game group the minimum strategy coordination game is played by n agents.

An agent corresponds to a neural network which is characterized by synaptic weights between two nodes in the neural network and thresholds which are parameters in the output function of nodes. Because a structure of neural networks is determined by the number of layers and the number of nodes in each layer, an agent is prescribed by the fixed number of parameters if these numbers are fixed. In our model, we form a string compound of these parameters which is identified with an agent, and the string is treated as a chromosome in an artificial genetic system.

A strategy of an agent is determined by the outputs of the agent's neural network. Namely, the outputs out_s, $s = 1,\dots,\bar{s}(= 7)$ of the neural network correspond to from the strategy 1 to the strategy 7, and the strategy s^* with the largest output out_{s^*} is chosen as the next strategy of the agent.

Inputs of the neural network are the following six values, and the additional two sets of inputs, input sets A1 and A2, are used in the simulation with respect to information given to players. Each of inputs is normalized in $[0,1]$. In the following inputs, the subscript i, $i = 1,\dots,n$ means the player i and the subscript j, $j = 1,\dots,m$ means the game group j. Thus, the subscript ij identifies a particular agent in the artificial genetic system.

[Input 1, x_{ij}] the strategy chosen by agent ij at the last period.

[Input 2, y_j] the minimum among the strategies chosen by all the agents in game group j at the last period.

[Input 3, π_{ij}] the payoff obtained by agent ij in the last period.

[Input 4, x_{ij}^T] the weighted most frequent strategy in the last T periods: $x_{ij}^T = \arg_{s\in\{1,\dots,7\}} \max \sum_{u=t-T+1}^{t} e_s^u w^{u-t}$, where $e_s^t = 1$ if agent ij chooses the strategy s at period t, otherwise $e_s^t = 0$, and w is the discount factor.

[Input 5, y_j^T] the weighted most frequent minimum strategy in the last T periods: $y_j^T = \arg_{s\in\{1,\dots,7\}} \max \sum_{u=t-T+1}^{t} f_s^u w^{u-t}$, where $f_s^t = 1$ if the strategy s is the minimum strategy in game group j at period t, otherwise $f_s^t = 0$.

[Input 6, π_{ij}^T] the weighted sum of obtained payoffs in the last T periods: $\pi_{ij}^T = \sum_{u=t-T+1}^{t} \pi_{ij}^u w^{u-t}$, where π_{ij}^u is the payoff of agent ij at period u.

[Input A1, (z_j, z_j^T)] for the treatments where agents are informed not only the minimum strategy but also the maximum strategy in game group j. Let z_j be the maximum strategy in game group j at the last period, and z_j^T be the most frequent maximum strategy in the last T periods where $z_j^T = \arg_{s\in\{1,\dots,7\}} \max \sum_{u=t-T+1}^{t} g_s^u w^{u-t}$, where $g_s^t = 1$ if the strategy s is the maximum strategy in game group j at period t, otherwise $g_s^t = 0$.

[Input A2, $(\mathbf{z}_j, \mathbf{z}_j^T)$] for the treatment where agents are informed the distribution of strategies chosen by all the agents in game group j. Let $\mathbf{z}_j$ be the distribution of strategies in game group j, where z_{js} is the number of agents choosing the strategy s at the last period where $\mathbf{z}_j = (z_{j1},\dots,z_{j7})$. Let $\mathbf{z}_j^T$ be the weighted distribution of chosen strategies in the last T periods where $\mathbf{z}_j^T = \sum_{u=t-T+1}^{t} \mathbf{z}_j^u w^{u-t}$, where $\mathbf{z}_j^u$ is the distribution of strategies in game group j at period u.

Concerning inputs 1 and 2, because human subjects in the experiment are informed the minimum strategy at the last game, and it is supposed that they remember the strategies selected by themselves, the strategy x_{ij} of agent ij and the minimum strategy y_j in game group j are given as inputs of the neural network; the payoff π_{ij} obtained by agent ij at the last period is also given as input 3. Supposing that a player does not remember an exact history of strategies in the past periods, but the player remembers at least the most frequent strategy in the past periods, we provide the weighted most frequent strategy x^T_{ij} in the last T periods as input 4 to the neural network. In the definition of x^T_{ij}, assuming that old memory is apt to decay, the discount factor w, $0 < w < 1$ is introduced. Similarly, as inputs 5 and 6, the weighted most frequent minimum strategy y^T_j and the weighted sum of obtained payoffs in the last T periods are also given.

Under the condition that agents are informed of only the minimum strategy as an outcome of the game, the set of inputs from 1 to 6 is given to the neural network. To examine the effect of contents of information, we arrange two other settings with more information; one is a case where the maximum strategy is informed as an aggregation information, and the other is a case where a distribution of strategies chosen by all the agents is informed. In the former, the maximum strategy at the last period and the weighted most frequent maximum strategy in the past periods are given as input A1; in the latter, a distribution of strategies at the last period and a weighted distribution of strategies in the past periods are given as input A2.

In the standard configuration of inputs, as we mentioned above, the neural network has the six units in the input layer and the seven units ($\bar{s} = 7$) in the output layer. Let m be the number of units in the hidden layer. Because the number of synaptic weights is $13m$ and the number of units in the hidden and the output layers is $m+7$, the neural network can be determined by the synaptic weights w_l, $l = 1,\ldots,13m$ and the thresholds θ_l, $l = 1,\ldots,m+7$. These parameters and the input values determine an action of the agent, and the synaptic weights and the thresholds are adjusted through the genetic algorithm so that the initial population evolves into the population of agents which can obtain larger payoffs.

3.2 Evolutionary Learning through the Genetic Algorithm

In our artificial genetic system for simulations, the minimum strategy coordination game is played by n agents in each of m game groups. Therefore, there are m agents for each type of players. There are $\bar{s}$ alternative strategies, and each of the agents chooses one among them. Then agents obtain payoffs defined by the payoff function (1). Repeatedly playing the game, agents obtaining larger payoffs are likely to survive; if this is not the case, such agents are easily culled out.

We start by describing how the parameters prescribing an agent are initialized. In the experiment conducted by van Huyck *et al.* [19], subjects understand the payoff table defined by the payoff function (1), and it is not true that they start to play the game without any prior knowledge of the game. Therefore it is natural for artificial agents in our system to have some knowledge of the game before playing it. To do so, by using the error back propagation algorithm with the teacher signals which are shown below, we adjust the parameters of the synaptic weights and the thresholds in the neural network.

Assume that in the last T periods agent ij continues to select a certain strategy s_{ij} and all of the other $n-1$ agents also continue to select the same strategy s'_{ij}. Then the set of inputs is given as $x_{ij} = x^T_{ij} = s_{ij}$ for inputs 1 and 4, $y_j = y^T_j = \min\{s_{ij}, s'_{ij}\}$ for inputs 2 and 5, and $\pi_i = \pi_i^T = a\min\{s_{ij}, s'_{ij}\} - bs_{ij} + c$ for inputs 3 and 6. The number of combinations of the inputs is $\bar{s}^2$.

The corresponding outputs out_s, $s = 1,\ldots,\bar{s}$ of the neural network are given as follows. On the assumption that the mixed strategy $(1/\bar{s},\ldots,1/\bar{s})$ which is the equiprobability distribution is selected by the opponent agents, the expected payoff of agent ij is

$$E(s) = \sum_{s'<s} \frac{1}{\bar{s}}(as' - bs + c) + \sum_{s' \geq s} \frac{1}{\bar{s}}(as - bs + c),\ s = 1,\ldots,\bar{s}. \quad (2)$$

The output corresponding the minimum strategy is set to 1.5 times the expected payoff; the outputs corresponding strategies around the minimum strategy to 1.2 times the expected payoff; and the other outputs to the expected payoff. Namely, by considering the expected payoff and the minimum strategy, the output out_s is given by

$$out_s = \begin{cases} 1.5E(s) & \text{if } s = \min\{s_{ij}, s'_{ij}\} \\ 1.2E(s) & \text{if } s = \min\{s_{ij}, s'_{ij}\} \pm 1 \\ E(s) & \text{otherwise.} \end{cases} \quad (3)$$

After initializing the parameters of the neural networks, the artificial agents repeatedly play the game, and accumulate the payoffs obtained in the iterated games.

The whole population is divided into m groups for playing the game. After playing the game, a subpopulation is formed by picking one agent from each group for the game; the ith subpopulation consisting of m agents corresponds to player i. The genetic operations are executed to each subpopulation separately. A fitness of each agent is a normalized payoff obtained by itself, $f(\pi_{ij}) = (\pi_{ij} - \min)/(\max - \min)$, where max and min are, respectively, the maximum and the minimum payoffs in the subpopulation.

4 Results of the Simulations

4.1 The Details of the Simulations

First, we provide the result of the simulation for examining sensitivity of the parameters of the payoff function. Second, we investigate the effect of information given to players. Finally, we examine the influence of the number of players on behavior of players. Simulation *Coefficients*, Simulation *Information*, and Simulation *Size* are arranged.

4.2 Simulation *Coefficients*

The payoff function (1) is characterized by the gain coefficient a with respect to the minimum strategy and the penalty coefficient b with respect to the strategy selected by self. The payoff is an increasing function with the minimum strategy and is a decreasing function with the strategy selected by self. Simulation *Coefficients* is composed of three treatments: treatment b is carried out by varying the value of b, fixing the value of a; in treatment a, the value of a is varied, fixing the value of b; finally in treatment *experience*, after putting agents in experiencing the payoff dominant equilibrium, fixing the value of a, a treatment similar to treatment b is performed.

Treatment b**.** In general as the value of b is made larger and the risk of paying the penalty increases, the payoff of an agent selecting a large strategy such as the payoff dominant strategy 7 becomes a small value, and therefore it is likely to fail in coordination. However, the risk-free game with $b = 0$ is not the case. In this treatment, fixing the value of a at $a = 0.2$, the value of the penalty coefficient b is varied; it is set at $b = 0.0, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1$. From the data observed in the treatment, we investigate transitions and steady states of the choice rate of each strategy, the realization rate of each individual equilibrium, and so forth.

The result of treatment b can be summarized as follows: 1) In the games without the risk of paying any penalty, the artificial agents successfully coordinate their strategies and the payoff dominant equilibrium is realized. 2) In the games with the risk of paying a substantial penalty, coordination among the artificial agents is failed, but they suitably predict strategies of the opponents and the secure equilibrium forms. 3) The games with the risk of paying the intermediate penalty are likely to bring outcomes of disequilibria. 4)As the value of b decreases, artificial agents shift choices of strategies stepwise from the secure strategy 1 to the payoff dominant strategy 7. 5) While the payoff dominant equilibrium is sensitive to increase of the value of b, the secure equilibrium is not so sensitive to decrease of the value of b.

Treatment a**.** While in treatment b the value of the penalty coefficient b is varied fixing the value of a at 0.2, in this treatment conversely fixing the value of b at 0.0 and 0.1, the value of the gain coefficient a is changed; the value of a is set at $a = 0.1, 0.2, 0.5, 1.0, 1.5, 2.0$. Especially, for the case of $b = 0.1$, we additionally vary the value of a from $a = 2.0$ to $a = 5.0$ at intervals of 0.5. We examine the data of the treatment as in treatment b.

The result of treatment a can be summarized as follows: 1) When $b = 0$, the artificial agents successfully coordinate their strategies, and as a result the payoff dominant equilibrium is realized. 2) In the case of $b = 0.1$, as the value of a increases, most of the artificial agents shift choices of strategies from the secure strategy 1 to the payoff dominant strategy 7 by way of the strategies 3 and 4. 3) When $a \geq 4.0$ where the effect of the risk of paying the penalty is relatively small, the artificial agents successfully coordinate their strategies and the payoff dominant equilibrium occurs at the rate of about 0.9. 4) When $a = 0.2$ where the effect of the risk of paying the penalty is relatively large, coordination of the artificial agents is failed but they successfully predict opponents' strategies; consequently the secure equilibrium is realized at the rate of about 0.9. 5) The games with $0.5 \leq a \leq 3$ where the risk of paying the penalty is intermediate are likely to bring outcomes of disequilibria. 6) The behavior of the artificial agents roughly depends on the ratio of the penalty coefficient b to the gain coefficient a; however, around $a/b = 10$, the influence of the penalty coefficient b seems to be larger.

5 Conclusions

We have focused on the coordination games characterized by the minimum strategy used in the experimental investigation by van Huyck *et al.* [19], and developed an simulation system, where artificial adaptive agents have a mechanism of decision making and learning based on neural networks and genetic algorithms, for the extensive range of analysis on behavior of artificial agents in the coordination games in order to complement the experiments with human subjects.

We summarize the findings from the simulation analysis as follows. (i) In treatment b of Simulation *Coefficients*, the result of the simulation supports that of the experiment with human subjects on the whole. As new insights from this simulation analysis, it is found that while the payoff dominant equilibrium is sensitive to increase of the value of the penalty coefficient b, the secure equilibrium is not so sensitive to decrease of the value of b. Moreover, in the games without the penalty or the games with a substantial penalty, the equilibria are likely to be realized; the games with the risk of paying an intermediate penalty are likely to bring outcomes of disequilibria. (ii) From treatment a of Simulation *Coefficients*, the behavior of the artificial agents roughly depends on the ratio of the penalty coefficient b to the gain coefficient a. (iii) If the penalty coefficient b is not so large, the experience of the payoff dominant equilibrium prompts the artificial agents to select larger strategies. (iv) The data from Simulation *Information* does not confirm the finding of the experiment with human subjects that disclosure of the distribution of strategies accelerates the convergence to selecting the secure strategy 1, but it is found that there exist some different effects of the additional information on the behavior of agents. (v) In Simulation *Size*, we find out that the equilibria are likely to be realize as the number of players decreases. The fact from the simulation supports the result of the experiment with human subjects.

References

1. Axelrod, R.: Advancing the art of simulation in the social sciences. In: Conte, R., Hegselmann, R., Terna, R. (eds.) Simulating Social Phenomena, pp. 21–40. Springer, Heidelberg (1997)
2. Andreoni, J., Miller, J.H.: Auctions with artificial adaptive agents. Games and Economic Behavior 10, 39–64 (1995)
3. Banerje, B., Sen, S.: Selecting parters. In: Parsons, S., Gmytrasiewicz, P., Wooldridge, M. (eds.) Game Theory and Decision Theory in Agent-Based Systems, pp. 27–42. Kluwer Academic Publishers, Boston (2002)
4. Cooper, R.W., DeJoong, D.V., Forsythe, R., Ross, T.W.: Communication in the battle of the sexes game: some experimental results. RAND Journal of Economics 20, 568–587 (1989)
5. Cooper, R.W., DeJoong, D.V., Forsythe, R., Ross, T.W.: Selection criteria in coordination games: some experimental results. American Economic Review 80, 218–233 (1990)
6. Dorsey, R.E., Johnson, J.D., Van Boening, M.V.: The use of artificial neural networks for estimation of decision surfaces in first price sealed bid auctions. In: Cooper, W.W., Whinston, A.B. (eds.) New Directions in Computational Economics, pp. 19–40. Kluwer, Dordrecht (1994)
7. Erev, I., Rapoport, A.: "Coordination, "magic," and reinforcement learning in a market entry game. Games and Economic Behavior 23, 146–175 (1998)
8. Erev, I., Roth, A.E.: Predicting how people play games: reinforcement learning in experimental games with unique, mixed strategy equilibria. The American Economic Review 88, 848–881 (1998)
9. Fundenberg, D., Levine, D.K.: The Theory of Learning in Games. MIT Press, Cambridge (1998)

10. Goeree, J.K., Holt, C.A.: An experimental study of costly coordination. Games and Economic Behavior 51, 349–364 (2005)
11. Holland, J.H., Miller, J.H.: Adaptive intelligent agents in economic theory. American Economic Review 81, 365–370 (1991)
12. Leshno, M., Moller, D., Ein-Dor, P.: Neural nets in a group decision process. International Journal of Game Theory 31, 447–467 (2002)
13. Ochs, J.: The coordination problem in decentralized markets: an experiment. The Quarterly Journal of Economics 105, 545–559 (1990)
14. Ochs, J.: Coordination problems. In: Kagel, J.H., Roth, A.E. (eds.) Handbook of Experimental Economics, ch. 3, pp. 195–251. Priceton University Press, Priceton (1995)
15. Rapoport, A., Seale, D.A., Winter, E.: Coordination and learning behavior in large groups with asymmetric players. Games and Economic Behavior 39, 111–136 (2002)
16. Roth, A.E., Erev, I.: Learning in extensive form games: experimental data and simple dynamic models in the intermediate term. Games and Economic Behavior 8, 163–212 (1995)
17. Straub, P.G.: Risk dominance and coordination failures in static games. The Quarterly Review of Economics and Finance 35, 339–363 (1995)
18. Sundali, J.A., Rapoport, A., Seale, D.A.: Coordination in market entry games with symmetric players. Organizational Behavior and Human Decision Processes 64, 203–218 (1995)
19. Van Huyck, J.B., Battalio, R.C., Beil, R.O.: Tacit coordination games, strategic uncertainty, and coordination failure. American Economic Review 80, 234–248 (1990)
20. Van Huyck, J.B., Battalio, R.C., Beil, R.O.: Strategic uncertainty, equilibrium selection, and coordination failure in average opinion games. The Quarterly Journal of Economics 106, 885–910 (1991)
21. Young, H.P.: Individual Strategy and Social Structure. Princeton University Press, Princeton (1998)

A Computational Method for Obtaining Stackelberg Solutions to Noncooperative Two-Level Programming Problems through Evolutionary Multi-Agent Systems

Kosuke Kato, Masatoshi Sakawa, Takeshi Matsui, and Hidenori Ohtsuka

Graduate School of Engineering, Hiroshima University,
1-4-1, Kagamiyma, Higashi-Hiroshima, 739-8527, Japan
{kosuke-kato, sakawa, tak-matsui}@hiroshima-u.ac.jp
http://www.sol.hiroshima-u.ac.jp/

Abstract. In management or public decision making, there often exist two decision makers (DMs) in the situation where one of them has the priority in decision over another. Such decision making situations are often formulated as two-level programming problems. Under the assumption that these DMs know the objective function and constraints for the other DM and do not have motivation to cooperate mutually, the Stackelberg solution is adopted as a reasonable solution. However, for even two-level linear programming problems as the simplest case, the problem solved to obtain Stackelberg solutions is a nonconvex programming problem with complex structures and is known as an NP-hard problem. In this paper, we propose an efficient approximate solution method for two-level programming problems based on an evolutionary multi-agent system.

Keywords: Two-level nonlinear programming, Stackelberg solution, evolutionary multi-agent system.

1 Introduction

In management or public decision making, there often exist two decision makers (DMs) in the situation where one of them has the priority in decision over another, e.g., the manager side DM and the factory side DM or the central government side DM and the local government side DM. Such decision making situations are often formulated as two-level linear programming problems with a DM at the upper level and another DM at the lower level, where each of DMs has his own decision variables and objective function, and the DM at the upper level has the priority in decision over the DM at the lower level. Various approaches for two-level programming problems could exist according to situations which the DMs are placed in. Under the assumption that these DMs do not have motivation to cooperate mutually, the Stackelberg solution [7] is adopted as a reasonable solution. However, for even two-level linear programming

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 639–648, 2009.

problems as the simplest case, the problem solved to obtain Stackelberg solutions is a nonconvex programming problem with complex structures and is known as an NP-hard problem. Thereby, an efficient computational method is needed to obtain (approximate) Stackelberg solutions for two-level programming problems in the real world.

As a solution method for noncooperative two-level linear programming problems, there were proposed Kth Best method by Bialas et al. [1] and Branch-and-Bound method by Hansen et al. [2], but their processing time may exponentially increases at worst as the size of problem increases since they are strict solution methods based on enumeration. As for solution methods for noncooperative two-level nonlinear programming problems, an approximate solution method based on particle swarm optimization (PSO) was proposed by Niwa et al. [6], but it seems time-consuming.

On the other hand, Socha et al. [8] proposed a fast computational method through an evolutionary multi-agent system (EMAS) for obtaining (approximate) Pareto optimal solution sets for multiobjective programming problems.

In this paper, we propose an efficient computational method for obtaining (approximate) Stackelberg solutions to two-level programming problems based on EMAS.

2 Two-Level Programming Problems and Its Solution Concept

In this paper, we focus on two-level programming problems formulated as:

$$\left.\begin{array}{l} \displaystyle\underset{\boldsymbol{x}_1}{\text{minimize}}\ f_1(\boldsymbol{x}_1,\boldsymbol{x}_2) \\ \text{where } \boldsymbol{x}_2 \text{ solves} \\ \displaystyle\underset{\boldsymbol{x}_2}{\text{minimize}}\ f_2(\boldsymbol{x}_1,\boldsymbol{x}_2) \\ \text{subject to } g_i(\boldsymbol{x}_1,\boldsymbol{x}_2) \le 0,\ \ i=1,2,\ldots,m \end{array}\right\} \quad (1)$$

where $\boldsymbol{x}_1$ is an n_1 dimensional decision variable column vector for the DM at the upper level (DM1), $\boldsymbol{x}_2$ is an n_2 dimensional decision variable column vector for the DM at the lower level (DM2), $f_1(\boldsymbol{x}_1,\boldsymbol{x}_2)$ is the objective function for DM1, $f_2(\boldsymbol{x}_1,\boldsymbol{x}_2)$ is the objective function for DM2 and $g_i(\boldsymbol{x}_1,\boldsymbol{x}_2)$, $i=1,2,\ldots,m$ are constraint functions. In general, $f_l(\cdot)$, $l=1,2$ and $g_i(\cdot)$, $i=1,2,\ldots,m$ are nonlinear. In (1), if the DM at the upper level (DM1) adopts a decision $\hat{\boldsymbol{x}}_1$, the DM at the lower level (DM2) is supposed to select a decision to minimize $f_2(\cdot)$ in the feasible region of (1) under the DM1's decision, $\hat{\boldsymbol{x}}_2(\hat{\boldsymbol{x}}_1)$, called a rational reaction. Then, the optimal solution (Stackelberg solution) to (1) is the point $(\boldsymbol{x}_1^*,\boldsymbol{x}_2^*(\boldsymbol{x}_1^*))$ which minimizes $f_1(\cdot)$ in the inducible region (IR) which is the set of points $(\hat{\boldsymbol{x}}_1,\hat{\boldsymbol{x}}_2(\hat{\boldsymbol{x}}_1))$ for all possible decisions $\hat{\boldsymbol{x}}_1$. Figure 1 illustrates an example of a Stackelberg solution for a two-level linear programming problem.

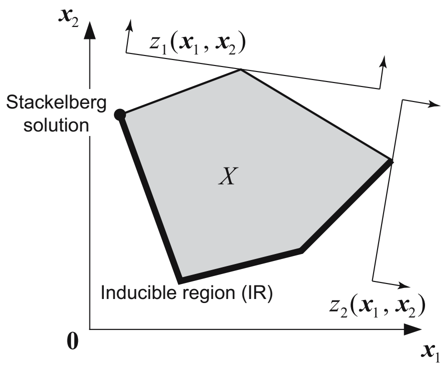


Fig. 1. An example of Stackelberg solution

3 Framework of EMAS for Two-Level Programming Problems

In this section, we outline the framework of a computational method through EMAS for obtaining Stackelberg solutions to two-level programming problems.

In the method based on EMAS, we use N agents $\boldsymbol{a}_r$, $r = 1, 2, \ldots, N$. Each of agents has the current position $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$, the upper level objective function value $f_1(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$, the lower level objective function value $f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$, the upper level energy e_r^U and the lower level energy e_r^L as attributes, and undergoes operations like energy exchange, reproduction and move.

First, carry out the search to the direction improving the lower level objective function to move each agent toward IR. To be more specific, in the upper level decision variable space, if there exists at least one agent in the neighborhood of each agent $\boldsymbol{a}_r$, select one of agents, $\boldsymbol{a}_{r'}$, as a communicating opponent and compare the lower level objective function value of $\boldsymbol{a}_r$ with that of $\boldsymbol{a}_{r'}$. Then, the superior agent gains the lower level energy from the inferior one. The inferior agent is killed if the lower level energy of it becomes empty. Otherwise, it is moved according to some rule. If there exists no agent in the neighborhood of $\boldsymbol{a}_r$, move $\boldsymbol{a}_r$ to the position where the lower level objective function value becomes better by changing $\boldsymbol{x}_{r2}$.

Next, carry out the search to the direction improving the upper level objective function to move each agent near IR toward a Stackelberg solution. For each agent $\boldsymbol{a}_r$, select an agent, $\boldsymbol{a}_{r''}$, which is the nearest agent around IR as a communicating opponent and compare the upper level objective function value of $\boldsymbol{a}_r$ with that of $\boldsymbol{a}_{r''}$. Then, the superior agent gains the upper level energy from the inferior one. The inferior agent is not killed even if the lower level energy of it becomes empty. It gains the same amount of the upper level energy as the initial value. This supplement is done to maintain the number of agents with nonzero upper level energy. After the exchange of the upper level energy between them, if the upper level energy of the inferior agent is sufficiently large, generate a new agent in the direction from the inferior one to the superior one which seems promising.

By repeating these procedures, agents gather around IR and the upper level energy of an agent near IR which is good in the sense of the upper level objective function value becomes large. Thereby, the current position of the best agent in the sense of the upper level objective function value among ones with large upper level energy can be regarded as the (approximate) Stackelberg solution.

The procedure is summarized as follows.

Step 1: Generate N agents $\boldsymbol{a}_r$, $r = 1, 2, \ldots, N$ at random.
Step 2: Let $T := 1$.
Step 3: Let $r := 1$.
Step 4: For the rth agent $\boldsymbol{a}_r$, carry out the search to the direction improving the lower level objective function value in order to move the current position of the agent toward IR.
Step 5: If the lower level energy of the rth agent $\boldsymbol{a}_r$ is greater than a threshold, i.e., the current position of the agent is regarded in IR, carry out the search to the direction improving the upper level objective function value in order to move the agent toward the Stackelberg solution.
Step 6: If $r = N$, go to step 7. Otherwise, let $r := r + 1$ and go to step 4.
Step 7: If $T = T_{\max}$, terminate the procedure and the current best solution is regarded as a Stackelberg solution. Otherwise, let $T := T + 1$ and go to step 3.

4 Consideration in Applying EMAS to Two-Level Nonlinear Programming Problems

In applying EMAS mentioned above to two-level nonlinear programming problems, there often occurs a problem that it is difficult to obtain initial agents which are feasible by generating them randomly in step 1. In addition, there is supposed to exist another problem that the judgment of the existence of an agent around IR by the amount of the lower level energy seems insufficient since the shape of IR for two-level nonlinear programming problems is expected to be complicated.

Thereby, in this paper, we adopt the homomorphous mapping used in [4]. By the use of the homomorphous mapping, we can easily generate feasible initial agents. In addition, we permit the reproduction in the infeasible region and the search of the infeasible region by infeasible agents in order to widen the search area.

On the other hand, we utilize the Kuhn-Tucker condition of problem (2) to obtain the rational reaction $\bar{\boldsymbol{x}}_2(\bar{\boldsymbol{x}}_1)$ corresponding to $\bar{\boldsymbol{x}}_1$ for the purpose of more accurately check whether an agent with the current position $(\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_2)$ exists around IR or not.

$$\left.\begin{array}{rl} \text{minimize} & f_2(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2) \\ \text{subject to} & g_i(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2) \leq 0, \ \ i = 1, 2, \ldots, m \end{array}\right\} \quad (2)$$

The Kuhn-Tucker condition of (2) is expressed as follows.

$$\nabla f_2(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2^*) = \mathbf{0}, \ \ \boldsymbol{\lambda}^* \geq \mathbf{0},$$
$$g_i(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2^*) \leq 0, \ \ \lambda_i^* g_i(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_2^*) = 0, \ \ i = 1, 2, \ldots, m$$

Since the Kuhn-Tucker condition is the necessary condition for the current position to be in IR, the check by the Kuhn-Tucker condition is expected to be more accurate and useful than that by the lower level energy amount. On the other hand, the best position $(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2)$ obtained through the search by EMAS satisfies Kuhn-Tucker condition but is not guaranteed to exist in IR. Thus, the following optimization problem with respect to the lower level objective function is solved in order to find a rational reaction for the upper level decision $\tilde{\boldsymbol{x}}_1$ of the best solution.

$$\left.\begin{array}{rl} \text{minimize} & f_2(\tilde{\boldsymbol{x}}_1, \boldsymbol{x}_2) \\ \text{subject to} & g_i(\tilde{\boldsymbol{x}}_1, \boldsymbol{x}_2) \leq 0, \ \ i = 1, \ldots, m \end{array}\right\} \tag{3}$$

Since (3) is a single-objective nonlinear programming problem, we use an approximate solution method based on particle swarm optimization (PSO) proposed by the authors [3,5] to check whether the (approximate) optimal solution $(\boldsymbol{x}_2^*)$ is equal to $\tilde{\boldsymbol{x}}_2$ or not. If $(\boldsymbol{x}_2^*) = \tilde{\boldsymbol{x}}_2$, we can regard $(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2)$ as the (approximate) Stackelberg solution since it exists in IR. Otherwise, repeatedly check whether the current position of an agent satisfying Kuhn-Tucker condition exists in IR or not in the same manner mentioned above in order of the quality of the upper level objective function value. If an agent whose current position exists in IR, the position is regarded as the (approximate) Stackelberg solution. By this procedure, the final solution is guaranteed to be in IR.

5 Detailed Procedure of the Proposed EMAS

In this section, we describe the computational procedure of the proposed EMAS for obtaining (approximate) Stackelberg solutions to noncooperative two-level programming problems.

In the method based on EMAS, we use N agents $\boldsymbol{a}_r$, $r = 1, 2, \ldots, N$. Each of agents has the current position $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$, the upper level objective function value $f_1(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$, the lower level objective function value $f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$, the energy e_r and the agent state variable k_r as attributes, and undergoes operations like energy exchange, reproduction and move. If Kuhn-Tucker condition is satisfied for the rth agent $\boldsymbol{a}_r$ whose current position is in the feasible region, let $k_r = 1$. If not, let $k_r = 0$. For agents in the infeasible region, let $k_r < 0$.

The procedure is summarized as follows.

Step 1: Generate N agents a_r, $r = 1, 2, \ldots, N$ by using the homomorphous mapping [4].
Step 2: Let $T := 1$.
Step 3: Let $r := 1$.

Step 4: If $k_r = 0$, go to step 5. If $k_r = 1$, go to step 6. If $k_r < 0$, go to step 8.

Step 5: Carry out the search to the direction improving the lower level objective function value. To be more specific, for the rth agent $\boldsymbol{a}_r$, choose an agent $\boldsymbol{a}_{r'}$ randomly. Then, compare $f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$ with $f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r'2})$. If $f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}) < f_2(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r'2})$, let $\boldsymbol{x}_{p2} := \boldsymbol{x}_{r2}$ and $\boldsymbol{x}_{p'2} := \boldsymbol{x}_{r'2}$. Otherwise, let $\boldsymbol{x}_{p2} := \boldsymbol{x}_{r'2}$ and $\boldsymbol{x}_{p'2} := \boldsymbol{x}_{r2}$. Then, update $\boldsymbol{x}_{p'2}$ by the following scheme:

$$\boldsymbol{x}_{p'2} := \boldsymbol{x}_{p'2} + 2R(\boldsymbol{x}_{p2} - \boldsymbol{x}_{p'2}) \tag{4}$$

where R is a uniform random number in $[0, 1]$. Repeat the comparison between $f_2(\boldsymbol{x}_{p1}, \boldsymbol{x}_{p2})$ and $f_2(\boldsymbol{x}_{p'1}, \boldsymbol{x}_{p'2})$ and the update of $\boldsymbol{x}_{p'2}$ n times. Let the final $\boldsymbol{x}_{p2}$ be $\boldsymbol{x}_{r2}$. If the current position of $\boldsymbol{a}_r$ satisfies Kuhn-Tucker condition, let $k_r := 1$ and go to step 6. Otherwise, go to step 9.

Step 6: If $k_r = 1$, carry out the search to the direction improving the upper level objective function value. To be more specific, choose an agent $\boldsymbol{a}_{r''}$ which satisfies Kuhn-Tucker condition at random as a comparing opponent and compare the upper level objective function value of $\boldsymbol{a}_r$ with that of $\boldsymbol{a}_{r''}$. Then, the superior agent gains the energy from the inferior one. Let the superior agent denote $\boldsymbol{a}_{r''}$ and the inferior one denote $\boldsymbol{a}_r$. If the inferior agent is killed by the disappearance of energy, go to step 7. Otherwise, go to step 9.

Step 7: Carry out the reproduction of the killed agent. Then, reproduce $\boldsymbol{a}_r$ with the current position which is determined as:

$$\begin{aligned}(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}) := &(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}) \\ &+2R((\boldsymbol{x}_{r''1}, \boldsymbol{x}_{r''2}) - (\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})).\end{aligned} \tag{5}$$

where R is a uniform random number in $[0, 1]$. If the current position of the reproduced $\boldsymbol{a}_r$ is infeasible, let $k_r := -1$ and go to step 8. Otherwise, let $k_r := 0$ and go to step 9.

Step 8: Carry out the search in the infeasible region. If $-t_1 \leq k_r < 0$, go to substep 8-1. If $-t_2 \leq k_r < -t_1$, go to substep 8-2. If $k_r = -t_2$, go to substep 8-3. Here, let $0 < t_1 < t_2$.

Substep 8-1: Let an agent with the current position $(x_{r,1}, x_{r,2}, \ldots, x_{r^*,i}, \ldots, x_{r,n_2})$ denote $\boldsymbol{a}_{r'}$. Here, r^* is randomly chosen from among $\{1, 2, \ldots, N\}$ and i is randomly chosen from among $\{1, 2, \ldots, n_1 + n_2\}$. Compare $f_1(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$ with $f_1(\boldsymbol{x}_{r'1}, \boldsymbol{x}_{r'2})$. Let the superior agent denote $\boldsymbol{a}_s$ and the inferior one denote $\boldsymbol{a}_{s'}$. Then, update $(\boldsymbol{x}_{s'1}, \boldsymbol{x}_{s'2})$ by the following scheme:

$$\begin{aligned}(\boldsymbol{x}_{s'1}, \boldsymbol{x}_{s'2}) = &(\boldsymbol{x}_{s'1}, \boldsymbol{x}_{s'2}) \\ &+2R((\boldsymbol{x}_{s1}, \boldsymbol{x}_{s2}) - (\boldsymbol{x}_{s'1}, \boldsymbol{x}_{s'2}))\end{aligned} \tag{6}$$

where R is a uniform random number in $[0, 1]$. Repeat the comparison between $f_1(\boldsymbol{x}_{s1}, \boldsymbol{x}_{s2})$ and $f_1(\boldsymbol{x}_{s'1}, \boldsymbol{x}_{s'2})$ and the update of $(\boldsymbol{x}_{s'1}, \boldsymbol{x}_{s'2})$ n' times. Let the final $(\boldsymbol{x}_{s1}, \boldsymbol{x}_{s2})$ be $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$. If $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$ is feasible, let $k_r := 0$. Otherwise, let $k_r := k_r - 1$. Go to step 9.

Substep 8-2: Choose an agent $a_{r'}$ randomly. Compare $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$ with $(\boldsymbol{x}_{r'1}, \boldsymbol{x}_{r'2})$ using the following function

$$h(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{i=1}^{m} g_i(\boldsymbol{x}_1, \boldsymbol{x}_2), \;\; g_i(\boldsymbol{x}_1, \boldsymbol{x}_2) > 0 \tag{7}$$

which is the degree of violation of the constraints. Let the superior position denote $(\boldsymbol{x}_{u1}, \boldsymbol{x}_{u2})$ and the inferior one denote $(\boldsymbol{x}_{u'1}, \boldsymbol{x}_{u'2})$. Then, update $(\boldsymbol{x}_{u'1}, \boldsymbol{x}_{u'2})$ by the following scheme:

$$\begin{aligned}(\boldsymbol{x}_{u'1}, \boldsymbol{x}_{u'2}) = (\boldsymbol{x}_{u'1}, \boldsymbol{x}_{u'2})\\ +2R((\boldsymbol{x}_{u1}, \boldsymbol{x}_{u2}) - (\boldsymbol{x}_{u'1}, \boldsymbol{x}_{u'2}))\end{aligned} \tag{8}$$

where R is a uniform random number in $[0, 1]$. Repeat the comparison between $h(\boldsymbol{x}_{u1}, \boldsymbol{x}_{u2})$ and $h(\boldsymbol{x}_{u'1}, \boldsymbol{x}_{u'2})$ and the update of $(\boldsymbol{x}_{u'1}, \boldsymbol{x}_{u'2})$ n'' times. Let the final $(\boldsymbol{x}_{u1}, \boldsymbol{x}_{u2})$ be $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$. If $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2})$ is feasible, let $k_r := 0$. Otherwise, let $k_r := k_r - 1$. Go to step 9.

Substep 8-3: Choose an agent $\boldsymbol{a}_{r'}$ whose current position is feasible randomly, and move $\boldsymbol{a}_r$ to the feasible region by the bisection method between $\boldsymbol{a}_{r'}$ and $\boldsymbol{a}_r$. Go to step 9.

Step 9: If $r = N$, go to step 10. Otherwise, let $r := r + 1$ and go to step 4.

Step 10: If $T = T_{\max}$, go to step 11. Otherwise, let $T := T + 1$ and go to step 3.

Step 11: Check whether the current position of the best agent exists in IR or not by solving (3). Since (3) is a single-objective nonlinear programming problem, we use the revised PSO method [3,5], which is one of most promising solution methods for nonlinear programming problems, to solve (3). If the current position of the best agent exists in IR, we can regard it as the (approximate) Stackelberg solution. Otherwise, repeatedly check whether the current position of an agent satisfying Kuhn-Tucker condition exists in IR or not in the same manner mentioned above in order of the quality of the upper level objective function value. If an agent whose current position satisfies Kuhn-Tucker condition and exists in IR, the position is regarded as the (approximate) Stackelberg solution and the solution procedure is terminated.

6 Numerical Examples

First, we compare the computational time of generating the initial population including 1000 agents by the homomorphous mapping [4] with that by the random method for a two-level nonlinear programming problem with 4 decision variables and 8 constraints (P1) and one with 6 decision variables and 10 constraints (P2). The results are shown in table 1. The results in table 1 show the effectiveness of the use of the homomorphous mapping in generating the initial population.

Table 1. Comparison of computational times of generating the initial population including 1000 agents

	Computational time (sec.)	
	P1	P2
Homomorphous mapping	1.500	3.109
Random method	177.650	2691.710

Table 2. The efficiency of substeps 8-2 and 8-3

	Upper level objective function	Lower level objecstive function
EMAS without 8-2 and 8-3	−14.999991	0.999997
EMAS with 8-2 and 8-3	−24.0	0.0
Optimal value	−24.0	0.0

Second, in order to investigate the efficiency of substeps 8-2 and 8-3, we apply EMAS without 8-2, 8-3 and one with 8-2, 8-3 to a two-level nonlinear programming problem. In the numerical experiment, the number of agents is 1000 and the maximal generation number is 1000. Results are shown in table 2. From the results in table 2, the use of substeps 8-2 and 8-3 seems efficient.

Next, in order to investigate the efficiency of substep 8-1 for improving the upper level objective function value, we apply EMAS without 8-1 and one with 8-1 to two-level nonlinear programming problems. In the numerical experiment, the number of agents is 1000 and the maximal generation number is 1000. To be more specific, problem A and B are formulated as:

Problem A: Upper level decision variables $\boldsymbol{x}_1 = (x_1, x_2, x_3, x_4)$, lower level decision variables: $\boldsymbol{x}_2 = (x_5, x_6, x_7, x_8)$.

$$\begin{array}{ll}
\underset{\boldsymbol{x}_1}{\text{minimize}} & f_1(\boldsymbol{x}_1, \boldsymbol{x}_2) = x_1^3 + (x_2-2)^2 - x_3x_4 - 2x_5^2 + x_6^4 + (x_7 - x_8)^2 \\
\text{where } \boldsymbol{x}_2 \text{ solves} & \\
\underset{\boldsymbol{x}_2}{\text{minimize}} & f_2(\boldsymbol{x}_1, \boldsymbol{x}_2) = -2x_1^2 + x_2^3 + 2x_3^2 - x_4^4 + 3x_5^3 - 2x_6 + 5x_7 + 4x_8^2 \\
\text{subject to} & 4x_1^2 - 3x_2^3 + 5(x_3-4)^2 - 6x_4 - x_5x_6 - 3x_7^4 + 5x_8 \le 0 \\
& 2x_1x_7 - x_2^2 - 3x_3 - 4x_4^2 + x_5^2 - x_6^3 - 2x_8^2 \le -12 \\
& 5x_1 + x_2 - 6x_3 + 4x_4 - 6x_5 - x_6 - 3x_7 + x_8 \le 5 \\
& -5 \le x_j \le 5, \;\; j = 1, \ldots, 8,
\end{array}$$

Table 3. The efficiency of substep 8-1

	Upper level objective function	
	Problem A (8 decision variables)	Problem B (20 decision variables)
EMAS without 8-1	453.620022	−0.202166
EMAS with 8-1	452.087664	−0.276156

Problem B: Upper level decision variables $\boldsymbol{x}_1 = (x_1, \ldots, x_{10})$, lower level decision variables $\boldsymbol{x}_2 = (x_{11}, \ldots, x_{20})$.

$$\begin{array}{ll}
\underset{\boldsymbol{x}_1}{\text{minimize}} & f_1(\boldsymbol{x}_1, \boldsymbol{x}_2) = -\left| \dfrac{\sum_{j=1}^{20} \cos^4(x_j) - 2\prod_{j=1}^{20} \cos^2(x_j)}{\sqrt{\sum_{j=1}^{20} j x_j^2}} \right| \\
\text{where } \boldsymbol{x}_2 \text{ solves} & \\
\underset{\boldsymbol{x}_2}{\text{minimize}} & f_2(\boldsymbol{x}_1, \boldsymbol{x}_2) = (x_1 - x_{14})(x_{17} - x_6) - (x_4 - x_{11})(x_{18} - x_7) \\
& \qquad + (x_8 - x_{12})(x_5 - x_{19}) - (x_{13} - x_3)(x_{10} - x_{16}) \\
& \qquad + (x_{20} - x_9)(x_2 - x_{15}) \\
\text{subject to} & 0.75 - \prod_{j=1}^{20} x_j \leq 0 \\
& \sum_{j=1}^{20} x_j - 7.5 \cdot 20 \leq 0 \\
& 0 \leq x_j \leq 10, \;\; j = 1, \ldots, 20.
\end{array}$$

From the results in table 3, the use of substep 8-1 is considered to be efficient.

Furthermore, in order to investigate the efficiency of the proposed EMAS, we compare the result obtained by it with that by two-level PSO method [6] in the application of both methods to a two-level nonlinear programming problem with 10 decision variables and 3 constraints. In the numerical experiment, the number of agents is 1000, the maximal generation number is 1000 and the number of trials

Table 4. Comparison of the proposed EMAS with two-level PSO method

	Upper level objective function value	Computational time (sec.)
Best	−199.515027	94.3028
Average	−196.160929	
Worst	−192.904405	
PSO	−184.600761	1357.313

Table 5. Effect of the number of agents and the number of generations on computational time (sec.)

	The number of agents		
	1000	2000	3000
Generation 1000	165.375	330.500	489.343
Generation 2000	344.203	689.984	1034.234
Generation 3000	525.718	1054.718	1587.437

is 10. Table 4 shows the best value, the average value, the worst value of the upper level objective function obtained by the proposed EMAS in 10 trials, the best value obtained by the two-level PSO [6] and the average computational time. The superiority of the proposed EMAS to PSO is shown in table 4.

Finally, table 5 shows the effect of the number of agents and the number of generations on computational time. Table 5 shows the computational time of EMAS linearly increases as the number of agents and the number of generations.

7 Conclusion

In this paper, we discussed an efficient approximate solution method based on evolutionary multi-agent systems to obtain Stackelberg solutions to noncooperative two-level programming problems. In particular, for the application to two-level nonlinear programming problems, we proposed a new EMAS by introducing the homomorphous mapping, the judgment of the rational reaction and the search of the infeasible region. Furthermore, we showed the efficiency of the proposed EMAS through its applications to numerical examples. In near future, we are going to extend the proposed method to noncooperative multi-level programming.

References

1. Bialas, W.F., Karwan, M.H.: On two-level optimization. IEEE Transactions on Automatic Control AC-27, 211–214 (1982)
2. Hansen, P., Jaumard, B., Savard, G.: New branch-and-bound rules for linear bilevel programming. SIAM J. Sci. Satist. Comput. 13, 1194–1217 (1992)
3. Kato, K., Matsui, T., Sakawa, M., Morihara, K.: An approximate solution method based on particle swarm optimization for nonlinear programming problems. Journal of Japan Society for Fuzzy Theory and Intelligent Informatics 20, 399–409 (2008) (in Japanese)
4. Koziel, S., Michalewicz, Z.: Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evol. Comput. 7, 19–44 (1999)
5. Matsui, T., Kato, K., Sakawa, M., Uno, T., Morihara, K.: Particle swarm optimization based heuristics for nonlinear programming problems. In: Proceedings of International MultiConference of Engineers and Computer Scientists 2007, pp. 2312–2317 (2007)
6. Niwa, K., Kato, K., Nishizaki, I., Sakawa, M.: Computational methods through particle swarm optimization for obtaining Stackelberg solutions to two-level nonlinear programming problems. In: Proceedings of 22nd Fuzzy System Symposium, pp. 229–230 (2006) (in Japanese)
7. Shimizu, K., Ishizuka, Y., Bard, J.F.: Nondifferentiable and Two-Level Mathematical Programming. Kluwer Academic Publishers, Boston (1997)
8. Socha, K., Kisiel-Dorohinicki, M.: Agent-based evolutionary multiobjective optimization. In: Proceedings of Congress on Evolutionary Computation, pp. 109–114 (2002)

Multi-Agent-System for General Strategic Interaction

Rustam Tagiew

Institute for Computer Science of TU Bergakademie Freiberg, Germany
tagiew@informatik.tu-freiberg.de

Abstract. The challenge addressed in this paper is designing of a software framework and associated language for interactions between real-world agents, if they reason strategically. The aspired result is general definition, providing and recording of strategic interactions, or also called games. The paper contains an overview of most important preliminary works. We present our approach for game management infrastructure. It is based on a multi-agent programing environment JADE. We also introduce our petri nets based language for definition of a subset of games of imperfect information restricted through finite number of states and actions. The language is additionally able to define time critical processes with discrete intervals. Game representation in our language can be also used for calculating game theoretic or heuristic solutions.

1 Introduction

Strategic interactions (SI) are best studied by game theorists [8]. Game theory is a model for rationality in SI. Game theorists assume common knowledge of rationality of interacting agents between interacting agents. Next assumption in classical game theory is existence of unbounded resources. The aim in game theoretic analysis is to solve the game. Solving games is finding equilibria. An equilibrium is a combination of behaviors of interacting agents - players. None player can improve his circumstances, if he deviates from equilibrium behavior going it alone. An equilibrium is a stable state for a game between rational agents. Rational agents are supposed to choose equilibrium behaviors.

Unfortunately, game theoretic predictions dismatch real-world SI [10]. Real-world agents are not fully rational and do not possess unbounded resources. There are many special cases, where artificial intelligence is used to study real-world SI. For instance, chess programs, which can beat human masters, use suboptimal heuristics. At time, nobody can say which color in chess is supposed to win. An another example are works about opponent modeling in games like Poker. In fact, opponent modeling is a suboptimal heuristic [3]. From game theoretical point of view, none rational player is special, because all rational player behave in same rational way predicted for current SI. Human behavior deviates more significantly from game theoretical predictions than the behavior of artificial agents [4]. The rationality in the behavior of human beings is disputed [5, p. 527–530].

Our approach is to create a software framework and associated language for definition, providing and recording of SI between real-world agents, humans and bots respectively. For this target, we need a SI description language (SIDL), which can be used to

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 649–658, 2009.

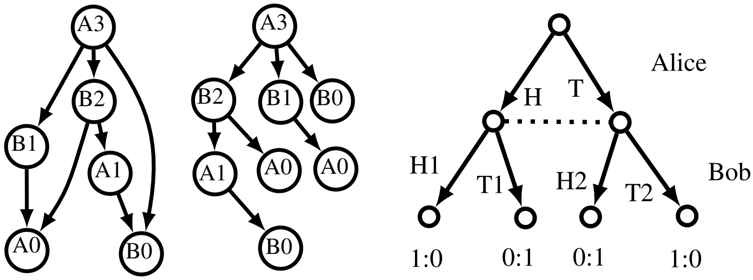


Fig. 1. Graph vs. tree representation and imperfect information

control a game management system and also to feed a game solver routine. These are two different tasks. In the first task, one needs a system for conducting evaluations of different possibly distributed players. For the other task, one need a library for calculating specific behavior. These two tasks are solved in most works game specifically using a pile of low level code. This paper addresses this problem and introduces a high level structural representation for SI.

Let us consider a simplified variant of game Nim with only one heap. At the beginning, one has a set of indistinguishable items[1]. This game is played by two players. We call further the first player Alice and the second Bob. They perform successively turns. Presentation of games as sequences of turn is called extensive Form. Every player can take at least one but not more than three items. If a player takes the last item, he loses. To model Nim we use a set of states. Every state is defined through a combination of the owner of the next turn and the amount of the items. For instance, state B0 means that it is Bob's turn, there are no more items and hence he wins. The states can be represented as nodes of a directed graph or of a tree with repeated states. Fig. 1 (left) depicts both representation possibilities. In the tree case, the states A0 and B0 are repeated and there is no more than one path between two states. If the game graph is cyclic then the game has an unlimited number of possible rounds. Every arc in the game graph corresponds to an action of a player. All outgoing arcs of a node have the same owner. Some of the states imply payoffs for the players. The payoffs are added to the account balances of the players. In Nim, the winner gets, what the loser losses. Nim is a game of perfect information. This means that every player knows the current state.

One another example is Matching Pennies. This game is also played by two players. Every player has only two actions. The players chose their actions secretly and then indicate them simultaneously. Alice wins if both players choose the same action, elsewise Bob. This is a game of imperfect information. It can be presented as set of payoff matrixes. Such representation is called strategic form. The graph of this game can be defined as in fig. 1 (right). The dashed line on this graph depicts Bob's uncertainness about Alice's choice. Actions H1 and H2 correspond to same choice Head. Bob can distinguish between actions H_ and T_, but not between _1 and _2 (_ is a variable). Bob is not sure about the state the game is in. Alice is also uncertain about Bob's turn. The

[1] e.g. matches.

players on this figure can be swapped, but it would not change the game structure. This presentation of games of imperfect information is in extensive form and commonly used by theorists [8]. Matching Pennies can be also performed between Alice and an ordinary count. Alice wants to guess the side of fallen count before or after it is thrown. The only change on fig.1 (right) is to replace Bob with Nature. Nature is an unpaid player with known distribution over his actions.

In some cases, at least one player is unsure about the structure of the game, viz. states, action and payoffs. Such cases are called games of incomplete information. Notions imperfect and incomplete are not the same in game theory. Games of incomplete information can be transformed to games of imperfect information, if a distribution over possible game structures is known. In this work we concentrate ourselves only on SI, which can be defined as games of imperfect information with finite states and actions and discrete time. An example of a game with infinite actions is a variant of Silverman's game. In Silverman's game both players have $\mathbb{N}_1^+$ as a set of actions. They choose simultaneously actions. A player wins, if his number is between opponent's number and opponent's number times three. The solution of this game is an equal distribution between 1, 2 and 5. Generally speaking, games like Silverman's game, which do not conform the restrictions, can not be computed on most current architectures.

Chapter 2 gives an overview of related but disjunct works relevant for our research. In chapter 3, we show the game management infrastructure based on JADE. Chapters 4 present petri nets and how they can be used for game definition. Chapter 5 presents examples modeled using our language. Chapter 6 presents YAML based syntax for game representation. In chapter 7, we present an algorithm for conducting SI. Future work, conclusion and implementation complete the work.

2 Related Works

There are different kinds of motivation for related works. As first, we introduce GALA system and GALA language [7]. GALA is designed for representation and solving of games. It provides a prolog-based GALA language for representing of games as braching programs. Games can be of imperfect information. The game solving algorithm is based on game tree generation and commercial linear program solving libraries. GALA is considered for games with finite states and actions, because only finite trees can be computed. GALA does not provide time dependent features of a game. It also does not provide an environment for evaluating real-world agents. GALA can also use GAMBIT for game solving. GAMBIT is a state-of-art tool for finding equilibria in games. For GAMBIT, games can be presented in normal or extensive form.

The next work, we introduce is ACE framework [11]. It is a general concept of software architecture for investigation of economic processes. ACE is an implemented design for a multiagent system consisting of four kinds of agents: worlds, markets, firms and consumers. This design is strong related to common economics. Worlds contain information about state of environment like laws, agreements and ownership. Markets are in their function similar to worlds. Firms and consumers are self-interested agents. This framework is used for studying economic systems.

General game playing (GGP) approach considers a system for competitions of artificial agents in diverse games [6]. A logic-based game description language (GDL) is invented for games with finite states and actions of perfect information without time dependent elements. Artificial agents are expected to read GDL input and develop an efficient strategy. A game is considered as a state machine. Transitions between states are actions of players. States are not monolithic. They are represented as databases and actions are rules for transformation of a database. This state machine is described in GDL. On the basis of GDL representation, the system bootstraps game management infrastructure. The main element, game master, is implemented as web service. The system and the language are already used in programing competitions.

3 SI Infrastructure

JADE [1] is used for our framework for management of SI (FRAMASI). JADE is a Java-based multi-agent programming environment compliant to FIPA[2] specifications. For FRAMASI, three types of agents are defined - periphery agents, worlds and players (similar to ACE). A world represents the SI himself. A world has 2 states, if he is alive - 'gathering players' and 'managing interaction' (fig.2). In state 'gathering players', world is registered in 'yellow pages'. In most cases a world needs more than one player. He deregisters himself in state 'managing interaction'. The goal of a world is to gather enough players, arrange SI and then die.

Players (fig.2) search for worlds and then contract with a world over FIPA-contract net-protocol (SC00026), about how and at which place to participate. If interaction participation conditions match, then the player is joined to a world. If a world filled his player slots, he begins to supply players with (partial) information about running SI. Players send their actions to the world. In contrast to worlds, player do not die after fullfilled SI. There is also an ability for putting players in 'nirvana' through external commands. In 'nirvana', players do nothing. Leaving worlds without completed SI is considered only for exceptions.

A couple of periphery agents add some extended abilities to our system. For instance, a protocoler agent can connect a world and save events in SI to a file or database. One can also consider a player as proxy agent for a human.

We designed abstract classes for worlds and players. Most important methods, which must be implemented to create a concrete world or a player are the following:

World.getAdditional→String: Get additional description of the game for sending to interested players.
World.getUpdate→Map<Player, String>: Update situation and return information to be send to players (periodically executed).
World.received(Player player, String action): Implement a player's command.
World.updatingFlag→Boolean: A flag used to pause world.
Player.chosePlace(Slots si, Additional description): Choose a slot for interaction. Description can be also read for making choice.

[2] Foundation for Intelligent Physical Agents.

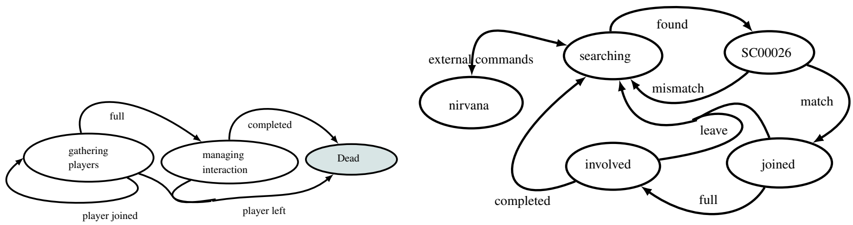


Fig. 2. World and player state transition diagrams

Player.update(String information): Get information about world's state and calculate the current command.

Player.act→String: Get the current command (periodically executed).

4 Petri Nets for SI

A PN is a labeled directed graph, formally represented through a tuple:

$$\text{PN} = (P, Q, F, W, M) \tag{1}$$

It consists of places P (circles), which can be filled with a positive amount of tokens (dots), and transitions Q (rectangles), which can be fired with an effect on places. Places and transitions are disjunct $P \cap Q = \oslash$. Places can not be connected to places and transition not to transitions. Arcs $F \subseteq (P \times Q) \cup (Q \times P)$ are weighted with positive natural numbers $W : F \rightarrow \mathbb{N}_1^+$. A transition can not be fired, when any place of incoming arcs have less tokens than its arc weight. Firing abolishes tokens of every incoming arc and produces tokens for every outgoing arc according to their weights. The fill levels of the places represent the state of the petri net. Vector $M \in \mathbb{N}^{|P|}$ is the current state of PN. Firing of a transition transforms current state M to M'. It can be defined as a vector addition:

$$M' = M + \widehat{q}^t, \text{where} \quad \widehat{q}^t(i) = W(t, p_i) - W(p_i, t) \wedge t \in Q \wedge p_i \in P \tag{2}$$

We discovered transition system (TS) of a PN as basis for game graph respresentation. TS is the following tuple [9]:

$$\text{TS} = (S, \Sigma, \rightarrow) \tag{3}$$

S - states, Σ - a set of transitions, $\rightarrow \subseteq S \times \Sigma \times S$ - directed arcs between states.

Theorem 1. *For every game TS with finite amount of states and actions, at least one PN can be constructed.*

Proof. (Sketch) Create for every action a transition and for every state a place. Every arc of this PN is weighted with 1. Create states as assignments of places. All places of a state have zero tokens, except of the place, which corresponds to this state. This place has one token. Find $\widehat{q}_t$ using equation 2 for every two states, which are connected by an action t. Derive outgoing and incoming arcs for a transition on basis of $\widehat{q}_t$. One can easily find a game TS, where one can construct more than one PN.

To model SI, PN is extended with tuple:

$$\text{SI} = (I, C, N, D, A, O, H, B) \tag{4}$$

I is a set of agents, empty element ε stands for nature or world accordingly.
$C \subseteq Q^*$ is a subset of sequences of transitions, called choice sets. Every transition is a member of only one element of C.
$N : C \rightarrow \mathbb{N}$ is a numbering function and not injective.
$D : \mathbb{N} \rightarrow (\mathbb{R}_0^1)^n$ is a function for firing probability distribution in a choice set, where $\sum(D(\_)) = 1$. n is number of elements of the related choice sets.
$O : \mathbb{N} \rightarrow I \cup \varepsilon$ denotes ownership.
$A : Q \rightarrow \mathbb{R}^{|I|}$ is the payoff vector of a transition, if it fires.
$H : P \rightarrow \{I\}$ provides for every place a subset of agents for which it is hidden. Agents can alter D for own numbers and see all unhidden places.
$B : I \rightarrow \mathbb{R}$ is the current account balance of agents.

Combination of PN (1) and SI (4) is the required strategic interaction language and also an infinite state machine PNSI[3]. The tuple (M, B) represents the current state of PNSI.

5 Examples

Fig.3(I) models choice between A and B for agent Alice with imperfect information about the state of the world. Dashed elements of this diagram represent the enhancement SI of PN. The choice set numbered with 0 is controlled by world and attached with a distribution over both events. The places can not be seen by Alice and she can only alter the distribution for all choice sets under number 1. Fig.3(II) shows modelling time critical concurrency using PNSI. At the start, transitions S1 and S2 have a firing probability of 1. Other transitions have a firing probability of 0. The token in resource can not be used. The agent, who alters his distributions first, can use the resource exclusively.

Fig.4(left) demonstrates an application of this formalism for a repeated variant of the already mentioned game Matching Pennies. Dashed diamonds are taken for the gathered payoff of players. Every transition, which is not in dashed box, is in a single element choice set and has a firing probability of 1. Altering of D must be done in right moment, because it has a default state and transitions firing algorithm does not wait till all agents made decisions. The places Ready and Time shows to the players, if they have or have not to react. The weight 'latency' between Time and Timer represents the time period for simultaneous decisions.

Fig.4(right) represents game Nim for one heap. At the beginning, Bob has turn. Alice and Bob have -1 and 1 as initial account balances. Arcs from place Items to players's actions are weighted with 1, 2 and 3. If Bob or Alice performs a turns, the account balances swap. If Items is empty, none of transitions can be fired. In this case, the game ends. As you can see, one can hack in PNSI and create very compact and exact representation of a game.

[3] [ˈpɛnˈzaɪ].

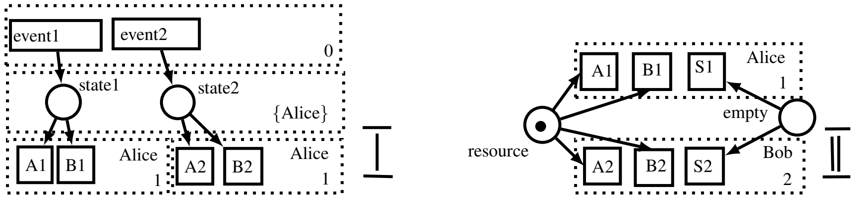


Fig. 3. Imperfect inform. & time critical concurrency

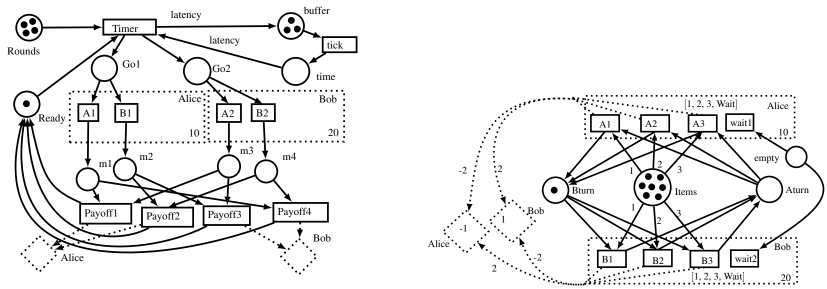


Fig. 4. Matching Pennies and Nim for one heap

6 Syntax for the Language

YAML[2] is used as basis for representation of the scenario in PNSI language. YAML is a human-friendly serialization and definition language. In YAML we can represent all kinds of nested collections. The advantage of YAML compared to other languages is its slight and highly readable syntax. For instance, ': ' means in YAML 'maps' in a map, '- ' means entry in a list or set and '#' means comment. Space leading a line measures its degree of nesting. Collections can be also written inline using '{}' for maps and '[]' elsewise. '- - -' stands at the begin of every document. Following commented code shows a representation of fig.3(I) in YAML:

```
---
agents:
  Alice: 0.0 # Agent -> initial Payoff
numbering:
  - actions: [Event1, Event2] # transitions
    number: 0
  - {actions: [A1, B1], number: 1}
  - {actions: [A2, B2], number: 1}
numbers:
  0: distribution: [0.5, 0.5]
  1: distribution: [0.5, 0.5] # default distribution
     aliases: [A, B] # optional aliases
     owner: Alice # Agent
places:
```

```
  State1:
    tokens: 0 # initial tokens
    hidden: [Alice] # set of agents
  State2: {tokens: 0, hidden: [Alice]}
transitions:
  Event1:
    outgoing:
      State1: 1 # Place -> Weight
  Event2: {outgoing: {State2: 1}}
  A1:
    incoming: {State1: 1}
    payoffs:
      Alice: 0.0 # Agent -> additional Payoff
  B1: {incoming: {State1: 1}}
  A2: {incoming: {State2: 1}}
  B2: {incoming: {State2: 1}}
```

7 Interaction Algorithm

As it is conceived, PNSI can be also used for client-server communication. First, the server sends the complete PNSI representations of the game. Then, the players apply for positions in this game. After all positions are filled with agents, the server starts to iterate Alg.1 (ruby similar syntax). As we see in lines 15–20, for every choice set no more than one transition can be fired in a step. The transition, which is to be fired, is chosen for every choice set separately. PNSI is closely related to stochastic PN with difference in the concept of choice set. In our application, we use the same algorithm for a PNSI based World agent. World dies, if its petri net is dead. This iteration can be interrupted, saved as a PNSI YAML file or in a data base and then loaded and restarted again. PNSI provides also an ability of persistent game evaluation.

8 Future Work

As it is already mentioned, we consider to implement a way for solving general SI defined using PNSI. To solve a game represented in PNSI one has to calculate the specific game tree first. The nodes of this tree correspond to the reachable assignments of the places and the account balances (M, B). Every reachable assignment corresponds to at least one tree node. An arc of this tree is a possible step of the algorithm presented above. The solution is a map between states and commands for changing the distributions for own numbers. To find it, one has to dump the game tree in GAMBIT compatible format.

The other direction is adding syntactical sugar to PNSI. PNSI can be also translated to a logic-based language and vice versa. Both mentioned works [7][6] about constructing a language for representing games are logic-based. Our approach uses at time no logic-based representation. The resulting problem is a restricted readability of textual representation. We experiment with constructing of PNSI game representation through renaming and merging small basic PNSI structures. Such basic PNSI structures can be cataloged using a logic-based approach.

Algorithm 1. PNSI based World.

```
Data: transitions, choice_sets, agents, amounts
while not time_period_expired do
    receive_altering_commands_from_players
    implement_received_commands
end
create_set(fireable)
foreach t in transitions do
    if enough_incoming_tokens(t) then
        fireable.add(t)
    end
end
if fireably.empty then
    complete_game
end
create_list(tobefired)
foreach c in choice_sets do
    th = c.choose_randomly_transition
    if fireable.contains(th) then
        tobefired.add(th)
    end
end
create_list(fired)
create_set(changed)
while not tobefired.empty do
    ta = tobefired.remove_at_index(random_value)
    if enough_incoming_tokens(ta) then
        abolish_incoming_tokens(ta)
        changed.add(ta.incoming)
        fired.add(ta)
    end
end
while not fired.empty do
    tp = fired.remove_first
    produce_outgoing_tokens(tp)
    changed.add(ta.outgoing)
    produce_payoffs(tp)
end
foreach a in agents do
    foreach p in changed do
        if not hidden(p, a) then
            add2message(a, p.id, p.value)
        end
    end
    foreach p in amounts do
        add2message(a, p.id, p.value)
    end
    send_message(a)
end
```

9 Conclusion

We created two mighty and independent concepts for managing SI - FRAMASI and PNSI. We also fused these two concepts to a multi-agent system for conducting studies of human behavior, managing AI programming contests and human-machine-interaction in general SI. The fused system runs distributed and is persistent. Finally, we discussed two directions of developing our system - Game Solver and SIDL.

Libraries based on JADE and YAML are implemented in Java and can be downloaded from our homepage. There are two main parts - FRAMASI and PNSI. Both can be used independently. The class, which connects both parts is *PetriWorld*. It is an implementation of a world using PNSI. The libraries are already successfully tested for a study conducted with humans.

References

1. Bellifemine, F., Poggi, A., Rimassa, G.: Jade: a fipa2000 compliant agent development environment. In: AGENTS 2001. ACM, New York (2001)
2. Ben-Kiki, O., Evans, C., Ingerson, B.: Yaml ain't markup language (YAML™) (2004), yaml.org/spec/
3. Billings, D., Burch, N., Davidson, A., Holte, R., Schaeffer, J., Schauenberg, T., Szafron, D.: Approximating game-theoretic optimal strategies for full-scale poker. In: IJCAI 2003 (2003)
4. Camerer, C.F.: Behavioral Game Theory. Princeton University Press, New Jersey (2003)
5. Eysenck, M.W., Keane, M.T.: Cognitive Pychology. Psychology Press, Hove (2005)
6. Genesereth, M.R., Love, N., Pell, B.: General game playing: Overview of the AAAI competition. AI Magazine 26(2), 62–72 (2005)
7. Koller, D., Pfeffer, A.: Representations and solutions for game-theoretic problems. Artificial Intelligence 94(1-2), 167–215 (1997)
8. Osborne, M.J., Rubinstein, A.: A course in game theory. MIT Press, Cambridge (1994)
9. Priese, L., Wimmel, H.: Petri-Netze. Springer, Heidelberg (2008)
10. Tagiew, R.: Simplest scenario for mutual nested modeling in human-machine-interaction. In: Dengel, A.R., Berns, K., Breuel, T.M., Bomarius, F., Roth-Berghofer, T.R. (eds.) KI 2008. LNCS, vol. 5243, pp. 364–371. Springer, Heidelberg (2008)
11. Tesfatsion, L., Judd, K.: Handbook of computational economics, vol. 2. Elsevier/North-Holland (2006)

Simulation Study of Public Goods Experiment*

Zhi Tang[1], Jian Wu[1], Wei Shi[1], Jing Wang[2], and Hanyi Xu[2]

[1] College of Computer Science, Zhejiang University, Hangzhou, China
[2] College of Science, Zhejiang University, Hangzhou, China
tangzhi2008@hotmail.com, wujian2000@zju.edu.cn,
weishi@zju.edu.cn,
{king3days, xuhanyi}@hotmail.com

Abstract. This paper built a simulation system of Public Goods Experiment to study the Conditional Cooperation. The study of this paper was divided into 3 parts including the execution of Public Goods Experiment in laboratory, the BPNN (Back Propagation Neural Networks) modeling of the resulting data of the former experiments and finally the developing of a simulation system of Public Goods Experiment. The results of the simulation show that displaying the number and name of people who have endowed to the Public Goods Pool instantly is the best condition for stimulating Conditional Cooperation which also means the best information displaying condition of avoiding free riders.

Keywords: Public Goods Experiment, Conditional Cooperation, BPNN, Simulation, Agent.

1 Introduction

In economics, a public good is defined to be a good that is non-rivaled and non-excludable which means respectively that one's consumption of the good will not influences others' consumptions of this good and everyone will not be excluded from consuming the good[1]. The Free Rider problem has been together with the public goods since it was produced. In order to demonstrate this, we considered the national defense which is public goods. People who have taken the military service will enjoy the benefit of national defense. However, who don't take military service will not be excluded from national defense. These who avoid of being included in military service are the very 'free riders'[2]. Since a lot of essential elements of living such as public education and public health can be considered to be public goods, if the rate of free riders becomes too large, the whole amount of public goods will be jeopardized and the benefits that can be enjoyed by individual will be reduced significantly. So in order to obtain the maximum amount of public goods, our goal is to find the solution that can reduce the rate of free riders maximally.

* This research is partially supported by the 863 program under grant of 2008AA01Z141, the National Natural Science Foundation of China under grant of 60873224, the National Key Technology R&D Program under grant of 2006BAH02A01. Corresponding author: Wei Shi.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 659–669, 2009.

Since public goods relate to many complex facts such as nation, religion and gender, it's hard to research relating problem of public goods under real environment. So researchers chose to simplify public goods in laboratory and finally Bohm established the classic Public Goods Experiment in 1972. Since then, the theoretical research of public goods has mainly been the research of Public Goods Environment.

This paper mainly studied the Conditional Cooperation in the Public Goods Experiment. People uses Conditional Cooperation to explain the phenomenon that people is "*willing to contribute the more to a public good, the more others contribute*"[3]. Since from the economic perspective, all people are considered to be reasonable man who behaviors only to magnify his own benefit. Conditional Cooperation is developed to explain the behavior of Social Man who is willing to benefit others in the Public Goods Experiment.

Our object is to reduce the number of free riders and increase the rate of cooperation in the experiments. According to the fact that people can't punish free riders in real life, we designed four kinds of experiment with no punishment to find the solution of generating and maintaining high rate of Conditional Cooperation. These four experiments are all the same except different information displaying conditions. This design will make it easier to classify the effects of different information displaying conditions and find the most efficient strategy of producing public goods.

The whole analysis of this paper can be divided into three parts. Part 1 is the design and execution of experiments. From these experiments, we got more than 30,000 groups of data and then in part 2 we preprocess the resulting data of the former experiments to get sample data firstly. Then we built a BP Neural Networks model and then use the sample date to train the model to generate a function which can forecast the probability of cooperation of every participant in the experiment. Based on this function, we designed and programmed a simulation system to simulate these experiments on the computer in part 3. Also in part 3 we analyzed the results of simulation to get our solution of avoiding free riders in Public Goods Experiment.

2 Related Works

The classic Public Goods Experiment was firstly mentioned by Bohm in 1972. Since that time, it has experienced a lot of revisions to meet researchers' different requirements. However, it had always been the most important way of researching public goods.

In 2000, Fehr published a paper on science, which is about the Public Goods Experiment with punishment. He divided 4 people into a group and found that punishment will obtain and maintain a complete cooperation. He also illustrated that free riding causes strong negative emotions in cooperators [4]. Since the huge success of Fehr, many researchers followed him idea of dividing 4 people into a group, this can help the players distinguish others' strategy of gaming quickly. We designed an experiment with 10 people in a group which will increase the difficulty of gaming and help the players hide their strategies. Also, this will magnify the impact of the number of people who have endowed to others during the decision-making process.

Traditionally, researchers favored statistical analysis to extract the inner rule of data [4]. This method will lose effect while the data are discrete. And the data of

Public Goods Experiment are discrete, so this paper chose the BPNN to find the inner relationship between the independent variables and dependent variable which will make the result more consistent to real life.

For Agent simulation of Public Goods Experiment, the Agent only has two behaviors: cooperation and defection. Searching the piles of documents, we can found that the decision-making algorithm is always Tit for Tat, ALLC (all cooperation) or ALLD (all defection) [5]. Historically, researchers like to set cooperation to be 1 and 0 for defection [6]. In this paper, if we choose the same 0-1 solution, we can't have enough samples for BPNN model to learn from. Therefore, we can't get the probability forecasting function. So we will preprocess the data before modeling by BPNN. In the simulation, Instead of gaming with neighbors and focusing on the analysis of neighbors' decision [5], this paper let the players game together. And the behavior of the Agent is based on the result of BPNN model.

The paper is organized in the following manner. Section 3 briefly deals with the design and execution of experiments, while in Section 4 is the details of the preprocessing of data and the BPNN model are discussed. Then in section 5 the design of the simulation system are presented. Result analysis and conclusion of simulation are showed in section 6 while concluding remarks are presented in Section 7.

3 Experimental Design

At first, we will introduce the classic Public Goods Experiment based on which we designed our own experiment. The following description is about the classic Public Goods Experiment.

There are n participants and every one get Z Yuan at first. Everyone can choose to endow nothing or R Yuan into the same Public Goods Pool in S seconds. When the nth people finished his/her investment, this very round was end. The Public Goods Pool has nothing at first. Since the Public Goods Pool has a benefit rate of ρ , all the money that are endowed at this very round is multiplied by ρ and then the benefit will be distributed to all the n participants averagely. After a single round, everyone's private goods will be Z minus the money that was endowed and plus the benefit from the Public Goods Pool.

The value of MPCR (Marginal Per Capital Return) can determine whether an individual can be free riders in the experiment. MPCR of every participant in Public Goods Experiment is $\rho \times 1/n$, and if $1/n$ <MPCR<1[7], individual will have the possibility of being Free Riders which means they can get revenue from the contribution of others without endowing anything to the Public Goods Pool.

In our experiment, we set ρ to be 2.0, and n=10, Z=5000, R=50 and S=30. The experiment will repeat 100 rounds for a group of 10 participants every time. At the same time, the amount of both public goods and individual goods will be cumulated. The MPCR of our experiment is 0.2 which is smaller than 1 and bigger than 0.1, so the participants in our experiment are possible to be Free Riders.

Then, we designed four types of information displaying condition. We used a parameter Environment to quantify the information displaying condition. The value of this parameter, the corresponding name and meaning are all shown in Table 1. During

Table 1. Four information displaying conditions

Environmental parameter	Name	Meaning
1	Testnone	Show nothing about others during the decision-making 30 seconds
2	Testnum	Show the instant number of people who have endowed at this round during the decision-making 30 seconds
3	Testname	Show the instant number and name of people who have endowed at this round during the decision-making 30 seconds
4	Testnoname	Show the instant number and name of people who have chosen not to endow at this round during the decision-making 30 seconds

the process of decision-making, despite of environment, every participant can know the number of round, the amount of public goods, the amount of their individual goods and the profit rate.

4 The BPNN Model

4.1 BPNN Model Design

BPNN is the ANN (Artificial Neural Networks) with back propagation algorithm. It comprises input layer, middle layer and output layer. While giving the BPNN some representative samples, the BPNN will learn from these samples. Theoretically, a good BPNN can approach any nonlinear mapping between input and output unlimitedly. The structure of BPNN is shown in Fig. 1.

In this paper, we chose the experience (the average number of people who have endowed in the past rounds) and present (the instant number of people who have endowed in a single round) as input. And the output is the probability of endowment. Input parameters can be achieved in the database directly, however, the output had to be calculated from the data in the database.

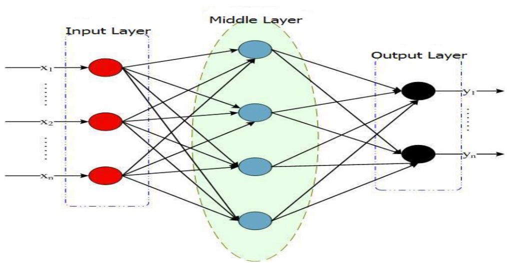


Fig. 1. Structure of BPNN

The formula of getting the probability of endowment is as follow:

$$p = m / (m + n) \ . \quad (1)$$

In this formula, n is the number of people who have chosen to not endow while m is the number of people who endowed to the Public Goods Pool. For instance, under the environment 1, when experience is 5 and present is 3, there are 10 chose to endow while 20 chose to not endow, we can get the probability of endowment at the condition (1,5,3) is10/(10+20)=33.3%.

Since there are four kinds of experiments in the research, and the participants of the experiment are complicated, in order to simplify the analysis, we ignored the differences among various individuals (such as gentle, age, and background) and treated the participants under the same environment as the same. So finally, we got 4 groups of data which were marked by Testnone, Testnum, Testname and Testnoname correspond to the four environmental parameters respectively. Then we took the use of the formula 1 to calculate the probability of endowment. From this, we generated the sample for the BPNN to learn from.

Since this model has 2 inputs and 1 output, the number of nodes of the input and output layer is 2 and 1 respectively. The most important and difficult thing is to decide the number of nodes of the middle layer. The middle layer will influence the accuracy of modeling significantly.

Since we didn't know the most suitable number of nodes of middle layer, we managed it by trial and error. We tried the BPNN with a middle layer has 3 to 11 nodes and tested the accuracy of them respectively. Finally, we found that a middle layer with 8 nodes is the most suitable architecture for us. So the ultimate architecture of our BPNN is 2-8-1.

4.2 The Result of the BPNN Modeling

After we have confirmed the construction of BPNN model, we provided it with the sample and got the formulas of the endowing probability forecasting function.

$$a_i = \frac{1}{1+\exp(-(\alpha \mathbf{w}_{i1}^{(1)} + \beta \mathbf{w}_{i2}^{(1)} - \boldsymbol{\theta}_i^{(1)}))} \ . \quad (2)$$

$$y = \frac{1}{1+\exp(-(a_1\mathbf{w}_{11}^{(2)} + a_2\mathbf{w}_{12}^{(2)} + a_3\mathbf{w}_{21}^{(2)} + a_4\mathbf{w}_{22}^{(2)} + a_5\mathbf{w}_{31}^{(2)} + a_6\mathbf{w}_{32}^{(2)} - \theta^{(2)}))} \ . \quad (3)$$

Where α and β is the input which represent present and experience respectively. Recall that present is the number of people who have chosen to endow at present while experience is the number of people who have chosen to input in the past rounds, these two parameters can be calculated during the simulating process. And y is the endowing probability while the input is (α, β).Let i=1,2,…,6. So in order to calculate y, the formula 2 will be repeated for 6 times. $\mathbf{w}_1$, $\mathbf{w}_2$ are 2×6 and 2×3 matrix respectively while $\boldsymbol{\theta}_1$ is a 6×1 matrix and $\boldsymbol{\theta}_2$ is a 1×1 matrix. All $\mathbf{w}_1$, $\mathbf{w}_2$, $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$

Table 2. $\mathbf{w}_1, \mathbf{w}_2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ for Testnone environment

$$\mathbf{w}_1 = \begin{bmatrix} -2.42719182083092 & -1.79634352581538 \\ -0.489820555057140 & 7.98046255221274 \\ 1.56617752156778 & -4.32640208198704 \\ -2.70679240097910 & -3.84750304389074 \\ 1.17270300472647 & 0.534069409346248 \\ 0.0252642165913305 & 3.90039094283573 \end{bmatrix} \quad \boldsymbol{\theta}_1 = \begin{bmatrix} -0.0492087717405787 \\ 3.69739448313514 \\ 0.677730619364981 \\ -0.123587469212692 \\ 0.706909995879256 \\ 0.167581476991997 \end{bmatrix}^T$$

$$\mathbf{w}_2 = \begin{bmatrix} 2.67958125413262 & 5.58277260354765 \\ 3.89770735165283 & 4.14337837678823 \\ -1.04236907859561 & -2.36702216058399 \end{bmatrix} \quad \boldsymbol{\theta}_2 = \begin{bmatrix} 1.440238198078728 \end{bmatrix}$$

are constant. They are generated by the learning of the BPNN model. Since there are 4 types of experiment and also 4 groups of sample data, we got 4 groups of value of $\mathbf{w}_1, \mathbf{w}_2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2$. In Table 2 we illustrate the value of $\mathbf{w}_1, \mathbf{w}_2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ for Testnone environment(Since the space are limited, we won't illustrate all the 4 groups).

From formula 2 and 3, we can calculate the endowing probability at any environment since the value of α and β are available. At the same time, in order to minimize errors, we should standardize the input and output to range them in (0, 1).

5 Simulation Program

5.1 The Platform of Simulation

This paper utilized the Swarm platform to implement the simulation. We took the Agent-Based method to simulate the four real experiments in computer. Swarm is a software toolkits developed by SFI (Santa Fe Institute). Normally, the program developed on Swarm platform contains two parts, the Model Swarm and the Observer Swarm.

Model Swarm is the most significant part of a Swarm simulation system. It generates Agents and regulates the behavior of these Agents. Agents are individuals which are adaptive to environmental changes. In this program, Agents represent the anticipants of Public Goods Experiment. Model Swarm also utilizes the data structure Action to define the execution sequences of Agents' various behaviors. Model Swarm should also contain the input and output of the program. Input includes all the parameters of the program while output is the result of the program including data and logs.

Observer Swarm is used to observe and set the parameters and attributes of the system before the simulation was started. Also, Observer Swarm illustrates the simulating result on the form of diagram.

5.2 Simulating Model Design

This paper developed a top-down model to simulate the endowment of Agents and the interaction between Agents and environment. Therefore, this model contains two parts, Agent and its environment.

Agent

Agent represents the participant of the experiment. An Agent comprises Attributes Set, Strategy Set, Decision-making Algorithm and Learning Algorithm.

The attributes of an Agent including Agent's location which is the coordinates of Agent, individual goods of the Agent, public goods, round, color, profit of each round and the number. The Strategy Set of this model only has cooperation and defection two strategies.

Whether an Agent chooses to cooperate or defect is decided by the Decision-making Algorithm which is also the result of BPNN modeling. This result can be formulated into a function on the form as follow:

$$p(n) = f(e_{n-1}, t_n) \ . \tag{4}$$

In formula 4, e_{n-1} is the average number of Agents who endowed in the last n-1 rounds, t_n is the instant number of Agents that have chosen to endow at round n. e_{n-1} and t_n must be standardized to fit in the function generated by the BPNN. After an Agent has generated a probability p from this formula, the program compares p with a random number x (0<x<1). If p is bigger than x, the Agent will endow. However, if p is smaller than x, the Agent will not endow.

After an Agent had chosen its strategy, that Agent will change its attributes according to the Learning Algorithm. The learning algorithm was shown as follows in Table 3.

In the learning algorithm below, $pr(n)$ means the amount of Public Goods at round n while $pr(n-1)$ is the amount of Public Goods at round n-1; c is the number of Agents that have chosen to endow, $ir(n-1)$ represents the individual goods at round n while $ir(n-1)$ represents that at round n-1; a is the number of all Agents; $e(n)$ is the average number of Agents that chose to endow in the past n rounds.

Environment

The environment is a black square grid in which Agents are located at and interacted with each other whose size can be determined by users before the simulation was started. At the beginning of the simulation, the Model Swarm generated Agents and located them in the grid randomly. The attributes of environment is the profit rate which is set to be 2.0 all the time. At the same time, we put the value of the Public Goods at the location (0, 0). During the simulation, all the Agents revise this value to make sure every Agent has the same value of Public Goods.

Table 3. Learning algorithm

if (have invested)	if (have not invested)
{	{
$pr(n) = pr(n-1) + 50 * c$	$pr(n) = pr(n-1) + 50 * c$
$ir(n) = ir(n-1) - 50 + 50 * 2 * c / a$	$ir(n) = ir(n-1) + 50 * 2 * c / a$
$e(n) = pr(n) / 50 / n$	$e(n) = pr(n) / 50 / n$
}	}

5.3 Program Design

After we have finished the design of model, we programmed to develop the simulation system. Eventually, the architecture of the simulation program is shown in Fig. 3.

From Fig. 3, we can get the relationship between each part of the program clearly. Observer Swarm is the top Swarm, it shows the Agents that are created by Model Swarm, environment and Agents' behavior. Observer Swarm also provides input panel through which users can set parameters including the size of the world, value of environment, number of Agents and the total number of round. This control panel was shown in Fig. 5.The sequences of the Agents' actions are shown in Fig. 4.

6 The Result of Simulation

After we had completed the programming work, we can utilize the system to simulate the Public Goods Experiment and analyze the results of simulation. Before the beginning of simulation, we should set the environmental parameter through the control panel firstly. Then we can simulate these 4 types of experiments easily through changing the environmental parameter.

We set the environmental parameter to be 1,2,3,4, the number of Agents to be 10, 50, 100, 200, 300, 400, 500 and the number of rounds to be 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000 respectively. Then according to the result of simulation, we get the following diagram of probability of endowment under these four environments.

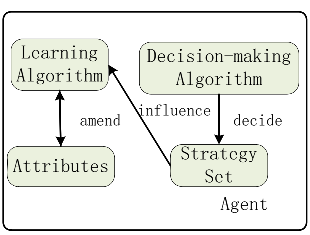


Fig. 2. Structure of Agent

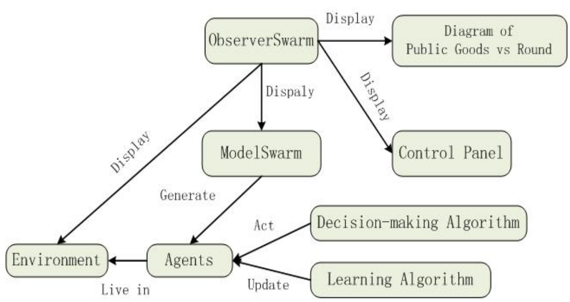


Fig. 3. Architecture of the simulation program

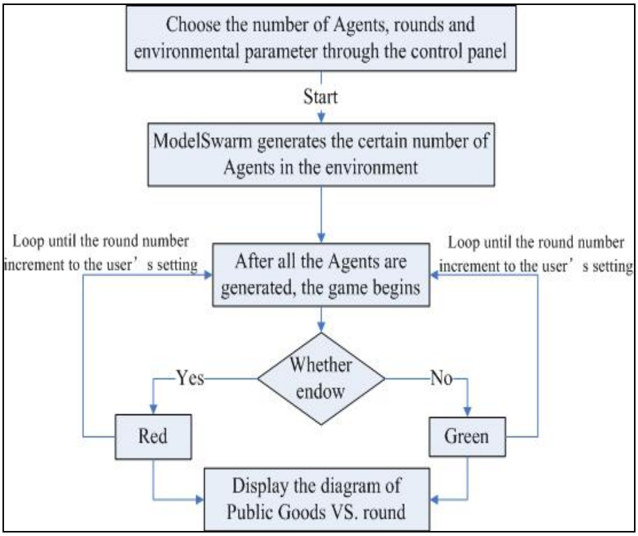


Fig. 4. Process of the simulation system

From Fig. 6-9, we can get the following results:

Result1: *While displaying nothing, the probability of conditional cooperation is the lowest.*

If the participants are given no hint, they distrust each other and have the lowest probability of endowment. From Fig. 6, we can see that Testnone environment has a probability of endowment 0.2 which is the lowest of the four environments. This proves that the players will think more from their own benefits and also believe that others will do this as well. Therefore, in order to protect their own benefits, they will choose defection instead of cooperation most of the time .

Result2: *While showing the instant number and name of people who chose not to endow can generate and maintain a higher probability of conditional cooperation in society.*

From Fig. 7, we can see that the average probability of endowment of Testnoname environment is 2 times higher than that of Testnone. Since the displaying of name of

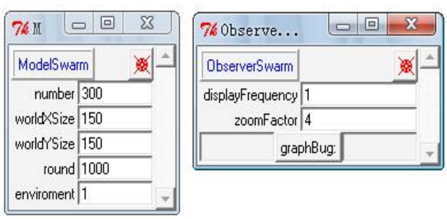


Fig. 5. Control panel of simulation system

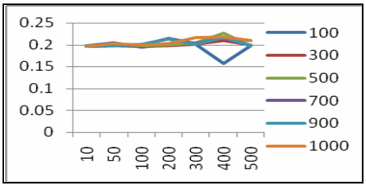


Fig. 6. Endowing probability of Testnone

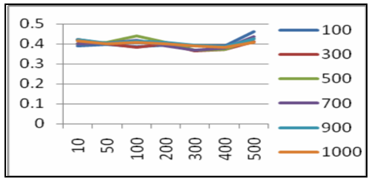


Fig. 7. Endowing probability of Testnoname

people who didn't endow can be seen as a punishment to people's fame, this result testified Fehr's conclusion that punishment can maintain conditional cooperation in Public Goods Experiment.

Result3: *The information displaying condition which shows the instant number of people who have endowed can stimulate individual's trust to others and generate higher probability of endowment.*

We can see from Fig. 8 that the average probability of endowment in Testnum environment is 0.4, this environment can stimulate conditional cooperation and trusts of the "Social Man" in the group. Also the probability of endowment is 0.4 which is not very high shows that the cooperation can't maintain high forever. Once there is treachery in the course of the n rounds gaming, "Social Man" will change their attitudes accordingly.

Result4: *Comparing to the effect of number, the displaying of instant number and name of people who have chosen to endow will cause the highest rate of conditional cooperation.*

From Fig. 9, we can see that while displaying the instant name and number of people who have chosen to endow at this round has the highest average probability of endowment. Since in a certain group, there are a fixed number of people who are the called "Social Man". Once others knew their name, they would like to follow the choice of these "Social Man" in the course of decision-making. Also the displaying of name can be seen as a reward to virtue and fame which will also stimulate the conditional cooperation.

Result 5: *Positive incentives can stimulate conditional cooperation more easily and generate higher amount of Public Goods than punishments and negative impacts.*

From the Fig. 8 and Fig. 9, the average probability of endowment of Testnoname is 0.4, while the Testname environment is 0.6. Therefore, we concluded that the

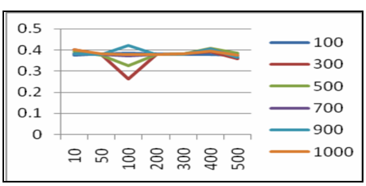


Fig. 8. Endowing probability of Testnum

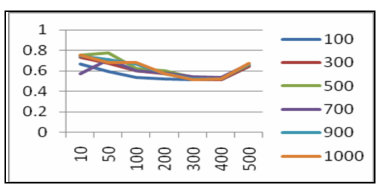


Fig. 9. Endowing probability of Testname

punishment to fame can not maintain the cooperation as effective as positive incentives.

7 Conclusion

After we have finished all the 3 parts of this paper, we found that different information displaying condition will cause remarkable differences to endowment probability and the rate of conditional cooperation. We found that people are more likely to cooperate while giving the name and number of people who have already chosen cooperation during the process of decision-making. At this circumstance, there are least free riders and the amount of public goods is the highest. The producer of public goods can use this theoretical discovery while producing and distributing public goods among the public.

References

1. Varian, H.R.: Microeconomic Analysis, pp. 415–431. W. W. Norton & Company (1992)
2. wiki cyclopedia, `http://en.wikipedia.org/wiki/Public_good`
3. Fischbacher, U., GÄachter, S., Fehr, E.: Are People Conditionally Cooperative? Evidence from a Public Goods Experiment. Economics Letters 71(3), 397–404 (2001)
4. Fehr, E., Gachter, S.: Cooperation and Punishment in Public Goods Experiments. American Economic Review 90, 980–994 (2000)
5. Axelrod, R., Ford, G.R.: Beyond Geography Cooperation with Persistent Links in the Absence of Clustered Neighborhoods. Pers. Soc. Psychol. 6, 341 (2002)
6. Chen, Y.-S.: Evolution of prisoner's dilemma strategies on scale-free networks (2006)
7. Wu, F., Sihai, Z.: Research on Social Norms in Public Goods Experiments (2006)
8. Epstein, J.M., Axtell, R.: Growing artificial societies. MIT Press Brookings, Cambridge (1997)
9. Shoam, Y.: Agent-Oriented Programming. Artificial Intelligence 60, 51–292 (1993)
10. Axelrod, R.: The evolution of cooperation. Basic Books, New York (1984)
11. Refenes, A.N., Zapranis, A., Francis, G.: Stock Performance Modeling Using Neural Networks: A Comparative Study with Regression Models. Neural Networks 7(2), 355–388 (1994)

Particle Swarm Optimization Using the Decoding Algorithm for Nonlinear 0-1 Programming Problems

Takeshi Matsui, Masatoshi Sakawa, and Kosuke Kato

Department of Artificial Complex Systems Engineering,
Graduate School of Engineering, Hiroshima University, Japan
{tak-matsui,sakawa,kosuke-kato}@hiroshima-u.ac.jp
http://www.sol.hiroshima-u.ac.jp/

Abstract. In general, actual various decision making situations are formulated as large scale mathematical programming problems with many decision variables and constraints. For programming problems that decision variables take value 0 or 1 in mathematical programming problems, we can get strict solution by the application of dynamic programming fundamentally. However, a number of the solution that we should search becomes increases by leaps and bounds as the scale of the problem becomes large. In particular, for nonlinear 0-1 programming problems, there are not general strict solution method or approximate solution method, such as branch and bound method in case of linear 0-1 programming problems. In this research, focusing on nonlinear 0-1 programming problems, we propose an approximate solution method based on particle swarm optimization proposed by Kennedy et al. To be more specific, we develop a new particle swarm optimization method which is applicable to discrete optimization problems by incorporating the decoding algorithm.

Keywords: decoding, nonlinear 0-1 programming, particle swarm optimization, swarm intelligence.

1 Introduction

In general, actual various decision making situations are formulated as large scale mathematical programming problems with many decision variables and constraints. In mathematical programming problems, for the programming problems that decision variables are 0 or 1, we can theoretically get strict solution by application of dynamic programming. In particular, for nonlinear 0-1 programming problems, there are not general strict solution method or approximate solution method, such as branch and bound method in case of linear 0-1 programming problems. In such a case, a solution method depended on property in problems is proposed. Thus, Sakawa [7] proposed genetic algorithms with double string representation based on updating basepoint solution using continuous relaxed as general approximate solution method for nonlinear 0-1 programming problems. In recent years, a particle swarm optimization (PSO) method was

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 670–676, 2009.

proposed by Kennedy et al. [2] and has attracted considerable attention as one of promising optimization methods with higher speed and higher accuracy tha those of existing solution methods. Hence, Kato et al. [1] proposed revised particle swarm optimization (rPSO) method solving drawbacks that it is not directly applicable to constrained problems and its liable to stopping around local solutions and showed its effectivity. Moreover we expanded revised particle swarm optimization method for application to nonlinear integer programming problems (rPSONLIP) [4] and showed more efficiency than genetic algorithm. And we expanded rPSO for application to nonlinear 0-1 programming problems (rPSONLZOP [5], rPSOmNLZOP [6]) and showed more efficiency than genetic algorithm and QUADZOP. However, in those researches, rPSONLZOP and rPSOmNLZOP method needs many computational time as same as genetic algorithm. In this research, we focus on nonlinear 0-1 programming problems and propose higher approximation solution method based on particle swarm optimization using the decoding algorithm than those methods.

2 Nonlinear 0-1 Programming Problems

In this research, we consider general nonlinear programming problem with constraints as follows:

$$\left.\begin{array}{rl} \text{minimize} & f(\boldsymbol{x}) = \boldsymbol{c}\boldsymbol{x} \\ \text{subject to} & g(\boldsymbol{x}) = A\boldsymbol{x} - \boldsymbol{b} \leq 0 \\ & \boldsymbol{x} \in \{0,1\}^n \end{array}\right\} \tag{1}$$

where $\boldsymbol{c} = (c_1, \ldots, c_n)$ is an n-dimensional row vector; $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ is an n-dimensional column vector of 0-1 decision variables; $A = [a_{ij}]$, $i = 1, \ldots, m$, $j = 1, \ldots, n$, is an $m \times n$ coefficient matrix; and $\boldsymbol{b} = (b_1, \ldots, b_m)^T$ is an m-dimensional column vector.

3 Particle Swarm Optimization

Particle swarm optimization [2] method is based on the social behavior that a population of individuals adapts to its environment by returning to promising regions that were previously discovered [3]. This adaptation to the environment is a stochastic process that depends on both the memory of each individual, called particle, and the knowledge gained by the population, called swarm.

In the numerical implementation of this simplified social model, each particle has three attributes: the position vector in the search space, the current direction vector, the best position in its track and the best position of the swarm. The process can be outlined as follows.

Step 1: Generate the initial swarm involving N particles at random.
Step 2: Calculate the new direction vector for each particle based on its attributes.
Step 3: Calculate the new search position of each particle from the current search position and its new direction vector.

Step 4: If the termination condition is satisfied, stop. Otherwise, go to Step 2.
To be more specific, the new direction vector of the i-th particle at time t, $\boldsymbol{v}_i^{t+1}$, is calculated by the following scheme introduced by Shi and Eberhart [8].

$$\boldsymbol{v}_i^{t+1} := \omega^t \boldsymbol{v}_i^t + c_1 R_1^t(\boldsymbol{p}_i^t - \boldsymbol{x}_i^t) + c_2 R_2^t(\boldsymbol{p}_g^t - \boldsymbol{x}_i^t) \tag{2}$$

In eq.(2), R_1^t and R_2^t are random numbers between 0 and 1, $\boldsymbol{p}_i^t$ is the best position of the i-th particle in its track at time t and $\boldsymbol{p}_g^t$ is the best position of the swarm at time t. There are three parameters such as the inertia of the particle ω^t, and two parameters c_1, c_2.

Then, the new position of the i-th particle at time t, $\boldsymbol{x}_i^{t+1}$, is calculated from eq.(3).

$$\boldsymbol{x}_i^{t+1} := \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1} \tag{3}$$

where $\boldsymbol{x}_i^t$ is the current position of the i-th particle at time t. After the i-th particle calculates the next search direction vector $\boldsymbol{v}_i^{t+1}$ by eq.(2) in consideration of the current search direction vector $\boldsymbol{v}_i^t$, the direction vector going from the current search position $\boldsymbol{x}_i^t$ to the best search position in its track $\boldsymbol{p}_i^t$ and the direction vector going from the current search position $\boldsymbol{x}_i^t$ to the best search position of the swarm $\boldsymbol{p}_g^t$, it moves from the current position $\boldsymbol{x}_i^t$ to the next search position $\boldsymbol{x}_i^{t+1}$ calculated by eq.(3). In general, the parameter ω^t is set to large values in the early stage for global search, while it is set to small values in the late stage for local search. For example, it is determined as:

$$\omega^t := \omega^0 - \frac{t \cdot (\omega^0 - \omega^{T_{\max}})}{0.75 \cdot T_{\max}} \tag{4}$$

where t is the current time, $T_{\max}$ is the maximal value of time, ω^0 is the initial value of ω^t and $\omega^{T_{\max}}$ is the final value of ω^t.

The search procedure of PSO is shown in Fig. 1. If the next search position of the i-th particle at time t, $\boldsymbol{x}_i^{t+1}$, is better than the best search position in its track at time t, $\boldsymbol{p}_i^t$, i.e., $f(\boldsymbol{x}_i^{t+1}) \leq f(\boldsymbol{p}_i^t)$, the best search position in its track is updated as $\boldsymbol{p}_i^{t+1} := \boldsymbol{x}_i^{t+1}$. Otherwise, it is updated as $\boldsymbol{p}_i^{t+1} := \boldsymbol{p}_i^t$. Similarly, if $\boldsymbol{p}_i^{t+1}$ is better than the best position of the swarm, $\boldsymbol{p}_g^t$, i.e., $f(\boldsymbol{p}_i^{t+1}) \leq f(\boldsymbol{p}_g^t)$, then the best search position of the swarm is updated as $\boldsymbol{p}_g^{t+1} := \boldsymbol{p}_i^{t+1}$. Otherwise, it is updated as $\boldsymbol{p}_g^{t+1} := \boldsymbol{p}_g^t$.

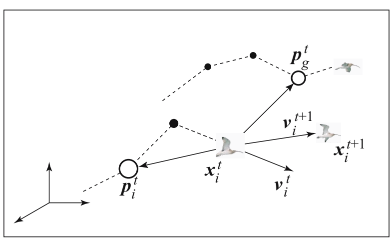


Fig. 1. Movement of an individual

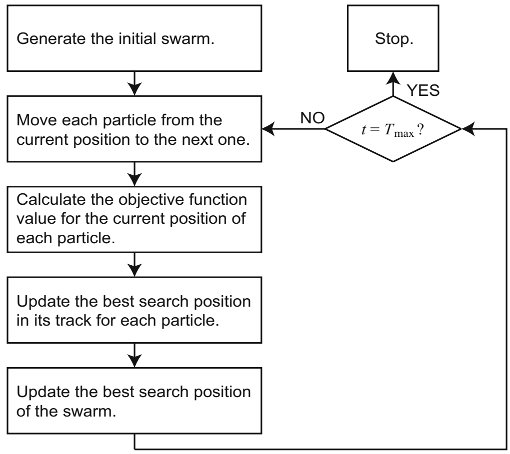


Fig. 2. The basic algorithm of PSO

Such the PSO method has two drawbacks. One is that particles stop at the best search position of the swarm and they cannot easily escape from the local solution since the move direction vector $\boldsymbol{v}_i^{t+1}$ calculated by eq.(2) always includes the direction vector to the best search position of the swarm. The other is that the next search position of a particle is not always feasible for constrained problems.

Thus, Kato et al. [1] proposed revised particle swarm optimization method solving drawbacks that it is not directly applicable to constrained problems and its liable to stopping around local solutions and showed its effectivity. Moreover we expanded revised particle swarm optimization method for application to nonlinear integer programming problems (rPSONLIP) and showed more efficiency than genetic algorithm [4]. Furthermore we expanded the PSO method for nonlinear 0-1 programming problems (rPSONLZOP) and showed more efficiency incorporating new move scheme [6]. However the rPSONLZOP method needs many computational time than the rPSONLIP method. Thus, we revised the rPSONLZOP method incorporating memory structures [5]. However, in that research, we showed reducing computational time but turned worse on accuracy. In this research, we propose high speed and efficiency particle swarm optimization method using decoding for nonlinear 0-1 programming problems.

4 Decoding Algorithm Using a Reference Solution with Backtracking and Individual Modification

In this research, we incorporate the decoding algorithm proposed by Sakawa [7]. In the algorithm, n, j, $s(j)$, $g_{s(j)}$, $x_{s(j)}$ and $\boldsymbol{L}_{s(j)}$ denote length of a string, a position in a string, an index of a varibale, the value of the element, 0-1 value of a variable with index $s(j)$ decoded from a string, and a $s(j)$-th column vector of the coefficient matrix A. By using the feasible solution $\boldsymbol{x}^0$ thus obtained, the decoding algorithm using a reference solution with backtracking and individual modification can be described as follows:

Step 1: Set $j = 1$, $l = 0$ and sum $= 0$.
Step 2: If $g_{s(j)} = 0$, set $j := j + 1$ and go to Step 4. If $g_{s(j)} = 1$, let sum = sum + $\boldsymbol{L}_{s(j)}$ and go to Step 3.
Step 3: If sum $\leq \boldsymbol{b}$, let $l := j$ and $j := j + 1$, and go to Step 4. Otherwise, let $j := j + 1$ and go to Step 4.
Step 4: If $j > n$, go to Step 5. Otherwise, return to Step 2.
Step 5: If $l > 0$, let $x^*_{s(j)} := g_{s(j)}$ for all j such that $1 \leq j \leq l$ and $x^*_{s(j)} := 0$ for all j such that $l + 1 \leq j \leq n$, and go to Step 6. Otherwise, let $\boldsymbol{x}^* := \boldsymbol{x}^0$ and go to Step 6.
Step 6: Let $j := 1$ and sum $:= \sum_{k=1}^{n} \boldsymbol{L}_{s(k)} x^*_{s(k)}$.
Step 7: If $g_{s(j)} = x^*_{s(j)}$, let $x_{s(j)} := g_{s(j)}$ and $j := j + 1$, and go to Step 9. If $g_{s(j)} \neq x^*_{s(j)}$, then go to Step 8.
Step 8: 1) If $g_{s(j)} = 1$ and sum + $\boldsymbol{L}_{s(j)} \leq \boldsymbol{b}$, let $x_{s(j)} := 1$, sum := sum + $\boldsymbol{L}_{s(j)}$ and $j := j + 1$. Here, if there exists at least one negative element in $\boldsymbol{L}_{s(j)}$, then go to Substep 1 for backtracking and individual modification. If not, go to Step 9. If $g_{s(j)} = 1$ and sum + $\boldsymbol{L}_{s(j)} > \boldsymbol{b}$, let $x_{s(j)} := 2$ and $j := j + 1$, and go to Step 9. 2) If $g_{s(j)} = 0$ and sum - $\boldsymbol{L}_{s(j)} \leq \boldsymbol{b}$, let $x_{s(j)} := 0$, sum := sum - $\boldsymbol{L}_{s(j)}$ and $j := j + 1$. Here, if there exists at least one positive element in $\boldsymbol{L}_{s(j)}$, then go to Substep 1 for backtracking and individual modification. If not, go to Step 9. If $g_{s(j)} = 0$ and sum - $\boldsymbol{L}_{s(j)} > \boldsymbol{b}$, let $x_{s(j)} := 2$ and $j := j + 1$, and go to Step 9.

<u>Substeps for backtracking and individual modification</u>

Substep 1: Set $h := 1$.
Substep 2: If $x_{s(h)} = 2$, go to Substep 3. Otherwise, let $h := h + 1$ and go to Substep 4.
Substep 3: 1) If $g_{s(j)} = 1$ and sum + $\boldsymbol{L}_{s(h)} \leq \boldsymbol{b}$, let $x_{s(h)} := 1$, sum := sum + $\boldsymbol{L}_{s(h)}$ and $h := h + 1$. Then, interchange $(s(j), g_{s(j)})^T$ with $(s(h), g_{s(h)})^T$. If there exists at least one negative element in $\boldsymbol{L}_{s(h)}$, then return to Substep 1. If not, go to Substep 4. 2) If $g_{s(h)} = 0$ and sum - $\boldsymbol{L}_{s(h)} \leq \boldsymbol{b}$, let $x_{s(h)} := 0$, sum := sum - $\boldsymbol{L}_{s(h)}$ and $h := h + 1$. Then, interchange $(s(j), g_{s(j)})^T$ with $(s(h), g_{s(h)})^T$. If there exists at least one positive element in $\boldsymbol{L}_{s(h)}$, then return to Substep 1. If not, go to Substep 4. If $g_{s(h)} = 0$ and sum - $\boldsymbol{L}_{s(h)} > \boldsymbol{b}$, let $h := h + 1$ and go to Substep 4.
Substep 4: If $h \geq j$, then go to Step 9. Otherwise, return to Substep 2.
Step 9: If $j > n$, let $h := 1$ and go to Step 10. Otherwise, return to Step 7.
Step 10: If $x_{s(h)} = 2$, let $x_{s(h)} := x^*_{s(h)}$ and $h := h + 1$, and go to Step 11. Otherwise, let $h := h + 1$ and go to Step 11.
Step 11: If $h > n$, stop. Otherwise, return to Step 10.

5 The Procedure of Revised PSO Using the Decoding Algorithm

The procedure of the revised PSO proposed in this research summarized as follows and shown in Fig. 3.

Step 1: Generate feasible initial search positions and decode each particle mentioned in Section 4. In addition, let the initial search position of each particle, $\boldsymbol{x}_i^0$, be the initial best position of the particle in its track, $\boldsymbol{p}_i^0$, and let the best position among $\boldsymbol{x}_i^0$, $i = 1, \ldots, N$ be the initial best position of the swarm, $\boldsymbol{p}_g^0$. Go to Step 2.
Step 2: Calculate the value of ω^t by eq.(4). For each particle, using the information of $\boldsymbol{p}_i^t$ and $\boldsymbol{p}_g^t$, determine the direction vector $\boldsymbol{v}_i^{t+1}$ to the next search position $\boldsymbol{x}_i^{t+1}$ by the modified move schemes. Next, move it to the next search position by eq.(3) and go to Step 3.
Step 3: If the evaluation function value $f(\boldsymbol{x}_i^{t+1})$ is better than the evaluation function value for the best search position of the particle in its track, $\boldsymbol{p}_i^t$, update

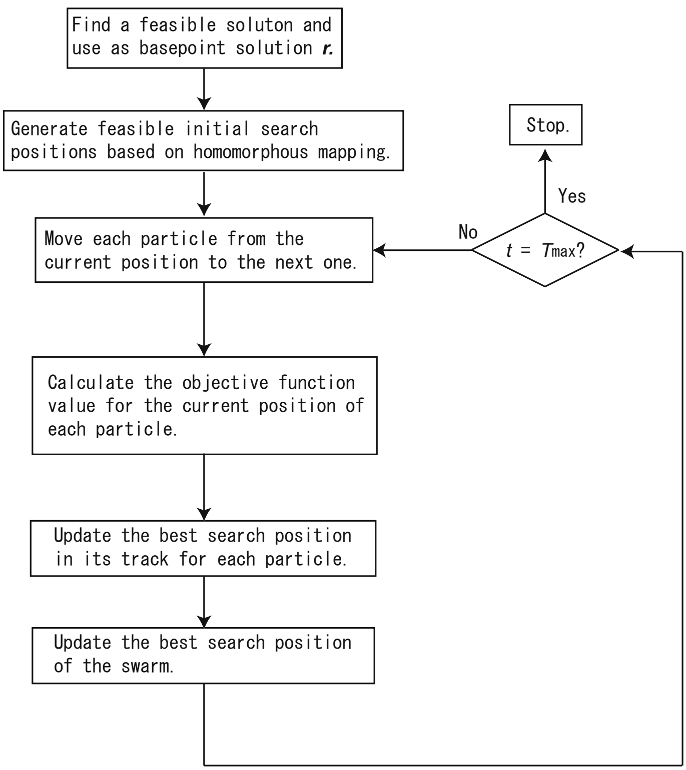


Fig. 3. The algorithm of proposed PSO

the best search position of the particle in its track as $\boldsymbol{p}_i^{t+1} := \boldsymbol{x}_i^{t+1}$. If not, let $\boldsymbol{p}_i^{t+1} := \boldsymbol{p}_i^t$ and go to Step 4.
Step 4: If the minimum of $f(\boldsymbol{x}_i^{t+1})$, $i = 1, \ldots, N$ is better than the evaluation function value for the current best search position of the swarm, $\boldsymbol{p}_g^t$, update the best search position of the swarm as $\boldsymbol{p}_g^{t+1} := \boldsymbol{x}_{i_{\min}}^{t+1}$. Otherwise, let $\boldsymbol{p}_g^{t+1} := \boldsymbol{p}_g^{t+1}$ and go to Step 5.
Step 5: Finish if $t = T_{\max}$ (the maximal value of time). Otherwise, let $t := t + 1$ and return to Step 2.

6 Conclusion

In this research, focusing on nonlinear 0-1 programming problems, we proposed new particle swarm optimization using genetic operators. In near future, we will report the result of the application of the proposed method to numerical examples.

References

1. Kato, K., Matsui, T., Sakawa, M., Morihara, K.: An approximate solution method based on particle swarm optimization for nonlinear programming problems. Journal of Japan Society for Fuzzy Theory and Intelligent Informatics 20(3), 399–409 (2008) (in Japanese)
2. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
3. Kennedy, J., Spears, W.M.: Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. In: IEEE Int. Conf. Evolutionary Computation (1998)
4. Matsui, T., Kato, K., Sakawa, M., Uno, T., Matsumoto, K.: Particle Swarm Optimization for Nonlinear Integer Programming Problems. In: International MultiConference of Engineers and Computer Scientists 2008, pp. 1874–1877 (2008)
5. Matsui, T., Sakawa, M., Kato, K.: Particle Swarm Optmization using Memory Structures for Nonlinear 0-1 Programming Problems. In: The 11th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty, pp. 57–62 (2008)
6. Matsui, T., Sakawa, M., Kato, K., Uno, T.: Particle Swarm Optmization for Nonlinear 0-1 Programming Problems. In: The IEEE International Conference on Systems, Man, and Cybernetics 2008, pp. 168–173 (2008)
7. Sakawa, M.: Genetic Algorithms and Fuzzy Multiobjective Optimization. Kluwer Academic Publishers, Dordrecht (2001)
8. Shi, Y.H., Eberhart, R.C.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation, pp. 69–73 (1998)

Agent Based Decision Support in Manufacturing Supply Chain

Per Hilletofth[1], Lauri Lättilä[2], and Olli-Pekka Hilmola[2]

[1] School of Technology and Society, University of Skövde, P.O. Box 408, SE-541 28 Skövde, Sweden, Fax: +46 500 44 87 99
per.hilletofth@his.se
[2] Lappeenranta University of Technology, Kouvola Research Unit, Prikaatintie 9, FIN-45100 Kouvola, Finland
lauri.lattila@lut.fi, olli-pekka.hilmola@lut.fi

Abstract. Supply Chain Management (SCM) is becoming increasingly complex and an intensified competition in the end-markets has started to create a situation where co-operation requirements between companies in a Supply Chain (SC) are increasing. The old mechanistic operations management solutions are becoming obsolete and advanced decision support is increasingly needed to realize efficient and effective management of complex SCs. The objective of this research is to contribute to the understanding of how Agent Based Modeling (ABM) can advance decision making and to discuss why ABM should be regarded as method to realize Information Fusion (IF). In this research work an agent based model of SCM has been implemented in a simulation platform to provide an approach for evaluation of decision and management alternatives. Research shows that this kind of decision support system is based on IF, since it collects and fuses information from different sources into a situation image that provides effective support for human decision making

Keywords: Supply chains, multi-agent systems, information fusion, simulation.

1 Introduction

Supply Chain Management (SCM) can be defined as a set of approaches utilized to efficiently integrate and coordinate the materials, information and financial flows across the Supply Chain (SC), so that merchandise is supplied, produced and distributed at the right quantities, to the right locations, and at the right time, in the most cost-efficient way, while satisfying customer requirements [1]. SCM is becoming increasingly complex due to several developments in the marketplace [2-3]. In addition the difficulties in controlling and coordinating logistics operations within and among firms (i.e. across the SC) are expected to increase, since the interdependence among cooperating firms is intensifying [4]. There are several factors contributing to an increased complexity of SCM, for instance, SCs nowadays need to satisfy more demanding customers and operate on international markets [5]. In order to realize efficient and effective management of complex SCs in this environment, managers, or

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 677–686, 2009.

decision makers, continuously need correct and updated information from different sources. They also need to be able to predict the outcome of their decisions, and how their decisions affect the whole SC. To realize this advanced decision support is needed.

One way to improve decision making is to generate business intelligence by fusing large amounts of data from various sources, i.e. Information Fusion (IF). The purpose of IF is typically to extract relevant information from several sources with known certainty to make better decisions than if fusion was not used. It can de defined as "the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making" [6]. One method to realize IF in complex SCs, which normally is not highlighted as an IF method, is Agent Based Modeling (ABM). It is related to IF in the way that information from different sources are collected and fused in a synergistic manner into a representation that provides effective support for human decision making. Empirical studies have shown that managers aided by agent based models can benefit in several ways [7]. For instance, agent based models can help managers to find the highest leverage among improvement alternatives, guide managers instinct and increase managers understanding of the impact of unscheduled factors. ABM is also expected to have comprehensive effects on the way that businesses use computers to support decision making.

In this research work an agent based model of SCM has been implemented in a simulation platform to provide a pragmatic approach for evaluation of decision and management alternatives. The agent model has been developed and described in an earlier paper [8]. The agent model was based on a case company, Alpha, a Swedish manufacturer operating on international basis in the white goods industry. In the simulation model, the SC processes are managed by a set of agents that are responsible for one or more activities; the simulation model is inspired by an actual case-study but additional data has been used to allow the simulation model to be developed. The model has been implemented through a leading agent based simulation software called Anylogic.

The overall purpose of this research is to contribute to the understanding of how ABM can advance decision making by an increased understanding of the problem domain and also generate a basis for decision making, thus improving the performance of the SC. Moreover, this paper aims to discuss why ABM should be regarded as method to realize IF by investigating where IF could be incorporated in agent based systems. The remainder of this paper is structured as follows: In Section 2, the concept of ABM is presented and discussed through existing literature. Thereafter, in Section 3 we introduce our agent based simulation model, which is based partially in our earlier case study results from the white goods manufacturer. Modeling logic in agent based systems is more complex, but scalability (e.g. adding more products, suppliers, and wholesalers etc.) is easier to complete. We illustrate this with the initial simulation results from two products with different demand characteristics through delivery capability as well as inventory development implications. In the final Section 4 we conclude our research work and propose avenues for further research.

2 Agent Based Modeling

ABM is a new modeling paradigm especially suited for complex and dynamic systems distributed in time and space [9]. It is expected to have comprehensive effects on the way that businesses use computers to support decision making. For instance, it provides a pragmatic approach for the evaluation of management alternatives [10]. Moreover, it is considered important for developing industrial applications in complex environments [11]. ABM implies that a real (observed) system is modeled in form of a couple of interacting agents within a certain environment (Fig. 1). The agent model is later implemented through different platforms, e.g. simulation or system development. This implies that an agent system consists of a couple of individual agents with specified relationships to one another within a certain environment. The agents are presumed to be acting in what they perceive as their own interests, such as economic benefit (i.e. they have individual missions), and their knowledge regarding the entire system (i.e. other agents and environment) is limited [12]. Still, the most important feature in an agent system is the agents' ability to collaborate, coordinate and interact with each other as well as with the environment to achieve common goals. By sharing information, knowledge, and tasks among the agents in the system, collective intelligence may emerge that can not be derived from the internal mechanism of an individual agent. Furthermore, the ability to coordinate makes it possible for agents to coordinate their actions among themselves, i.e. taking the effect of another agent's actions into account when making a decision about what to do. The term Multi-Agent System (MAS) is commonly used for agent systems including several interacting and collaborating agents.

An agent based model implemented through simulation can be used to simulate the actions and interactions of autonomous agents in an agent system to evaluate the agents' effects on the system as a whole as well as to evaluate the system in general (Fig. 2). The implemented model basically consists of the interacting agents and some performance and risk indicators. The utilized data in the model can be collected from databases or from sensors within the real system. Managers, or decision makers, can set parameters in the simulation model, run the simulation and evaluate the results. Based on the retrieved information/knowledge they can make decisions regarding how to handle the real system. They could also continually alter different parameters

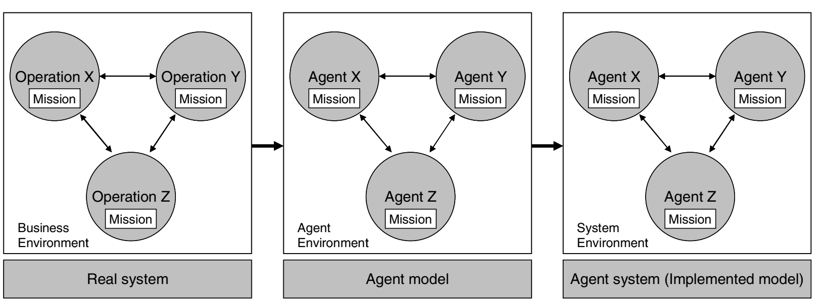


Fig. 1. The process of agent based modeling

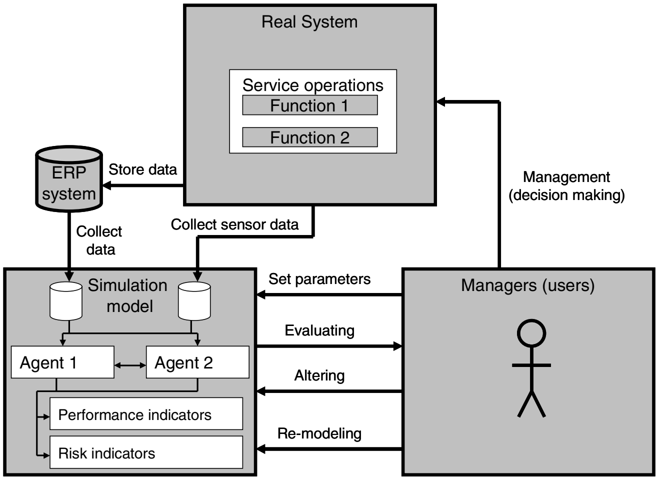


Fig. 2. Agent based decision support system

and simulate again to evaluate different management alternative, or perhaps initiated a re-modeling of the real system. This implies that ABM basically collects and fuses information from different sources in a synergistic manner into a representation that provides effective support for human decision making. Therefore, it could be regarded as an IF method.

Nilsson and Darley [7] conclude in their empirical study that managers aided by agent based models and simulations can benefit in several ways. Firstly, they acquire an increased understanding of the impact of unscheduled factors such as breakdowns, accidents and changes of demands. Such factors are often found in reality, but usually reduced or even ignored when transferred to most traditional models. This implies that the optimized solutions from these traditional models mislead managers into believing in future scenarios, which do not reflect reality. Secondly, ABM based simulations can guide managers instinct, since interactive agents generate an emergent pattern, which can be explained and understood and therefore beneficial for the improvement of decision making in companies. Thirdly, ABM can help managers to find where the highest leverage is to be gained among improvement alternatives. This is based on the fact that ABM allows models to encompass several business functions and how they affect each other. Finally, there are sometimes even opportunities to improve predictability based on the scenarios generated.

There is a growing interest in using ABM in several business related areas, such as manufacturing, maintenance, and SCM [8]. Macal and North [12] have discussed a couple of reasons to the increased utilization of ABM. Firstly, the observed systems are becoming more complex in terms of their interdependencies which imply that traditional approaches no longer are as applicable as they once were. Secondly, some systems have earlier been too complex to model, but now possible to model through ABM. Thirdly, data are becoming organized into databases at finer levels of granularity. Micro-data can now support micro simulations. Finally, but most importantly, computational power is advancing rapidly. Even if the interest in implementing ABM in varies types of business is increasing, is it currently a quite conceptual concept. Based on literature reviews Davidsson et al. [13] and Cantamessa [14] conclude that very few field experiments and developed systems can be reported to be found in the academic literature. Davidsson et al. [13] reviewed the maturity of agent approaches presented in the literature and used following four main levels: (i) conceptual proposal, (ii) simulation experiment, (iii) field experiment, and (iv) deployed system. In their sample of 56 journal articles published between 1992 and 2005, it was only identified one level 4 and three level 3 research works. A more recent literature review confirms that this situation still exists. In Hilletofth et al. [8] sample of 33 journal articles published between 2000 and mid 2008 only one paper included empirically verified results after the implementation of ABM. Furthermore, the literature review showed that very few agent based frameworks were developed based on empirical information and most of the articles only included developed prototype systems.

3 Simulation Model and Initial Results

In this section the structure of the developed simulation model is described as well as some initial results.

3.1 Structure of the Simulation Model

The current simulation model contains five different kinds of agents: wholesalers, order management, production planning, requirements planning and suppliers. The wholesalers communicate only with order management. Order management also communicates with production planning, while production planning can further communicate with requirements planning. The requirements planning agent is connected to the final agents, the suppliers. This current logic is shown in Fig. 3.

In the simulation model only two products is modeled. One of the products has a cyclical demand while the other one has a life-cycle demand. By life-cycle demand we mean that the product will have an increasing amount of sales before the markets become saturated. After this point the demand for the product will start to slowly decrease.

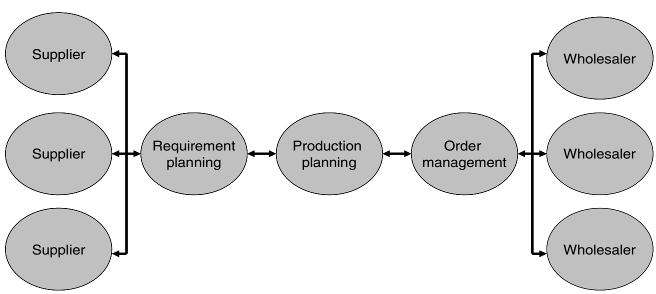


Fig. 3. Agent connectedness in the simulation model

Wholesalers create a 12 week long forecast with the help of exponential smoothing. This forecast is then used with the item inventory to create a delivery plan. The wholesalers want to have a safety stock of 100 units and use a delivery batch size of 20 units. As soon as the order management agent has received all of the delivery plans from the wholesalers the agent checks the amount of end item inventory at the plant and aggregates the total demand for the production planning agent. After these procedures the information is sent to production planning agent who can then start to work on the Material Requirements Planning. The order management agent also sends the aggregated demand to the suppliers, so they can more easily manage their own raw material purchases. Production has a two week long lead time, which is also used in the MRP calculations. MRP has a production batch size of thirty units and the plant does not want to hold any excess end item inventory. During the MRP run a rough cut capacity calculation is also done to ensure that sufficient capacity exists for the plan. When the production planning agent finishes the MRP calculations, it sends the current production plan to requirements planning. Requirement planning checks the amount of raw materials at the manufacturing unit. The lead time for all of the raw materials is four weeks, the manufacturing unit wants to have a safety stock of 1000 units and the order batch is 500 units. With this information it is possible to do mrp calculations and requirements planning agent modifies the production plan according to the raw material availability. When the requirements planning agent finishes the mrp calculation, it will have the raw material orders, which are sent to the suppliers. Requirements planning finishes the production plan, which is sent to order management in order to create the confirmed deliveries for the wholesalers. As the suppliers have access to the end item forecast, they use this information in their own MRP calculations. The suppliers have a three week long lead-time with their own suppliers and they want to have safety stock of 15000 products in raw materials and the materials are ordered with a 5000 unit batch size. The suppliers send their own purchases to their suppliers, but they have not been modeled, so it is not shown in the main view.

There is an actual case behind the simulation model but some of the data has not been gathered from the case-company. The bill-of-material and batch-sizing rules (and their respective suppliers) have been made-up as it was not available for the researchers when the simulation model was constructed. However, data regarding delays corresponds to actual delays and the company also uses 12 week long forecasts, which are used in this simulation model.

As highlighted above each one of the agents performs different sorts of tasks. This implies that every agent in the model has its own internal mechanisms and also a specific mission. When the agents collaborate, coordinate and interact with each other a collective intelligence may emerges that can not be derived from the internal mechanism of an individual agent. This implies that IF occurs at several places in the model. Firstly, it takes place within an agent to decide actions in order to achieve the agent specific goals. However, it also takes place on a holistic level when the agents generates a situation image by communicate and interact with each other to achieve system overall goal. Each agent’s specific tasks and information usage is presented in Table 1, while all of the communicated information is presented in Table 2.

Table 1. Agent framework environment description

Agent	Tasks	Information (source)
Wholesaler	Receive Orders Create/Edit Forecast Create/Edit Delivery Plan	Demand (system generated) Previous Demand (database/variable) Safety Stock Size (database/variable) Delivery Batch Size (database/variable) Confirmed Delivers (order management)
Order management	Create/Edit Aggregated Delivery Plan Confirm Deliveries	Delivery Plans (wholesalers) Finished Inventory (database/variable) Confirmed Production Plan (requirement planning)
Production planning	Create/Edit Production Plan Perform Capacity Planning	Aggregated Delivery Plan (order management) Lead Time (database/variable) Batch Size (database/variable)
Requirement planning	Create/Edit Purchase Orders Confirm Production Plan	Production Plan (production planning) Raw Material Inventory (database/variable) Bill Of Materials (database/variable) Lead Time (database/variable) Batch Size (database/variable) Safety Stock Size (database/variable)
Supplier	Create/Edit Production Plan Create/Edit Purchase Orders Confirm Purchase Orders	Aggregated Delivery Plan (order management) Lead Time (database/variable) Batch Size (database/variable) Safety Stock Size (database/variable)

Table 2. Information flows between agents in the simulation model

Sender	Receiver	Information
Wholesaler	Order Management	Delivery plan
Order management	Wholesaler	Confirmed deliveries
Order management	Supplier	Aggregated delivery plan
Order management	Production planning	Aggregated delivery plan
Order management	Production planning	Finished inventory
Production planning	Requirements planning	Production plan
Requirements planning	Order management	Confirmed production plan
Requirements planning	Suppliers	Purchase orders

As a lot of information is communicated between the agents, there is a need to have a numerous variables to store this information in different agents. Most of the information in the model is stored in matrixes (e.g. multiple products, agents, and time). The information is communicated between the agents using ports and these ports can deliver different kinds of information (e.g. matrixes, individual values, text).

3.2 Results

Fig. 4a shows the sales generated by the wholesalers. It also shows the final forecasts and current forecasts. During the simulation it is possible to show how the forecast changes during the simulation. As there is sufficient capacity in the manufacturing system, the whole SC can relatively well meet the market's demand. From Fig. 4a it is possible to notice how the whole SC has difficulties when the demand for the cyclical product starts to rise again. This is due to the lags in the SC. As the forecast is lower than the actual demand, there is insufficient amount of raw materials in the factory, when the up-turn starts again. The sales for the life-cycle product are also impacted as many raw materials are used by both products. As the manufacturing unit cannot cope

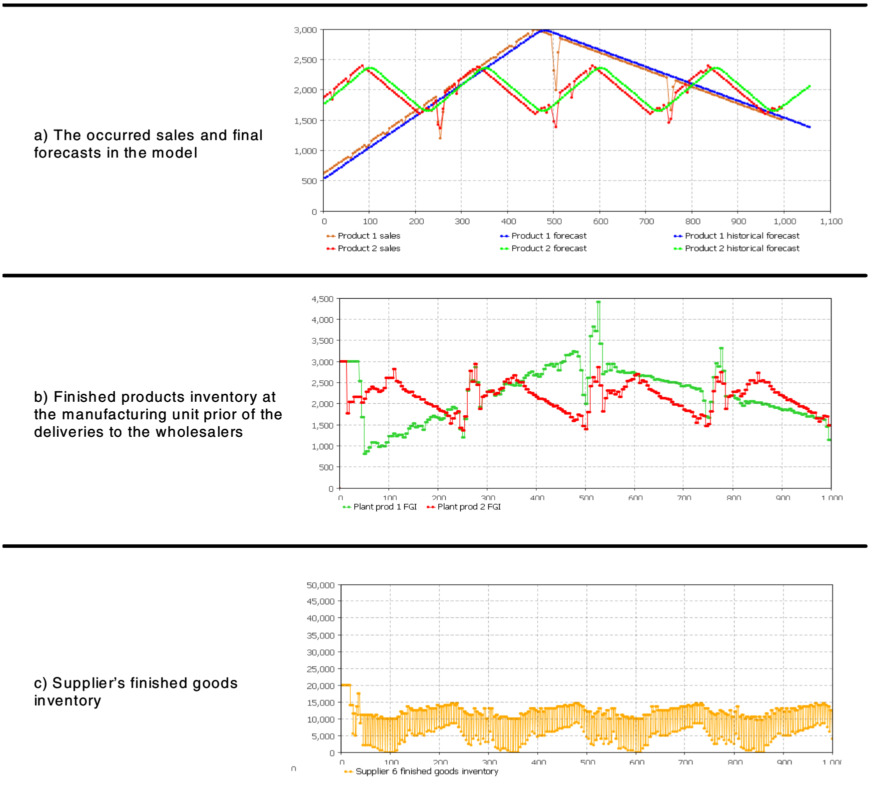


Fig. 4. Statistics from simulation analysis

with the demand, there is going to be a huge backlog at the plant. As soon as the manufacturing unit receives a sufficient amount of raw materials, it can manufacture the backlogged units. This can be seen from Fig. 4b.

Fig. 4b shows the finished product inventory at the manufacturing unit before the deliveries to the wholesalers. The amount of finished product inventory clearly shows what kind of demand is behind both products. The manufacturing overshoots, when the demand for the cyclical product changes and the reaction is very intensive for the life-cycle product during its peak demand. The overreactions are similar to the Forrester effect [15], even though information is shared very openly in the 4c. Fig. 4c shows the finished goods inventory at the suppliers.

The suppliers cannot reach their desired finished goods inventory level of 20000 and it fluctuates very heavily. Even though the order batch size for the manufacturing unit is 500 units, the average purchase is a lot larger. The supplier in Fig. 4c is supplying parts, which are only used in the life-cycle product, but the supplier's finished goods inventory clearly reacts to the cyclical product's demand: they seem to have a negative correlation. As the demand for the cyclical product is low, the end item inventory is a lot higher at the supplier. This indicates that the whole supply-network reacts as a whole to changes in one part of the chain.

4 Discussion and Conclusions

To realize efficient and effective management of complex SCs, decision makers, continuously require correct and up to date information. The possibility to predict the outcome of their decisions is beneficial, also what effect these decisions will have on the SC operations. This research shows that an effective decision support system can be developed through Agent Based Modeling (ABM). This implies that the real system of interest is modeled using agent principles and implemented in simulation software, this model consists of interacting agents and some performance and risk indicators. In this agent based decision support system, decision makers, can set parameters, run the simulation and evaluate the results, iteratively and to evaluate different management alternatives. Based on the retrieved information/knowledge they can make decisions regarding how to handle the real system. This implies that an agent based decision support system collects and fuses information from different sources in a synergistic manner into a representation (situation image) that provides effective support for human decision-making.

In this research work a decision support system has been developed for planning of SC operations using ABM. This research shows that this kind of decision support system is based on IF since it collects and fuses information from different sources into a situation image that provides effective support for human decision making. The information comes from databases and from various messages generated by different agents. Fusion may occur on agent or system level, and the result of this fusion is a collective intelligence that cannot be derived from the internal mechanism of an individual agent.

The main benefit of an agent based decision support system for the decision maker comes mainly from the ability to see how different situations flow through the company. Normally a decision maker can only observe his own small decision making node while the simulation model allows the decision maker to see how decisions are generated and how different situations will show up in different parts of the organization. For instance, the supplier for the life-cycle product will be impacted by the demand for the cyclical product. This kind of behavior would be impossible to analyze without proper IF as the supplier would assume that the fluctuations are due to demand cycles in the life-cycle product. It would also be possible to see what happens if some of the information would be distorted and how it would escalate in the decision making chain.

The proposed system is still a simple one; only showing the necessary information flows in an organization that uses a MRP-system. Thus, the model only has a low amount of total IF possible. However, the model can easily be expanded to include more information flows and if the manufacturing unit would also be modeled, it would be possible to connect the simulation model to be part of the decision support system. Most of the information for the simulation model could directly come from the real databases of the company and from real information flows. This is a clear advantage for an agent based simulation model as there would be no need to gather any additional information from the organization. Also, situation awareness of individual decision makers would improve as they could see how the same initial information is used in different parts of the organization.

Initial simulation results seem to be promising and we will continue in pursuing to have more strategic and tactic decision making completed from this environment. Also real-life demand data among lead time, capacity addition, production lead times, and change over times etc. are planned to be gathered in the future to establish real-life simulation platform, from where management could make better decisions, and possibly apply agent based logic in operative decisions. We need also to evaluate in the future with past data, what kind of decisions simulation model has proposed, and how case company has actually in traditional mechanistic manner completed these.

References

1. Gibson, B.J., Mentzer, J.T., Cook, R.L.: Supply Chain Management: The Pursuit of a Consensus Definition. Journal of Business Logistics 26(2), 17–25 (2005)
2. Christopher, M.C.: The agile supply chain: competing in volatile markets. Industrial Marketing Management 29(1), 37–44 (2000)
3. Nilsson, F.: Logistics management in practice - towards theories of complex logistics. International Journal of Logistics Management 17(1), 38–54 (2006)
4. Prater, E., Biehl, M., Smith, M.A.: International supply chain agility – tradeoffs between flexibility and uncertainty. International Journal of Operations & Production Management 21(5/6), 823–839 (2001)
5. Christopher, M., Lowson, R., Peck, H.: Creating agile supply chains in the fashion industry. International Journal of Retail and Distribution Management 32(8), 367–376 (2004)
6. Boström, H., Andler, S.F., Brohede, M., Johansson, R., Karlsson, A., van Laere, J., Niklasson, L., Nilsson, M., Persson, A., Ziemke, T.: On the Definition of Information Fusion as a Field of Research. Technical report, University of Skövde (2007)
7. Nilsson, F., Darley, V.: On complex adaptive systems and agent-based modeling for improving decision-making in manufacturing and logistics settings. International Journal of Operations & Production Management 26(12), 1351–1373 (2006)
8. Hilletofth, P., Aslam, T., Hilmola, O.-P.: Multi-agent based supply chain management: Case-study of requisites. International Journal of Networking and Virtual Organisations (accepted, forthcoming)
9. Lim, M.K., Zhang, Z.: A multi-agent based manufacturing control strategy for responsive manufacturing. Journal of Materials Processing Technology 139(1/3), 379–384 (2003)
10. Swaminathan, J.M., Smith, S.F., Sadeh, N.M.: Modeling supply chain dynamics: a multi-agent approach. Decision Sciences 29(3), 607–632 (1998)
11. Davidsson, P., Wernstedt, F.: A multi-agent system architecture for coordination of just-in-time production and distribution. The Knowledge Engineering Review 17(4), 317–329 (2002)
12. Macal, C.M., North, M.J.: Tutorial on Agent-based Modeling and Simulation Part 2: how to Model with Agetns. In: Proceedings of the 2006 Winter Simulation Conference (2006)
13. Davidsson, P., Henesey, L., Ramstedt, L., Trönquist, J., Wernstedt, F.: An analysis of agent-basedapproaches to transport logistics. Transportation Research Part C 13(4), 255–271 (2005)
14. Cantamessa, M.: Agent-based modelling and management of manufacturing systems. Computers in Industry 34(2), 173–186 (1997)
15. Forrester, J.W.: Industrial dynamics – a major breakthrough for decision makers. Harvard Business Review 36(4), 37–66 (1958)

Simulation of Resource Acquisition by e-Sourcing Clusters Using NetLogo Environment

Konrad Fuks and Arkadiusz Kawa

Poznań University of Economics,
al. Niepodległości 10, 61-875 Poznań, Poland
{konrad.fuks, arkadiusz.kawa}@ue.poznan.pl

Abstract. In the last few years many business clusters have been formed. Thanks to the geographical proximity as well as close cooperation companies are able to gain a lot of benefits that they would not reach if they worked on their own. The ICT[1] has given a possibility to go beyond traditional cooperation and has allowed various geographically spread companies to create alliances. Especially the agent technology seems very promising. The software agents are able to efficiently support e-marketplaces, including partners who offer one another the best cooperation conditions at a given time[1][2]. The main contribution of this paper is a verification of the improved e-sourcing strategy which was presented in [3]. Simulations of this model have been carried out in the *NetLogo* environment. It has been shown that the usage of e-sourcing cluster can inconceivably boost up the process of resource acquisition.

Keywords: agent based simulation, e-sourcing, clusters, logistics, electronic supply chains, NetLogo.

1 Introduction

Procurement is a process of buying raw material, resources, semi-finished and finished products ready for commercial and manufacturing activity and other goods needed for an enterprise to function. The purchasing process is convoluted and requires a lot of laborious activities, such as: creating an offer enquiry, finding a supplier, collecting offers from particular suppliers, choosing an appropriate offer, monitoring orders, paying, etc. Due to this complexity of the purchase process as well as the potential large savings it may allow special attention should be paid to it. An exact identification of the future purchase needs enables to limit the time, prepare the company to finance the purchase in the most profitable way and limit the warehouse costs. In order to reduce the costs of resource acquisition even more competitive enterprises start to cooperate in the purchase process. Together they can achieve better prices than if they were operating individually.

In this paper we are offering to use the e-sourcing cluster (eSoC) concept to acquire resources needed by companies. Before we proceed to presenting the eSoC, however, let us explain what the cluster is, according to the traditional approach.

---

[1] Information and Communication Technology.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 687–696, 2009.

Generally, the cluster can be described as a group or bunch of objects. Merriam-Webster Online Dictionary defines the cluster as a number of similar things that occur together.[4] Cluster applications are multiple. The concept may be found in astrophysics (star cluster), biology and health sciences (cancer cluster), computing (computer cluster), music (tone cluster), economics (business cluster), the military (cluster bomb), etc. This paper focuses on the business cluster. Intuitively the business cluster can be specified as a group of companies which cooperate in order to achieve some advantages.

The term, business cluster, also known as an industry cluster, competitive cluster, or Porterian cluster, was introduced and popularized by Michael Porter in his work, *The Competitive Advantage of Nations*.[5] He defines a cluster as ''…a geographically proximate group of interconnected companies and associated institutions in a particular field, linked by commonalities and complementariness''.[6]

The clusters are formed by enterprises linked through vertical (enterprises connected with others in the relation "supplier-buyer") and horizontal (e.g. competitors creating an alliance to achieve the strategic goal) relationships with the main players located in the same place. The geographical proximity is seen as a way of facilitating the transmission of knowledge, supplies and the development of institutions, which in turn may enhance cluster effectiveness.[7] According to M. Porter, it creates competitive advantages for small and medium enterprises (SMEs) which cooperate and compete closely, since a host of linkages among cluster members results in a whole greater than the sum of its parts.[6] So, all the companies can benefit directly from being part of a cluster.

Unfortunately, clusters understood in this way have some limitations. The definition presented above and many other existing in the professional literature and papers [5][7][8][9][10] suggest that the characteristic feature of the business cluster is the geographical proximity. This vicinity as well as the informal communication and face-to-face contacts connected with it still matter and create the competitive advantage.[11] However, not every enterprise (especially SMEs) can be the member of the business cluster due to a high entry costs. Moreover, it is impossible to physically locate all enterprises in one place. Further, the clusters are formed for years, that is why there is a high exit costs.

The structure of the paper is as follows. In Section 2 e-sourcing, as a modern way of resources acquisition using new ICT, is presented. Section 3 presents the e-sourcing cluster (eSoC) concept and general model of resources acquisition and division using eSoC. Section 4 describes simulation experiment background, variables and constants. Additionally, structure of simulation model implemented in NetLogo environment, numerical results from conducted simulations and trends/dependences analysis are presented. Section 5 concludes the paper and shows future work.

2 E-Sourcing

Geographical proximity still plays an important role in quick information interchange but the emergence of the ICT, especially the Internet, changed this completely. Thanks to ICT, communication costs declined, while trading networks have increased the global market participation of firms not depending on their size, experience, incomes etc. So, the Internet has revolutionized traditional supply chain management systems and has allowed various geographically spread companies to create alliances and to assist in the purchase of goods and services.[12] The rapid advances in ICT tend to develop virtual links between enterprises contributing towards realizing collaborative relationships with trading partners and easing the virtual sourcing processes, thus clusters do not necessarily have to be locally defined entities. The geographical proximity loses importance because of the easier access to information, yet some valuable, non-codified, but tacit, knowledge can still be exclusively obtained within a cluster.[7][13]

The aforementioned virtual sourcing can be defined as electronic sourcing (e-sourcing) which is the process of finding potential new suppliers using the Internet in general or, more specifically, a B2B internet platforms. The companies using the Internet in the procurement process have an access to a greater number of trading partners and can improve their capabilities in value-added processes. For example, they can acquire resources by participating in e-marketplaces in which they are enabled to find appropriate products or services very quickly.

Information systems that support e-sourcing can be classified into four major segments: buy-side applications, sell-side applications, content applications and e-marketplace applications.[14] The most complex and problematic are the last solutions so we will focus on them in this paper. The e-marketplace platform merges different entities in order to create competitive advantage among suppliers as well as to create access to an abundance of buyers. Although e-marketplaces offer a lot of facilities and have many strengths, there are some limitations of this solutions supporting e-sourcing.

Firstly, today it is very easy to establish a marketplace on the Internet. Thus there are hundreds of globally operating e-marketplaces[15]. This quantity makes it impossible for humans to browse through all of them. Owing to the scope of offers, it is very difficult to sort through all of the information about potential suppliers from many e-marketplaces and to compare one with another in order to make a final decision. Additionally, there are also differences between the requirements of prospective buyers and sellers. The expectations of the former are different from the offers presented by the latter. Moreover, the environment of e-marketplaces is constantly changing – some e-marketplaces appear, some evolve and some disappear. The growing competitiveness and demand changeability make the best offer difficult to find.[3]

3 E-Sourcing Cluster

The task of finding new, reliable suppliers on many e-marketplaces is costly and time-consuming, especially when the company acts alone. Thus, we propose the concept of an e-sourcing cluster - eSoC. The e-sourcing cluster (type of business cluster) is a group of enterprises which are looking for the same type of resource (e.g. steel, plastic, packaging, etc.), so they cooperate in order to acquire it more quickly and more effectively. They communicate and exchange the information automatically using software agents. In such clusters the geographical proximity does not matter at all. Enterprises do not have to be located in one place because they cooperate using the Internet. The eSoC consists of the net of enterprises from the same or similar branch, which are characterized by the capability of quick and dynamic adaptation to the changing market and different requirements. The eSoC concept can be very attractive to SMEs, because it refutes high entry/exit costs barrier from traditional cluster theory.[16] Another distinctive feature of such clusters is decentralization and the lack of one central point around which enterprises are concentrated.

Within the e-sourcing cluster SMEs can coopete[2] with big enterprises which use their size and purchasing power to get cheaper supplies and offer inexpensive goods. The entities joined in eSoC share useful information and knowledge with other members in order to achieve better mutual understanding, acquire wider offers range and develop a base for mutual trust that may eventually lead to collaboration in achieving the members' individual as well as collective goals.[16] Thanks to that, it is easiest to reconfigure elements of sourcing processes according to changing output requirements and the rise of a new market.[17]

Obviously, it must be remembered that in the case of eSoC some transaction costs are higher (e.g. the transportation costs are boosted due to the longer distances), but can be compensated by lower cost of searching and choosing suppliers, contracts management, data interchange, effects of synergy, and, what is most important, lower prices of resource.

In our approach e-sourcing clusters are created on e-marketplaces. All eSoC data is stored within the e-marketplace database. Therefore, e-marketplaces are a base for enterprises to find potential resource/suppliers. E-marketplaces can be accessed by an arbitrary number of software agents (e.g. representing many suppliers) which work independently, or collaborate with others.[2] Main assumption of the model is that e-marketplaces are based on service-oriented architecture (SOA) which provides system interoperability. That means that software agents do not have any problems with migration from one to another e-marketplace. At the same time each e-marketplace is searched only by one enterprise representative (software agent) from each eSoC.

The algorithm of the resource searching and acquisition by e-sourcing cluster in network of e-marketplaces has been presented in detail in work [3]. Let us remember about the most important foundations and assumptions of this model. They are essential to understand the simulation described in the further part of this paper.

First of all, a company creates a profile (equipped with a group of software agents) on one of available e-marketplaces. Then a delegated agents look for sourcing cluster/s. If they do not find it, they create one on each of indicated e-marketplaces. In the next step *leader agents* (type of software agents) of each company assigned to the e-sourcing cluster delegate other, subordinate to them, *scout agents* (another type of software agents) to search for resource with defined conditions (quantity, quality, price, shipping terms, etc.). After that, *scout agents* visit all potential e-marketplaces and look for resources that meet the predefined conditions.

If *scout agent* finds proper resource, it compares resource supply (*VRs*) with it company's demand (*Dm*). Than it reserves *Rv* units of the resource, where *Rv* value depends on:

1) $Dm >= VRs => Rv = VRs$
2) $Dm < VRs => Rv = Dm$.

[2] Coopetition is a neologism which matches cooperating and competition.

In first scenario only *scout agent* of company acquires the resource. Its demand can by partially (*Dm* > *VRs*) or fully satisfied (*Dm* = *VRs*). If *Dm* > *VRs scout agents* keep looking for next resource offers and its *leader agents* monitor eSoCs' information pool for findings of other agents.

When supply of the resource exceeds demand of *scout agent*, that company satisfies whole demand. Next, *scout agent* comes back to the eSoC from which it originate and informs other agents about the fitness, direction and distance of resource. Than it passes the information to other eSoCs in which it participates. Notifying other agents from all of its clusters about resource results in two main advantages. On the one hand, found resource is almost always used. On the other hand the possibility of resource acquisition should increase, if the company is a member of greater number of e-sourcing clusters. As section 5 will show the upper range of clusters number is not infinite.

In the next step *leader agents* sum individual demands of all, interested in resource, companies in eSoCs and set cluster demand (*D*) for found resource. Than *D* is compared with left resource supply (*R* = *Vrs* – *Dm*). Found resources can be divided in several ways. If total demand of the eSoCs for the resource is lower than or equal to left supply of the resource (*D* <= *R*), all *foraging agents* (third type of software agent presented) can migrate to the specified e-marketplace and start the buying process of needed resource. In this case they fulfill all their demands. In the contrary situation, if *D* > *R,* authors distinguished three groups of resource division methods:

1) E-sourcing cluster perspective:
   a. *First-come, First-buy.*
   b. *Proportionally.*
2) Individual company perspective:
   a. *On the income basis* – enterprise with the biggest income satisfies its demand first.
   b. *On the p-level basis* (priority level which can depend on: purchase/sell volume or value, length of e-marketplace participation period, company size, annual income, country of origin, etc.).
3) Auction perspective (Vickrey, Dutch, English).

After that *scout agents* visit again potential e-marketplaces and look for the resource that meets the predefined condition. They repeat this process as long as their demands are satisfied.

4 Simulation

In order to check the capabilities of e-sourcing cluster were carried out the simulations of the model in NetLogo environment. NetLogo is a programmable agent based modeling environment for simulating natural and social phenomena. It is the next generation of the series of multi-agent modeling languages that started with *StarLogo*. The environment allows to give instructions to a lot of independent agents which can interact with each other and perform multiple tasks.[18] In NetLogo the turtles (mobile agents) move on a grid of *patches*, which are also programmable, but static, agents[3]. All agents can simultaneously carry out their own activities.[19] NetLogo also works on time basis, so changeability of the simulated model over time can be observed and analyzed. Implicitly time in *Netlogo* is measured in *tick* units. *Tick* can represent any possible real time unit (second, hour, day, year, specific time period, etc.).

Two kinds of *breeds* were distinguished: 1) resource offers (*ress*), 2) companies (*coms*). Thanks to that we could define different behaviors and *agentsets* of both breeds. Three kinds of resource fitness were introduced, there were: price (*price*), place (*place*), and shipping time (*ship-time*). Values of each fitness were randomly generated for each individual in each breed.

The fitness of sought after resource was compared with the fitness of found one according to the rules:

1) *Price* of found resource cannot exceed *price* declared by the company.
2) *Place* of found resource must be the same as region of company.
3) *Ship-time* of found resource cannot exceed *ship-time* declared by the company.

It is worth noticing that sellers give different prices, some of which include other additional costs, but others do not. A lower price is offset by significantly higher acquisition costs such as those of delivery, monitoring, coordination and other administrative tasks.[20][21] Thus it is very difficult to compare them with one another. In our model *price* comprised all those costs.

The basic number of potential e-marketplaces was assumed to be 900[4], but for additional comparison and model verification simulations with 9000 e-marketplaces were carried out. E-marketplaces are represented by *patches*. Other parameters, which were constant, are as follows:

- *Cnum = 100* – initial number of companies.
- *Rnum = 10* – initial number of resources.
- *Max-cval = 20* – maximum value of company demand.
- *Max-rval = 200* – maximum value of finding resources.
- *Max-res-life = 500* – maximum resource life.
- *Max-ship-weeks = 1* – maximum randomization of number of shipping weeks.

The most significant variable is *max-clu-num* (maximum clusters number), which was assumed to vary from 0 to 20 and its influence on companies demand satisfaction is the main experiment measurement. Participation in a cluster is based on probability (in this case probability is 50%), which is set individually for each company's cluster. Additional important assumptions for this simulation are:

- Method of new resources generation: every 100 ticks random *rnum*. resources are generated - from 0 to 9 resources.

[3] In NetLogo there are four types of agents: turtles, patches, links, and the observer.

[4] eMarket Services (http://www.emarketservices.com) has about 600 e-marketplaces in its database. Authors increased this value by 50% in connection with internet based B2B solutions popularization.

- Relation between *cnum* and *rnum* – 10:1.
- Relation between *max-cval* and *max-rval* – 1:10.
- *Scout agents* only chooses resources that can satisfy their company's demand (fitness of found resource/s must meet *price*, *place* and *ship-time* rules presented above).
- Method of resource/s division is *First-come, First-buy*.

The simulation was run 1000 times for each case (changing *max-clu-num* value by 1), that gives 21000 simulations. Additionally, each case (static value of *max-clu-num*,

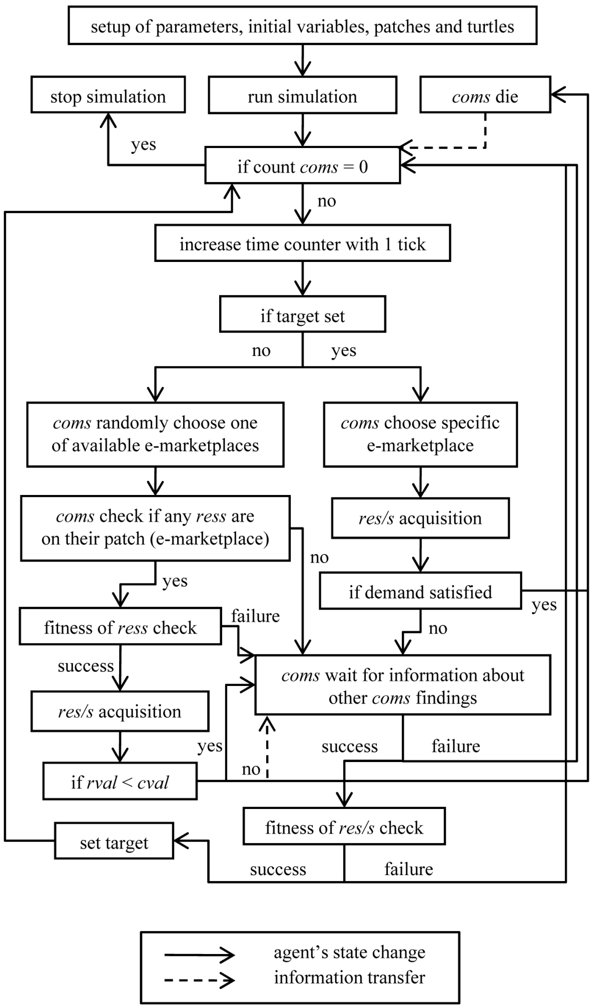


Fig. 1. NetLogo simulation algorithm of eSoC model

Table 1. Time (ticks) after demand of specific percent of companies was fulfilled (900 e-marketplaces)

Clusters	*100*	*90*	*80*	*70*	*60*	*50*	*40*	*30*	*20*	*10*
0	5704	1882	1193	873	660	509	387	292	195	102
1	4980	1496	933	665	517	400	305	224	147	75
5	3786	749	465	321	225	164	115	74	45	22
10	2850	461	312	222	155	107	73	47	27	15
20	2086	382	239	171	123	84	57	40	25	12

i.e. 5) was examined ten times (each 10% of *cnum* decrease) to get better view of *cnum* changeability over time (ticks). When we add cases with 9000 e-marketplaces, it will give us total number of 420000 simulations in eSoC experiment.

Figure 1 shows structure of simulation model implemented in NetLogo. Solid lines shows change of agent's state and dashed ones illustrates only information passing.

Simulation results were compared for 900 and 9000 e-marketplaces scenarios. Numbers presented in table 1 shows numerical data from five cases of the experiment: no clusters, one, five, ten, twenty potential cluster/s. Analysis shows that increasing number of e-marketplaces ten times (to 9000) increases time of population demand satisfaction also about ten times. It can lead to first conclusion from experiment, that time of population demand satisfaction is directly proportional to number of e-marketplaces. Of course this conclusion is true when we have got the same initial parameters of the simulation as well as methods of interactions and dynamic model evolution.

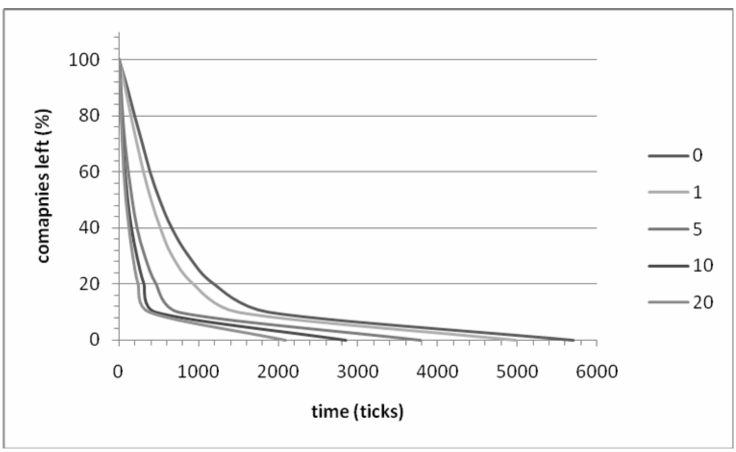


Fig. 2. Population demand satisfaction over time dependent on maximum number of clusters in which one company can participate. Legend shows maximum number of potential clusters.

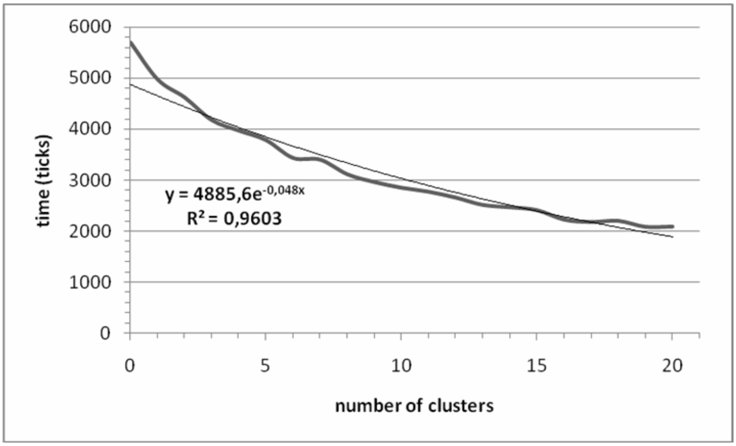


Fig. 3. Total population demand satisfaction time dependent on maximum number of potential e-sourcing clusters

Another conclusion is that increasing maximum number of potential clusters (eSoCs) do not decrease time of population demand satisfaction proportionally. As figure 2 shows differences between 10 and 20 *max-clu-num* are marginal. It corroborates the Gossen's First Law also known as the law of diminishing marginal utility. Additionally, when entry and stay in cluster costs will be taken into consideration, it can be stated that for this particular resource model maximum value of *max-clu-num* is 10.

Other analysis of the experiment is comparison of total population demand satisfaction time and maximum number of potential e-sourcing clusters. Figure 3 shows that increasing *max-clu-num* influences on total population demand satisfaction time and decreases it on specific trend basis (96% trend fit).

5 Conclusions

The experiment shows that eSoC model can significantly boost company's resources acquisition process. Additionally, lower than traditional Porterian cluster entry/exit costs can indirectly result in a lower risk factor of investing in a new business model. While this can lead to better cooperation, cross-companies relations and collaboration in achieving the eSoC's members' individual as well as collective goals.

References

1. Wieczerzycki, W.: Polymorphic Agent Clusters – the Concept to Design Multi-Agent Environments Supporting Business Activities. In: Mařík, V., William Brennan, R., Pěchouček, M. (eds.) HoloMAS 2005. LNCS(LNAI), vol. 3593, pp. 233–245. Springer, Heidelberg (2005)

2. Fuks, K., Kawa, A., Wieczerzycki, W.: Dynamic Configuration and Management of e-Supply Chains Based on Internet Public Registries Visited by Clusters of Software Agents. In: Mařík, V., Vyatkin, V., Colombo, A.W. (eds.) HoloMAS 2007. LNCS(LNAI), vol. 4659, pp. 281–292. Springer, Heidelberg (2007)
3. Fuks, K., Kawa, A., Wieczerzycki, W.: Improved e-Sourcing Strategy with Multi-Agent Swarms. In: Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC 2008). IEEE Computer Society, Washington (in press, 2009)
4. http://www.merriam-webster.com/dictionary/cluster
5. Porter, M.: The Competitive Advantage of Nations. The Free Press, New York (1990)
6. Porter, M.: Clusters and the new economy of competition. Harvard Business Review 76(6) (1998)
7. Karaev, A., Koh, S.C.L., Szamosi, L.T.: The cluster approach and SME competitiveness: a review. Journal of Manufacturing Technology Management 18(7) (2007)
8. UNIDO, Promoting Enterprise through Networked Regional Development, United Nations International Development Organization, UNIDO Publications, Vienna (2000)
9. Belleflamme, P., Picard, P., Thisse, J.F.: An economic theory of regional clusters. Journal of Urban Economic 48(1) (2000)
10. Pouder, R., St John, C.H.: Hot spots and blind spots: geographical clusters of firms and innovations. The Academy of Management Review 21(4) (1996)
11. Andersson, T., Serger, S.S., Soervik, J., Hansson, W.E.: The Cluster Policies Whitebook, IKED, Malmo (2004)
12. Folinas, D., Manthau, V., Sigala, M., Vlachopoulou, M.: E-volution of a supply chain: cases and best practices. Internet Research 14(4) (2004)
13. Preissl, B., Solimene, L.: Innovation clusters: virtual links and globalization. In: Proceedings of the Conference on Clusters, Industrial Districts and Firms: The Challenge of Globalization held at University of Modena and Reggio Emilia (2003)
14. Hawking, P., Stein, A., Wyld, D.C., Foster, S.: E-Procurement: Is the Ugly Duckling Actually a Swan Down Under? Asia Pacific Journal of Marketing and Logistics 16(1) (2004)
15. http://www.emarketservices.com
16. Chung, W.W.C., Yam, A.Y.K., Chan, M.F.S.: Networked enterprise: A new business model for global sourcing. International Journal of Production Economics 87 (2004)
17. Piore, M., Sabel, C.: The Second Industrial Divide. Basic Books, New York (1984)
18. Wilensky, U.: NetLogo itself, NetLogo, Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL (1999), http://ccl.northwestern.edu/netlogo/
19. Tisue, S., Wilensky, U.: NetLogo: Design and Implementation of a Multi-Agent Modeling Environment. Presented at SwarmFest, Ann Arbor, May 9–11 (2004), http://www.ccl.sesp.northwestern.edu/papers/netlogo-swarmfest2004.pdf
20. Cannon, J.P., Homburg, C.: Buyer–supplier relationships and customer firm costs. Journal of Marketing 65(1) (2001)
21. Steinle, C., Schiele, H.: Limits to global sourcing? Strategic consequences of dependency on international suppliers: Cluster theory, resource-based view and case studies. Journal of Purchasing & Supply Management 14 (2008)

Analysing Supply Chain Strategies Using Knowledge-Based Techniques

Areti Manataki and Yun-Heh Chen-Burger

Centre for Intelligent Systems and Applications, School of Informatics,
University of Edinburgh, Informatics Forum, 10 Crichton Street, Edinburgh EH8 9AB, UK
A.Manataki@sms.ed.ac.uk, Jessicac@inf.ed.ac.uk

Abstract. We are experiencing a digital revolution that is rapidly changing the way business is conducted. In this new economy, information is shared speedily and many participants are collaborating. Effective and seamless collaboration between distributed supply chain members is therefore crucial. To understand how this may be done, we describe a knowledge based framework for abstracting, enriching, analysing and improving supply chain models. We employ business process modelling as a useful methodology for capturing and analysing supply chain strategies. In addition, we use a meta-interpreter and workflow engine for simulating business processes to help understand business scenarios. Its declarative approach makes business rationale more transparent. Our work is demonstrated by looking at a case study of Dell's supply chain management that depicts its PC supply chain operation logics and strategies.

Keywords: supply chain management, business process modelling, simulation, workflow engine, Dell, FBPML, declarative approach, logical formal method.

1 Introduction

We have entered a Digital Economy era. The emergence of the Web and the general transformational impact of information and communication technologies upon business activity have lead us to an epoch where e-business, e-commerce, and electronic goods and services are perceived as standard practices. In fact, as "defined by the changing characteristics of information, computing and communications, the digital economy is now the preeminent driver of economic growth and social change" [1]. In this new business reality, competition is consumer-driven and there is a need for closer collaboration between companies. The Internet is recognised to "provide considerable opportunities for firms to streamline their business operations" [2], and to collaborate or even virtually integrate, especially among supply chain (SC) members. We are experiencing integrated value chains and emerging business models, including autonomous virtual enterprises. In the meantime, this digital revolution "makes commerce less local, more interstate and more global" [1].

In such a highly distributed environment, where there is a need for integration, it is important to assure coordination among the different business actors, especially among SC members. We believe that a third-party broker could successfully

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 697–704, 2009.

undertake this task, given that he is strategy-aware, flexible and reliable to changes. In this case, an important question is what sort of knowledge is needed for such a broker.

In this paper we regard integrated supply chain management as an integral part of the digital economy, and we focus on knowledge capture for a SC broker. We, thus, suggest a framework for abstracting, enriching and analysing supply chain strategies with the use of knowledge-based techniques. Business process modelling is recognized as a useful vehicle for analysing SC strategies, and a cyclic modelling framework is introduced. A workflow engine is presented to carry out the simulation of business processes, and thus will be used within the suggested framework for reasoning about and comparing SC strategies. Finally, recognising Dell as the coordinator and manager of its supply chain, we introduce its case study to show how our framework can be applied in real-world business scenarios.

2 Background Information

Supply Chain Management (SCM) is defined as "the integration of key business processes from end user through original suppliers that provides products, services and information that add value for customers and other stakeholders" [3]. A supply chain, i.e. all parties involved in fulfilling a customer request, has to manage the flow of information, products and funds throughout its tiers so that it has the right products in the right quantities at the right time and at the minimal cost. The digital economy has enforced and facilitated the surge of SCM, hence "SCM consciousness is accelerating up the corporate agenda" [4]. In fact, experts state that "enterprises are currently competing through their supply chains" [5], thus reflecting the significance of SC coordination, alignment and integration.

Dell is a successful SCM example, as it uses the knowledge of coordination to improve SC performance by operating as a SC organiser. Dell "coordinates information sharing, incentive alignment and collective learning to focus on direct selling and build to order" [6]. Its *direct model* involves selling PCs directly to customers through its website Dell.com, hence bypassing the cost-adding distributors and retailers while getting access to valuable marketing information. Moreover, Dell differentiates its strategic approach towards different market segments (i.e. large organizations, small and medium businesses, and personal consumers), with a main focus on large customers. Dell was one of the first computer companies to highlight *component assembly* as a respectful core activity. The *build to order* strategy means that a computer is built only after an order has been placed, a practice requiring close collaboration between Dell and its suppliers. According to Michael Dell, "*Dell's partners are treated as if they're inside the company*" [7], thus virtual integration is accomplished and real-time information is shared through ValueChain.Dell.com. Furthermore, the high selection and evaluation criteria that Dell sets for its suppliers guarantees long-lasting relationships between them [8]. Hence, we believe that Dell has a threefold role: it is a computer manufacturer, a point of contact for customers and an overall SC manager. The last two points upgrade Dell to a SC broker in the digital economy.

Business process modelling is a popular tool for enhancing business performance, as it "allows the capturing, externalisation, formalization and structuring of knowledge about enterprise processes" [9], thus enhancing knowledge management. *Workflow Management Systems* involve process modelling, process reengineering, and workflow implementation and automation [10]. According to Mentzas et al. [11] there are three basic categories of workflow techniques: communication and activity based, and hybrid techniques. In our work, we follow the activity-based approach.

A language that supports business process modelling and workflow system development is the *Fundamental Business Process Modelling Language* (FBPML) [12]. It is a merger and adaption of two standard languages: PSL and IDEF3, that results with both formal process execution semantics and rich visual modelling methods. These characteristics make it suitable for our work. Hence, FBPML is used in the modelling and simulation procedures. Interested readers are referred to [12] and [13] for further specification and notation of FBPML.

3 Capturing Supply Chain Processes and Strategies

3.1 A Cyclic Modelling Framework

Existing documents on SCM are often written in natural or structural languages and often intended to facilitate business studies focusing on certain aspects of the SC. They, therefore, cover only partial information needed to enable fully automated SC analysis and automation. To address this issue, we used a systematic, cyclic modelling framework that allows us to incrementally capture and enrich SC processes and strategies. A schematic abstraction of our framework is provided in Fig. 1.

In this *cyclic modelling framework*, the modeller is to first identify processes and data from the described organizational environment that is relevant to business operations and SC strategies. He then creates a suitable initial model and evaluates it to identify information gaps. If the existing model is satisfactory then it constitutes the final BPM; otherwise the modeller needs to find ways to fill these gaps. Once new

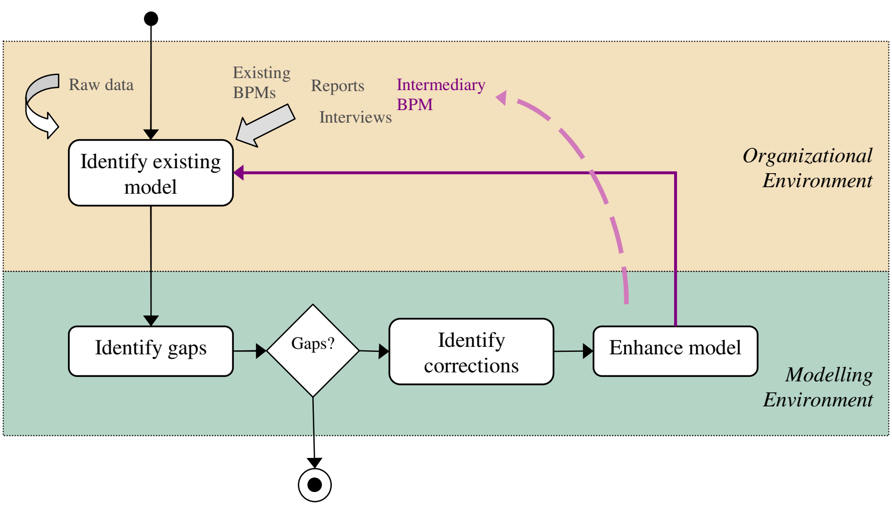


Fig. 1. Cyclic modelling framework

filler information is identified, he adds it to the existing BPM and creates an improved version of the model, which is provided as input for the next cycle of the modelling procedure. When no gap is identified, the modelling framework is ended. To identify information gaps, we suggest the following evaluation criteria and corresponding questions: i) soundness: does the BPM follow correctly the modelling specification, and are the semantics reliable?, ii) completeness: does it cover all important SC processes, strategies and business scenarios? iii) realism: does it correspond to real business and SC strategies? iv) level of detail: is it abstract enough to provide an overview of SC strategies and detailed enough to derive interesting information?

3.2 Development of Dell's BPM

While most of the literature that covers Dell's SC strategies is theoretical, we suggest business process modelling as a useful methodology for capturing and making those strategies concrete. We have applied the cyclic modelling framework explained above to the case study of Dell in order to develop a BPM that provides insight into the company's successful SC strategies, allowing us to perform interesting analysis. The initial input data was the MIT Process Handbook [14, 15] and its specific entry for Dell, as well as other literature that covers Dell's business and SC strategies and operations. By applying the cyclic modelling framework, we have encountered three distinctive larger loops. Each loop is identified with the lack of specific type of information and subsequently is provided and enriched with new information.

We first examined the initial process model, BPM#0, for Dell as provided by the MIT Process Handbook and found that the lack of process sequence in the model was an important limitation. To rectify this problem, we introduced sequences based on business logics and created a new version of the initial BPM, BPM#1. This new version was used as input for the second iteration of the cycle. Subsequently, BPM#1

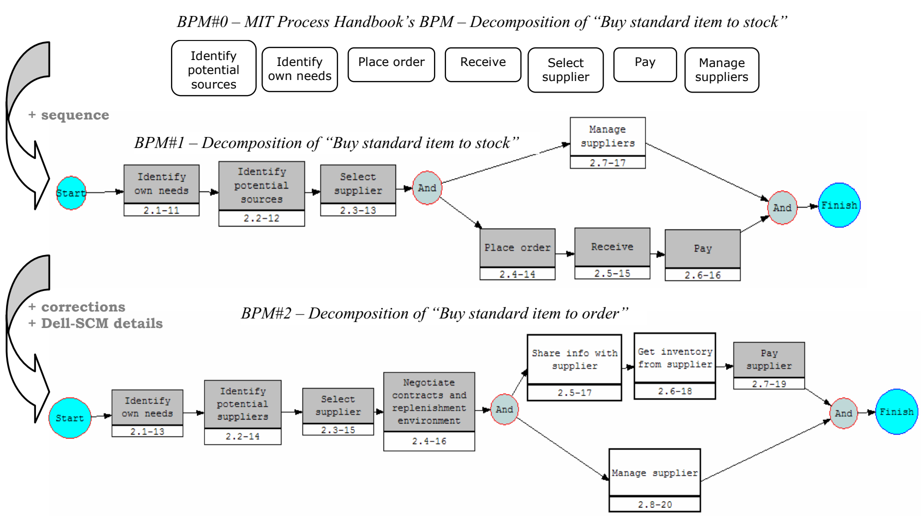


Fig. 2. Example processes of Dell's procedure "Buy standard item to order" throughout the modelling framework

was evaluated against the criteria in Sec. 3.1, and several further information gaps have been identified: i) some of Dell's SC strategies were not covered, e.g. virtual integration with different customer segments, ii) at several points it was too high-level to provide interesting SCM information and enable automation, iii) a mistake was identified, i.e. instead of buy-to-order, Dell was presented as buy-to-stock that is contradictory to Dell's well-known SC strategies, and iv) the assembly procedure of computers was not presented, which is a core operation of Dell. After suitable corrections were made, a new version of the BPM was created, BPM#2. Again, BPM#2 was used as input of the third loop repetition and was found to be satisfying for capturing Dell's SC strategies. In Fig. 2, we provide these three versions of the model of the same business process – one for each BPM version. Interested readers can refer to [16] for further information on the complete BPM.

4 Workflow Engine

4.1 Workflow Engine Design and Architecture

We will now present the Declarative Workflow Engine, DeWE, that has been designed for business process simulation, and which will later be used for SCM analysis. Since time and cost are important SC performance measures, DeWE will be given an appropriate *business context*, and calculate the total time and cost for a SC, given that we know the time and cost of each process involved. Moreover, when comparing different strategies through BPM simulation, one is interested in the business operations in the average case, so that e.g. when comparing the computer assembling procedure of two companies, such as Dell and IBM, it is possible to compare the average time and cost. To start the workflow engine, we use the notion of event that can be used to represent external customer order or internal events. We define the minimum values for processing time and cost to be 1 and 0, respectively.

Our workflow system has been implemented by adopting a declarative approach and it consists of three main components: a specification of the process model, a specification of the current world state, and the workflow engine algorithm. Note that DeWE treats each process in the BPM as a class definition, where instances of these processes are created and changed at run time due to events or (internal) computational changes.

4.2 Workflow Engine Reasoning Engine and Logical Representation

DeWE's reasoning engine was implemented in Prolog, and a simplified set of rules is provided below. Once the BPM simulation is started, DeWE is instantiated with BPM and its world state is initialised. A step execution involves scheduled actions', junctions' and processes' execution. A reached junction is executed, if its executions semantics are satisfied, according to FBPML specification [12, 13]. For instance, an and-joint junction is executed, only if all preceding processes have completed their executions. A process instance is executed, if its preconditions and trigger conditions hold. A trigger represents a request to invoke a particular process that can be caused by an internal or external event, while a precondition is a requirement that makes sure that process actions can be carried out successfully.

R1: IF start-up THEN (read process model AND read world state AND initialise ScheduledActions, ActionAgenda, ReadyProcesses and ReachedJunctions to [])
R2: IF found event THEN (IF its triggers are true THEN create a corresponding process instance AND execute-step is true for this event)
R3: IF execute-step THEN (execute ActionAgenda AND execute ReachedJunctions AND execute ReadyProcesses UNTIL no further action can be done; in that case update the time)
R4: IF (no event is left AND ScheduledActions and ReachedJunctions and ReadyProcesses are []) OR end of BPM is reached) THEN execute-step is false for the corresponding event
R5: IF execute ActionAgenda THEN (execute all actions in ActionAgenda, IF their preconditions and triggers are true AND no conflict between them is found)
R6: IF execute ReachedJunctions THEN (IF its process logic is true THEN extend all following branches to ReadyProcesses)
R7: IF execute ReadyProcesses THEN (expand its actions to ActionAgenda, IF their preconditions and triggers are true)
R8: IF (reached junction AND junction semantics is satisfied) THEN add Junction to ReachedJunctions
R9: IF (reached process AND process semantics is satisfied) THEN add process to ReadyProcesses

The declarative representation of the *process model* includes i) the *junction specification*, i.e. the control sequence of processes in a BPM, and ii) the *process specification*: inc. attributes of each process in a BPM, with respect to FBPML. The *world state* is declaratively specified through entity and event specification. Example predicates of junction and process specifications are provided below.

```
junction(Jid, Type, PreProcesses, PostProcesses).
junction(j4, and_joint, [p16,p17],[j5]).
process(Pid,Name,Triggers,Preconditions,Actions,Time,Cost).
process(p2_2,identifyPotentialSuppliers,
[exist(event_occ(needForNewXInventory))],
[not_exist(entity_occ(supplierX)),exist(data(needsOnXinventor
y)),exist(data(relevantXsuppliers))],[create_data(potentialXs
uppliers,[sup1_good, sup2_ok])],21,1000).
entity_occ(EntityName, EntityId, EntityAttributeList).
event_occ(EventId, EventName, TimePoint).
```

4.3 Workflow Engine Evaluation

We have used DeWE to simulate several BPMs, among which Dell's BPM. Based on those, we have evaluated it across four criteria, i) soundness, ii) completeness, iii) coverage and support, and iv) ease of use, thus answering the following questions: 1) Does it behave correctly? 2) Does it cover all needed concepts and functions? 3) Does it cover all possible BPM execution scenarios and thus support useful BPM analysis? and 4) Is it easy to use, for both experienced and inexperienced users?

The evaluation procedure can be found in [16], and it is beyond the scope of this paper to present it here. However, we mention the evaluation results, according to which our workflow engine is suitable as a tool for reasoning about SC strategies. So, it was found to be *sound*, *adequately complete*, *relatively easy to use* and *covering all scenarios* prescribed by its design manifestation.

5 Analysing Supply Chain Strategies

A generic framework we deployed for analysing SC strategies consists of three steps: First, we develop a BPM that captures the company's SC strategies, and choose the most interesting processes on which analysis will be performed. Second, BPMs are analysed and improvements sought. We experiment with parallelizing sequential processes, given physical and business constraints, with the aim to confirm whether it is already at its optimal or can still be improved. And third, to examine SC strategies and compare them with alternative ones; thus by creating an alternative BPM. We found BPM can be streamlined and performance improved where there is no (hard) information or business constraints.

Using the developed BPM for Dell, we identified two interesting processes for analysis, reflecting Dell's two important SCM characteristics: virtual integration with its suppliers and direct sales model: the "buy standard items to order" and "sell directly to large business and public sector customers" processes. During our experiments, no improvement was identified, which confirms Dell's operation has reached its optimal. We have thus used Dell's model as a golden standard to compare with the BPM of a conventional PC company. The comparison of the simulation results showed that Dell's choice of buy-to-order and virtual integration with its suppliers result in higher cost and time in the case of finding a new supplier; but it leads to much shorter duration and cost overall than that of a traditional PC company (that buys to stock) on the long run, because of its two above distinctive strategies.

6 Conclusions

Recognising the significance of a SC broker in the digital economy era, we have suggested in this paper a cyclic framework for abstracting, enriching and analysing SC strategies with the use of knowledge-based business process modelling techniques. We have developed a declarative workflow engine that simulates BPM execution and calculates the related time and cost that can be used as a tool to assist SC analysis. The suggested frameworks have also been applied to the case study of Dell and PC sector. We have shown that knowledge-based techniques can help us understand SCM. With the vision of a more business-aware workflow engine, our future work is to use the suggested process-based approach within a knowledge-enriched agent context, consisting of communication, process and reasoning layers.

References

1. Brynjolfsson, E., Kahin, B.: Understanding the Digital Economy: Data, Tools, and Research. MIT Press, Cambridge (2000)
2. Sharma, S.K.: Socio-Economic Impacts and Influences of E-Commerce in a Digital Economy. In: Kehal, H., Singh, V. (eds.) Digital economy: Impacts, influences and challenges, ch. 1. Idea Group Publishing, London (2005)

3. Lambert, D.M., Cooper, M.C.: Issues in Supply Chain Management. Industrial Marketing Management 29, 65–83 (2000)
4. Storey, J., Emberson, C., Godsell, J., Harrison, A.: Supply chain management: theory, practice and future challenges. Int. Journal of Operations & Production Management 26(7), 754–774 (2006)
5. Harrison, A., van Hoek, R.: Logistics Management and Strategy: Competing Through the Supply Chain. Prentice Hall, FT (2008)
6. Simatupang, T.M., Wright, A.C., Sridharan, R.: The knowledge of coordination for supply chain integration. Business Process Management J. 8(3), 289–308 (2002)
7. Magretta, J.: The power of virtual integration: An interview with Dell Computer's Michael Dell. Harvard Business Review 76(2), 73–84 (1998)
8. Fugate, B.S., Mentzer, J.T.: Dell's Supply Chain DNA. Supply Chain Management Review 8, 20–24 (2004)
9. Kalpic, B., Bernus, P.: Business process modelling in industry - the powerful tool in enterprise management. Computers in Industry 47(3), 299–318 (2002)
10. Georgakopoulos, D., Hornick, M., Sheth, A.: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases 3(2), 119–153 (1995)
11. Mentzas, G., Halaris, C., Kavadias, S.: Modelling business processes with workflow systems: an evaluation of alternative approaches. Int. J. Information Management 21(2), 123–135 (2001)
12. Chen-Burger, Y.-H., Tate, A., Robertson, D.: Enterprise Modelling: A Declarative Approach for FBPML. In: Knowledge Management and Organisational Memories Workshop, Proceedings of the 15th ECAI Conference. IOS Press, Amsterdam (2002)
13. Chen-Burger, Y.-H., Stader, J.: Formal Support for Adaptive Workflow Systems in a Distributed Environment. In: Fischer, L. (ed.) Workflow Handbook 2003, pp. 93–118. Future Strategies Inc., Florida (2003)
14. Malone, T.W., Crowston, K., Herman, G.A.: Organizing Business Knowledge: The MIT Process Handbook. MIT Press, Cambridge (2003)
15. MIT Process Handbook, `http://process.mit.edu/Activity.asp?ID=990929112221UF15790`
16. Manataki, A.: A Knowledge-Based Analysis and Modelling of Dell's Supply Chain Strategies. MSc Thesis, School of Informatics, University of Edinburgh (2007), `http://www.inf.ed.ac.uk/publications/thesis/online/IM070456.pdf`

Knowledge Management in Digital Economy

Lilia Georgieva and Imran Zia

MACS, Heriot-Watt University

Abstract. Knowledge management is the process of gathering, refining, organising, and disseminating knowledge through which organisations generate value from their intellectual and knowledge-based assets. In this paper we study standard formal models for knowledge management and discuss their transfer to digital economy. We propose a framework for knowledge management applicable to online trading agents. The framework relies on identification of processes which support knowledge management. We express knowledge exchange in epistemic logic. We then translate multi-agent dialogues in a protocol language which is verifiable by model checking.

1 Introduction

In this paper we propose a general framework for verification and simulation of knowledge management processes in an online trading environment. We study formal models for knowledge management and address the challenge of transferring these models to digital economy.

Knowledge Management is the process of critically managing knowledge in order to meet existing needs of a business or organisation, identify and exploit existing and acquired knowledge assets, and develop new opportunities. We determine crucial processes which support knowledge management in a digital economy and propose a general framework for management and verification of these processes. Our aim is to model check knowledge exchange among agents, consequently checking correctness properties of knowledge management processes.

We develop a formal framework for modelling knowledge exchange between agents. Our contribution is twofold. We identify fundamental knowledge streams essential for knowledge sharing and exchange among agents in digital economy and prove that the exchange is correct.

We achieve this by first formalising the processes in epistemic logic, followed by encoding of the formalisation in an agent modelling protocol language, and model checking the protocol using SPIN [4].

Our framework has the advantage of being provably correct, i.e. we can verify that correctness properties, which correspond to both the behavioural aspects and the data exchange used in the specification of the scenarios of knowledge exchange, are preserved, for example we can checked processes for consistency, termination, and fairness.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 705–713, 2009.

The structure of the paper as follows. First, we give background on knowledge management and discuss its importance. Next, we present three of the most popular approaches to modelling knowledge and comment on their advantages and disadvantages in online setting. Further, we propose framework for knowledge management and discuss model checking agent knowledge exchange processes.

2 Background

Knowledge is a key asset to any organisation, but organisations are still at an early learning stage of understanding the implications of knowledge management [1].

Knowledge management is critical for the success of an organisation both in real and digital environment. Economic innovations are frequently direct result of knowledge exchange and sharing. New technical possibilities provide opportunities for efficient operations and yet more effective knowledge sharing.

Digital economy is based on electronic transactions for goods and services produced by an e- business and traded through e-commerce. The transactions are conducted through internet and web technologies and present new challenges in addition to existing ones. Recognised issues in effective knowledge management, costing thousands of pounds per annum, are for example online security or identity theft. Not less important and costly are issues such as denial of service, or delay of service online. In this context effective and secure knowledge management is paramount.

In this paper we analyse the underlying knowledge management processes and discuss an approach to proving their correctness. We clarify that we take a process, rather than project based perspective to knowledge management. In the process perspective, an organisation measures the value of knowledge assets and the impact of knowledge management and then accesses the knowledge from external sources in order to generate new knowledge. Consistent with this approach, we view knowledge management as a collection of suitably identified processes.

Standardly knowledge management models utilise the concepts *tacit* and *explicit* knowledge. Tacit knowledge cannot be codified, but can only be transmitted via training or gained through personal experience. It involves learning and skill but not in a way that can be written down.

Explicit Knowledge is codified knowledge that can be transmitted in formal, systematic language. It is captured in records and is assessed on a sequential basis. It can be expressed in words and numbers and shared in the form of data, scientific formulate, and specifications, manuals. This kind of knowledge can be readily transmitted between individuals formally and systematically [8].

Formal models, which allow for non-digital knowledge management, have been proposed. We next discuss three prominent models. They are the spiral model [6], the VDT model [9], and computational model [3]. The models are developed knowledge flow in non-digital economy. We discuss which of the models are immediately transferrable to digital economies and what challenges and issues are faced when attempting to model knowledge management in digital economy.

2.1 Spiral Model

The spiral model describes a spiral of dynamic interaction between tacit and explicit knowledge along an epistemological dimension and characterises four processes of knowledge management, i.e. socialisation, combination, externalisation and integration. The processes enable the individual knowledge to be amplified and thus affect the organisational knowledge [6].

The characterisation of knowledge flow dynamics remains static in terms of this representational model; the model does not use a dynamic representation of knowledge flow, and consequently important dynamic interactions between model elements remain obscured through descriptive models based upon natural language texts and figures.

The framework of digital economy delivers a different purpose. Dynamic evaluation of processes is crucial for online transactions. For example, in an online trading environment, supported by eBay or Amazon market place, the winning bid is continuously re-evaluated and changed depending on the time limit, bided amount or even a number of participating agents.

2.2 VDT Model

The VDT modelling environment relies on information processing and embedding it in software tools [9]. Similarly as in the multi-agent approach in the VDT environment knowledge exchange partners are modelled as actors communicating with one another and performing tasks. The qualitative behaviours of VDT models correspond closely to an enterprise processes when put in practise. This is done by embedding software tools that can be used to design organisations. Multiple virtual prototypes of the knowledge management system are modelled and analysed. The VDT model emulates the dynamics of the organised behaviours.

This approach is transferrable to online environment. A challenge is the flexibility in an organization in which actors can communicate with anyone they choose inside or outside their local "organization" hence creating many prototypes and significantly increasing the complexity of the validation of such model. Such organizational form will have links which form and either persist or dissolve in cyberspace.

The validation in a VDT model is achieved via a representation of knowledge processes through simulation which gets increasingly more complex as more agents and behaviours are involved.

2.3 Computational Models

The computational model describes properties of the knowledge flow independently of the environment [3]. Knowledge flow has been associated with complex processes for example within software development which can be simplified as process types with specific properties in a computational model.

Computational models are often based on modal logics in general and specifically on logics for knowledge modelling in multi-agent systems, or epistemic logics. For reasoning purposes and to ensure reasonable complexity or decidability, the agents are often resource bound and have restricted logical capabilities, e.g. time, memory. Feasible reasoning is ensured by weakening the epistemic logic, i.e. the agent only knows some "obvious" logical truths but not the complicated ones. From this we can assume that the agent can draw all "obvious" consequences, but not any arbitrary consequence, i.e. the deduction mechanism of the agents is not complete. The agents are thus not powerful enough to draw all the logical consequences of their knowledge, which can be challenging in digital economy.

Nevertheless for simple trading behaviours, epistemic logic can capture knowledge exchange and is a suitable framework for modelling of agents' behaviour.

3 Framework for Knowledge Management

We present a framework defined in terms of knowledge management processes. In the framework we can model capturing and sharing of knowledge among agents.

The framework consists of structured streams which correspond to processes integral to knowledge exchange. The framework is tested through model checking. Model checking is fast and fully automatic verification technique. In the context of knowledge management model checking ensures correctness and termination of the underlying processes. The added benefits in a model checked knowledge management framework is that non-terminating processes and processes with deadlocks will be detected and counterexamples generated.

Market efficiency is the central motivation and justification of the liberalization policies devised worldwide in digital economy. The final economic outcome is dependent on practical implementation of processes and successful successive strategic interactions between the involved agents. Our framework allows us to consider and study knowledge management processes but also to get feedback from an unexpected event such as process deadlock and evaluate its consequences.

Consider for example a scenario where agents bid for goods online (See Fig. 1 [11]).

The protocol assumes a single auctioneer agent and a variable number of bidder agents, which is a realistic scenario for online trading agents (for example on Ebay). The auction begins with the auctioneer sending out the starting value for a particular auction item. Each bidder then makes an internal decision whether to bid at the current value, and makes a bid if appropriate. When the auctioneer receives a valid bid, if there is a higher bidder, the bid value is incremented and the new value is sent to all of the bidders. The bidders then make a decision to bid at the new value. The auction continues until no further bids are received by the auctioneer and a timeout occurs. At this point the winning bidder is notified and the auction concludes.

We use epistemic logic to formalising the exchange of information and bids among agents. We use SPIN to model and simulate a real-life online trading and identify whether a delay or a deadlock in the trading among agents occurs.

Our framework consists of the following processes: determining knowledge and available knowledge, determining knowledge gap, knowledge lock, knowledge sharing, knowledge utilisation and evaluation.

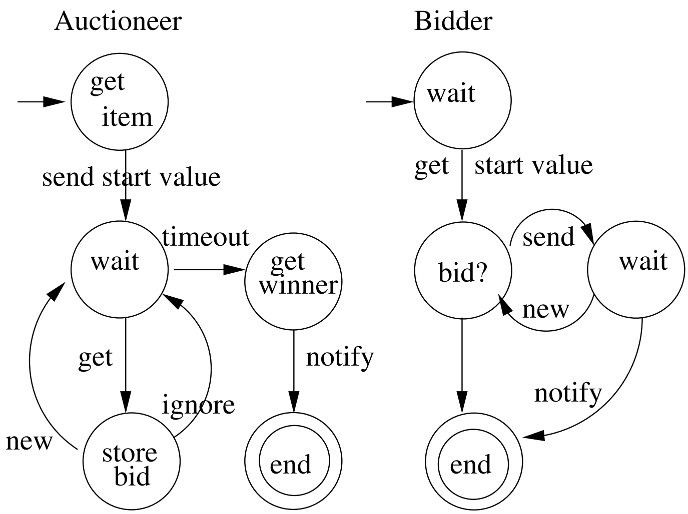


Fig. 1. Online trading agents

We next briefly discuss the knowledge exchange processes that we model.

Determine knowledge. Determining knowledge is important prior to any knowledge exchange as it includes the knowledge necessary for the exchange to happen. Consistency checks with respect to the objectives of the exchange to be determined are also part of this process.

Issues to be addressed when transferring determining knowledge to digital economy are: the kind of knowledge compulsory for the exchange (for example unique id or credit card guarantee) and the security constraints imposed on it; the type of knowledge that is needed in order for the exchange to happen (for example password, PIN or other type of identification); the goals of the exchange (for example transfer of goods or services).

Determine the available knowledge. This process looks at what knowledge is already available and its strategic implications of, for example, the direction of the exchange in online trading environment. In digital economy, this translates to, for example, keeping track of goods purchased by customers and basing future recommendations on previous purchases. More advanced software allows for personalised recommendations and statistics (currently in use for example by Amamzon and Amazon market place) where recommendations for future purchases or rentals are based on purchase or rental history.

Determine the knowledge gap. The space between the current knowledge and target knowledge defines knowledge gap. The objective for successful business development is that current knowledge equals target knowledge. There are two ways this can happen: the current knowledge can be increased or the target knowledge reduced. Online applications, handling customer details aspire to both. Increasing current knowledge is achieved via cookies that store information of the transactions made by a customer and building a reliable customer profile. Reducing the knowledge amounts to targeting the customer with specific to his interest goods or services and hence reducing by selection.

Knowledge lock. Knowledge lock in an enterprise means that the business have retained the knowledge into position, e.g. type of software or a type of system in place which the organisation has in place. Locking the knowledge in digital economy is achieved by storing user profiles unchanged until a new transaction or search takes place, hence updating the user details and preferences.

Knowledge sharing. Standardly knowledge is a resource for preserving valuable heritage, but also for learning new ideas, solving problems, creating core competences, and initiating new situations for both individual and organisations now and in the future.

Knowledge sharing means distribution of knowledge in the organisation which includes locked knowledge. It is vital that correct knowledge gets to the right person at the right time. This is dependent on the culture and structure of the organisations, arguably distribution of knowledge is straightforward in organisations with flat structure such as Microsoft or Google. In digital economy knowledge sharing can be achieved effortlessly, provided that the correct security checks are in place. A recognised challenge is secure communication and information exchange.

Knowledge sharing in digital economy encourages knowledge exchange and creation in the organisations in order to recognise their competitive advantages such as intellectual capital.

Knowledge sharing in organisations depends not only on technological means, but is also related to behavioural factors. Cognitive and motivational factors are thought to interfere with knowledge sharing both online and in standard face to face communication.

Knowledge utilisation. Utilisation of knowledge is largely based on the culture of the organisation. Crucial for online economy and trading is the speed and security of the exchange.

Knowledge evaluation. The evaluation concerns knowledge which is already retained in the organisation, and can be utilised. Knowledge can be evaluated e.g. through auditing, conducting satisfaction studies and benchmarking various parts of services [7]. Such evaluation is continuously carried out online.

3.1 Modal Epistemic Logic

In this section we give a brief introduction to our modelling language epistemic logic, based on model logic.

Epistemic logic is core to the development of agent communication languages [2]. The logic has sound formal semantics and high expressive power, it can be enriched for example by operators modelling commitment, intention, or trust which are integral to business communication in real or digital world.

The semantics of epistemic logic is based on modal logic semantics. Indexed knowledge modalities capture the properties of knowing by an individual agent, i.e. the fact that the agent i knows a fact ϕ is modelled by the epistemic formula $K_i\phi$.

Informally a modal epistemic logic for N agents is obtained by joining together N modal logics, one for each agent. For simplicity it is usually assumed that the agents are homogeneous, i.e. they can be described by the same logic. So an epistemic logic for N agents consist of N copies of a certain modal logic [3]. The following axioms and rules of inference are applicable to modal epistemic logic.

- (PC) All propositional tautologies
- (K) $K_i\alpha \wedge K_i(\alpha \rightarrow \beta) \rightarrow K_i\beta$
- (MP) Modus ponens: from α and $\alpha \rightarrow \beta$ infer β
- (NEC): From α infer $K_i\alpha$

Extensions of the logic can express properties such as the truth axiom (T) $K_i\alpha \rightarrow \alpha$ which states that knowledge must be true or consistency of knowledge (D) $K_i\alpha \rightarrow \neg K_i \neg \alpha$ requiring that agents are consistent in their knowledge, awareness of knowledge or lack of it (4) $K_i\alpha \rightarrow K_i K_i\alpha$; (5) $\neg K_i\alpha \rightarrow K_i \neg K_i\alpha$.

In our framework, the language representing the dialogue between agents is the MAP language. The MAP is a lightweight dialogue protocol and is used to represent multi-agent dialogue. MAP a replacement for the state-chart representation of protocols in Electronic Institutions and is designed by Foundation for Intelligent Physical Standardisation (FIPA 02). MAP is suitable for online information exchange. It permits only the agents that are involved in the scene to take place and excludes the ones that are not relevant.

Conveniently for digital economy additional security measures may be placed or introduced into the scene for e.g. placing entry and exit conditions on the agent. If we place a barrier condition on the agents, a knowledge exchange cannot begin until all the agents are present and the agents cannot leave the scene until the dialogue is complete. The protocols in the knowledge exchange are constructed from operations known as actions.

This controls the flow of the protocol and this also controls the actions which have side effects and can result in failure. Interaction is performed between agents by exchanging messages [10].

The semantics of message passing corresponds to reliable, buffered, non-blocking communication. Sending a message or sharing a message will succeed immediately if an agent matches the definition, and the message (share) will be stored in a buffer on the recipient. Receiving a message requires an additional unification step. The message supplied in the definition is treated as a template to be matched against any message in the buffer.

The foundation of intelligent protocol agents (FIPA) semantics sets the standard for the MAP protocol. FIPA agents communication language (FIPA ACL) is one of the most popular agent communication languages. In this language the interaction between agents is based on the exchange of messages. This defines the sets of performatives known as message types that express the intended meaning of the message. The language does not define the actual content of the message but assumes a reliable method of message exchange.

FIPA ACL message contains a set of one or more message parameters. Precisely which parameters are needed for effective agent communication varies 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 order to prove that the knowledge exchange among agents is reliable, and non-blocking we encode MAP in PROMELA similarly to model checking agent dialogues [10]. The translation into MAP is not new. Our contribution is in using the MAP encoding of formally defined epistemic processes in knowledge management setting. We benefit from using a language that has well-defined syntax and semantics and is especially designed to model properties of knowledge exchange and agent communication. Alternative specifications can be used. For example, in [5] a specially designed specification is used for verification of business processes using SPIN.

We then model-check the translation. There are a number of similarities between MAP and PROMELA, for example, both are based on the notion of asynchronous sequential processes or in our case (agents) both assume that communication is performed through message passing or exchange of information to the high-level similarities significantly simplify the translation, as the translation is from MAP agents directly into PROMELA processes and agent communication into message passing over buffered channels. However, the translation of the low level details of MAP is not straight forward. The difference is that there are significant semantic differences in the execution behaviour of the languages, for example the order of execution of the statements. MAP assumes that messages can be retrieved in an arbitrary order (by unification) while the PROMELA enforces a strict queue of messages. Using model checking, we can ensure, for example, that the knowledge sharing process is terminating when modelling online trading agents simulating behaviour on online market places. We have carried out experiments with SPIN for knowledge management dialogues, including knowledge sharing and knowledge gap. In future work we plan to verify the behaviour of more processes relevant to digital economy.

References

1. Beijerse, P.: Knowledge management in small and medium-sized companies: Knowledge management for entrepreneurs. Journal of Knowledge Management 4, 162–182 (2000)
2. Cohen, P., Levesque, H.: Intention is choice with commitment. Journal of Artificial Intelligence 42, 213–261 (1990)
3. Halpern, J., Moses, Y.: A guide to completeness and complexity for modal logics of knowledge and belief. Journal of Artificial Intelligence 54, 319–380 (1992)

4. Holzmann, G.: The SPIN model checker, Primer and Reference Manual. Addison-Wesley, Reading (2003)
5. Janssen, W., Mateescu, R.: Verifying business processes using spin. In: Proceedings of the 4th International SPIN Workshop (1998)
6. Nissen, M., Levitt, R.: Dynamic models of knowledge flow dynamics, center for integrated facility engineering, stanford university, working paper (manuscript, 2002)
7. Nonaka, I.: Toward middle-up-down management: accelerating information creation. Sloan Management Review 29, 9–18 (1998)
8. Polanyi, M.: Knowing and Being. University of Chicago Press (1969)
9. Stanford University. The virtual design team research group
10. Walton, C.D.: Model Checking Agent Dialogues. In: Leite, J., Omicini, A., Torroni, P., Yolum, p. (eds.) DALT 2004. LNCS(LNAI), vol. 3476, pp. 132–147. Springer, Heidelberg (2005)
11. Walton, C.: Model checking for multi-agent systems. Journal of Applied Logic, Special Issue on Logic-Based Agent Verification (2006)

Forming Buyer Coalitions with Bundles of Items

Laor Boongasame[1], Ho-fung Leung[2], Veera Boonjing[3], and Dickson K.W. Chiu[4]

[1] Department of Computer Engineering, Bangkok University, Bangkok, Thailand
[2] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, P.R. China
[3] Department of Mathematics and Computer Science, King Mongkut's Institue of Technology Ladkrabang, Bangkok, Thailand
[4] Dickson Computer Systems, 7 Victory Avenue, Kowloon, Hong Kong, P.R. China

laor.b@bu.ac.th,lhf@cse.cuhk.edu.hk,dicksonchiu@ieee.org

Abstract. There are several existing buyer coalition schemes, but they do not consider bundles of items. This research presents an algorithm for forming a buyer coalition with bundles of items, called the GroupBuyPackage scheme, in order to maximize the total discount. Our simulation results show that the total discount of the coalitions in this scheme are close to that in the optimal scheme.

Keywords: group buying, coalition formation, cooperative game theory, bundles of items.

1 Introduction

A buyer coalition is a group of buyers who join together to negotiate with sellers for a bulk purchasing of items at a larger discount [7]. Buyer coalitions have expanded rapidly on the Internet because it helps reduce communication costs and increase traders' profits in joining a coalition. Buyers can benefit from purchasing the items in large lots or bundles of items through buyer coalitions if the price of the lots or price of the bundles of items is less than the normal retail price. Additionally, the buyer coalitions may help to reduce cost of stock of the items in case that the items are not produced yet. On the other hand, sellers can benefit from selling the items at large lots or bundles of items[1] via buyer coalitions if the cost of wholesale marketing (such as the advertising or bidding costs) is less than that of the retail marketing [7] or their market power are preserved or increased [3].

[1] Bundles of items refers to the practice of selling two or more goods together in a package at a price that is below the sum of the independent prices [3].

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 714–723, 2009.

There are several existing buyer coalition schemes [4, 5, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17]. However, these schemes do not consider forming a buyer coalition with bundles of items. This practice can be observed very often in the real world such as restaurants (e.g., McDonald's Happy Meal), durable consumer goods (e.g., personal computer options), and non-durable consumer goods (e.g., dishwasher detergent and rinse aid packages).

Consider the following motivating example that motivates this research. There are three buyers b_1, b_2 , and b_3 who want to purchase any item (g_1 or g_2) within the bundle of items $Pk = \{g_1, g_2\}$. The price of this bundle of items is \$1800, while the retail price of g_1 and g_2 is each \$1000. Buyer b_1 needs one unit of item g_1 at most \$950. Buyer b_2 needs a unit of item g_2 at most \$900, while buyer b_3 needs a unit of item g_2 at most \$925. The total discount, the difference between the total reservation prices of items of all buyers and the cost of the items, of $\{b_1,b_2\}$ and $\{b_1,b_3\}$ are \$50 and \$75 respectively. So, the coalition structure $\{b_1,b_2\}$ that gives the maximum total discount should be formed. Based on this motivating example, we present an algorithm for forming a buyer coalition with bundles of items, called the GroupBuyPackage scheme, which aims at maximizing the total discount. The effectiveness of this scheme is demonstrated by the results of simulation. To guarantee satisfactory performance of the GroupBuyPackages scheme, we compare our results with those of the optimal scheme.

This rest of the paper is organized as follows. Section 2 compares related work. Section 3 details the design of our GroupBuyPackages scheme. Section 4 presents the simulation results before we discuss the conclusion with our future works in Section 5.

2 Related Works

2.1 The Research Concerning Forming a Buyer Coalition

There are several existing buyer coalition schemes. The mentioned scheme of GroupBuyAuction [17] forms a buyer coalition based on item categories. The Combinatorial Coalition Formation scheme [13] considers an e-market where buyers bid on item combinations with reservation prices while sellers offer price discounts for each item based on volume. Hyodo, Matsuo, and Ito [8] optimally allocate buyers to several group buying sites that are selling similar (or the same) goods by a genetic algorithm. Matsuo, Ito, and Shintani [15] integrate buyers with multi-attribute preferences (utility) into a coalition so that the system supports buyers' decision-making by using an analytic hierarchy process. He and Ioerger [4] assume that each buyer needs to buy different goods as a bundle, while sellers offer discount policies based on the total cost of goods sold in one transaction. They propose some optimal purchasing strategies that maximized the discount to buyers. Further, they present a buyer coalition scheme [16] where each buyer places a bid for all possible sets of items with reservation prices, while sellers offer price discounts for each item based on volume.

2.2 Game Theory and the Set Covering Problem

There are several theories such as the stable set [9], the core [2], the shapely value [6], and the bargain set [1] concerning the formation of a coalition. The main activities in forming coalitions are classified into two steps. First, members in a group of buyers are selected into a coalition. Second, the total discount of the coalition is divided among the buyers in the coalition. These theories focus mainly only on the stability in dividing the total discount, but not on designing any practical algorithm.

The problem of buyers coalitions with bundles of items addressed in this paper is similar to the Set Covering problem. The Set Cover is one of the oldest and most studied problem-solving in the area of combinatorial optimization and game theory [2,5,12]. Nevertheless, the Set Cover is inapplicable in this research because it is NP-hard.

3 Forming Buyer Coalitions with Bundles of Items

3.1 Example 1

The purchasing office of an enterprise, called Bangkok, wants to get a discount on purchasing bundles of items for all its subsidiaries. Therefore, when any subsidiary orders some items within a bundle of items, the purchasing office will gather these orders to purchase such bundles of items with a larger discount. Suppose that some subsidiaries want to purchase computer equipment of a particular computer package. The price list for the computer packages are shown in Table 1. For instance, the unit price of printer, CPU, Monitor, or RAM is $1000 each. When the printer, CPU, and RAM are sold together, the price is $2700 per package. Table 2 shows the subsidiaries' required computer equipment and their reservation prices, i.e., the maximum price that the buyer b_k is willing to pay for a unit of each item. For instance, subsidiary Bangkok-B wants to purchase a unit of CPU at $900 or lower and a unit of monitor at $900 or lower.

Table 1. Price list for individual computer equipment and packages

Item	Package / Item	Unit price ($)
1	Printer, CPU, Monitor	2700
2	Printer, CPU, RAM	2700
3	Printer	1000
4	CPU	1000
5	Monitor	1000
6	RAM	1000

Table 2. Subsidiaries' required computer equipments and their reservation prices

Subsidiary	rs_i of Printer	rs_i of CPU	rs_i of Monitor	rs_i of RAM
Bkk-A	1000			
Bkk-B		900	900	
Bkk-C	800	1000		800
Bkk-D	900		1000	900

3.2 Problem Formulation

Given a set of bundles of items $Pk = \{pk_1, pk_2, ..., pk_m\}$, each bundle of items $pk_l \in Pk$ is a component of items $Gd_l = \{gd^l_1, gd^l_2, ..., gd^l_{ql}\}$ and its unit price Pr_l. Additionally, given a set of buyers $B = \{b_1, b_2, ..., b_k\}$ in a community, each buyer $b_i \in B$ wants to purchase only one unit of each item within a set of items $Gd_i = \{gd^i_1, gd^i_2, ..., gd^i_{ti}\}$. Each buyer $b_i \in B$ places a set of reservation prices $Rs_i = \{rs^i_1, rs^i_2, ..., rs^i_{ti}\}$. A buyer can place only a set of seal bids to an honest coordinator Ł. Different buyers generally want different items and have different bids. A coalition C is a subset of buyers $C \subseteq B$ who want to join together to purchase the bundles of items pk_l with a larger discount. The coalition C has a set of items G_C, which is the sum of the items that the coalition buyers want. The coalition C can be formed to purchase bundles of items pk_l only if the set of items of the coalition members satisfies $\exists gd^i_h \in Gd_l$, when $b_i \in C$ and $gd_h \in Gd_l$.

The problem that we solve is forming buyer coalitions $C_i \subseteq B$ for purchasing item packages $pk_l \in Pk$, such that $\forall b_i \in B$ can purchase required items and $\sum_i V_i$ is maximal, where V_i is the joint utility that members of C_i can reach by cooperating via coalitional activity for purchasing a specific bundles of items. Formally, the utility V of a coaliton C in purchasing item packages pk_l is defined as $V_{cl} = \sum rs^i_h$ - Pr_l when 1) $rs^i_h \geq rs^j_h$ for $\forall b_j \in C$ and $gd_h \in pk_l$; and 2)$\forall b_i \in C$.

3.3 Algorithm

1. Calculate all the permutations that include up to k buyers. This is the set of all potential coalitions PC.
2. In each coalition, the coalitional potential items vector Gd^{PC}_C is calculated by summing up the unused items of the numbers of the coalition. Formally, $Gd^{PC}_C = \sum_{b_i \in C} Gd_i$.
3. For each bundle of items $pk_l \in Pk$, perform:
   - Check what items Gd_l are wanted for the satisfaction of pk_l.
   - Compare Gd_l with the sum of the unused items of the members of the coalition Gd^{PC}_C, thus finding the packages pk_l that can be satisfied by coalition C. Coalition C will be formed to purchase the package pk_l, if there is at least one member of coalition C that wants to purchase any item in the package pk_l or $\exists gd^i_h \in C$, when $b_i \in C$ and $gd_h \in Gd_l$.
   - V_{Cl} , a discount that the coalition C gets from forming to purchase the packages pk_l, will be calculated. Formally, $V_{cl} = \sum rs^i_h$ - Pr_l

when condition1) $rs_h^i \geq rs_h^j$ for $\forall b_j \in C$ and $gd_h \in pk_l$ and condition2) $\forall b_i \in C$.

4. V_{Cl} that gives the maximal will be chosen.
5. Update the items-vectors of all the members of $C^{"}$ according to their contribution to the package execution.

Example of condition 1. Based on the set of buyers intending to form a coalition to purchase any item package pk_l and their reservation price shown in Table 2, Bkk-B and Bkk-D want to purchase one unit of monitor within the bundle of items pk_l. Since the reservation price of buyer {Bkk-D} is higher than that of buyer {Bkk-B} or $rs_{Monitor}^{Bkk-D} \geq rs_{Monitor}^{Bkk-B}$, buyer {Bkk-D} will be selected to member of the coalition first.

Example of condition 2. From Table 2, the set of buyers {Bkk-B, Bkk-C, Bkk-D} want to form a coalition to purchase item package pk_1. {Bkk-D, Bkk-C} can be satisfied by condition 1 and therefore form a coalition to purchase the bundle of items pk_1. They can purchase items within package pk_1 as follows: 1) buyer Bkk-D can form the coalition to purchase one unit of printer; 2) buyer Bkk-C can form the coalition to purchase one unit of CPU; and 3) buyer Bkk-D can form the coalition to purchase one unit of monitor. However, only Bkk-C and Bkk-D in the set of buyers {Bkk-B, Bkk-C, Bkk-D} are selected in forming the coalition. Therefore, discount of such a coalition is not calculated. In other words, the discount for the set of buyers {Bkk-C, Bkk-D} is calculated instead of that of {Bkk-B, Bkk-C, Bkk-D}.

3.4 Example Revisited

Example 1 is considered again. From step 1 to step 5 of round 1, the algorithm gives the results of this round as shown in Table 3 and Table 4.

Table 3. The maximum utilities of potential coalitions

The set of potential coalitions PC	v_{cl}
Bkk-A	0
Bkk-B	-100
Bkk-C	0
Bkk-D	0
Bkk-A, Bkk-B	100
Bkk-A, Bkk-C	100
Bkk-A, Bkk-D	-700
Bkk-B, Bkk-C	0
Bkk-B, Bkk-D	100
Bkk-C, Bkk-D	200
Bkk-A, Bkk-B, Bkk-C	200
Bkk-A, Bkk-B, Bkk-D	200
Bkk-A, Bkk-C, Bkk-D	300

Table 4. Subsidiaries' required computer equipments and their reservation prices after the execution of round 1

Buyer—Items	rs of Printer	rs of CPU	rs of Monitor	rs of RAM
Bkk-A	~~1000~~			
Bkk-B		900	900	
Bkk-C	800	~~1000~~		800
Bkk-D	900		~~1000~~	900

Total discount $v_{tot} = 300(pk_1)$

We repeat this process until all buyers in the group of buyers can purchase items that they require as shown in Table 5.

Table 5. Subsidiaries' required computer equipments and their reservation prices after execution of round 5

Buyer—Items	rs of Printer	rs of CPU	rs of Monitor	rs of RAM
Bkk-A	~~1000~~			
Bkk-B		~~900~~	~~900~~	
Bkk-C	~~800~~	~~1000~~		~~800~~
Bkk-D	~~900~~		~~1000~~	~~900~~

$$v_{tot} = 300(pk_1) + 0(pk_1) + (-100)(pk_6) + (-200)(pk_3) + (-200)(pk_6)$$

4 Simulation

In this section, we present simulation results in order to evaluate the performance of our algorithm.

4.1 Setup of Experiments

We have conducted 100 of runs of the simulation for various simulation parameters such as the numbers of packages and the number of buyers. Table 6 summarizes the simulation parameters.

4.2 Results and Analysis

Fig. 1 compares the mean of the total discount of any coalition in the GroupBuyPackage scheme. The horizontal axis represents the average number of required items for each buyer / the average number of both item necessary for bundles of items and single item of the seller (IB/AIS). The vertical axis represents the mean of the total discount of any coalition in the GroupBuyPackage scheme. The results of the simulation are divided into three categories: 1) the number of buyers is smaller than the number of packages, 2) the number of buyers is equal to the number of packages, and 3) the number of buyers is greater than the number of packages.

Table 6. Simulation Parameters

		Parameter	Range	Range
Seller		The number of sellers		1
	Packages	The number of packages		6
		The number of single item		4
		The number of bundles of items		2
			The number of items necessary for each bundle of item	3
		The average number of both item necessary for bundles of items and single item of the seller (AIS)		2
Buyers		The number of buyers		2,4,6
	Required Items for each buyer	The average number of required items for each buyer (IB)		1,2,4

From Fig. 1, it is observed that the mean of the total discount of any coalition in the GroupBuyPackage scheme with the third category are higher than that in the GroupBuyPackage scheme with the first category and the second category.

Fig. 2 compares the ratio of the number of formed single item to one by bundles of items. The horizontal axis represents the average number of required items for each buyer / the average number of both item necessary for bundles of items and single item of the seller (IB/AIS). The vertical axis of this graph is the ratio of the number of formed single item to one by bundles of items.

From Fig. 2, it is observed that the ratio of the number of formed single item to one by bundles of items in the GroupBuyPackage scheme with the third category

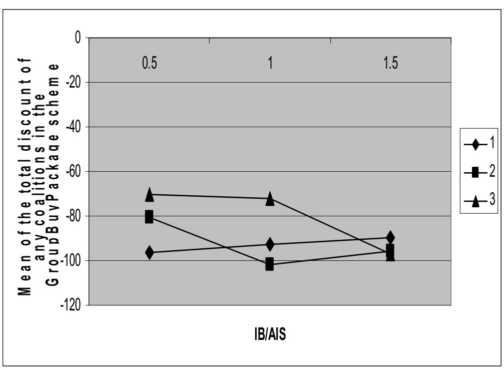


Fig. 1. Comparison of mean of the total discount of any coalition in the GroupBuy-Package scheme

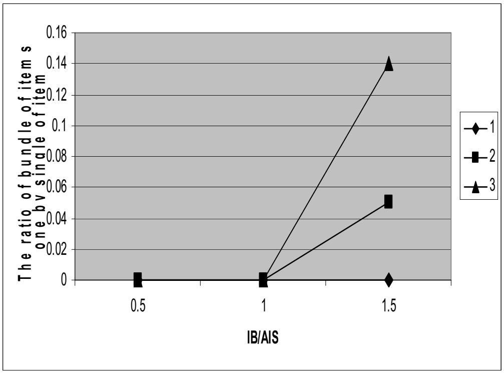


Fig. 2. Comparison of the ration of the number of formed bundles of items to one by single item

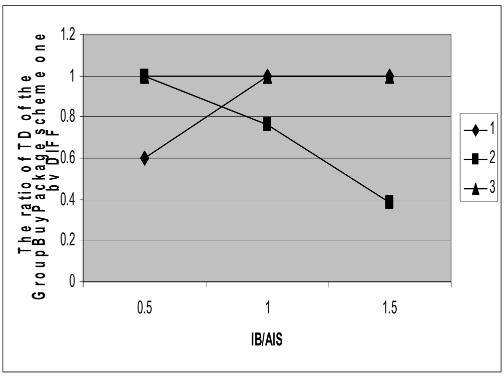


Fig. 3. The performance ratio of TD of the GroupBuyPackage scheme to DIFF

is higher than that in the GroupBuyPackage scheme with the first category and the second category.

From both graphs, we conclude that the total discounts of any coalition in this schemes is high as the number of buyers were more than the number of packages (third category). This is because the ratio of the number of formed single item to one by bundles of items in the GroupBuyPackage scheme with the third category is higher than that in the GroupBuyPackage scheme with the first category and the second category.

We denote the mean of the total discount of any coalition as TD and the difference between TD of the GroupBuyPackage scheme and TD of the optimal scheme as DIFF. The performance ratio of TD of the GroupBuyPackage scheme to DIFF is illustrated in Fig. 3. The horizontal axis represents IB/AIS, while the vertical axis represents the ratio of TD of the GroupBuyPackage scheme to that by DIFF. The value 1 means that the two schemes have the same performance; a value below 1 indicates that the optimal scheme is better; and a value above 1 shows the opposite.

From Fig. 3, it is observed that the ratio of TD of the GroupBuyPackage scheme to one by DIFF is close to one on all values of IB/AIS. In other words, the mean of the total discount of any coalition of the GroupBuyPackage scheme is close to that of the optimal scheme.

5 Discussion and Conclusion

In this paper, we present a new algorithm for forming buyer coalitions with bundles of items, called the GroupBuyPackage scheme. This algorithm is suitable for cases where buyers cooperate in order to maximize a total discount, especially when individual buyers cannot buy a whole bundle of items by themselves. Nevertheless, they may get more discounts (or utilities) when the discounts from buying the items individually is lower than the discounts from purchasing bundles of items. This algorithm can be stopped at any time to supply a solution though the quality of the solution increases with computation time.

To guarantee the performance of this algorithm, we compare its results with those of the optimal algorithm. In the simulation, main effective forming factors were IB, and AIS. From Fig. 1 - 3, the total discount of any coalition in this algorithm is close to that in the optimal algorithm in almost all values of IB and AIS.

However, this algorithm has some restrictions for forming a buyer coalition as follows: 1) the buyer coalition is formed concerning only the price attribute; 2) each buyer can purchase only one unit of the item; 3) there is only one seller who can supply unlimited units of the items; 4) the buyer coalition is suitable for only the case where the difference between discounts from purchasing a bundle of items and those from buying the items individually is high; 5) the period of time for forming the buyer coalition is not limited, therefore, this parameter should be considered in future research because it may change the results of forming the buyer coalition; and 6) the bundling in this algorithm is mixed bundling in which the bundle items are also sold separately. In our future research, pure bundling, whereby only the item bundle is offered and the individual items in the bundle cannot be purchased on their own, is considered in forming the buyer coalition. We are working on overcoming these restrictions in future research.

References

1. Aumann, R.J., Maschler, M.: The Bargaining Set for Cooperative Games. J. Advances in Game Theory (Annals Of Mathematical Studies 52), 443–476 (1964)
2. Gillies, D.B.: Solutions to General Non-Zero-Sum Games. J. Theory of Games (Annals Of Mathematical Studies 40) 5, 47–85 (1959)
3. Gurler, U., Oztop, S., Sen, A.: Optimal Bundle Formation and Pricing of two products with limited stock. J. International Journal of Production Economics (2008)

4. He, L., Ioerger, T.: Combining Bundle Search with Buyer Coalition Formation in Electronic Markets: A Distributed Approach through Explicit Negotiation. J. Electronic Commerce Research and Applications 4, 329–344 (2005)
5. Anand, K.S., Aron, R.: Group Buying on the Web: A Comparison of Price-Discovery Mechanisms. J. Management Science 49, 1546–1562 (2003)
6. Shapley, L.S.: A Value of N-Person Games. J. Theory of Games (Annals Of Mathematical Studies 28) 2, 307–317 (1953)
7. Tsvetovat, M., Sycara, K.P., Chen, Y., Ying, J.: Customer Coalitions in Electronic Markets. In: Dignum, F.P.M., Cortés, U. (eds.) AMEC 2000. LNCS(LNAI), vol. 2003, pp. 121–138. Springer, Heidelberg (2001)
8. Hyodo, M., Matsuo, T., Ito, T.: An Optimal Coalition Formation among Buyer Agents based on a Genetic Algorithm. In: Chung, P.W.H., Hinde, C.J., Ali, M. (eds.) IEA/AIE 2003. LNCS(LNAI), vol. 2718, pp. 151–157. Springer, Heidelberg (2003)
9. Von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behaviour. John Wiley and Sons, New York (1964)
10. Indrawan, M., Kijthaweesinpoon, T., Srinivasan, B., Sajeev, A.: Coalition Formation Protocol for E-Commernce. In: Proceedings of the International Conference on Intelligent Sensing and Information Processing, pp. 403–407 (2004)
11. Kraus, S., Shehory, O., Tasse, G.: Coalition Formation with Uncertain Heterogeneous Information. In: Proceedings of the Second international Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1–8 (2003)
12. Kraus, S., Shehory, O., Taase, G.: The Advantages of Compromising in Coalition Formation with Incomplete Information. In: Proceedings of the Third international Joint Conference on Autonomous Agents and Multiagent Systems, New York, pp. 588–595 (2004)
13. Li, C., Sycara, K.: Algorithm for Combinatorial Coalition Formation and Payoff Division in an Electronic Marketplace. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 120–127 (2002)
14. Li, C., Rajan, U., Chawla, S., Sycara, K.: Mechanisms for Coalition Formation and Cost Sharing in an Electronic Marketplace. In: Proceedings of the 5th International Conference on Electronic Commerece, pp. 68–77 (2003)
15. Matsuo, T., Ito, T., Shintani, T.: A Buyers Integration Support System in Group Buying. In: Proceedings of the IEEE international Conference on E-Commerce Technology, pp. 111–118 (2004)
16. Matsuo, T., Ito, T., Shintani, T.: A Volume Discount-based Allocation Mechanism in Group Buying. In: Proceedings of the 2005 international workshop on data engineering issues in e-commerce, pp. 59–67 (2005)
17. Yamamoto, J., Sycara, K.: A Stable and Efficient Buyer Coalition Formation Scheme for E-Marketplaces. In: Proceedings of the 5th International Conference on Autonomous Agents, pp. 576–583 (2001)

Distributed Data Reduction through Agent Collaboration

Ireneusz Czarnowski

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

Abstract. Distributed data mining (DDM) is an important research area. One of the approaches suitable for the DDM is to select relevant local patterns from the distributed databases. Such patterns, often called prototypes, are subsequently merged to create a compact representation of the distributed data repositories. In the paper the local prototype selection is carried out independently at each site where instances and features are selected simultaneously by teams of agents. To assure obtaining homogenous prototypes the feature selection requires collaboration of agents. In the paper two agent collaboration strategies producing a common set of features are proposed and experimentally validated. The paper includes a detailed description of the proposed approach and a discussion of the computational experiment results.

1 Introduction

Traditional data mining algorithms are based on the assumption that all the training data can be pooled together in a centralized data repository. In the real life there are, however, numerous cases, where the data have to be physically and geographically distributed. In the real life it is often unrealistic or unfeasible to pool together the distributed data for centralized processing. Thus, applying traditional data mining tools to discover knowledge from the distributed data sources may not be possible [8].

Creating systems that automatically learn from distributed data sets is one of the current research focus in the field of the distributed data mining (DDM) [8]. The demand and, at the same time, opportunity for developing such systems is a result of rapid growth of the number and size of various business and scientific databases. Their users need automated methods and tools for extracting useful knowledge from massive and distributed data repositories.

This paper focuses on classification tasks understood as the process of finding a set of models (or functions) that describe and distinguish data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown [7]. In DDM the process of finding a classification model is referred to as a distributed learning.

The problem of learning from distributed data can be summarized as follows: Given data sets $D_1,\ldots,D_K$ stored in K separate sites, a set of hypothesis H, a performance criterion P, the learning algorithm L outputs a hypothesis $h \in H$ that optimize P.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 724–733, 2009.

In the pattern classification application, h is a classifier (i.e. decision tree, artificial neural network, naive Bayes, k-nearest neighbor, etc.). Each data set D_i (where i=1,…,K) consists training examples, also called training instances. Each example is described by a set of attributes A. The task of the distributed learner L is to output the hypothesis $h \in H$ that optimize performance criterion P (e.g. function of accuracy of classification, complexity of the hypothesis, classification cost or classification error) using data sets $D_1,\ldots,D_K$ located in K sites.

Recently, several approaches to distributed classification have been proposed. One of basic approaches to the distributed data mining is applying the technique known as the meta-learning. This technique assumes combining the global classifier from independent local classifier, where each one local classifier is learned from the separated data set [10].

The other approach to learning from distributed data sets is based on moving all of the data to a central site, merging the data and building a single global model. However, moving all data into a centralized location may not be feasible due to restricted communication bandwidth among sites or high expenses involved, or some other reasons. Selecting out of the distributed databases only the relevant data can eliminate or reduce the above restrictions and speed up the global knowledge extraction process. Selection of the relevant data is the process often referred to as the data reduction with an objective to find patterns, also called prototypes or references vectors, or regularities within certain attributes (see, for example [9]). Thus, the goal of data reduction approaches is to reduce the number of instances in each of the distributed data subsets, without loss of the extractable information and where the datasets $D_1,\ldots,D_K$ are replaced by the reduced sets $S_1 \subset D_1,\ldots,S_K \subset D_k$ of local prototypes. The reduced datasets enable either pooling the data together and using some mono-database mining tools or effectively applying some meta-learning techniques.

The process of learning from the distributed data can be further complicated by differences between attributes among the distributed datasets. Differences between attribute sets can be also introduced during the data reduction process, when prototypes are selected by simultaneously reducing data set in the two dimensions through selecting instances and removing irrelevant attributes at each local site. In such case data reduction can result in the reduced datasets, which are not necessarily homogenous. Formally, let $A_1,\ldots,A_K$ denote sets of attributes which values are stored at the distributed sites $1,\ldots,K$ respectively. Obviously, there is a possibility that some attributes are shared across more then one reduced data set S_i, where $i=1,\ldots,K$.

If the reduced datasets are heterogeneous in the above sense then some techniques would be required to deal with such situation. One possible approach is applying a special combiner strategy which is responsible for combining and integrating a number of classifiers learned from heterogeneous sets of prototypes available at the global level (see comparison of various strategies for combining classifiers in [3]).

Another possible approach is to assure that prototypes obtained at each local site are homogenous in the discussed sense, that is are characterized by identical set of attributes. In such case, if at the local level sites the data reduction process including simultaneously reducing data set in the two dimensions is carried out, a special technique for assuring the selection of homogenous prototypes is needed. Such technique can be based, for example, on the distributed feature discrimination test using tree induction algorithm supported by some special agent-mediator whose role is to facilitate the communication process between agents involved in the distributed data mining process [2].

In this paper the distributed data reduction through agent collaboration is proposed. The approach is to extend the agent-based distributed learning classifier proposed in [4] adding features that allow collaboration between agents with a view to obtain the homogenous set of locally selected prototypes. The proposed solution involves two stages. At the local level the selection of prototypes from distributed data is carried out. Prototypes are selected by simultaneously reducing data set in two dimensions through selecting reference instances and removing irrelevant attributes. The instance selection is carried out independently at each site through applying the agent-based population search but the feature selection is managed and coordinated through the process of interaction and collaboration between agents.

The paper is organized as follows. Section 2 explains the proposed agent-based data reduction approach and provides details on how the interaction among agents is achieved. Section 3 provides the details on the computational experiment setup and discusses its results. Finally, the last section contains conclusions and suggestions for future research.

2 An Agent-Based Distributed Data Reduction

2.1 Main Features of the Proposed Approach

It is well known that instance reduction, feature selection and also learning classifier from distributed data are computationally difficult combinatorial optimization problems (see, for example [6], [12]). To overcome some of the difficulties posed by computational complexity of the distributed data reduction problem it is proposed to apply the population-based approach with optimization procedures implemented as an asynchronous team of agents (A-Team). The A-Team concept was originally introduced in [13]. The design of the A-Team architecture was motivated by other architectures used for optimization including blackboard systems and genetic algorithms. Within the A-Team multiple agents achieve an implicit cooperation by sharing a population of solutions, also called individuals, to the problem to be solved. An A-Team can be also defined as a set of agents and a set of memories, forming a network in which every agent remains in a closed loop. All the agents can work asynchronously and in parallel. Agents cooperate to construct, find and improve solutions which are read from the shared, common memory.

In our case the shared memory is used to store a population of solutions to the data reduction problem encountered at a given local site. Each solution is represented by the set of prototypes i.e. by the compact representations of the original datasets available at the considered site. A feasible solution to the data reduction problem at a local site is encoded as a string consisting of numbers of selected reference instances and numbers of selected features. The first part of such string, where numbers of selected instances are stored, is constructed at the initial population generation phase. When the initial population is generated at first, for each instance from the original set, the value of its similarity coefficient, proposed in [5], is calculated. Then instances with identical values of this coefficient are grouped into clusters. Further on, selection of the representation of instances through population-based search carried out by the team of agents for each cluster and removal of the remaining instances constitute basic steps of the first stage of the proposed procedure. The other part of the string representing a solution to the data reduction problem and containing numbers of the selected features is, at this stage, generated randomly. However, at the second stage a special agent called the feature manager is activated. Its role is to coordinate collaboration between other agents with a view to unify the set of selected features. Feature manager actions are described in the next subsection.

2.2 Agents Involved in Solving the Distributed Data Reduction Problem

All the required steps of the proposed approach are carried out by program agents of the four following types:

- global manager - agent responsible for managing the process of the distributed learning,
- optimizing agents and solution manager - agents responsible for instance and feature selection,
- feature manager - agent responsible for the feature selection coordination.

The proposed approach deals with several data reduction subproblems solved in parallel. Solutions to subproblems are not independent. Outcomes of the data reduction at local sites influence overall quality of the distributed learning. Hence, some form of coordination between solutions at local sites is required. The process of solving the data reduction problem is managed by the global manager, which is activated as the first one. The global manager is responsible for managing all stages of the distributed learning. At the first step it identifies the distributed data mining task that is to be solved and allocates optimizing agents to local sites using the available agent migration procedure. Than the global manager runs, in parallel, all subtasks, that is data reduction processes at local sites. It should be, however, noted that solving a data reduction task requires in fact obtaining solution to the independent learning classifiers problem.

When all the subtasks have been solved, solutions from the local level are used to obtain a global solution. Thus, the global manager creates the global set of prototypes by integrating local solutions and finally producing the global classifier, called also the meta-classifier.

The process of solving the data reduction is carried out by optimizing agents and the solution manager. Each optimizing agent is an implementation of certain improvement algorithm, and the problem of data reduction at local sites is solved by A-Team, that is a team of optimizing agents possibly of different kinds supervised by the solution manager.

To solve the data reduction problem four types optimizing agents representing different improvement procedures have been implemented. These procedures include: local search with tabu list for instance selection, simple local search for instance selection, local search with tabu list for feature selection and hybrid local search for simultaneous instance and feature selection. All of them have been proposed and discussed before in [4].

The solution manager, is responsible for organizing the data reduction process at a local site through managing the population of solutions called individuals and updating them when appropriate. Each solution manager is also responsible for selecting the best obtained solution for the supervised data reduction subproblem. The solution manager is also responsible for randomly generating the initial population of solutions and storing it in the common memory. During the data reduction process the solution manager continues reading individuals (solutions) from the common memory and storing them back after attempted improvement until a stopping criterion is met. During this process the solution manager keeps sending single individuals (solutions) from the common memory to optimizing agents. Solutions forwarded to optimizing agents for improvement are randomly drawn by the solution manager from the common memory. Each optimizing agent tries to improve quality of the received solution and afterwards sends it back to the solution manager, which, in turn, updates common memory by replacing a randomly selected individual with the improved one.

While data reduction subproblems are being solved in parallel the feature selection is coordinated by the special agent called feature manager. The feature manager is responsible for the final integration of features selected locally by optimizing agents. The feature manager actions include receiving candidate features form solution mangers, collecting the candidate features from all solutions to data reduction subproblems and finally deciding on the common set of features to be used at both – the local and the global levels. Such feature set will be called the winning set of features.

The feature manager performs its tasks in collaboration with solution managers supervising the data reduction processes at local sites. Such collaboration involves information exchange and decision making. Each solution manager, after having supervised a prescribed number of iterations within the data reduction process at a local site is obliged to send to the feature manager a set of the candidate features. This set contains features from the current best local solution. The value of fitness of this solution is also forwarded to the feature manager. This value is calculated as the estimated classification accuracy of the classifier induced from the reduced dataset at a local site. The number of iterations carried out by a local A-Team is the parameter set by the user. One iteration cycle covers all A-team activities between two successive replacements of a single solution from the common memory by the improved one transmitted from an optimizing agent.

After having receiving all candidate features from local sites the feature manager decides on the winning set of features. Its decision is based on a predefined strategy selected from the set of possible strategies by the user. Once the winning set of features have been chosen the feature manager passes the outcome of its decision to all solution managers, which role now is to update the respective individuals by correcting accordingly numbers of features in strings representing solutions in the current population.

Once all subtasks have been solved and the unified set of features selected and corrected at the local level, best solutions from this level are merged into the global dataset by the global manager.

The simplest strategy for deciding on the winning set of features is to accept all features that have been proposed at the local level. In such case the procedure of updating features is equivalent to adding to each string representing the best solution at a local site numbers of features that have been selected from the best solutions at other sites. The other strategy for deciding on the winning set of features is feature voting. In such case the winning set contains features that have been proposed by a majority

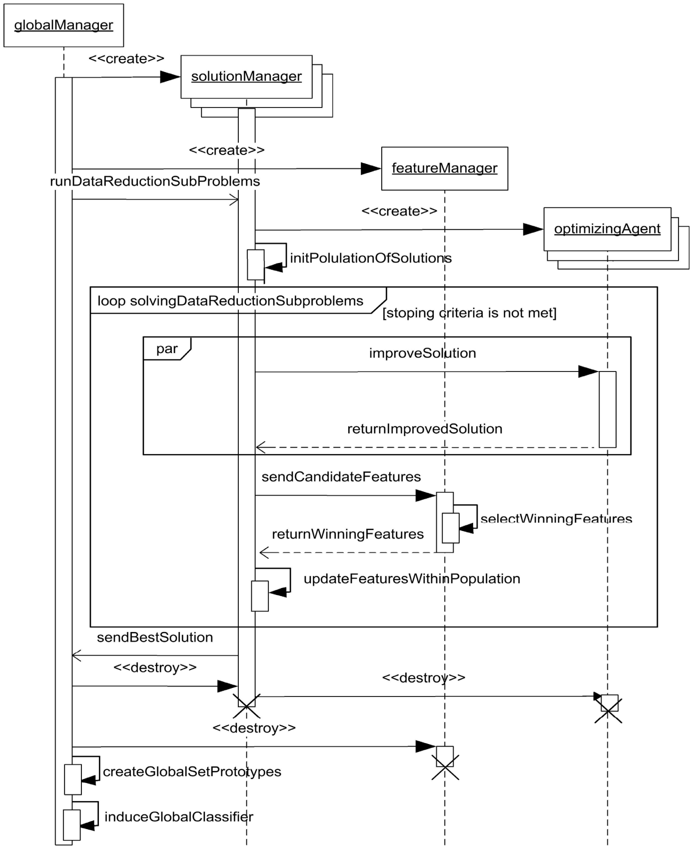


Fig. 1. Sequence diagram of agent interactions

of the local level solutions. Of course there are many other strategies possible. In the next section several example strategies for deciding on the winning set of features are discussed and evaluated.

Functional interactions between the agents used for solving the distributed data reduction problem are shown on the use case diagram in Fig. 1.

3 Computational Experiment

3.1 Computational Experiment Setting

The aim of the computation experiment was to evaluate how the choice of strategy for deciding on the winning set of features can influence the classification accuracy of the global classifier induced from the set of prototypes selected from autonomous distributed sites. It has been decided to implement and evaluate the following strategies:

– Strategy of pooling together prototypes selected from distributed databases through the simultaneous instance and feature selection and by using the global classifier computed by the combiner strategy based one simple feature voting (see [4]).
– Strategy named static feature selection shown in Fig. 2.
– Strategy named the dynamic feature selection as shown in Fig. 3.

```
begin
  // solution manager - i-th local level
  for each solution manager do
    Solve the data reduction subproblem using optimizing agents;
    Let βi=f(Ai) denote the value of the fitness function and Ai
    the set of candidate features from the best solution, where
    i=1,…,K;
    Send Ai and βi to the feature manager;
  end for
  // feature manager
  Select the winning set of features A^win: A^win = arg max_{Ai} Σ_{i=1}^{K} β(Ai);
  Return A^win to solution managers;
  // solution manager - i-th local level
  for each solution manager do
    Replace features from the best solution in the current
    population by features from A^win;
    Solve the instance reduction subproblem using optimizing agents;
    Select the best solution from the current population;
    Send the best solution to global manager;
  end for
  // global manager
  Create the global set of prototypes by integrating local
  solutions and produce the global classifier;
end
```

Fig. 2. Pseudo-code of the static feature selection strategy

3.2 Experiment Results and Their Discussion

Classification accuracy of the global classifiers obtained using the above described feature selection strategies have been compared with:

- results obtained by pooling together all instances from the distributed databases, without data reduction, into the centralized database,
- results obtained by pooling together all instances selected from distributed databases through the instanced reduction procedure only.

Generalization accuracy has been used as the performance criterion. The learning tool used was C 4.5 algorithm [11].

The experiment involved three datasets - C*ustomer* (24000 instances, 36 attributes, 2 classes), *Adult* (30162, 14, 2) and W*aveform* (30000, 21, 2). For the first two datasets the best known reported classification accuracies are respectively 75.53% and 84.46% [1], [14]. The above datasets have been obtained from [1] and [14]. The reported computational experiment was based on the ten cross validation approach. At

```
begin
 Set the temporary set of features: A^temp=∅;
 iteration=1;
 While !terminationCondition do
   // solution manager - i-th local level
   for each solution manager do
    if (iteration=1) then
      Solve the data reduction subproblem using all optimizing
      agents;
    else
      Solve the data reduction subproblem using only optimizing
      agents responsible for instance selection;
    end if
    Select the best solution from the current population;
    Let βi=f(Ai) denote the value of the  fitness function and Ai
    the set of candidate features from the best solution, where
    i=1,…,K;
    Send Ai and βi to the feature manager;
   end for
   // feature manager
   Select the winning set of features A^win:  A^win = arg max_{Ai} Σ_{i=1}^{K} β(Ai) ;

   if (A^win= A^temp and random()<0.5) then
     Select randomly {a}⊂A^win;
     Select randomly {a'}⊂A', where A' = ( ∪_{i∈{1,...,K}} Ai ) \ A^win ;

     Modify the set of winning features: A^win= (A^win \{a})∪{a'};
   end if
   Update the temporary set of features: A^temp= A^win;
   Return A^win to solution managers;
   // solution manager - i-th local level
   Replace features from the best solution in the current
   population by features from A^win;
   iteration++;
 end while
 // solution manager - i-th local level
 for each solution manager do
   Select the best solution from the current population;
   Send the best solution to the global manager;
 end for
 // global manager
 Create the global set of prototypes by integrating local
 solutions and produce the global classifier;
end
```

Fig. 3. Pseudo-code of the dynamic feature selection strategy

first, the available datasets have been randomly divided into the training and test sets in approximately 9/10 and 1/10 proportions. The second step involved the random partition of the previously generated training sets into the training subsets each representing a different dataset placed in the separate location. Next, each of the obtained datasets has been reduced. The reduced subsets have been then used to compute the global classifier using the proposed strategies. The above scheme was repeated ten times, using a different dataset partition as the test set for each trial.

Table 1. Average accuracy (%) of C 4.5 results obtained for the distributed datasets

Problem	Number of datasets			
	2	3	4	5
	Selection of references instances at the local level			
Customer	68.45±0.98	70.40±0.76	74.70±2.12	75.21±0.70
Adult	86.20±0.67	87.20±0.45	86.81±0.51	87.10±0.32
Waveform	75.52±0.72	77.61±0.87	78.32±0.45	80.67±0.70
average	76.72	78.40	79.94	80.99
	Strategy based on a simple feature voting			
Customer	69.10±0.98	73.43±0.72	75.35±0.53	77.20±0.49
Adult	88.90±0.41	87.45±0.31	91.13±0.23	91.58±0.41
Waveform	80.12±1.03	82.46±0.98	85.04±0.73	83.84±0.64
average	79.37	81.11	83.84	84.21
	Static feature selection strategy			
Customer	70.12±1.28	71.22±1.46	72.10±1.10	73.21±0.70
Adult	86.00±1.02	85.30±1.15	87.10±0.90	87.00±0.90
Waveform	76.12±0.94	78.21±0.91	78.32±1.00	80.37±1.10
average	77.41	78.24	79.17	80.70
	Dynamic feature selection strategy			
Customer	71.14±0.38	72.12±1.10	73.50±1.30	74.21±1.30
Adult	87.00±0.90	87.10±1.40	88.56±1.10	89.15±1.30
Waveform	78.43±1.24	79.40±1.20	82.08±0.78	81.92±1.40
average	78.86	79.54	81.38	81.76

The above scheme has been repeated four times for the four different partitions of the training set into a multi-database. The original data set was randomly partitioned into, respectively, the 2, 3, 4 and 5 multi-databases of approximately similar size.

The experiment results are shown in Table 1. It should be noted that the proposed strategies produce reasonable to very good results as compared with the case where, at the local level, the prototypes are selected only by instance selection and with the case where all instances are pooled together. Out of the investigated feature selection strategies the one based on a simple feature voting has performed best.

4 Conclusion

Computational experiment results confirmed that collaboration between agents involved locally in the data reduction process can result in improvement of the distributed data mining quality. The extend of such an improvement depends on the choice of strategy used for integrating locally selected reference instances. Since, however, the experiment involved only three datasets it would be premature to draw any valid conclusion as to the superiority of any single strategy out of the investigated ones. The author's belief is that proposed agents' interaction framework gives possibility to extend and validate even more advanced strategies for choosing the appropriate winning feature set common for all the distributed sites. This should be the focus for future research.

Acknowledgments. This research has been supported by the Polish Ministry of Science and Higher Education with grant for years 2008-2010.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Bala, J., Weng, Y., Williams, A., Gogia, B.K., Lesser, H.K.: Application of Distributed Mining Techniques for Knowledge Discovery Dispersed Sensory Data. In: 7th Joint Conference on Information Science, JCIS 2003, Cary, USA (2003)
3. Czarnowski, I., Jędrzejowicz, P.: A Comparison Study of Strategies for Combining Classfiers from Distributed Data Sources. In: International Conference on Adaptive and Natural Computing, ICANNGA 2009, Kuopio, Finland (in print, 2009)
4. Czarnowski, I., Jędrzejowicz, P.: Distributed Learning Algorithm Based on Data Reduction. In: Proceeding of the 1st International Conference on Agents and Artificial Intelligence (ICAART), Porto, pp. 189–203 (2009)
5. Czarnowski, I., Jędrzejowicz, P.: An Approach to Instance Reduction in Supervised Learning. In: Coenen, F., Preece, A., Macintosh, A. (eds.) Research and Development in Intelligent Systems XX, pp. 267–282. Springer, London (2004)
6. Dash, M., Liu, H.: Feature Selection for Classification. Intelligence Data Analysis 1(3), 131–156 (1997)
7. Han, J., Kamber, M.: Data Mining. Concepts and Techniques. Academic Press, San Diego (2001)
8. Kargupta, H., Park, B.-H., Hershberger, D., Johnson, E.: Collective data mining: A new perspective toward distributed data analysis. In: Kargupta, H., Chan, P. (eds.) Accepted in the Advances in Distributed Data Mining. AAAI/MIT Press (1999)
9. Liu, H., Lu, H., Yao, J.: Identifying Relevant Databases for Multidatabase Mining. In: Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 210–221 (1998)
10. Prodromidis, A., Chan, P.K., Stolfo, S.J.: Meta-learning in Distributed Data Mining Systems: Issues and Approaches. In: Kargupta, H., Chan, P. (eds.) Advances in Distributed and Parallel Knowledge Discovery, ch. 3. AAAI/MIT Press (2000)
11. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, SanMateo, CA (1993)
12. Rozsypal, A., Kubat, M.: Selecting Representative Examples and Attributes by a Genetic Algorithm. Intelligent Data Analysis 7(4), 291–304 (2003)
13. Talukdar, S., Baerentzen, L., Gove, A., de Souza, P.: Asynchronous Teams: Co-operation Schemes for Autonomous, Computer-Based Agents, Technical Report EDRC 18-59-96, Carnegie Mellon University, Pittsburgh (1996)
14. The European Network of Excellence on Intelligence Technologies for Smart Adaptive Systems (EUNITE) – EUNITE World Competition in domain of Intelligent Technologies (2002), http://neuron.tuke.sk/competition2

Solving the RCPSP/max Problem by the Team of Agents

Piotr Jedrzejowicz and Ewa Ratajczak-Ropel

Department of Information Systems
Gdynia Maritime University, Poland
{pj,ewra}@am.gdynia.pl

Abstract. The paper proposes an agent-based approach to solving instances of the resource constrained project scheduling problem with minimal and maximal time lags (time windows). Since the problem is known to be NP-hard, the proposed solution takes advantage of the parallel and distributed computations abilities of the E-JABAT which is a middleware supporting the construction of the dedicated A-Team architectures based on the population-based approach. In this paper the E-JABAT-based A-Team architecture dedicated for solving the RCPSP/max problem instances is proposed and experimentally validated. The paper contains the RCPSP/max problem formulation, overview of the E-JABAT environment, a detailed description of the E-JABAT architecture implemented for solving the RCPSP/max problem instances and a discussion of the computational experiment results.

1 Introduction

The paper proposes an agent approach to solving instances of the resource constrained project scheduling problem with minimal and maximal time lags (time windows) known in the literature as the RCPSP/max problem. RCPSP/max has attracted a lot of attention and many exact and heuristic algorithms have been recently proposed for solving it (see for example [5], [11], [14]). Since the problem is known to be NP-hard the approaches proposed so far produce either approximate solutions or can be applied for solving instances of the limited size. Hence, searching for a more effective algorithms and solutions to the RCPSP/max problem is still a lively field of research. One of the promising directions of such research is to take advantage of the parallel and distributed computation abilities, which are the feature of the contemporary multiple-agent systems.

The multiple-agent systems are an important and intensively expanding area of research and development. There is a number of multiple-agent approaches proposed to solve different types of optimization problems. One of them is the concept of an asynchronous team (A-Team), originally introduced in [13]. The A-Team paradigm was used to develop the JADE-based environment for solving a variety of computationally hard optimization problems called E-JABAT ([2],[3]).

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 734–743, 2009.

E-JABAT is a middleware supporting the construction of the dedicated A-Team architectures based on the population-based approach. The proposed system is accessible via the Internet and gives a possibility of using open computational resources over the Web. Moreover, the mobile agents used in E-JABAT allow for decentralization of computations and use of multiple hardware platforms in parallel, resulting eventually in more effective use of the available resources and reduction of computation time.

In this paper the E-JABAT-based A-Team architecture dedicated for solving the RCPSP/max problem instances is proposed and experimentally validated. To solve instances of the RCPSP/max optimization agents are used. Optimization agents represent heuristic algorithms such as various local search procedures or path relinking algorithm. The approach has been validated experimentally. Section 2 of the paper contains the RCPSP/max problem formulation. Section 3 gives some details on E-JABAT environment. Section 4 provides details of the E-JABAT architecture implemented for solving the RCPSP/max problem instances. Section 5 describes computational experiment carried-out with a view to validate the proposed approach. Section 6 contains conclusions and suggestions for future research.

2 Problem Formulation

In the resource-constrained project scheduling problem with minimal and maximal time lags (RCPSP/max) a set of $n+2$ activities $V = \{0, 1, \ldots, n, n+1\}$ is considered. Each activity has to be processed without interruption to complete the project. The dummy activities 0 and $n+1$ represent the beginning and the end of the project. The duration of an activity j, $j = 1, \ldots, n$ is denoted by d_j where $d_0 = d_{n+1} = 0$. There are r renewable resource types. The availability of each resource type k in each time period is r_k units, $k = 1, \ldots, r$. Each activity j requires r_{jk} units of resource k during each period of its duration where $r_{1k} = r_{nk} = 0$, $k = 1, ..., r$. Each activity $j \in V$ has a start time s_j which is a decision variable. There are generalized precedence relations (temporal constraints) of the start-to-start type with time lags $s_j - s_i \geq \delta_{ij}$, $\delta_{ij} \in Z$, defined between the activities.

The structure of a project can be represented by an activity-on-node network $G = (V, A)$, where V is the set of activities and A is the set of precedence relationships. The objective is to find a schedule of activities starting times $S = [s_0, \ldots, s_{n+1}]$, where $s_0 = 0$ (project always begins at time zero) and resource constraints are satisfied, such that the schedule duration $T(S) = s_n + 1$ is minimized.

The RCPSP/max as an extension of the RCPSP belongs to the class of NP-hard optimization problems ([1], [4]). The objective is to find a makespan minimal schedule that meets the constraints imposed by the precedence relations and the limited resource availabilities.

3 E-JABAT Environment

E-JABAT is a middleware allowing to design and implement A-Team architectures for solving various combinatorial optimization problems. The problem-solving paradigm on which the proposed system is based can be best defined as the population-based approach.

E-JABAT produces solutions to combinatorial optimization problems using a set of optimization agents, each representing an improvement algorithm that is a procedure generating approximate solution to the problem at hand using some initial solution as an input. Each improvement algorithm when supplied with a potential solution to the problem at hand, tries to improve this solution. To escape getting trapped into the local optimum an initial population of solutions (individuals) is generated or constructed. Individuals forming an initial population are, at the following computation stages, improved by independently acting agents, thus increasing chances for reaching the global optimum. Main functionality of the proposed environment includes organizing and conducting the process of search for the best solution. It involves a sequence of the following steps:

– Generating an initial population of solutions and storing them in the common memory
– Activating optimization agents which apply solution improvement algorithms to solutions drawn from the common memory and store them back after the attempted improvement, using some user defined replacement strategy
– Continuing reading-improving-replacing cycle until a stopping criterion is met

To perform the above described cycle two main classes of agents are used. The first class called OptiAgent is a base class for all optimization agents. The second class called SolutionManager is used to create agents or classes of agents responsible for maintenance and updating individuals in the common memory. All agents act in parallel. Each OptiAgent is representing a single improvement algorithm (for example: simulated annealing, tabu search, genetic algorithm, other local search heuristics etc.). The main agents and their roles in the E-JABAT are presented as use case diagram in the Figure 1.

Other important classes in E-JABAT include: Task which represents an instance or a set of instances of the problem and Solution representing the solution. To initialize the agents and maintain the system the TaskManager class is used. To maintain different platforms the PlatformManager class is used. Objects of the above classes also act as agents. E-JABAT offers an opportunity of using the open computational resources over the Web. The use of mobile agents in E-JABAT can bring decentralization of computations, resulting in a more effective use of available resources and a reduction of the computation time. From the user point of view, E-JABAT is a web application which provides the functionality required to send in and solve optimization problems. After having registered to the system the user gets opportunity to solve any particular instance of the

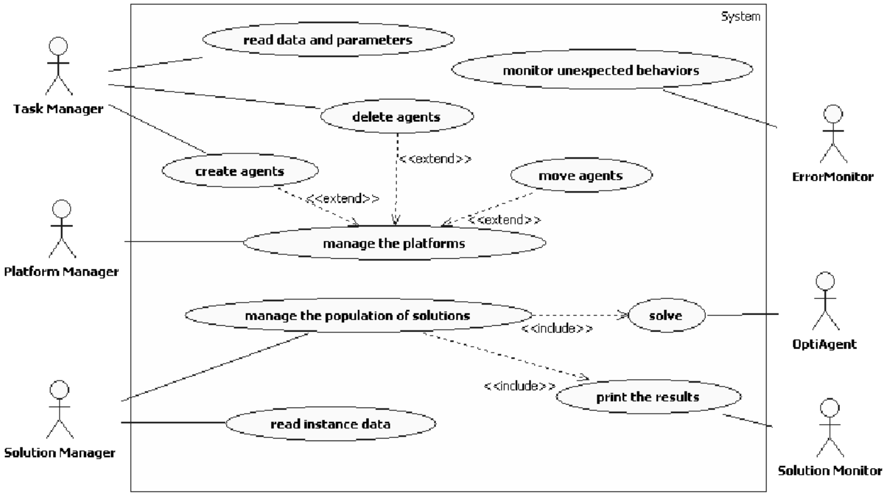


Fig. 1. E-JABAT use case diagram

problem providing a set of the optimizing agents have been designed and implemented. Using WWW interface the user can upload files with instance data and choose a set of parameters defining in what manner the search for a solution is carried out by the system. This, for example, may include a selection of optimization agents to be used and definition of the strategy used to maintain the population of solutions. The user provides these parameters using the special form available on the system Web page. After computation process has been stopped the report with results becomes accessible to the user on the Web page. Up to now the E-JABAT environment has been used to solve instances of the following problems: the resource-constrained project scheduling problem (RCPSP), the traveling salesman problem (TSP), the clustering problem (CP), the vehicle routing problem (VRP).

E-JABAT environment has been designed and implemented using JADE (Java Agent Development Framework), which is a software framework supporting the implementation of multi-agent systems. More detailed information about E-JABAT environment architecture and its implementations can be found in [8], [2] and [3].

4 E-JABAT Architecture to Solving the RCPSP/max Problem

E-JABAT environment was successfully used for solving RCPSP and MRCPSP problems (see [9]). Based on the above approach, the architecture designed and implemented to solve instances of the RCPSP/max is proposed in this paper. Development of such an architecture using the E-JABAT environment requires designing and coding of the set of the optimization agents which are specialized in solving the problem at hand. In addition, the user should define and implement the strategy of maintenance and evolution of the set of solutions stored in the common memory.

To solve the RCPSP/max problem instances the proposed architecture includes classes required to describe the problem, and optimization procedures searching for the improved solutions. All these have been implemented as E-JABAT objects. The above set of classes forms the package called RCPSPmax. The RCPSPmax contains the following classes:

– RCPSPmaxTask inheriting form the Task class,
– RCPSPmaxSolution inheriting from the Solution class,
– Activity class,
– Resource class.

The RCPSPmaxTask identifies instances which attributes include a list of activities, and a list of the required renewable and non-renewable resources. The RCPSPmaxSolution identifies solutions to RCPSPmax instances which attributes include a list of activities starting times. The Activity identifies activities, which attributes include the number and the lists of predecessors and successors. Finally, the Resource identifies both - available renewable and non-renewable resources, storing values representing numbers of the resource units. It should be noted that the classes described above are different from the classes implemented in [9] for solving RCPSP and MRCPSP instances and hence can not be inherited from them.

Additionally, the RCPSPmaxTaskOntology and RCPSPmaxSolutionOntology classes have been also defined through over-ridding the TaskOntology and SolutionOntology, respectively. Both - TaskOntology and SolutionOntology are provided by the designers of E-JABAT. Ontology is the class enabling definition of the vocabulary and semantics for the content of message exchange between agents.

To solve the RCPSPmax three optimization agents representing three different heuristic algorithms have been proposed and implemented:

– Local Search Agent (LSA),
– Path Relinking Agent (PRA),
– Gene Expression Programing Agent (GEPA).

Pseudo codes of the above algorithms are presented in Figures 2, 3, 4 respectively.

The LSA is a simple local search algorithm (Figure 2) which finds the local optimum by moving each activity to all possible places in the schedule. For each combination of activities the value of possible solution is calculated. The best schedule is returned.

The PRA is an implementation of the path-relinking algorithm (Figure 3). For a pair of solutions a path between them is constructed. The path consists of schedules obtained by carrying out a single move from the preceding schedule. The move is understood as moving one of the activities to a new position. For

```
LSA(InitialSolution, IterationNumber)
{
  Copy InitialSolution to s
  for(i=1; i<IterationNumber; i++)
    for(p1=0; p1<n-1; p1++)
      for(p2=1; p2<n; p2++)
      {
        Move activity from position p1 to position p2 in the schedule s
        Find solution for the schedule s
        Remember solution if it is better then the best one found so far
        Move activity from position p2 to position p1 in the schedule s
      }
}
```

Fig. 2. Pseudo code of LSA algorithm

```
PRA(InitialSolution1, InitialSolution2)
{
  Remember InitialSolution1 in s1
  Remember InitialSolution2 in s2
  for(p=1; p<n; p++)
  {
    Get activity i from position p in the schedule s1
    Find position r of activity i in the schedule s2
    Move activity i from position p to r in s1
    Find solution for schedule s1
    Remember solution if it is better then the best one found so far
  }
}
```

Fig. 3. Pseudo code of PRA algorithm

each schedule in the path the value of the possible solution is checked. The best schedule is remembered and finally returned.

The GEPA is an implementation of the gene expression program (Figure 4) based on the gene expression programming idea proposed in [6] and [7]. The initial population of the GEPA individuals is partly selected from the current content of the E-JABAT common memory where population of possible solutions is stored, and partly generated randomly. Individuals (solutions) from the E-JABAT population are transformed into GEP individual representation. Next, the whole population is evolved using GEP operators: one point, two point and n-point recombination, restricted permutation, inversion, gene deletion/insertion and mutation. Value of each of these parameters shows percentage of solutions evolved using the respective GEP operator. All the parameters are chosen experimentally. To calculate the fitness of an individual the function $f_x = T_g - t_x + 1$ is used, where x is the individual, t_x is the length of the schedule and T_g is the length of the largest schedule encoded in the chromosomes of the current population. The best schedule is remembered and finally returned.

```
GEPA(GenerationNumber, PopulationSize, InitialSolutions)
{
  Create initial population Pop of size PopulationSize-2
  Add InitialSolutions to Pop
  for(i=1; i<GenerationNumber; i++)
  {
    Pop.mutationConstruct(0.02)
    Pop.mutationLC(0.08)
    Pop.geneDelIns(0.04)
    Pop.inversion(0.04)
    Pop.inversion2g(0.02)
    Pop.restrictedPermutation(0.06)
    Pop.onePointRecombination(0.6)
    Pop.twoPointRecombination(0.8)
    Pop.nPointRecombination(0.8)
    Pop.geneRecombination(0.5)
    Remember the best individual
    Create next generation
  }
}
```

Fig. 4. Pseudo code of GEP algorithm

All optimization agents co-operate together using the E-JABAT common memory during the process of searching for the best solution to the problem instance. In the common memory the population of individuals (solutions) is stored. The initial population in the common memory is generated randomly. Because it might be difficult to obtain feasible solution for some RCPSP/max problem instances, the random drawing of an individual could be repeated several times. If this does not produce a feasible solution the unfeasible ones are added to the population in the common memory. All the proposed improvement algorithms can use unfeasible solutions as an initial seed. Individuals in the common memory are represented as schedules of activities. The final solution is obtained from the schedule by SGSU (Serial Generation Scheme with Unscheduling) [10]. The time and frequency an agent of each kind receives a solution or set of solutions from the common memory with a view to improve its quality is determined by the user defined strategy. For solving the RCPSP/max problem instances the basic strategy has been used where individuals forwarded to optimization agents for improvement are randomly chosen from the population stored in the common memory. Such individuals are being sent to these optimization agents which are ready to start searching for a better solution. After computation the improved individual replaces the worst one stored in the common memory. The set of configuration variables defining the user strategy includes the kind and number of optimization agents used, selection procedure, population size and computation time allowed.

5 Computational Experiment Results

To validate the proposed approach the computational experiment has been carried out using benchmark instances of RCPSP/max from PSPLIB [12]. The experiment involved computation with the fixed number of optimization agents representing LSA, PRA and GEPA. The results are shown in Table 1. The discussed results have been obtained using 6 optimization agents: two of the LSA, two of the PRA and two of the GEPA. At the beginning the population of solutions in the common memory consisting of 50 individuals has been generated randomly. In case of the GEPA two solutions are drawn by an optimization agent and they are incorporated into the population of 48 solutions generated internally by this agent. Such population is then evolved in 100 generations. If the best individual produced has been improved as compared with the solution/solutions originally drown from the common memory then the improved solution replaces the worse individual drawn from the E-JABAT common memory.

The computation results have been evaluated in terms of the mean relative error (Mean RE) calculated as the deviation from the lower bound, percent of feasible solutions (% FS) and mean computation time (Mean CT). Each instance has been solved three times and the results presented in Table 1 have been averaged over all solutions. The maximum computational time for one instance of the problem has, in all cases been not longer then 1 second during computations

Table 1. Experiment results obtained by the proposed E-JABAT based A-Team

Number of activities	Mean RE	% FS	Mean CT
10	0.62 %	96.48 %	0.46 s
20	4.11 %	95.02 %	0.64 s
30	9.05 %	83.91 %	0.91 s

Table 2. Results obtained by ISES [5]

Number of activities	Mean RE	% FS	Mean CT
10	0.99 %	100.00 %	0.71 s
20	4.99 %	100.00 %	4.48 s
30	10.37 %	100.00 %	22.68 s

Table 3. Results obtained by B&B [5]

Number of activities	Mean RE	% FS	Mean CT
10	0.00 %	100.00 %	–
20	4.29 %	100.00 %	–
30	9.56 %	98.92 %	–

Table 4. Results obtained by C-BEST [5]

Number of activities	Mean RE	% FS	Mean CT
10	0.00 %	100.00 %	–
20	3.97 %	100.00 %	–
30	8.91 %	100.00 %	–

on 2 PC's with 2.0 GHz processors. The results obtained by the E-JABAT can be compared with the results reported in the literature. Such a comparison are presented in Tables 2, 3, and 4.

6 Conclusions

Experiment results show that the proposed E-JABAT implementation is an effective tool for solving instances of the resource-constrained project scheduling problem with minimal and maximal time lags. Presented results are comparable with the best solutions known from the literature. It is worth noticing that the computation time using the proposed architecture is very short, even for more complex problem instances. Implementing the proposed approach is quite simple, but some effort is needed to select GEP operators and fine-tune their parameters. The most important disadvantage of this approach is possible lack of solutions in some cases. This can be partly eliminated by using more then one agent executing each of the algorithm.

Future research will concentrate on finding, testing and adding to E-JABAT more different optimization agents and eliminating a possibility of not finding a feasible solution at all. Some more experiments are needed to establish rules on finding an optimal configuration and number of optimization agents best suited to the available computational resources.

References

1. Bartusch, M., Mohring, R.H., Radermacher, F.J.: Scheduling project networks with resource constraints and time windows. Annual Operational Research 16, 201–240 (1988)
2. Barbucha, D., Czarnowski, I., Jedrzejowicz, P., Ratajczak, E., Wierzbowska, I.: JADE-Based A-Team as a Tool for Implementing Population-Based Algorithms. In: Proc. VI Int. Conf. on Intelligent Systems Design and Applications, vol. 3. IEEE Computer Society, Los Alamitos (2006)
3. Barbucha, D., Czarnowski, I., Jedrzejowicz, P., Ratajczak-Ropel, E., Wierzbowska, I.: E-JABAT – An Implementation of the Web-Based A-Team. In: Intelligent Agents in the Evolution of Web and Applications. Studies in Computational Intelligence, vol. 167, pp. 57–86. Springer, Heidelberg (2009)
4. Blazewicz, J., Lenstra, J., Rinnooy, A.: Scheduling subject to resource constraints: Classification and complexity. Discrete Applied Mathematics 5, 11–24 (1983)

5. Cesta, A., Oddi, A., Smith, S.F.: A Constraint-Based Method for Project Scheduling with Time Windows. Journal of Heuristics 8, 108–136 (2002)
6. Ferreira, C.: Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence, on-line book (2002), http://www.gene-expression-programming.com/GepBook/Introduction.htm
7. Ferreira, C.: Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence, 2nd edn. Springer, Heidelberg (2006)
8. Jedrzejowicz, P., Wierzbowska, I.: JADE-Based A-Team Environment. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3993, pp. 719–726. Springer, Heidelberg (2006)
9. Jedrzejowicz, P., Ratajczak, E.: Agent-Based Approach to Solving the Resource Constrained Project Scheduling Problem. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 480–487. Springer, Heidelberg (2007)
10. Neumann, K., Schwindt, C., Zimmermann, J.: Project Scheduling with Time Windows and Scarce Resources. Springer, Heidelberg (2002)
11. Neumann, K., Schwindt, C., Zimmermann, J.: Resource-Constrained project Scheduling with Time Windows, Recent developments and new applications. In: Perspectives in Modern Project Scheduling, pp. 375–407. Springer, Heidelberg (2006)
12. PSPLIB, http://129.187.106.231/psplib
13. Talukdar, S., Baerentzen, L., Gove, A., de Souza, P.: Asynchronous Teams: Cooperation Schemes for Autonomous, Computer-Based Agents, Technical Report EDRC 18-59-96, Carnegie Mellon University, Pittsburgh (1996)
14. Valls, V., Ballestin, F., Barrios, A.: An evolutionary algorithm for the resource-constrained project scheduling problem subject to temporal constraints. In: Proc. of PMS 2006, Tenth International Workshop on Project Management and Scheduling, Nakom, Poznan, pp. 363–369 (2006)

AGV Mind Model and Its Usage for Autonomous Decentralized FMS by Change of Mind

Hidehiko Yamamoto[1], Takayoshi Yamada[2], and Shinsuke Kato[3]

[1] Department of Human and Information System, Gifu University, Japan
yam-h@gifu-u.ac.jp
[2] yamat@gifu-u.ac.jp
[3] Hokuto Corporation, Japan

Abstract. We propose the mind modeling which is the complicated expression of combinations from the three elements: Unit, Load, Stimulation. The mind changes like the human mind behavior. The Unit has two states that are regular and active which can be changed. The change of Mind is expressed by the change of the Unit. Even if the same situation happens, it takes different actions when the mind is changed. By using the mind model for the autonomous decentralized FMS, it is ascertained that it can reduce the collisions between each AGV.

Keywords: Autonomous decentralized system, FMS, AGV, mind, collision avoidance.

1 Introduction

The actions of robots are made by pre-decided control rules. When many robots try to perform complicated tasks, it is difficult for them to cooperate with each others by using the pre-decided control rules. On the other hand, an aggregation of creatures can perform such complicated tasks by an analyze based on thinking. It is because the human beings have a mind that enables them to think by themselves [1-2].

Instead of using the pre-decided control rules, if we can provide a mind to the robots, they can cooperate to perform complicated operations and will be able to adapt to every situation. As a result, information processing cost will be decreased.

However, until today, it is difficult to provide to the robot such a mind because there is no clear definition of what a mind is. In this paper, we describe the basic concept of a mind. First, we propose the mind model. Second, by using the mind model, we solve an Automated Guided Vehicle (AGV) collision problem in an Autonomous Decentralized Flexible Manufacturing System (AD-FMS) as an application example of the mind model[3-4]. By providing the mind model to the AGVs, we will prove the correctness of the proposed mind.

We have still three collision avoidance research for AGVs. They are central control methods that consider paths of all AGV, limit the number of AGVs in its path and exchange each knowledge of AGVs[5-8]. The original of this study is that the AGV has

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 744–753, 2009.

a mind and repeat change of mind. In this meaning, our research is different from the conventional three methods. Also, the problem this paper deals is not solved by the conventional optimal method because the problem includes unexpected situations.

2 Mind Model

2.1 Basic Mind

It is said that a mind is defined as an abstract object which has an invisible intelligence, feelings, affectivity and wills[9]. A mind is not always kept in a constant condition, but it can change at any time. The operation of a mind can be realized through a physical output of the human being. To express a mind, the abstract object, which cannot be easily defined, we propose the mind model and express change of mind by using the mind model.

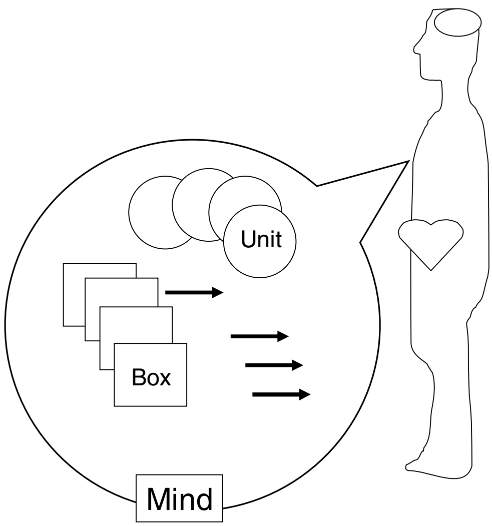


Fig. 1. Mind model

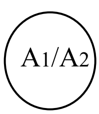


Fig. 2. Unit

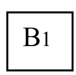


Fig. 3. Box

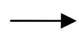


Fig. 4. Chain

We define the basic model of mind as a set of three elements, that is, the combinations of 3 elements consist of units, boxes and chains as shown in Fig. 1. The chain is the function that links between the units and the boxes. Next, we describe each element of the mind model. Fig. 2 shows the schematic figure of the unit that consists of two numerical values, an input value and a threshold value. A1 and A2, in Fig. 2, represent respectively the input value and the threshold value. The input value varies with a positive value while the threshold value is constant. The operation of the unit is described as follow:

When a signal is sent to the unit, the input value will be increased or decreased. The increase or decrease depends on the signal. If the input value exceeds the threshold value, it will be decreased to the threshold value and the unit will send an output signal in the direction of a chain arrow. In this way, the unit function is to receive an input signal and to send an output signal.

When the input and the threshold values are equal, the unit is called as an awakening condition. Meanwhile, when the input is lower than the threshold value, the unit is called as a sleeping condition.

The box shown in Fig. 3 keeps one value of number. The unit is linked with the box by a chain following an arrow direction. The input value of the unit is added or reduced by the box value. The chain function is to link between the unit and the box. It sends a signal to the unit or box by following the arrow direction as shown in Fig. 4.

In this way, by expressing a mind, we can define a mind and can express one of the characteristics of mind, that is change of mind. In general, change of mind suddenly comes without calculating something. To express this kind of change of mind, we adopt the mind changes from an awakening condition to a sleeping condition. Or, otherwise, from a sleeping condition to an awakening condition.

The mind model structure seems to be similar to the neural network (NN) model. However, they are different. Because information flows between each node of NN is one way. On the contrary, the node of the mind model has dynamic many ways from input to output.

2.2 Arrogant and Modest Mind of AGV

Although we have some types of change of mind, this paper deals with arrogant and modest mind as change of mind. The arrogant mind does something by force. On the contrary, the modest mind does not so. We adopt the two kinds of change of mind to AGV which moves in AD- FMS.

The AGV mind which carries out change of mind is shown in Fig. 5. The chain has the function to link between the units U1, U2 and the box C. The mind depends on the

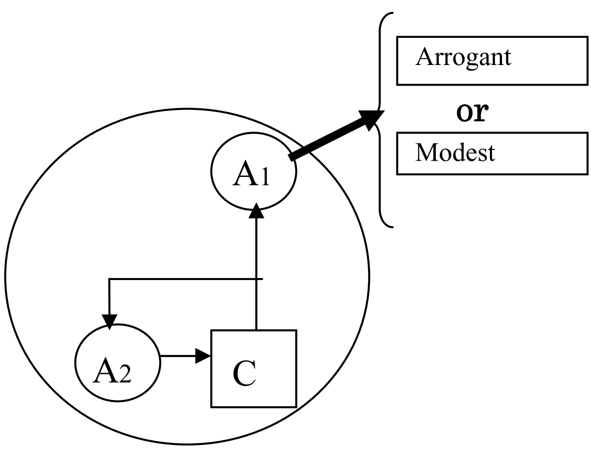


Fig. 5. Model of AGV's Mind

change of Unit A1 situations. The situations have two states, awake and asleep. When, A1 keeps an awakening condition, the AGV has a modest mind. The AGV with modest mind gives priority to the other AGVs and goes ahead without ignoring other AGVs moving. When A1 keeps a sleeping condition, an AGV has an arrogant mind. The AGV with the arrogant mind moves through in its way without giving priority to any AGVs. We try to control AGV movements smoothly by this change of mind. Hereafter, we call the AGV with the modest mind as the modest AGV and the AGV with the arrogant mind as the arrogant AGV.

3 Collision Avoidance by Change of Mind

In this chapter, we can explain how AGV mind can avoid the collisions between AGVs with the model of mind described above. The basics of collision avoidance of AGV with mind are the actions that the modest AGV gives the route to the arrogant AGV and the arrogant AGV advances first.

3.1 Arrogant AGV and Arrogant AGV

We explain how to avoid the collision in the case that two arrogant AGVs are facing each other. If the two AGVs that are arrogant AGV and modest AGV are getting close, the modest AGV gives the route to the arrogant AGV and the arrogant AGV advances first. However, if the two AGVs are both arrogant ones, they will be collided. By using the AGV mind as shown in Fig. 5, the collision will be avoided. Fig. 6 shows the two arrogant AGV-1 and AGV-2 are about to stop by the prediction of a collision. In this case, both arrogant AGVs send a signal to the unit A1 of its mind and as a result, the unit A1 gets the input value increase of 1. After a certain period of time, keeping the above situation, the arrogant AGV sends a signal to its unit A1. Once more, the input value of the unit A1 will get increased by 1. Because the arrogant AGV continues to send a signal to its unit A1, when the arrogant AGV keeps stopping, the input values and the threshold values become equal and one unit A1 varies from a sleeping condition to an awakening condition. As a result, an arrogant AGV becomes the modest AGV. Because the input values of unit A1 of AGV-1 are closer to the threshold value of its unit A1 than the input values of unit A1 of AGV-2 in Fig. 6, the unit A1 of AGV-1 becomes an awakening condition earlier than unit A1 of AGV-2. AGV-1 changes into the modest AGV from the arrogant AGV, and, by the work of the modest mind, AGV-1 leaves the way. Finally, AGV-2 has the right to pass first. In this way, the problem of a collision avoidance between arrogant AGVs can be solved, as shown in Fig. 7.

3.2 Modest AGV and Modest AGV

Next we explain how to avoid the collision in the case that the same modest AGVs are facing each other. Fig. 8 shows that the two modest AGV-1 and AGV-2 are about to stop by the work of the modest mind. In this case, both modest AGVs send a signal to the unit A2 of their own mind and, as a result, the unit A2 gets its input value increased by 1. After a certain period of time, keeping the above situation, the modest AGV sends a signal to its unit A2 again and the value of this unit will get increased by 1 once more.

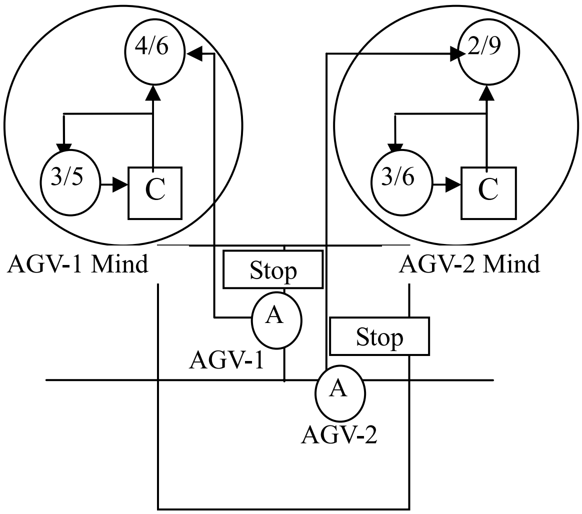


Fig. 6. Arrogant and Arrogant

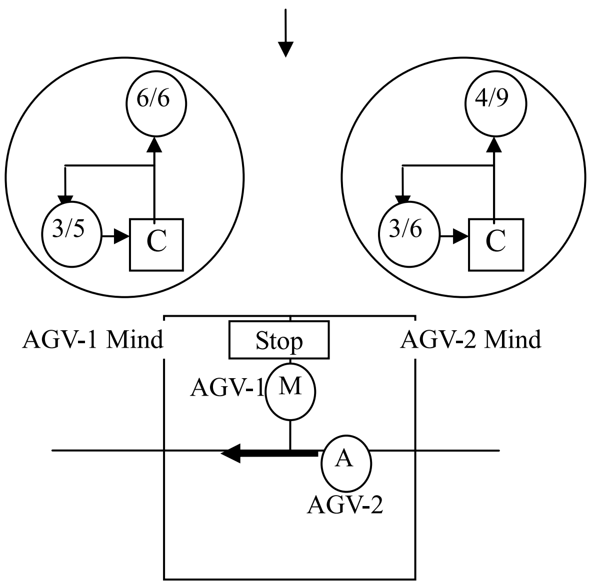


Fig. 7. Example 1 of change of mind

Because the modest AGV continues to send a signal to the unit A2 when the modest AGV keeps stopping, the input values and the threshold values become equal and the unit A2 varies from a sleeping condition to an awakening condition. Furthermore, when a signal is sent to the unit A2 of an awakening condition, the unit A2 sends a signal to box C. Whenever a signal is sent to the box C, it randomly chooses the value from -3 ~ -10 and the input values, for example, of the unit A1 and A2 are decreased by the

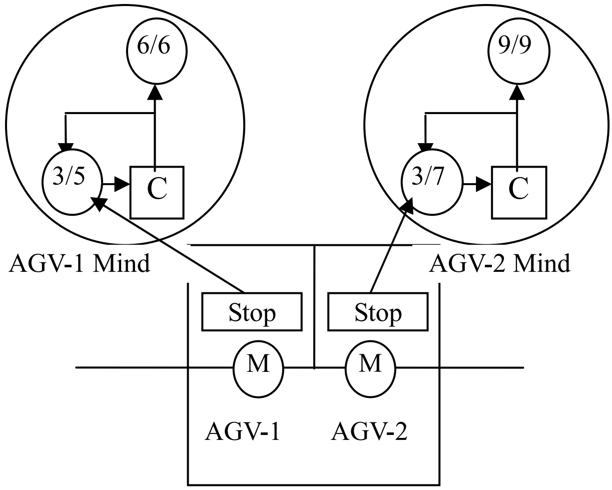


Fig. 8. Modest and Modest

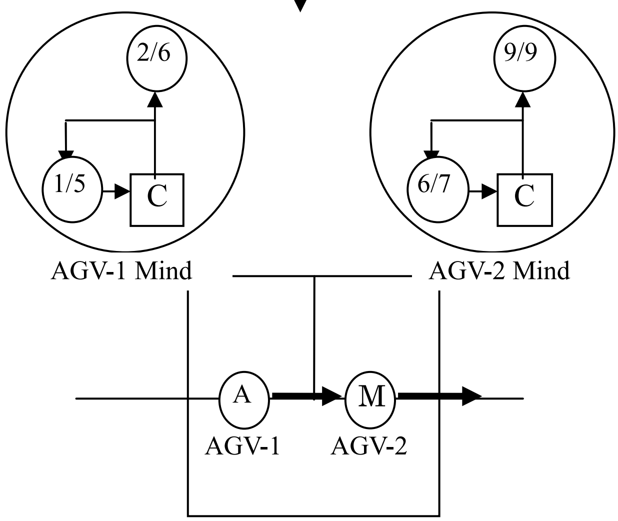


Fig. 9. Example 2 of change of mind

selected numerical values. As a result, the units A1 and A2 are changed from an awakening condition to a sleeping condition.

For example, Fig. 8 shows that the unit A2 of AGV-1 is in an awakening condition and when a signal is sent to Box C, it selects a numerical value of -4 that decreases the input value of the units A1 and A2 by – 4. As a result, the units A1 and A2 turn from an awakening condition to a sleeping condition. By this way, AGV-1 changes its character from the modest AGV to the arrogant AGV as shown Fig. 9. That means by using the mind change from the modest mind to the arrogant mind, collisions between two modest AGVs can be avoided.

Furthermore, when two AGVs that have the same threshold values and the input values are in stop situation, change of mind occurs at the same time but it cannot solve the stop situation. However, the difference of the input value between the AGVs that keep to stop occurs because the Box C decides to select a numerical value randomly. As a result, the problem can be solved.

4 Simulation Experiment

4.1 Simulation Condition

In this paper, we apply the mind that we proposed in the last section to AGVs in AD-FMS built on a computer and carried out the simulations with 3 production conditions. Parts Warehouse, Products Warehouse and MCs of FMS are located as show in Fig. 10. The black squares, ■, of Fig. 10 indicate Machining Center (MC) and the number of AGVs is changed by 3 different simulation states. Eight hours simulation time were adopted and the numbers of AGVs were 3, 4, 5 for each simulation state. The first of the 3 production conditions adopts three kinds of products P1, P2, P3 and their production ratio is P1:P2:P3=5:6:3.

The second of the 3 production conditions adopts six kinds of products P1, P2, P3, P4, P5, P6 and their production ratio is P1:P2:P3:P4:P5:P6=5:6:3:3:2:1.

The third of the 3 production conditions adopts nine kinds of products P1, P2, P3, P4, P5, P6, P7, P8, P9 and their production ratio is P1:P2:P3:P4:P5:P6:P7:P8:P9=5:
6:3:3:2:1:4:5:2.

4.2 Initialization of Mind

As the initial conditions of AGVs mind moving in AD-FMS, the threshold values of the units, A1 and A2 are randomly chosen from among 3~10 of an integer value and the input value of the units A1 and A2 was decided as 0 as shown in Fig. 11. In other words, because the units A1s of all AGVs are sleeping condition, the all AGVs start the simulations as the arrogant AGVs.

4.3 Simulation Results

We summarized the simulation result under the condition in section 5.1 in Table 1 and Table 2. Table 1 shows the result of the collision avoidance without using a mind for AGV[10]. On the other hand, table 2 shows the result of the collision avoidance with using change of mind for AGVs. The condition1-1, 1-2,1-3 in each table show condition1 in Fig. 10. The condition2-1, 2-2, 2-3 in each table show condition2 in Fig. 10 and the condition3-1, 3-2, 3-3 show condition3. For each condition, we ran 3 types of simulations with different numbers of AGVs that were 3, 4, 5. For condition-1, the conditions 1-1, 1-2 and 1-3 imply for 3, 4, 5 AGVs respectively. Similarly, for condition-2, the conditions 2-1, 2-2 and 2-3 imply using 3, 4, 5 AGVs respectively, and for condition-3, the conditions 3-1, 3-2, and 3-3 imply using 3, 4, 5 AGVs respectively.

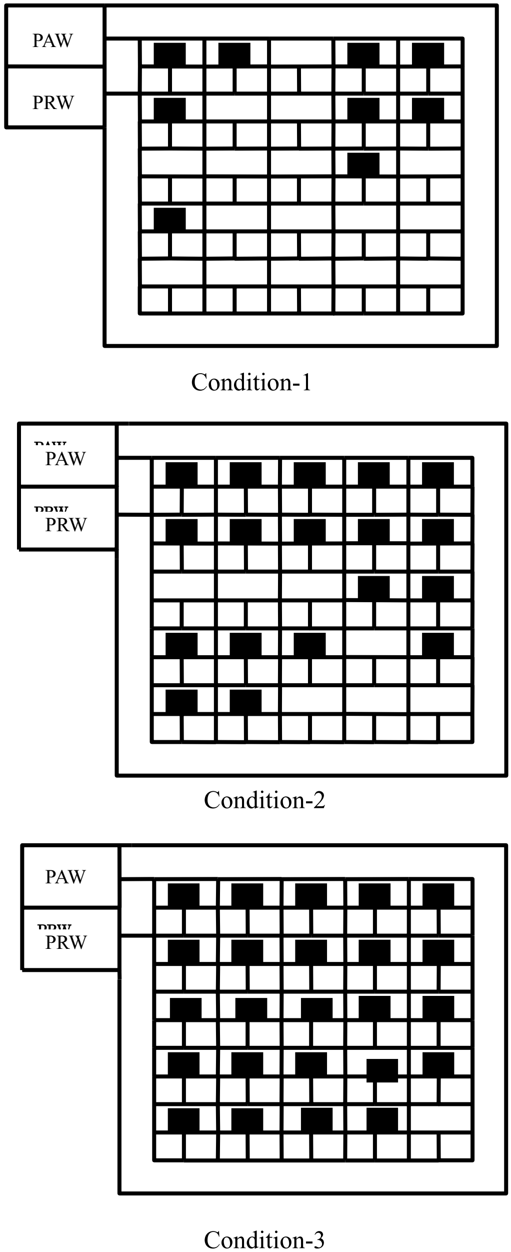


Fig. 10. AD-FMS types

From the results of table 1 and 2, for any unexpected conditions, the AGV that uses the collision avoidance with mind was able to avoid collision, which improves the production rate and MC efficiency rate. Compared with the conventional method [10],

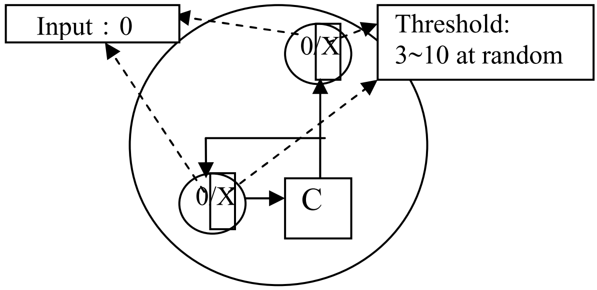


Fig. 11. Initialization of AGV's mind

Table 1. Results of AGV without mind

Condition	1-1	1-2	1-3	2-1	2-2	2-3	3-1	3-2	3-3
MC average efficiency (%)	21.2	22.9	25.8	11.7	11.4	16.1	9.3	11.6	13.6
Collisions	161	396	423	105	276	465	123	234	477
Outputs	93	86	141	48	45	56	36	48	61

Table 2. Results of AGV with mind

Condition	1-1	1-2	1-3	2-1	2-2	2-3	3-1	3-2	3-3
MC average efficiency (%)	35.3	41.7	46.8	19.3	25.7	30.7	16.7.	22.0	26.9
Collisions	0	0	0	0	0	0	0	0	0
Outputs	264	311	349	170	225	272	146	195	235

AGVs of the conventional rule based collision avoidance system stopped and did not move because many AGVs gathered at one point, for example in front of PAW, and did not get out of there.

5 Conclusions

In this paper, we proposed the mind model of AGV in AD-FMS and the action control of AGV with collision avoidance in AD-FMS by using the mind model, not using the usual rule and control laws of collision avoidance. When AGVs avoid collision in AD-FMS, AGV with the mind model take 2 actions, "go away from other AGVs" and "go to destination". After some simulations of AD-FMS, it was found that AGVs with mind did not have any kind of collisions. Also, the MC average efficiency and outputs are improved.

In other words, it is easy for AGV to take a vague action with simplicity rather than using elaborated and control rules with complicated operations.

References

1. Toya, Y., Uozumi, E., Sagawa, Y., Sugie, N.: From Consciousness to Unconsciousness – a Model of Motor Skill Acquisition. In: Proceedings of IFIP INTERACT 2001: Human-Computer Interaction 2001, Tokyo, Japan, pp. 708–709 (2001)
2. De Wall, F.B.M., Aureli, F., Judge, P.G.: Psychology of the congestion. Nikkei Science 8, 22–29 (2000) (in Japanese)
3. Wataya, H., Kawano, K., Hayashi, K.: The Cooperating Autonomous Decentralized System Architecture. In: Proceeding of the 2nd International Symposium on Autonomous Decentralized System, pp. 40–47 (1995)
4. Shen, W.: Distributed Manufacturing Scheduling Using Intelligent Agents. IEEE Intelligent Systems 17(1), 88–94 (2002)
5. Endo, S., Konishi, M.: Motion Planning for Multiple Mobile Robots by a Petri Net approach. In: Proceedings of 1998 Japan-USA Symposium on Flexible Automation, pp. 587–594 (1998)
6. Borzer, Y.M., Srinivasan, M.M.: Tandem AGV systems: a partitioning algorithm and performance comparison with conventional AGV systems. European Journal of Operational Research 63(2), 173–191 (1992)
7. Ho, Y.C.: A dynamic-zone strategy vehicle-collision prevention and load balancing in an AGV system with single-loop guide path. Computers in Industry 42, 159–176 (2000)
8. Morinaka, et al.: A Distributed Routing Method for Multiple AGVs for Motion Delay Disturbances. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2001–2006 (2005)
9. Maeno, T.: How to Make a Conscious Robot –Fundamental Idea Based on Passive Model. Journal of Robotics on Japan 23(1), 51–62 (2005) (in Japanese)
10. Yamamoto, H., Ramli, R.B.: Real-time Decision Making of Agents to Realize Decentralized Autonomous FMS by Anticipation. International Journal of Computer Science and Network Security 6(12), 7–17 (2006)

Agent-Based Dantzig-Wolfe Decomposition

Johan Holmgren, Jan A. Persson, and Paul Davidsson

Department of Systems and Software Engineering, Blekinge Institute of Technology,
Soft Center, 372 25 Ronneby, Sweden
{johan.holmgren,jan.persson,paul.davidsson}@bth.se

Abstract. We present an agent-based approach to Dantzig-Wolfe column generation (a decomposition approach), which is applied to an integrated production, inventory, and distribution routing optimization problem. The decomposition model has been implemented in the Java programming language, using the Java Agent DEvelopment Framework (JADE) and the ILOG CPLEX mixed integer linear optimization problem solver. The model has been validated on a set of realistic scenarios and based on the results, we estimate the potential performance improvement which can be obtained by using a completely distributed implementation. We analyze the overhead, in terms of communication costs, that is imposed by an agent-based approach.

1 Introduction

It has been argued that the strengths and weaknesses of agent-based approaches and classical optimization techniques complement each other well for dynamic resource allocation problems [1]. The authors sketched a number of ways to combine the approaches in order to capitalize on their respective strengths, and two of these were implemented and evaluated. In the first approach, an optimization technique was used for coarse planning and agents for operational replanning, i.e., for performing local adjustments of the initial plan in real-time to handle the actual conditions when and where the plan is executed. In the second approach, optimization was embedded in the agents. This can be seen as a further development of distributed optimization where agent technology is used, e.g., to improve coordination.

In this paper, we describe another approach, namely how an optimization approach can be "agentified" into a multi-agent system (MAS) by using principles of Dantzig-Wolfe (DW) decomposition, which is a classical optimization (reformulation) technique. The purpose is to detail the concept of agentifying an optimization approach, and to validate that some positive characteristics can be achieved by this type of approach. The main characteristics studied are confidentiality aspects of information, i.e. to keep information locally when possible, and the potential to achieve performance improvements by distribution and parallelization.

In a related approach [2], an agent-based approach was used for solving the *Generalized Mutual Assignment Problem*. The author presented a distributed solution protocol based on Lagrangean decomposition and distributed constraint satisfaction. The agents solve individual optimization problems which are coordinated in order to improve the global solution.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 754–763, 2009.

Another possibility is to optimize a system without using advanced mathematical optimization, e.g. by using agent-based negotiation as in an application to virtual collaboration networks [3]. In yet another approach [4], agent technology was used for solving real-world dynamic transportation problems by means of a neighborhood search algorithm and auction-based negotiations.

Our approach has been applied to a real-world integrated production, inventory and distribution routing problem (IPIDRP). The problem includes production planning, vehicle routing, inventory planning, and when and what to deliver, to be able to satisfy the product demands of a set of customers. The objective is to minimize the total cost for production, transportation and inventory holding, while meeting the customer demands with respect to delivery time etc.

In next section the integration of agent technology and decomposition is outlined and in section 3, we present the decomposition approach for the identified IPIDRP. Our computational experiments are discussed in section 4 and finally, in section 5 some conclusions and directions for future research are given.

2 Decomposition and Agent Technology

Decomposition approaches such as Dantzig-Wolfe column generation [5] for solving large linear optimization problems can be used together with *branch-and-price* [6] to solve mixed integer linear optimization problems (MILPs). In DW decomposition, the LP-relaxation (MP) of a MILP is reformulated into a *restricted master problem* (RMP) containing a subset of the variables in MP, and a set of subproblems which produce new solutions (columns) that are coordinated by RMP. In an iterative process, subproblem solutions based on optimal dual variables (a control mechanism) are added as improving variables to RMP.

We suggest a multi-agent-based DW decomposition approach, with a coordinator agent that corresponds to the master problem and planner agents that represent the subproblems. A planner agent is responsible for planning some set of resources thereby assisting the coordinator agent in its search for the global optimal solution. The planning procedure is guided by the values of the dual variables, which is used by the coordinator to assist the planners in finding (globally) improving plans. Hence, the coordinator is not explicitly dictating the characteristics of the plans to be produced by the planner agents, and the planner agents are thereby to some extent autonomous. The coordinator is responsible for coordinating the plans that are produced by the planners, and it decides if and when the planner agents should be given some directives of partially restricting the plan in a branch-and-price strategy, which is based on tree searching. This procedure of restricting plans is necessary in cases where the DW-decomposition does not guarantee a feasible (integer) optimal solution. The planners are forced to obey the partial restrictions in the branch-and-price strategy when they create future plans. Note that agents are assumed to be benevolent, and they do not cheat, by providing incorrect costs for generated plans, in order to benefit from other agents.

It has been seen in a real world application that planning tasks are performed by different organizations, why confidentiality is important. Traditionally, a decomposition algorithm runs in a single process (on a single computer). In such an approach confidentiality cannot be achieved, since the coordinator must be given access to information that might be considered sensitive for the planners. With an agent-based approach however, where the different problems are represented by different individual agents, it is often possible to run the optimization with less need for sharing sensitive information. In our case, we have a coordinator agent who coordinates production, transportation and inventory. The coordinator needs access, e.g., to customer demand forecasts, but it does not need to know any underlying details about how transportation plans and production plans are created, because such information belongs to the subproblem solvers.

Obviously, the use of agents causes a loss in performance due to an increased need for communication. In a distributed solution approach, there is however a potential gain in computation time and in some cases solution quality since more computing power allows for solving more complex subproblems. The reason is that the subproblems can be solved in parallel, which is impossible when the optimization runs on a single processor. The ease with which an agent system can be distributed over several computers gives another possible advantage over classical decomposition approaches. A decentralized approach further makes the system less vulnerable to single point failures. In the agent-based approach, where the coordinator agent typically retains control of all decisions, a failure of the coordinator agent is fatal, because the presence of the coordinator agent is absolutely necessary. However, in the case of a planner failure, the rest of the planners might still be able to produce new (improving) plans to be considered by the coordinator.

In section 3 we demonstrate the agent-based decomposition approach by presenting a case study for an IPIDRP. We have chosen a decomposition formulation which we find attractive particularly from the perspective of a natural interpretation of dual prices and real-world correspondence to the sub-problems. In general, a decomposition scheme including master/subproblem formulations, variable restriction strategy, and termination criteria needs to be designed in such a way that the special characteristics of the studied problem can be utilized. We would like to point out that some problems can be extremely difficult to even formulate as a single mixed integer linear problem. Solving them can be even more difficult, since they are typically NP-hard.The studied problem class captures the difficulties with distributed decision making since typically, information and/or resources are distributed and the exact conditions, for instance, the demand and the availability of resources, are not known in advance and are changing.

3 Case Specific Decomposition Formulation

Our IPIDPR is formulated as a MILP with the objective to minimize the costs for production, distribution, and inventory holding, while satisfying the product demands of a set of customers throughout a given planning horizon. The problem formulation includes production planning at the producer depots, transportation planning including route choice, quantities and times for deliveries to customers and pickups from the producers, and inventory planning of finished products. The presented approach focuses on *Vendor Managed Inventory* (VMI) [7] due to its potential of improved resource utilization. However, it is not limited to VMI, since a tight specification of customer inventories mimics a non-VMI situation.

The decomposition formulation includes a linear master problem, a set of transportation subproblems for construction of transportation plans and a set of production scheduling subproblems for construction of production plans. A coordinator maps to the master problem, a production planner handles a set of production scheduling subproblems (one subproblem for each production line in one producer depot), and a transportation planner handles a set of transportation subproblems (one subproblem for each vehicle in its vehicle fleet). In addition to the practical issues, like for example confidentiality, the reasons for not formulating the problem as one big MILP are:

1. Such a model would most probably be difficult/impossible to solve to optimality due to its high complexity.
2. It is more practical to use a decomposition approach since, e.g., it allows for more flexibility to reformulate the subproblems without modifying the master problem.

We let D^P denote the set of producer depots, D^C the set of customer depots, $D = D^P \cup D^C$ the set of all depots, V the set of (inhomogeneous) vehicles, P the set of product types, and L the set of production lines available in the model. The planning horizon is represented by an ordered set $T = \{1, 2, \ldots, \overline{t}\}$ of discrete time periods with uniform length τ. A *transportation plan* for a vehicle is defined as the amount of each product delivered to each customer depot and picked-up from each producer depot by that vehicle in each time period throughout the planning horizon. The set of all transportation plans for a vehicle $v \in V$ is denoted R_v, the cost for using plan $r \in R_v$ is denoted ψ_r, and x_{dptr} is the amount of product p that is delivered/picked-up to/from depot d by vehicle v by plan r in period t. Similarly, we define a *production plan* for a production line as the amount of each product that is produced by that production line in each time period throughout the planning horizon. We let S_l denote the set of all valid production plans for production line $l \in L$, where ω_s denotes the cost for using plan $s \in S_l$. For production plan $s \in S_l$, which represents a production line located at some depot $d = d(s)$, we let y_{dpts} represent the amount of product p produced in period t. Furthermore, parameter ϱ_{dpt} denotes the demand for product $p \in P$ at customer depot $d \in D^C$ in time period $t \in T$. Hence, the parameter ϱ_{dpt} specifies the amount of product p removed from the inventory in time period t.

Each depot $d \in D$ has an inventory level modeled by variable z_{dpt}. An inventory cost ϕ_{dp} is considered for each unit of product $p \in P$ in stock at depot $d \in D$ between two subsequent periods. For depot d, the inventory level of product p in time period t must not fall below a lower bound $\underline{z}_{dpt}$ (which typically corresponds to a safety stock level) and must not exceed an upper bound (typically a maximum capacity) of $\overline{z}_{dpt}$ units. To allow violating the safety stock levels and maximum allowed inventory levels, we introduce variable u_{dpt} to represent how much the inventory level of product p falls below the safety stock level at depot d in period t, q_{dpt} how much it exceeds the maximum allowed inventory level, and corresponding penalty costs M^u and M^q for breaking the inventory constraints.

Binary decision variables are used to determine which transportation plans and production plans to use, exactly one transportation plan for each vehicle and exactly one production plan for each production line is allowed. Decision variable v_r determines if transportation plan $r \in R_v$ is used ($v_r = 1$) or not ($v_r = 0$), and w_s if production plan $s \in S_l$ is used ($w_s = 1$) or not ($w_s = 0$).

We formulate our DW Master Problem (MP) as the LP-relaxation of the following:

$$
\text{(IMP)} \begin{cases}
\min & \sum_{p\in P}\sum_{d\in D}\sum_{t\in T}(\phi_{dp}z_{dpt} + M^q q_{dpt} + M^u u_{dpt}) + \\
& \quad \sum_{v\in V}\sum_{r\in R_v}\psi_r v_r + \sum_{l\in L}\sum_{s\in S_l}\omega_s w_s, & (1)\\
\text{s.t.} & z_{dpt-1} + \sum_{v\in V}\sum_{r\in R_v} v_r x_{dptr} + u_{dpt} - q_{dpt} - \varrho_{dpt} = z_{dpt}, & \\
& \qquad t\in T, d\in D^C, p\in P, & (2)\\
& z_{dpt-1} - \sum_{v\in V}\sum_{r\in R_v} v_r x_{dptr} + \sum_{l\in L}\sum_{s\in S_l} w_s y_{dpts} + u_{dpt} - & \\
& \quad q_{dpt} = z_{dpt}, \qquad t\in T, d\in D^P, p\in P, & (3)\\
& \underline{z}_{dpt} \le z_{dpt} \le \overline{z}_{dpt}, \qquad d\in D, p\in P, t\in T, & (4)\\
& \sum_{r\in R_v} v_r = 1, \qquad v\in V, & (5)\\
& \sum_{s\in S_l} w_s = 1, \qquad l\in L. & (6)
\end{cases}
$$

In IMP, which represents our main problem, $v_r \in \{0,1\}\,\forall\, r \in \bigcup_{v\in V} R_v$, $w_s \in \{0,1\}\,\forall\, s \in \bigcup_{l\in L} S_l$ and $z_{dpt} \ge 0\,\forall\, d \in D, p \in P, t \in T$. The first component of the objective function (1) models the inventory costs with penalties, and the second and third components represent the costs of the transportation and production plans respectively. Constraint sets (2) and (3) express the customer and producer depot inventory balances, (4) assures that the inventory levels are always kept between the minimum and maximum allowed levels, and (5) and (6) assure that only one transportation/production plan is used for each vehicle/production line.

We let $R'_v \subseteq R_v$ denote the set of all currently known transportation plans for $v \in V$, and $S'_l \subseteq S_l$ the set of all currently known production plans for $l \in L$. RMP (the restricted master problem) is identical to MP, except for that we replace all occurrences of R_v with R'_v, all occurrences of S_l with S'_l. For RMP, we let λ denote the dual variables for constraints group (2), μ the dual variables for constraints (3), δ the dual variables for constraints (5), and θ the dual variables for constraints (6). A dual variable expresses the change of the optimal objective value per unit increase of the right hand side of the corresponding constraint. For example, $\bar{\lambda}_{dpt} > 0$ represents the value of having one extra unit of product p in depot d in period t.

We have developed a column generation based solution algorithm where RMP is updated iteratively with new improving production and transportation plans that is generated by the planner agents based on the current optimal values of λ, μ, δ, and θ. Dual variables are obtained from the optimal solution when using a standard solver (typically based on the simplex method). A flow chart of the main algorithm, which is used by the coordinator agent, is presented in Fig. 1 and a diagram showing how the coordinator communicates with the planners is given in Fig. 2. In a distributed approach, it is possible (and preferable) to send the transportation and production requests in parallel to allow the planners to solve subproblems simultaneously. Plans with negative reduced cost are added as improving columns/variables to RMP, and when no improving plan can be produced, the optimal solution of RMP is reached. This optimal solution (or near optimal depending on termination criteria) typically contains fractional combinations of plans, which is infeasible in our integer problem IMP. To be able to find an optimal or, at least a heuristically "good" integer solution to IMP, some delivery/pickup (depot, period, product, and vehicle) or production (period, product and production line) must be restricted (fixed) to some integer quantity, whenever a fractional optimal solution of RMP is found. Fixings are determined by the coordinator (from RMP) and communicated to the planners to prevent them from violating the fixings when generating future plans. Throughout the procedure of the algorithm, more and more quantities are fixed. Eventually, when the solution space of the subproblems is restricted enough by fixed variables, the algorithm terminates with an integer solution. A more detailed description of our decomposition approach is presented in [8]. To guarantee that the optimal solution of IMP will be generated, it is necessary to branch on all restrictions using a search tree. However, for performance reasons, we chose to explore only one branch in the search tree.

The described decomposition approach is for integrated production and transportation, where inventory levels between production and transportation, and between delivery and customer usage are considered. This is a rather general type of problem and it is independent of the choice of production and transportation subproblems. The subproblems must, however, be able to take the dual variables as input and be able to produce new production or transportation plans.

4 Computational Experiments

The column generation algorithm was developed inside a multi-agent-based simulation tool called TAPAS [9]. TAPAS was implemented in the Java programming language using the Java Agent DEvelopment Framework (JADE) platform [10] and the mixed integer linear problem solver ILOG CPLEX[1] 10.0. In the experiments, TAPAS runs on a single computer but a distributed implementation (with different agents running on different computers) is straight forward. This would increase the communication overhead, while potentially reducing the overall solution time due to the availability of more processors.

Due to the slow convergence speed caused by the complexity of the studied problem, we found it unreasonable to solve RMP to optimality before fixing some variable. Instead fixings are performed whenever the average relative improvement of a certain number of the latest generated plans falls below some required relative improvement level. This restricts the solution space of RMP, but hopefully, heuristically good integer solutions to IMP will be found.

4.1 Scenario Description

We solved our problem on 15 scenarios with a transportation network with 2 producer depots (with one production line each), 6 customer depots, 9 vehicles, 4 products, and

[1] http://www.ilog.com/

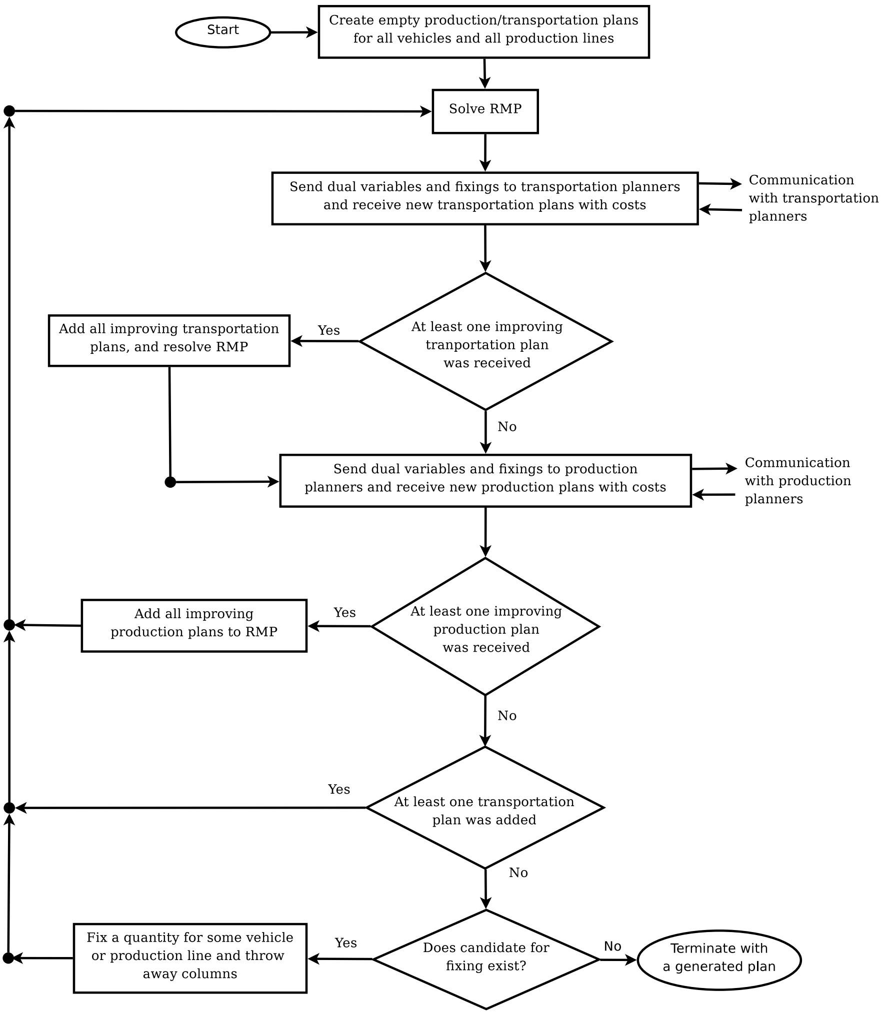


Fig. 1. Coordinator agent (main algorithm) flow chart

a planning horizon of 72 time periods with uniform length 2 hours (6 days). All vehicles are planned by the same transportation planner, i.e. in the scenarios, there is one transportation planner agent and two production planner agents (each representing one production depot). The scenarios were generated randomly, and they vary in average customer demands, minimum/maximum/initial storage levels, and the number of VMI customers.

4.2 Time Performance Analysis

In each iteration, the coordinator agent sends one plan request message to each planner agent, and each planner returns with a response message which contains either a

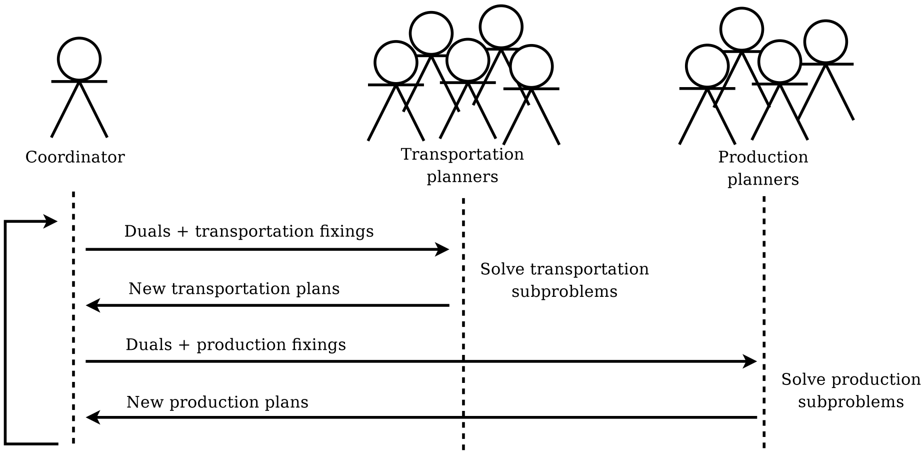


Fig. 2. Communication diagram

generated plan or a failure notification. For our 15 scenarios, which used an average of 2171 master problem iterations to reach the final solution, in average 13023 messages had to be sent. The number of messages can be used as a measure of the overhead in the system, in comparison to a non-agent-based implementation approach. The size of each message depends on the actual application and the problem size of course, since a bigger problem requires that more information (e.g. dual variables) need to be communicated.

For each of the scenarios, we estimated an upper bound of the overhead imposed by the agentification of our decomposition algorithm, calculated as the total running time minus the estimated time for performing tasks related to decomposition, mainly communication. We measured an average total running time of 9509 seconds and an average lower bound estimation of 6583 seconds for the time spent in the actual decomposition algorithm. This gives an average estimated overhead of approximately 35.7% of the total running time. Here the average estimated overhead is taken as the average over the estimated overhead (in percentage) of the 15 scenarios.

For the 15 scenarios, we estimated the expected performance improvement that can be achieved when all subproblems are distributed to, and solved in parallel on, different computers. This is one of the purposes with the agent-based decomposition approach, and for the scenarios, we got average estimated total solving times of 3856 seconds for the transportation subproblems and 17 seconds for the production scheduling subproblems. In theory, ignoring the increased communication time imposed by a parallelization, an average potential time reduction from $(3856 + 17) \approx 3873$ seconds to 429 seconds can be achieved by solving the subproblems in parallel. The reduced time for a scenario is calculated as the number of master problem iterations times the maximum taken of the average solve time of one transportation subproblem and the average solve time of one production subproblem. Since this calculation uses average subproblem solving times, the potential time reduction is an upper bound of the actual potential time reduction. In our scenarios, this would constitute an average time reduction of

$3873 - 429 = 3444$ seconds to a total running time of 6064 seconds. Here, we assume a usage of 12 computers, 9 for transportation subproblems, 2 for production scheduling subproblems and 1 for the master problem. In this case, solving the production scheduling subproblems and the master problem on the same computer would give the same improved running time, but confidentiality of information would be weakened. Note that we cannot use the main algorithm as it is presented in Fig. 1 to utilize a complete parallelization. The algorithm needs to be slightly modified so that the dual variables is sent to the transportation and production planners in parallel without resolving RMP in between.

5 Conclusions and Future Work

Our experiments show that it is possible to agentify a DW decomposition method. Some rather obvious positive effects of the agent-based approach are increased confidentiality, robustness and possibility of distributed computing. Negative effects are overhead in processing and communication. The tasks of the agents could be better supported by a distributed agent-based decomposition approach than by a centralized approach, one advantage is the ability to locally specify and modify the subproblems. Further, a local agent can always provide the most recent subproblem solution to the local planners (even though based on old dual variables).

In order to capitalize on the use of the multi-agent-based approach for achieving performance improvements, we find it interesting to experiment with a distributed system where different agents run on different computers. This is a natural representation of the real-world, where the actors (agents) are geographically separated and each agent typically only has access to its own data and its own computer. The estimations presented in section 4.2 indicate that a parallelization can give significant improvement on the time performance. Also, a parallelization would allow for solving more complicated subproblems which would make it possible to capture more real-world details. Note that our time performance improvement estimation assumes perfect parallelization with zero communication time. Future work includes further validation regarding communication overhead.

Currently, a planner agent may go down temporarily, while the coordinator never is allowed to fail. Future work includes putting some effort into refining the algorithm in order to allow it to deal with permanent failures of all agents types (the coordinator as well as the planners).

The described decomposition approach is for integrated production and transportation, where inventory levels are considered. The studied IPIDRP is rather general and it is independent of the choice of production and transportation subproblems. The planners (subproblems) must, however, be able to take dual variables as input and be able to produce new production or transportation solutions that can be communicated to the coordinator (master problem). By choosing customized production and transportation subproblems, our solution approach can be used to solve IPIDRPs where:

1. Decisions about inventories are taken centrally and detailed decisions about transportation and production are taken locally.

2. The master problem can produce dual variables and receive production and transportation plans.

We conclude that our agent-based solution approach has the potential to work as a framework for this type of IPIDRPs.

References

1. Davidsson, P., Persson, J.A., Holmgren, J.: On the integration of agent-based and mathematical optimization techniques. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS, vol. 4496, pp. 1–10. Springer, Heidelberg (2007)
2. Hirayama, K.: Distributed lagrangean relaxation protocol for the generalized mutual assignment problem. In: AAMAS 2006: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, pp. 890–892. ACM, New York (2006)
3. Karageorgos, A., Mehandjiev, N., Weichhart, G., Hämmerle, A.: Agent-based optimisation of logistics and production planning. Engineering Applications of Artificial Intelligence 16(4), 335–348 (2003)
4. Dorer, K., Calisti, M.: An adaptive solution to dynamic transport optimization. In: AAMAS 2005: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, pp. 45–51. ACM, New York (2005)
5. Dantzig, G., Wolfe, P.: The decomposition algorithm for linear programs. Econometrica 29, 767–778 (1961)
6. Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H.: Branch-and-price: Column generation for solving huge integer programs. Operations Research 46, 316–329 (1998)
7. Daugherty, P.J., Myers, M.B., Autry, C.W.: Automatic replenishment programs: an empirical examination. Journal of Business Logistics 20(2), 63–82 (1999)
8. Holmgren, J.: Multi-Agent-Based Simulation and Optimization of Production and Transportation. Licentiate Dissertation No. 2008:05, Blekinge Institute of Technology, Sweden (2008)
9. Davidsson, P., Holmgren, J., Persson, J.A., Ramstedt, L.: Multi agent based simulation of transport chains. In: AAMAS 2008: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems, Richland, SC, pp. 1153–1160. International Foundation for Autonomous Agents and Multiagent Systems (2008)
10. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE. Wiley Series in Agent Technology. John Wiley & Sons, Chichester (2007)

A-Team Middleware on a Cluster

Ireneusz Czarnowski, Piotr Jędrzejowicz, and Izabela Wierzbowska

Department of Information Systems,
Gdynia Maritime University, Morska 83, 81-225 Gdynia, Poland
{irek,pj,iza}@am.gdynia.pl

Abstract. JABAT (Jade-based A-Team) is a middleware that allows to design and implement multiagent architectures for solving combinatorial optimization problem. So far JABAT has been implemented in environments of local/global networks, in which the users were able to adjoin computer to the system for the work of the system's agents.

The paper describes implementation of the middleware called JABAT (JADE-based A-Team) on a cluster of computers at the TASK Academic Computer Centre. It also presents the results of first tests carried out in this new environment.

Keywords: JABAT, A-Team, optimization, computionally hard problems, multi-agents systems, cluster.

1 Introduction

Agent technology has already shown many advantages in developing flexible, adaptive and sophisticated solutions in most difficult areas. Its application in the field of optimization, in the concept of asynchronous team (A-Team), originally introduced by Talukdar [17], has resulted in many successful applications, and the advances in this field has by no means reached its limits yet.

The multi agent architecture of A-Team forms a framework that enables users to easily combine disparate problem solving strategies. Autonomous agents may cooperate to evolve diverse and high quality solutions [16], and they may achieve it by dynamically evolving a population of solutions [17]. Each agent works to create, modify or remove solutions from the population. The quality of the solutions gradually evolves over time as improved solutions are added and poor solutions are removed. Cooperation between agents emerges as one agent works on solutions produced by another. Each agent encapsulates a particular problem-solving method along with methods to decide when to work, what to work on and how often to work.

The reported implementations of the A-Team concept include two broad classes of systems: dedicated A-Teams and platforms, environments or shells used as tools for constructing specialized A-Team solutions. Such specialized A-Teams are usually not flexible and can be used for solving only the particular types of problems they are designed for. Among example A-Teams of such type one can mention the OPTIMA system for the general component insertion

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 764–772, 2009.

optimization problem [15] or A-Team with a collaboration protocol based on a conditional measure of agent effectiveness designed for Flow optimization of railroad traffic [5].

Among platforms and environments used to implement A-Team concept some well known include IBM A-Team written in C++ with own configuration language [16] and Bang 3 - a platform for the development of Multi-Agent Systems (MAS) [14]. Some implementations of A-Team were based on universal tools like Matlab [18]. Some other were written using algorithmic languages like, for example the parallel A-Team of [6] written in C and run under PVM operating system.

The above discussed platforms and environments belong to the first generation of A-Team tools. They are either not portable or have limited portability, they also have none or limited scalability. Agents are not in conformity with the FIPA (The Foundation of Intelligent Psychical Agents) [10] standards and there are no interoperability nor Internet accessibility. Migration of agents is either impossible or limited to a single software platform.

In the second group of platforms a middleware called JABAT (Jade-based A-Team) was proposed in [12]. It was intended to become the first step towards next generation A-Teams which are portable, scalable and in conformity with the FIPA standards. JABAT allowes to design and implement an A-Team architecture for solving combinatorial optimization problems. JABAT can be also extended to become a fully Internet-accessible solution [3], with the ability to not only to access the system in order to solve own combinatorial problems, but also to add own computer to the system as to increase the resources used for solving the problems.

All previous implementations of JABAT has been done in MS Windows environment, although the architecture used (Java and JADE) allowed for incorporating computers with different systems. The most recent application has been installed on a cluster of computers at the Academic Computer Centre TASK in Gdask, and this paper presents the first results obtained. The paper is organized as follows: Section 2 gives a short overview of the JABAT features. Section 3 describes the details of the cluster implementation. Section 4 offers some results obtained for ANN Training. Finally, Section 5 contains conclusions and suggestions for future research.

2 Main Features of the JABAT Middleware

JABAT is a middleware supporting design and development of the population-based applications intended to solve difficult computational problems. The approach is based on the A-Team paradigm.

Main features of JABAT include:

- The system can solve instances of several different problems in parallel.
- The user, having a list of all algorithms implemented for the given problem may choose how many and which of them should be used.

- The optimization process can be carried out on many computers. The user can easily add or delete a computer from the system. In both cases JABAT will adapt to the changes, commanding the optimizing agents working within the system to migrate JABAT may also clone some already working agents and migrate the clones, thus increasing the number of concurrently operating agents.
- The system is fed in the batch mode - consecutive problems may be stored and solved later, when the system assesses that there is enough resources to undertake new searches.

JABAT produces solutions to combinatorial optimization problems using a set of optimizing agents, each representing an improvement algorithm. The process of solving of the single task (i.e. the problem instance) consists of several steps. At first the initial population of solutions is generated. Individuals forming the initial population are, at the following computation stages, improved by independently acting agents, thus increasing chances for reaching the global optimum. Finally, when the stopping criterion is met, the best solution in the population is taken as the result.

How the above steps are carried out is determinded by the "strategy". There may be different strategies defined in the system, each of them specyfying:

- how the initial population of solutions is created (in most cases the solutions are drawn at random),
- how to choose solutions which are forwarded to the optimizing agents for improvement,
- how to merge the improved solutions returned by the optimizing agents with the whole population (for example they may be added, or may replace random or worst solutions),
- when to stop searching for better solutions (for example after a given time, or after no better solution has been found within a given time).

For each user's task the system creates report files that include:

- the best solution obtained so far (that of the maximum or minimum value of fitness),
- average value of fitness among solutions from current population,
- the actual time of running and list of changes of the best solution (for each change of the best solution in the process of solving the time and new value of fitness is given).

The report on the process of searching for the best solution may be later analysed by the user. It can be easily read into a spreadsheet and converted into a summary report with the use of the pivot table.

3 JABAT Implementations

The JABAT middleware is built with the use of JADE (Java Agent Development Framework), a software framework proposed by TILAB [19] for the development and run-time execution of peer-to-peer applications. JADE is based on the agents paradigm in compliance with the FIPA [10] specifications and provides a comprehensive set of system services and agents necessary to distributed peer-to peer applications in the fixed or mobile environment. It includes both the libraries required to develop application agents and the run-time environment that provides the basic services and must be running on the device before agents can be activated [4].

JABAT itself has also been designed in such a way, that it can be easily instantiated in new environments where Java and JADE are available. To do so, JABAT with at least one set of objects (eg. task, solutions, optimizing agent) dedicated to a chosen problem has to be set up on a computer, and in the minimal configuration it is enough to solve the problem. Such a minimal configuration may be extended by adding

– more optimizing agents,
– more replacement strategies,
– objects (agents) for solving more problems,
– more resourses, eg. adjoining some computer to which the optimizing agents could migrate to work on.

It is worth mentioning that JABAT has been designed in such a way, that all above extentions that require some amount of programming (like solving new problems, solving them with new optimizing algorithms or with the use of new replacement strategies) may be done with relatively little amount of work: JABAT makes it possible to focus only on defining these new elements, while the processes of communication and population management procedures will still work. More detailed information about extending the functionality of JABAT can be found in [1].

The previous implementations of JABAT have been working in MS Windows environment, and have been available either within a local network, or through a web interface, making the original functionality accessible for users from all over the world. The latter implementation with the web interface was called e-JABAT and was described in more detailes in [3].

To validate the various preceding implementations a number of experiments has been conducted. Experiments have involved a variety of combinatorial optimization problems. The results reported in [11], [2], [9], [7], [8] have proved ability and efectiveness of JABAT with regard to solving computationally hard problems.

3.1 JABAT on a Cluster

The most recent implementation of JABAT is working on Holk - a cluster of computers at the Academic Computer Centre TASK in Gdaņsk. Holk is build on 256 Intel Itanium 2 Dual Core with 12 MB L3 cache processors, with Mellanox InfiniBand interconnections with 10 Gb/s bandwidth.

The implementation started with installing required Java and Jade, then the whole engine that had been used in e-JABAT was moved to Holk (eg. e-JABAT without the web interface). It required only minimal number of steps:

- changing the configuration file in which current paths to folders for data given and read by potential users had to be set up, and
- recompiling the whole system.

The process went smoothly and did not bring about problems of any kind.

Then the system has been tested with all most strategies and problems that had already been implemented in it's previous implementations. The problems that can be solved on the cluster are the following:

- resource constrained project scheduling problem with single and multiple node (RCPSP, MRCPSP),
- clustering problem (CP),
- euclidean planar traveling salesman problem (TSP),
- vehicle routing problem (VRP) and
- artificial neural network training (ANN).

For these problems the following optimizing agents are available (each representing a different optimization algorithm):

- the local search algorithm, algorithm based on the simple evolutionary crossover operator and the tabu search algorithm for the RCPSP,
- the Lin-Kerninghan algorithm and the evolutionary algorithm for the TSP,
- the 2-optimum algorithm operating on a single route, the λ interchange local optimization method and the local search algorithms for the VRP,
- the random local search, the hill-climbing local search and the tabu search algorithms for the CP,
- the agent-based algorithm for the ANN described in subsection 3.2.

In the implementation on Holk, the JABAT engine operates on one of the computers of the cluster, while other computers can be adjoined in required number. Adding these computers to the system has been facilitated by preparing a set of ready to use dsh (distributed shell) scripts.

The tests that has been carried out showed that the new implementation of JABAT is fully operating.

3.2 Experiment Results

The environment of the cluster Holk is more powerful than the previous ones, so the efficiency of the whole system increased. For example it could be easily observed for all problems that have been solved that the time of searching for solutions decreased, unless the stopping criterion was set as the time limit.

To evaluate the performance of the proposed JADE-based implementation a computational experiment has been carried out. It involved the JABAT implementation of agent-based neural network training algorithm for ANN. The algorithm was described in more details in [9].

The algorithm uses seven agent types representing seven different improvement procedures, including standard mutation, local search, non-uniform mutation, gradient mutation, gradient adjustment, a single point crossover and

Table 1. The values of the training errors

Problem	Training error
Sonar	0.25
Credit	0.13
Cancer	0.035
Heart	0.12

arithmetic crossover. Out of this set the crossover agents take care of information exchange and diversification while the remaining agents are used to directly improve the fitness of individuals drawn from the common memory of solutions.

The experiment involved training of the MLP type artificial neural networks aimed at solving benchmark datasets including instances of four well known classification problems - Cleveland heart disease (303 instances, 13 attributes, 2 classes), credit approaval (690, 15, 2), Wisconsin breast cancer (699, 9, 2) and sonar problem (208, 60, 2). The respective datasets have been taken from [13].

The algorithm has been previously investigated in [9]. Then its performance was measured in the environment consisting of several PC computers with Pentium IV 1.7 GHz processors and 256 MB RAM, connected within the local area network (Fast Ethernet). The results may be compared against the analogous results obtained in the cluster environment. In both cases (both environments) the system performance has been investigated for different number of optimizing agents (run on separate or the same nodes or computers). In each case the time of the computation of a solution with some predefined quality was measured (eg. the classification accuracy).

A critical (that is the required) value of the training error for each problem type has been set, which, in turn, has been used to evaluate the computational speed-up factor resulting from adding additional computers and enabling agents

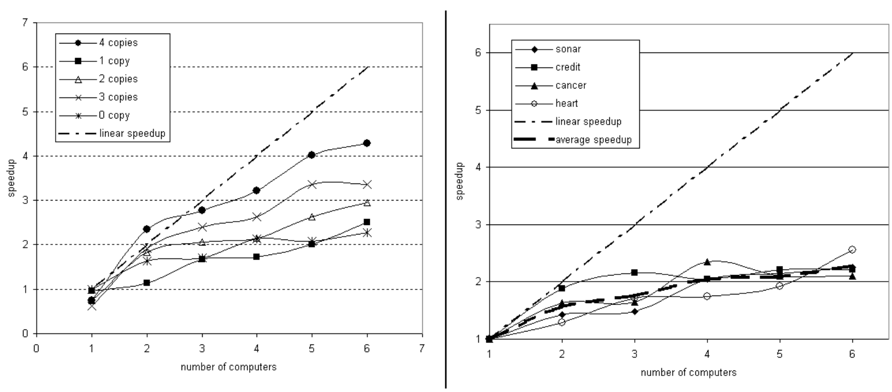


Fig. 1. PC environment: The speed-up factor for various numbers of computers and copies of optimizing agents (left box) and for various problems and numbers of computers (right box)

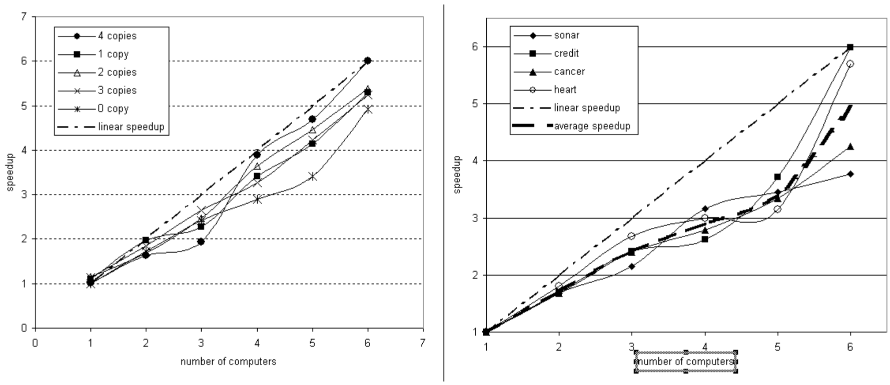


Fig. 2. Cluster environment: The speed-up factor for various numbers of computers and copies of optimizing agents (left box) and for various problems and numbers of computers (right box)

migration. The values of the respective critical values of the training errors has been set at the levels shown in Table 1.

In both experiments the training procedure of the respective ANN classifier terminated as soon as the classifier had been able to reach the respective critical value of the training error. In each run computation times were registered.

Figure 1 and Figure 2 show dependency between the speed-up factor and the number of containers (computers or nodes) used by JABAT. As the right boxes of the figures show, in the case of PCs environment mean speed-up factor can be estimated as 1.6 to 2.3 for 2 to 6 adjoined computers respectively. In case of the cluster, the factor can be estimated as 1.7 to 4.9 for 2 to 6 nodes respectively. These results have been achieved with 7 optimization agents distributed among adjoined computers or nodes. Increasing the number of copies of the optimization agents can still improve the speed-up factor. This is shown in the left boxes of the same figure, where the speed-up factor depending on the number of computers and the allowed number of agent clones is shown. The results are shown for 0, 1, 2, 3 and 4 additional clones of each optimizing agent, corresponding, respectively, to 7, 14, 21, 28 and 35 optimization agents distributed among available platforms. Data for this figure has been averaged over all problem types.

For both environments the parallelization efficiency has been estimated. These results are shown in Table 2. The data show significant improvement in the latter case. It also shows that the approach of JABAT for dividing the work

Table 2. The parallelization efficiency

Number of processore	2	3	4	5	6
PC environment	78%	58%	51%	42%	38%
cluster environment	86%	80%	72%	68%	82%

between agents, not typical for parallel computations (only optimizing agents are distributed to other nodes), gives quite good results.

Finally the experiment has shown that cluster environment is indeed more efficient in terms of the time required to achieve comparable results: on average the speed of solving tasks was 9.9 times better.

4 Conclusions

The middleware JABAT has been designed to simplify the development of distributed A-Teams composed of autonomous entities that need to communicate and collaborate in order to solve difficult combinatorial problems. It is a framework that hides all complexity of the distributed architecture plus a set of predefined objects are available to users, who can focus on the logic of the A-Team application and effectiveness of optimization algorithms rather than on middleware issues, such as discovering and contacting the entities of the system. Among the most important features of the system are flexibility and scalability, allowing for seting up the system in various environments. The goal of the research presented in this paper was to check in practice its usability in these aspects.

The test carried out in the environment of cluster Holk confirmed that the architecture of JABAT may be easily moved to new environments, proving that the approach resulted in achieving functional, scalable, flexible, efficient, robust, adaptive and stable A-Team architecture. Hence, JABAT can be considered as a step towards next generation A-Team solutions. Also it has beem shown that in the new environment the efficiency of solving task significantly increased.

Acknowledgments. Calculations have been preformed in the Academic Computer Centre TASK in Gdansk.

References

1. Barbucha, D., Czarnowski, I., Jędrzejowicz, P., Ratajczak, E., Wierzbowska, I.: JADE-Based A-Team as a Tool for Implementing Population-Based Algorithms. In: Chen, Y., Abraham, A. (eds.) Intelligent Systems Design and Applications, IDSA, Jinan Shandong China, pp. 144–149. IEEE, Los Alamitos (2006)
2. Barbucha, D., Jędrzejowicz, P.: An experimental investigation of the synergetic effect of multiple agents solving instances of the vehicle routing problem. In: Grzech, A. (ed.) Proceedings of the 16th International Conference on Systems Science, Wroclaw, vol. II, pp. 370–377 (2007)
3. Barbucha, D., Czarnowski, I., Jędrzejowicz, P., Ratajczak, E., Wierzbowska, I.: Web Accessible A-Team Middleware. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part III. LNCS, vol. 5103, pp. 624–633. Springer, Heidelberg (2008)
4. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE. A White Paper, Exp. 3(3), 6–20 (2003)
5. Blum, J., Eskandarian, A.: Enhancing intelligent agent collaboration for flow optimization of railroad traffic. Transportation Research Part A 36, 919–930 (2002)

6. Correa, R., Gomes, F.C., Oliveira, C., Pardalos, P.M.: A parallel implementation of an asynchronous team to the point-to-point connection problem. Parallel Computing 29, 447–466 (2003)
7. Czarnowski, I., Jędrzejowicz, P.: An agent-based approach to the multiple-objective selection of reference vectors. In: MLDM 2007. LNCS(LNAI), vol. 4571, pp. 117–130. Springer, Heidelberg (2007)
8. Czarnowski, I., Jędrzejowicz, P.: An agent-based algorithm for data reduction. In: Bramer, M., et al. (eds.) Research and Development of Intelligent Systems, vol. XXIV, pp. 351–356. Springer, London (2007)
9. Czarnowski, I., Jędrzejowicz, P.: Implementation and Performance Evaluation of the Agent-Based Algorithm for ANN Training. In: Nguyen, N.T., Grzech, A., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2007. LNCS(LNAI), vol. 4496, pp. 131–140. Springer, Heidelberg (2007)
10. The Foundation for Intelligent Physical Agents, http://www.fipa.org/
11. Jędrzejowicz, P., Ratajczak-Ropel, E.: Agent-Based Approach to Solving the Resource Constrained Project Scheduling Problem. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 480–487. Springer, Heidelberg (2007)
12. Jędrzejowicz, P., Wierzbowska, I.: JADE-Based A-Team Environment. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3993, pp. 719–726. Springer, Heidelberg (2006)
13. Merz, C.J., Murphy, P.M.: UCI repository of machine learning databases. University of California, Department of Information and Computer Science, Irvine (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
14. Neruda, R., Krusina, P., Kudova, P., Rydvan, P., Beuster, G.: Bang 3: A Computational Multi-Agent System. In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2004 (2004)
15. Rabak, C.S., Sichman, J.S.: Using A-Teams to optimize automatic insertion of electronic components. Advanced Engineering Informatics 17, 95–106 (2003)
16. Rachlin, J., Goodwin, R., Murthy, S., Akkiraju, R., Wu, F., Kumaran, S., Das, R.: A-Teams: An Agent Architecture for Optimization and Decision-Support. In: Muller, J.P., et al. (eds.) ATAL 1998. LNCS(LNAI), vol. 1555, pp. 261–276. Springer, Heidelberg (1999)
17. Talukdar, S.N., de Souza, P., Murthy, S.: Organizations for Computer-Based Agents. Engineering Intelligent Systems 1(2) (1993)
18. Talukdar, S., Baerentzen, L., Gove, A., de Souza, P.: Asynchronous teams: cooperation schemes for autonomous agents. Journal of Heuristics 4, 295–321 (1998)
19. Jade - Java Agent Development Framework, http://jade.tilab.com/

Simulating Activities of the Transportation Company through Multi-Agent System Solving the Dynamic Vehicle Routing Problem

Dariusz Barbucha

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

Abstract. A wide range of problems arising in practice of transportation companies has a dynamic character, i.e. the required information is not given a priori to the decision maker but is revealed concurrently with the decision-making process. One of the most important group of such problems are dynamic vehicle routing problems, which involve dynamic decision making with respect to vehicle routing in response to the flow of customer demands. The goal of such routing is to provide the required transportation with minimal service cost subject to various constraints. The paper proposes a new approach to solving the dynamic vehicle routing problem using the multi-agent environment.

Keywords: multi-agent systems, dynamic vehicle routing problem, heuristics.

1 Introduction

Classical vehicle routing problem (VRP) is probably one of the best known among problems from the transportation area in which a set of given customers is to be served by the fleet of vehicles in order to minimize the service cost and satisfying several customer's and vehicle's constraints.

In practice of transportation companies there are many problems which extend in natural way the classical problem by adding additional constraints. Typical examples of such extended problems are Vehicle Routing Problem with Time Windows (VRPTW) or Pickup and Delivery Vehicle Routing Problem (PDVRP). Another possibility of extension of the VRP is to take account of the fact that some data of classical problem may depend on time. For example, new requests may occur while the system is already running, vehicle speed may depend on road's conditions or customers may arise with given probability. Such problems are called *dynamic vehicle routing problems* (DVRP).

In the dynamic vehicle routing problem considered in the paper a set of given customers with a non-negative demand is to be served by the fleet of capacited vehicles originally located at the depot in order to minimize the total cost of travel and such that each route starts and ends at the depot, each customer

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 773–782, 2009.

is serviced exactly once by a single vehicle, and the total load on any vehicle associated with a given route does not exceed the vehicle capacity. Additionally, it is assumed that a certain number of customers' requests are available in advance (static requests) and the remaining requests arrive at different time instances while the system is already running (dynamic requests). Moreover, it is also assumed that all requests have to be served in predefined period of time.

During recent years there have been many important advances in the field of classic VRP. Because of the fact that this problem is computationally difficult, most of them are based on heuristics or metaheuristics [6]. Definitely much less works have been published with respect to solving DVRP (see for example [7] and [4]). Among the methods proposed for solving VRP and DVRP, only few approaches based on using intelligent agents have been suggested (see for example [9] and [5]).

The goal of the paper is to propose a new approach for solving the dynamic vehicle routing problem based on the multi-agent paradigm. To support searching for a satisfactory solution a multi-agent platform simulating activities of the transportation company has been designed and implemented.

The paper is organized as follows. Section 2 describes main features of the proposed multi-agent approach proposed. Section 3 reports on the results of the computational experiment. Finally, Section 4 contains conclusions and suggestions for future research.

2 Multi-Agent Approach to Solving the DVRP Instances

The proposed procedure of solving the DVRP instances proposed in this paper is based on a dedicated multi-agent platform developed to simulate a transportation company activities. The platform described in details in [1] is further on referred to as *MAS-DVRP*. The platform allows a decision maker to observe values of many performance indicators, like the cost of servicing, the time a customer must wait before its request is completed, the time spent by each vehicle in the system, the waiting time of each vehicle, etc.

The platform is built using JADE (Java Agent Development Framework), a software framework proposed by TILAB [11] for the development and run-time execution of peer-to-peer applications. JADE is based on the agents paradigm in compliance with the FIPA [10] specifications and provides a comprehensive set of system services and agents necessary to distributed peer-to peer applications in the fixed and mobile environment. It includes both the libraries required to develop application agents and the run-time environment that provides the basic services and that must be running on the device before agents can be activated [2].

Within the platform a several types of autonomous agents are used. Each agent encapsulates a particular abilities and during the process of solving the problem agents play their roles cooperating in order to achieve the common goal.

2.1 Agent Types

The following types of agents have been implemented within the proposed system:

ACompany (AC). An agent which runs first and initializes all others agents.

ARequestGenerator (ARG). An agent which generates (or reads) new orders and sends them to the *ARequestManager* agent.

ARequestManager (ARM). An agent which manages the list of requests received from *ARG*. It maintains two lists:

- list of all static requests - l_stR
- list of actual dynamic requests - l_dynR

After receiving the new request, *ARM* announces it to each *AVehicle* agent and chooses the best one from offers returned by *AVehicle* agents.

AVehicle (AV). An agent that represents a vehicle and is characterized by:

- depot's coordinates on the plane - $(x0, y0)$,
- the capacity of the vehicle - W,
- actual route assigned to this vehicle - R,
- actual cost of the route - $cost(R)$,
- actual available space - Wr,
- speed - v,
- vehicle state (*waiting*, *driving*, *stopped*) - sV,
- total time spend by vehicle in the system - tsV,
- vehicle's total waiting time - twV.

Periodically AV receives customer's request from the ARM one at a time, tries to assign it to the existing route in order to minimize the cost, and sends back its offer (i.e. calculated cost of insertion) to the *ARM*. If the offer turns out to be the best, the respective request is added to the actual route. Most of its lifetime, a vehicle spends serving requests. It starts after receiving and accepting the first request. After reaching the nearest customer it proceeds to the next customer belonging to the route. If the vehicle reaches the last customer on the current route, it waits in the location until a new request arrives. When all requests are dispatched among the available vehicles, the waiting vehicle returns back to the depot.

It is easy to see, that all agents acting in the system operate on *Customer* requests. Each such request is characterized by:

- coordinates on the plane - (x, y),
- required demand - d,
- arrival time - t,
- time in which a request is served - ts,
- state of the request (*available*, *blocked*, *finished*, *canceled*) - sC,
- customer's waiting time for servicing - twC.

The process of communication between the agents acting in the system takes place at two levels. The first level includes messages exchanged between agents while creating agents, changing the state of the vehicle or request, etc. The second one includes the messages exchanged between agents during the process of solving the DVRP instance.

2.2 Process of Solving the DVRP Instance

The whole process of solving an instance of the DVRP carried out by agents using in the proposed platform is divided into several steps, which are presented in form of the following pseudocode.

MAS-DVRP procedure

```
BEGIN
   Allocate static requests
   All vehicles with the assigned requests start moving
   WHILE (system is waiting for the next event)
      IF (request_event)
         CASE "new request":
            allocate new request to available vehicles
         CASE "end of requests":
            all waiting vehicles return to the depot
      ELSE IF (vehicle_event)
         CASE "vehicle v(i) start moving":
            vehicle v(i) is en route the first location
         CASE "vehicle v(i) reached the location p":
            IF (NOT location p is the last location)
               vehicle v(i) proceeds to the next location
            ELSE
               vehicle v(i) waits in location p
         CASE "vehicle v(i) is waiting":
            IF (end of requests)
               vehicle v(i) is moving to the depot
         CASE "vehicle v(i) is stopped at the depot":
            IF (all vehicles returned to the depot)
               STOP
      Do intra-route and inter-routes operations between events
END
```

Allocating static requests to available vehicles requires a solution to the classical version of the VRP. In the proposed approach such a solution is obtained by the procedure based on the polar representation of each vertex (customer with its request), which uses the idea originated from the *split* phase of the *sweep* algorithm [3].

Assignment of static requests to available vehicles is carried out in the form of messages exchanged between *ARequestManager* and *AVehicle* agents.

After allocation of static requests to the available vehicles, vehicles start moving to serve these requests. At the same time the system starts waiting for an event. There are two possible kinds of events which can arise in the system: *request events* and *vehicle events*.

Request events are generated by the *ARequestGenerator* agent which announces each new arriving request to the *ARequestManager* or reports on reaching the end of the list of requests. In the first case, the process of assigning new request to the available vehicles is starting. In the second case, all waiting vehicles are moving to the depot.

Vehicle events are generated by *AVehicle* agents which report on their starting, reaching next customer on its route, waiting for the next new request in last visited customer's location after servicing all already assigned requests and finishing their routes at the depot.

As it was mentioned above, after receiving new customer request, the system is trying to assign it to the available vehicles in order to minimize the total cost of servicing. Of course, assigning a set of dynamic requests to the available vehicles is carried out in a dynamically changing environment, where all vehicles are serving the customers already assigned to their routes.

First, *ARG* reads (or generates) a new dynamic request and sends it to the *ARM*. In a simulation mode it is assumed that arriving requests are uniformly distributed on the plane. After receiving a new request from the *ARG*, *ARM* initializes a session using the *Contract Net Protocol (CNP)* [8] and starts communication between *ARM* and *AV* agents. As *Initiator* it announces the request to each *AV* agent sending around a call for proposal (`cfp`) message. *AV* (as *Participants* or *Contractors*) are viewed as potential contractors. Each *AV* agent after receiving the request (with customer data) from the *ARM*, calculates the cost of inserting a new customer into the existing route. If an insertion of a new customer into the existing route does not violate the vehicle's capacity, as well as the estimated time of servicing all requests assigned to this vehicle does not exceed predefined time, the calculated cost of insertion is sent back (as `propose` message) to the *ARM*. Otherwise, the *AV* sends back the rejection (`reject`) message. *ARM* after receiving proposals from all *AV* agents, chooses the one with the lowest cost of insertion. Next, it sends the `accept-proposal` message to the *AV* which is awarded and the `reject-proposal` to the others. *AV* which receives the `accept-proposal` message, inserts the customer into the current route and sends the `inform-done` message if the operation is performed successfully and `failure` message, otherwise. The above steps are repeated for each new request.

In order to improve a solution obtained by the assignment procedures for static and dynamic requests, two kinds of the improvement procedures are additionally defined for *ARM* and *AV* agents and performed during the time gap. Each *AV* agent executes a set of operations that aim at improving the cost of its route (*intra-route* operations which operate on one selected route). In addition, the *ARM* agent also periodically performs global moves that aim at improving the global solution (*inter-routes* operations which operate on at least two selected routes).

Three *intra-route* operations include: *v1_2opt*, *v1_relocate* and *v1_exchange*. *v1_2opt* is the implementation of *2-opt* algorithm where the sequence of customers visited by the vehicle on given route is changed by eliminating two edges and reconnecting the two resulting paths in a different way to obtain a new route. In *v1_relocate* the sequence of customers visited by the vehicle on a given route is changed by moving one randomly chosen customer from its current position to another one. In *v1_exchange* two selected customers from current route are swapped. All possible moves are considered in above operations and moves that shorten the current route are accepted. The resulting route with the greatest reduction of the total cost is accepted as a new tour of the vehicle.

The above *intra-route* operation could be initialized by the *ARM* agent and next performed by *AV* agent or directly performed by *AV* agent. In the first case, *ARM* decides whether and when use the operation and sends the proper message to the particular *AV*. In the second case, *AV* autonomically decides about performing the operation.

Two *inter-routes* operations are proposed and implemented in the system: *v2_relocate*, where one selected customer from one route is moved to the second route, and *v2_exchange*, where two selected customers from two different routes are selected and swapped. Both operations are initialized an performed by the *ARM* agent.

Here, it is worth noting, that although the idea of *intra-route* and *inter-routes* operations implemented in the proposed system are similar to the ones known from the local optimization methods operating on one or two different routes, the approach based on intelligent software agents presented in the paper does not assume existence of central mechanism to control relocating or exchanging processes. Each agent is an independent entity and autonomically decides about its behavior trying to achieve the common goal.

3 Computational Experiment

To validate the proposed approach computational experiment has been carried out. One of the goal of the experiment has been to determine the quality of the solution (global cost of serving all requests in the predefined time) proposed by the system measured as a mean relative error (MRE) from the best known solution to the static case. Additionally, for dynamic cases, the influence of the degree of dynamism of the problem instance and frequency of the customer request arrivals on the solution have been observed. Degree of dynamism (*DoD*) is measured here as a proportion of the number of dynamic requests to the number of all requests [7]. The second goal of the experiment has been to determine values of certain indicators describing some important features of the fleet of the solutions. In the experiment, the number of vehicles needed to serve all requests in the predefined time, as well as the utilization of vehicles' space have been evaluated for different levels of dynamism and different frequencies of the customer requests arrivals.

The proposed agent-based approach was tested on the classical VRP datasets transformed into the dynamic version through not revealing all requests in advance. The experiment has involved 5 benchmark instances (*vrnpc1* - *vrnpc5*) available from the *OR-Library* [12]. Each benchmark instance includes information about the number of customers, capacity of vehicles, coordinates of depot and coordinates and demands of customers. The selected instances involve 50-199 customers located randomly over the plane and have only capacity restriction.

Table 1. The performance of the proposed agent-based system (the *MRE* (%) calculated from the best known result for the static case and the number of served requests *#R*)

		vrpnc1 (50)		*vrpnc2* (75)		*vrpnc3* (100)		*vrpnc4* (150)		*vrpnc5* (199)	
DoD	λ	*MRE*	*#R*	*MRE*	*#R*	*MRE*	*#R*	*MRE*	*#R*	*MRE*	*#R*
0%		0,5%	50	3,8%	75	3,8%	100	5,1%	150	4,6%	199
20%	6	15,3%	50	10,4%	75	10,6%	100	10,1%	150	9,2%	199
	10	12,7%	50	9,5%	75	9,3%	100	9,9%	150	8,8%	199
	20	2,2%	50	7,9%	75	4,5%	100	6,6%	150	7,7%	199
	30	2,0%	50	6,1%	75	4,2%	100	6,3%	150	6,3%	199
	60	1,9%	50	4,7%	75	3,9%	100	5,4%	150	4,9%	199
50%	6	46,2%	50	31,3%	75	28,3%	100	-	134	-	165
	10	32,4%	50	31,2%	75	37,8%	100	-	145	-	193
	20	14,2%	50	21,3%	75	22,3%	100	37,5%	150	27,3%	199
	30	14,1%	50	15,3%	75	18,4%	100	27,6%	150	20,1%	199
	60	7,3%	50	10,4%	75	11,9%	100	19,4%	150	18,3%	199
80%	6	65,3%	50	44,6%	75	-	92	-	98	-	118
	10	24,4%	50	39,7%	75	-	98	-	124	-	156
	20	13,4%	50	38,1%	75	33,5%	100	-	149	-	180
	30	12,3%	50	29,9%	75	32,3%	100	45,7%	150	34,4%	199
	60	10,1%	50	23,4%	75	28,4%	100	36,5%	150	28,1%	199
100%	6	73,6%	50	-	69	-	79	-	102	-	65
	10	38,5%	50	-	74	-	93	-	111	-	93
	20	22,4%	50	51,0%	75	49,8%	100	-	146	-	179
	30	19,2%	50	42,9%	75	44,7%	100	41,4%	150	54,2%	199
	60	18,3%	50	34,0%	75	39,2%	100	29,8%	150	43,8%	199

The proposed simulation model was run for the number of dynamic requests varying from 0% (pure static problem) to 100% (pure dynamic problem) of all requests (0%, 20%, 50%, 80%, 100% in the reported experiment). Additionally, for each dynamic instance, it has been assumed that dynamic requests may arrive with various frequencies. In the experiment arrivals of the dynamic requests have been generated using the Poisson distribution with λ parameter denoting the mean number of requests arriving in the unit of time (1 hour in the experiment). For the purpose of the experiment value of λ was set to 6, 10, 20, 30, 60.

It is also assumed that all requests have to be served in the predefined time T (in experiment $T = 12$ hours).

The above assumptions produced 21 test instances (1 static and 20 dynamic) for each instance, giving in total 105 test instances. Moreover, each test instance have been solved five times and mean results from these runs were recorded. The experiment results are presented in Tab. 1-2.

Tab. 1 shows values of the percentage increase of costs resulting from allocating all requests to available vehicles in the predefined time as compared with the cost of the best known solution to the static case. Moreover the number of requests served by available vehicles in the predefined time is shown and denoted

Table 2. The performance of the proposed agent-based system (number of vehicles $\#V$ and vehicle space utilization factor U) for selected instances of the DVRP

		vrpnc1 (50)		*vrpnc2* (75)		*vrpnc3* (100)		*vrpnc4* (150)		*vrpnc5* (199)	
DoD	λ	$\#V$	U	$\#V$	U	$\#V$	U	$\#V$	U	$\#V$	U
0%		5	97.1%	11	88.6%	8	91.1%	12	93.1%	5	93.7%
20%	6	5	97.1%	11	88.6%	8	91.1%	12	93.1%	17	93.7%
	10	5	97.1%	11	88.6%	8	91.1%	12	93.1%	17	93.7%
	20	5	97.1%	11	88.6%	8	91.1%	12	93.1%	17	93.7%
	30	5	97.1%	11	88.6%	8	91.1%	12	93.1%	17	93.7%
	60	5	97.1%	11	88.6%	8	91.1%	12	93.1%	17	93.7%
50%	6	5	97.1%	11	88.6%	8	91.1%	-	-	-	-
	10	5	97.1%	11	88.6%	8	91.1%	-	-	-	-
	20	5	97.1%	11	88.6%	8	91.1%	12	93.1%	17	93.7%
	30	5	97.1%	11	88.6%	8	91.1%	12	93.1%	17	93.7%
	60	5	97.1%	11	88.6%	8	91.1%	12	93.1%	17	93.7%
80%	6	5	97.1%	12	81.2%	-	-	-	-	-	-
	10	5	97.1%	11	88.6%	-	-	-	-	-	-
	20	5	97.1%	11	88.6%	8	91.1%	-	-	-	-
	30	5	97.1%	11	88.6%	8	91.1%	12	93.1%	18	88.5%
	60	5	97.1%	11	88.6%	8	91.1%	12	93.1%	17	93.7%
100%	6	5	97.1%	-	-	-	-	-	-	-	-
	10	5	97.1%	-	-	-	-	-	-	-	-
	20	5	97.1%	12	81.2%	8	91.1%	-	-	-	-
	30	5	97.1%	11	88.6%	10	72.9%	14	79.8%	20	79.7%
	60	5	97.1%	11	88.6%	8	91.1%	12	93.1%	18	88.5%

as $\#R$. The first two columns of the table show degree of dynamism (in %) and mean number of requests per hour.

By analyzing the results obtained by the proposed agent-based system for the static case of the VRP as presented in the first row of Tab. 1 one can conclude that the system performs well. The average value of MRE as compared with the best known solution is not greater than 5%, but it depends on the instance and for most instances is smaller. Minimal value of the MRE observed during the experiment is equal to 0% for *vrpnc1* instance or close to 2-3% for most of the instances. Although the costs obtained by the system are slightly higher than costs of best solutions produced by leading methods, like tabu search algorithms [6], the proposed multi-agent system still remains competitive to many other approaches.

Further analysis of results obtained during the experiment and presented in Tab. 1 shows that the overall cost strongly depends on the degree of dynamism and frequency of dynamic request arrivals for all tested instances. In most cases the total cost for the dynamic case are substantially higher than for the static one. For instances with the mean number of requests arriving in the time unit at a low level it has been often impossible to serve all arriving requests. Since it has been required that all requests are served during the allowed time the respective costs are not shown in Tab. 1 in such cases.

In Tab. 2 the performance of the proposed agent-based system measured in terms of the number of vehicle needed to serve all requests within the required time horizon ($\#V$) as well as vehicle space utilization factor (U) are shown. These values has been calculated for different levels of dynamism and different frequencies of the customer requests for each tested instance of the problem. Simulation tests have been carried out for several levels of dynamism and mean number of requests per hour, and similar as in Tab. 1, the first two columns of the table show both variables. For instances without possibility of serving all requests within the allowed time the values of indicators are shown.

By analyzing the results shown in the table one can conclude that the number of vehicles needed for serving all requests does not depend on DoD or λ for instances with a relatively small number of customers (here *vrpnc*1 and *vrpnc*2). In all cases the same number of vehicles is required. The utilization space of vehicles exceeds 90%. But the situation changes when we consider instances where the number of customers exceeds 100 customers. First of all, for small values of λ, a certain number of requests can not be served. Next, for some instances it can be observed that the number of vehicles needed grows with the number of dynamic requests. The obvious consequence is the decrease of vehicle utilization factor. During the simulation it has been observed especially for instances *vrpnc*4 and *vrpnc*5 that the number of vehicles needed grows especially at the end of operational time, when new requests, in order to be served, have to be assigned to new vehicles.

It should be noted that comparisons of solutions to dynamic cases can not be directly compared with the approaches proposed by other authors. This is mainly due to differences in problem formulation and differences in datasets used for evaluating the results. In case of static instances the results are, however, fully comparable.

4 Conclusions

The main contribution of the paper is proposing a new approach to solving dynamic vehicle routing problem based on the multi-agent paradigm. This approach uses a multi-agent platform which enables simulating activities of the transport company and analyzing various scenarios with respect to the dynamic routing of the fleet of vehicles.

The proposed approach extends the range of existed algorithms for solving DVRP and, as computational experiment has proved, it can offer solutions of a good quality for the static case and gives a certain number of performance indicators for observing how dynamic nature of the problem can influence the realization of all customer requests in a given period of time. Additionally, results from simulation tests allow the decision maker to find nearly optimal values of some important factors, like for example number of vehicles needed to serve all coming requests.

Despite good performance of the proposed system, in authors' opinion there is still a place for research on improvement algorithms used as intra-route and inter-routes operations. Metaheuristics, like tabu search, evolutionary algorithms or ant optimization algorithms included in agent-based environment with specific features of it, like autonomy of agents, ability to increase computational efficiency through parallelization and possibility of using distributed environment could further improve overall performance of the system.

References

1. Barbucha, D., Jędrzejowicz, P.: Multi-agent platform for solving the dynamic vehicle routing problem. In: Proc. of the 11th IEEE International Conference on Intelligent Transportation Systems (ITSC 2008), Beijing, pp. 517–522. IEEE Press, Los Alamitos (2008)
2. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE. A White Paper. Exp. 3(3), 6–20 (2003)
3. Gillett, B.E., Miller, L.R.: A heuristic algorithm for the vehicle dispatch problem. Operations Research 22, 240–349 (1974)
4. van Hentenryck, P.: Online Stochastic Combinatorial Optimization. MIT Press, Cambridge (2006)
5. Kozlak, J., Creput, J.C., Hilaire, V., Koukam, A.: Multi-agent Approach to Dynamic Pick-up and Delivery Problem with Uncertain Knowledge about Future Transport Demands. In: Fundamenta Informaticae XXI, pp. 1001–1010. IOS Press, Amsterdam (2005)
6. Laporte, G., Gendreau, M., Potvin, J., Semet, F.: Classical and modern heuristics for the vehicle routing problem. International Transactions in Operational Research 7, 285–300 (2000)
7. Larsen, A.: The on-line vehicle routing problem. Ph.D. Thesis, Institute of Mathematical Modelling, Technical University of Denmark (2001)
8. Smith, R.G.: The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers 29(12), 1104–1113 (1980)
9. Thangiah, S.R., Shmygelska, O., Mennell, W.: An agent architecture for vehicle routing problem. In: Proc. of the ACM Symposium on Applied Computing (SAC 2001), Las Vegas, pp. 517–521 (2001)
10. The Foundation for Intelligent Physical Agents, http://www.fipa.org/
11. Jade - Java Agent Development Framework, http://jade.tilab.com/
12. OR-Library, http://people.brunel.ac.uk/~mastjjb/jeb/orlib/vrpinfo.html

Solving Global Optimization Problems Using MANGO*

Akın Günay[1], Figen Öztoprak[2], Ş. İlker Birbil[2], and Pınar Yolum[1]

[1] Boğaziçi University, İstanbul, Turkey
akin.gunay@boun.edu.tr, pinar.yolum@boun.edu.tr
[2] Sabancı University, İstanbul, Turkey
figen@su.sabanciuniv.edu, sibirbil@sabanciuniv.edu

Abstract. Traditional approaches for solving global optimization problems generally rely on a single algorithm. The algorithm may be hybrid or applied in parallel. Contrary to traditional approaches, this paper proposes to form teams of algorithms to tackle global optimization problems. Each algorithm is embodied and ran by a software agent. Agents exist in a multiagent system and communicate over our proposed MultiAgent ENvironment for Global Optimization (MANGO). Through communication and cooperation, the agents complement each other in tasks that they cannot do on their own. This paper gives a formal description of MANGO and outlines design principles for developing agents to execute on MANGO. Our case study shows the effectiveness of multiagent teams in solving global optimization problems.

1 Introduction

Many powerful algorithms for solving global optimization problems exist [1]. Some of the algorithms propose a single, unique technique to solve a problem, while others propose manually-integrated, hybrid approaches. In both cases, algorithms can be run independently or in parallel. Many such approaches have their own advantages over others. Depending on context, one algorithm can defeat others in finding a solution. One intuitive perspective is to enable these algorithms to exist in teams so that they can complement each other in tasks they cannot do well on their own. Depending on the context, one algorithm can help the other algorithm (e.g., if one gets stuck in a local optimum) to continue operations successfully [2,3,4].

(Software) Agents are autonomous computations that can perceive their environment, reason, and act accordingly [5]. Agents are most useful when they exist in a multiagent system so that they can interoperate with other agents. Interoperation requires agents to speak a common language to coordinate their activities and to cooperate if they see fit. Autonomy of the agents implies that agent can choose how and with whom they want to interact. These properties of multiagent system make it an ideal candidate to realize teams of algorithms. While multiagent systems have been used in many areas, their use in solving global optimization problems, as we discuss here, is new.

* This research has been partially supported by the Scientific and Technological Research Council of Turkey by a CAREER Award under grant 107M455. A preliminary version of this paper appeared at AAMAS OPTMAS Workshop 2008.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 783–792, 2009.

Our proposed software environment is MANGO, which provides the necessary utilities to develop agents that can participate in a multiagent system to solve global optimization problems. MANGO enables agents to find each other through a directory system. MANGO contains an extendible protocol for agents to communicate with each other. The protocol messages are related to solving problems, such as exchanging current best points, signaling areas already explored by others, and so on. Hence, agents can find others and cooperate with them on their own. As a result, agent teams are formed and exploited to solve global optimization problems.

There are three main contributions of this paper: (i) We summarize MANGO, our proposed environment for developing multiagent systems that solve global optimization problems cooperatively. (ii) We provide design principles for developing an agent that can participate in a multiagent environment for solving global optimization problems. (iii) We develop a case study in which agents are developed following the above design principles and show how they can actually solve a given global optimization problem.

The rest of this paper is organized as follows: Section 2 explains MANGO in detail, concentrating on the architecture, services, and messaging. Section 3 identifies the design principles for developing a MANGO agent. Section 4 develops a case study to show how MANGO agents can cooperate to solve global optimization problems. Finally, Section 5 gives our conclusions and future research ideas.

2 MANGO Framework

MANGO is a multiagent global optimization framework. It is implemented in Java and uses Java Messaging Service (JMS) technology. The aim of MANGO is to provide a development and experiment environment for global optimization research. Because of this, contrary to conventional multiagent development environments, the agents that can be developed using MANGO are targeted for solving global optimization problems. Concepts that are crucial for global optimization such as problem definitions, problem solutions, and so on are first class entities in MANGO. In the rest of this section we explain the MANGO environment and working principles of its components in detail.

2.1 Architecture Overview

We present the architecture of the MANGO environment in Figure 1. The fundamental entity of the MANGO architecture is the agent. Every task, such as solving global optimization problems, visualization of results and administrative tasks in the MANGO environment is performed by agents.

An agent is implemented as a regular Java executable that uses MANGO API in order to work in the MANGO environment. Agents communicate through JMS, however MANGO API hides details of the JMS communication by providing its own communication methods in order to simplify development process. MANGO uses the service concept of the service oriented architecture (SOA) [5]. In this manner agents may provide services to other agents. For instance, in a typical MANGO environment one agent may provide a service to solve global optimization problems using a specific algorithm whereas another agent might provide a visualization service to graphically represent the results of the optimization algorithm. A third agent might use these two services

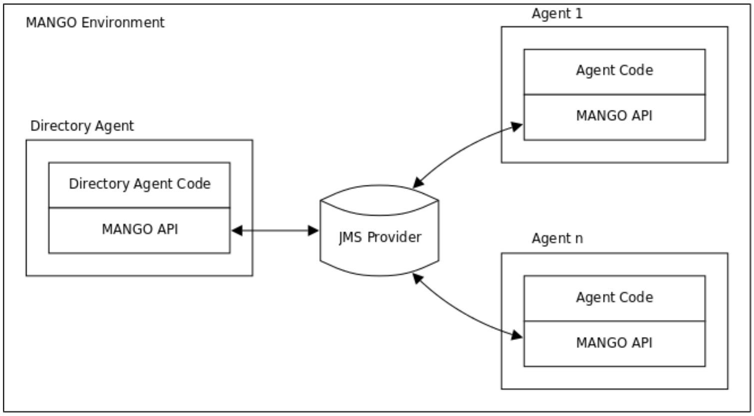


Fig. 1. MANGO Architecture

in combination to solve its global optimization problem and to visualize the results. MANGO environment itself provides a directory agent for management and service discovery purposes.

2.2 Directory Service

In MANGO environment a directory agent is a special agent with administrative and managerial capabilities and each MANGO environment has one directory agent. It keeps track of all other agents and their services. Using this information it also acts as a service matchmaker and provides service discovery service to other agents in the environment. It is also responsible for low level managerial tasks such as maintenance of communication resources.

2.3 MANGO API

The MANGO API is a fully extendible API that provides all necessary facilities for the developers to implement their own agents for the MANGO environment. Figure 2 shows the basic components of the MANGO API. There are four main libraries of classes as the *agent templates, protocol, optimization and service*. Agent templates library provides a set of basic agents with communication capabilities. Developers can implement their own agents through extending these agent templates without considering details such as messaging. Classes in the protocol library are the predefined set of protocols and related messages for agent communication. These protocols are further divided into two libraries as system and optimization protocol libraries. System protocols are mainly used by the agents to communicate with the directory agent for resource management and service discovery purposes. On the other hand optimization protocols are used between the agents to solve these problems in cooperation. Classes under the optimization library provide facilities for global optimization. These include common problem and solution definitions that allow interchange of global optimization problem

Agent Templates	Protocol		Optmization		Service	
JMS Communication	System	Optimization	Problem	Utilities	System	Optimization

Fig. 2. MANGO API

knowledge between agents and utility classes used in these solution and problem definitions. The classes in service library provide support to define agent services, both for administration and optimization.

2.4 Protocols and Messages

MANGO environment provides a set of extendable protocols to coordinate communication between agents. Each protocol specifies a set of message types as classes to specify the content to be exchanged by the agents during the execution of the protocol. We divide the protocols into two categories as system and optimization protocols. System protocols are mainly used between individual agents and the MANGO environment. Some examples of system protocols are agent-register-protocol executed when a new agent joins to the environment, service-register-protocol executed when an agent starts to provide a new service, and service-discovery-protocol executed when an agent searches for a new service. On the other hand optimization related protocols provide simple message exchange blocks that can be combined in order to realize complex high-level cooperative optimization strategies explained in Section 4. Some examples of optimization protocols are solution-request protocol executed when an agent requests the solution of a specific problem from another agent and refrain-request protocol executed when an agent informs another agent not to search a specific region in a given search space, and explore-request protocol executed when an agent requests another agent to explore a certain area.

2.5 Agent Lifecycle

The lifecycle of an agent starts by registering itself to the directory agent. While registering the new agent to the MANGO environment, the directory agent creates necessary communication facilities for the new agent. After the registration process the new agent is ready to act in the MANGO environment. In general an agent can act in three different ways after this point. First, it can use services provided by other agents in the environment to perform its own tasks. Second, it can provide services to other agents. Third, it can do both in parallel. When the agent decides to use services from other agents, it queries the directory agent for the available services. According to the results of this query it communicates with the agents that provide the required service. On the other hand, if the agent decides to provide its own service to other agent, it must register the service to the directory agent in order to inform other agents about its service. An agent can use any number of services provided by other agents at any point of its lifecycle. Similarly, an agent can provide any number of services and do this at any point of its lifecycle. The lifecycle of an agent ends when the agent unregisters all of its services and also itself from the directory agent.

3 Developing a MANGO Agent

When a MANGO agent is being designed, there are three decision points that need to be considered.

Optimization Algorithm: The first point is the agent's main algorithm for attacking the global optimization problem. This algorithm may be any known or newly-developed algorithm for solving a global optimization problem. The agent designer decides on this algorithm and implements it in the agent.

Outgoing Messages: The second component is related to when and with whom the agent is going to communicate during its execution. The communication is necessary for various reasons, but most importantly for coordination. That is, it is beneficial for an agent to position itself correctly in the environment. That is, generally two agents may not want to be searching the same area since probably if they search two different areas they may find a solution faster.

- Needed Services: The questions of when and with whom to communicate are strictly related to the optimization algorithm that the agent is using. If the agent's own algorithm cannot handle certain tasks, the agent would need others' services to handle these. For example, if the agent's optimization algorithm cannot perform local search well, the agent may find it useful to find other agents that can offer local search service. As explained before, whether an agent does offer this service can be found out by querying the directory agent that keeps track of the services associated with each agent. After finding out the agents that offer the service, the agent may contact one of them to receive the service. Alternatively, an agent that can do local search well may be interested in finding out new areas to search when it finishes its local search. Hence, it may be interested in finding others that can suggest new areas to search.
- Played Roles: An agent may decide to take a leader role in the multiagent system and influence the others by suggesting areas to explore or refrain from. The choice of taking this role is up to the agent, but is also affected by the particular algorithm the agent is executing. That is, some algorithms can identify potential "good" areas quickly and thus it is reasonable for the agent to take this role and to inform others about the potential of these areas.

Incoming Messages: The third decision point is related to if and how the agent is going to handle incoming messages. One naive approach is to always answer or follow the incoming messages. For example, if an agent receives an explore message, it can always jump to the areas that is being suggested for exploration. Or, whenever it is prompted for the best solution it has found, it can return its current best solution. However, the following play an important role in how the incoming messages can be handled intelligently.

- Exploration State: The exploration state corresponds to how well the agent has explored the environment. This is important in answering questions, since an agent may prefer not to answer question if it has not explored the environment well or conversely prefer not to follow orders (such as refrain messages) if it has explored the environment carefully. For example, in the beginning of the execution, when the agent did not have enough time to search properly, it may decide not to answer incoming messages related to the best solutions it has found, since its solution may not be representative.

- Agent Sending the Message: Over the course of exaction, an agent may model other agents based on the types of messages they are sending. Based on this model, an agent may decide how and if it is going to handle a message. For example, if an agent sends frequent explore messages to a second agent, the receiving agent may mark the sender agent as a “spammer” and decide to ignore messages coming from that agent.

4 A Case Study

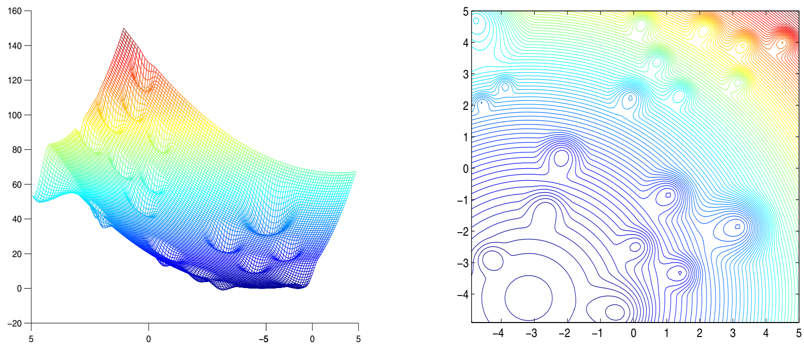
In this section, we provide an illustrative system that has been established to solve a global optimization problem with three cooperative agents in MANGO.

Motivations for the Example: The strategies in this example are motivated by two challenges in global optimization. An important issue that makes very well-known efficient local optimization methods useless for global optimization problems is that the objective function may have multiple local minima. A local optimization method finds one of them, which may or may not be the global one, depending on the initial point from where it has started its search. A first idea to overcome this problem to a certain degree is starting local search from several initial points. But the obvious drawback of this straightforward approach is that many searches may end up in the same local minimum point, i.e., the same local minimum may be rediscovered for several times.

Position of the global minimum is another issue. When the global minimum lies at the bottom of a large basin, i.e., the attraction region, it is relatively easier to find it out since an initial point is near to that large attraction region with a higher probability. A situation at least as hard as a narrow attraction region is a narrow attraction region placed within the attraction region of another local minimum. In this case, there is a significant risk of ending up at the more attractive local minimum point even when we start very close to the global minimum. If we escape from the larger attraction region not to rediscover it, then we may never approach to the global minimum and waste our efforts in irrelevant faraway parts of the search space.

The Problem: We select a two-dimensional problem for our illustration. The problem is produced by the GKLS generator [6]. It has 20 local minima and finding the global minimum is quite hard in the sense that it has a relatively small attraction region and it is located within the attraction region of another local minimum (see Figure 3).

The Agents: The three agents we run in this example are all local optimization agents [7]. We name the three agents as *BFGS*, *TR*, and *PTR*. The BFGSAgent applies BFGS quasi-Newton algorithm: a line search method which progresses by taking steps through directions that provide decrease in the objective function value, so it calculates a direction vector and a step size at each iteration. TRAgent applies a trust region method. It



(a) Mesh plot of the objective function (b) Contours of the objective function

Fig. 3. The problem

generates a model function which is a quadratic approximation to the original objective function. It accepts that the model function is a good approximation within a *trust region*, a Δ-radius ball, and it minimizes the model function in that region. It updates the model function and the trust region radius at each iteration. Finally, PTRAgent applies a perturbed trust region method: It applies a trust region algorithm like TRAgent. But it works more sensitive, i.e., the maximum radius value allowed is small. Also, it follows a perturbed direction in some iterations to increase the chance of finding an unvisited minimum point. That is, the iterates are not moved along the direction as in the regular trust region method; instead, the trust region direction is distorted randomly.

Cooperation Strategies: The cooperation strategies applied by three agents are illustrated in Figure 4. They follow three basic ideas to improve their performance:

- penalize approaching to already discovered local minima (BFGS)
- do not enter regions searched by others (TR)
- conduct a more sensitive search in the region searched by the other agents, so that a possible global optimum near a local one is not missed (PTR)

In this context, the cooperation procedures applied by each agent are summarized as follows:

- BFGSAgent

  ListenMessage: If an INFORM_SOLUTION message is received, then add the received point x_r to the *penalized* list so that approaching to x_r increases the objective function.

  SendMessage: If converged to a point x_f, starting from a point x_0, then send the ball with center x_f and radius $\|x_f - x_0\|$ to both TRAgent and PTRAgent as a REFRAIN_REGION message.

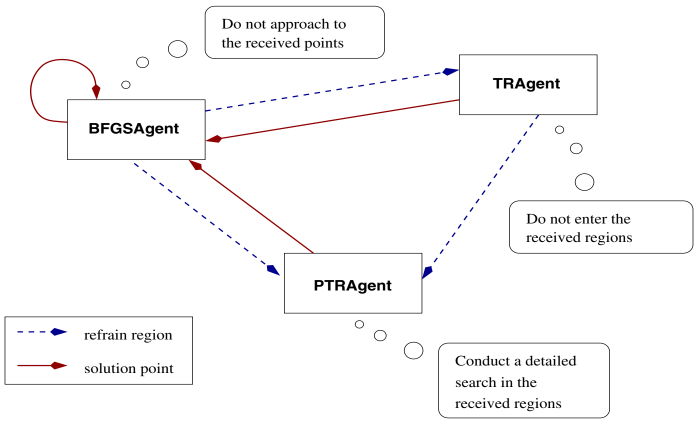


Fig. 4. The cooperation strategies

- TRAgent

  ListenMessage: If a REFRAIN_REGION message is received, then add the received region to the *refrained* list so that if case of entrance to that region leave the ongoing search and start a new search from some random point in the solution space.

  SendMessage: If converged to a point x_f, then send it to BFGSAgent as an INFORM_SOLUTION message. If the radius of the last trust region is Δ_f, then send the ball with center x_f and radius $k\Delta_f$ to PTRAgent as a REFRAIN_REGION message, $k \geq 1$.
- PTRAgent

  ListenMessage If a REFRAIN_REGION message is received, then it is added to the *explore* list so that it is going to be searched for a minimum other than the center of that region.

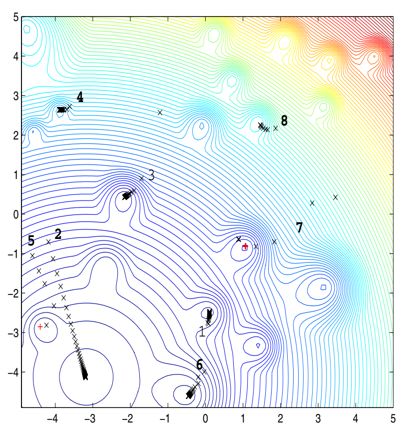


Fig. 5. The penalizing strategy applied by BFGSAgent

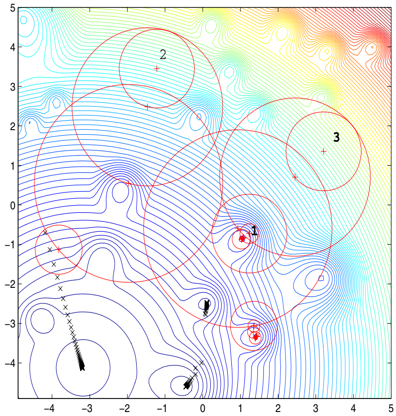


Fig. 6. The leaving strategy applied by TRAgent

SendMessage If converged to a point x_f, then send it to BFGSAgent as an INFORM_SOLUTION message.

Observations: We next illustrate how those strategies work on the above mentioned problem with 20 minima. In Figure 5, the search paths of BFGSAgent has been marked with $\times$. The points marked by $+$ are the minima that have been already discovered by other agents and sent to BFGSAgent. The consecutive searches conducted by BFGS Agent are numbered. As the figure points out, the search is discouraged to approach to the previously converged points, either by BFGSAgent or by the other agents.

In Figure 6, we illustrate the second idea. The search paths of TRAgent are marked with $+$ signs, the circles are the trust regions. The paths are numbered at their starting points. During its second run, TRAgent has entered to the refrain region sent by the

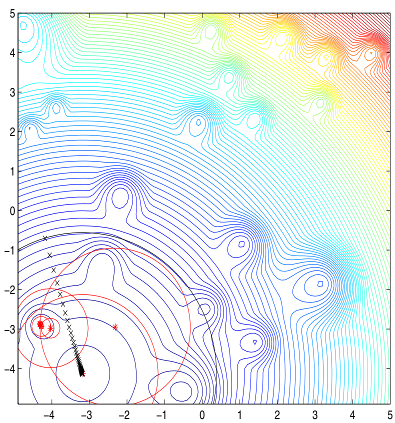


Fig. 7. The research strategy applied by PTRAgent

BFGSAgent at its forth step. Thus, it has left the second path at that point and started a new search from another point so that it has spent its effort for discovering another local minimum (at the end of the third path).

Finally, in Figure 7, we can see the steps of PTRAgent marked by $*$ signs. It has started a new search in the refrain region sent by BFGSAgent, which has provided finding the global minimum point.

5 Conclusions

We have introduced a new multi-agent environment for global optimization. The proposed environment provides a flexible mechanism that can be used to design new cooperation strategies among different global optimization algorithms. We have demonstrated on an illustrative example that the design of different cooperation strategies can significantly enhance the performance of individual algorithms. In the future, we intend to focus on different strategies and demonstrate their performances with empirical results.

References

1. Floudas, C.A.: Deterministic Global Optimization: Theory, Methods and Applications, 2nd edn. Nonconvex Optimization and Applications, vol. 37. Kluwer Academic Publishers, Dordrecht (2000)
2. Talukdar, S., Baerentzen, L., Gove, A., Souza, P.D.: Asynchronous teams: Cooperation schemes for autonomous agents. Journal of Heuristics 4(4), 295–321 (1998)
3. Tyner, K., Westerberg, A.: Multiperiod design of azetropic seperation systems i: An agent based approach. Computers and Chemical Engineering 25, 1267–1284 (2001)
4. Siirola, D., Hauan, S., Westerberg, A.: Toward agent-based process systems engineering: Proposed framework and application to non-convex optimization. Computers and Chemical Engineering 27, 1801–1811 (2003)
5. Singh, M.P., Huhns, M.N.: Service-Oriented Computing: Semantics, Processes, Agents. John Wiley & Sons, Chichester (2005)
6. Gaviano, M., Kvasov, D.E., Lera, D., Sergeyev, Y.D.: Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization. ACM Transactions on Mathematical Software 29(4), 469–480 (2003)
7. Nocedal, J., Wright, S.: Numerical Optimization. Springer, Heidelberg (2006)

Awareness-Based Learning Model to Improve Cooperation in Collaborative Distributed Environments

Mauricio Paletta[1] and Pilar Herrero[2]

[1] Centro de Investigación en Informática y Ciencias de la Computación (CITEC), Universidad Nacional Experimental de Guayana (UNEG), Av. Atlántico, Ciudad Guayana. 8050, Venezuela
mpaletta@uneg.edu.ve

[2] Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo S/N. 28.660 Boadilla del Monte, Madrid, Spain
pherrero@fi.upm.es

Abstract. In the great family of distributed applications, collaborative systems are distinguished by the fact that users, usually agents, are working together to fulfill a common goal. Therefore they have a critical need to interact closely with each other through: information exchange, sharing of applications, control of some over others, and sharing computer resources, among others. However, the success of achieving the common goal depends on the implementation of appropriate collaborations in these dynamic distributed environments. But proper collaborations also depend on several factors that have to do, mainly, with the way in which the environment is at any given time, as well as provide answers to questions like: "With whom it must collaborate?" "Who should collaborate?" On the other hand, awareness is a process where users recognize each other's activities with the promise of co-presence, for example, "What are they doing?", "Where are they?" This paper describes a learning model for collaborative distributed environments that takes into account the experience of awareness collaborations occurring in the environment for achieving the most appropriate future awareness situations.

Keywords: Learning, collaboration, distributed environment, multi-agent system, awareness.

1 Motivation and Related Work

A collaborative environment is one in which multiple users, usually agents, participate in an activity shared by all users. Usually these agents are in remote locations, and then this surround becomes a Collaborative Distributed Environment (CDE). One of the most recent examples of CDEs has to do with Grid Computing [3, 6]. The success of achieving the common goal in proper time and/or to obtain a higher quality of results depends on the implementation of appropriate collaboration mechanisms in these dynamic and distributed environments.

On the other hand, according to CSCW (Computer Supported Cooperative Work), awareness is one of the most interesting topics to achieve cooperation and collaboration

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 793–802, 2009.

by increasing communication opportunities [14]. In this matter, a collaboration process, in CSCW, is lead from five processes [12, 13]: co-presence, awareness, communication, collaboration and coordination. Co-presence gives the feeling that the user is with someone at the same time in a shared environment. Awareness is a process where users recognize each other´s activities on the premise of co-presence; for example, “What are they doing?” “Where are they working?”. Communication and collaboration permit users to collaborate between each other for accomplishing the tasks and common goals. Finally, coordination is needed to resolve the conflicts towards effective collaboration.

Researches in CSCW have already proposed awareness to: 1) give information on the surroundings of the target user [4]; 2) provide common or public space where users can gather and meet [5]; 3) simulate informal communicative opportunities in real world using computers [15]. These awareness-based models are implemented using multi-media technologies to bond physically distributed environments.

In CSCL (Computer Supported Collaborative Learning), awareness is also very important for effective collaborative learning and it plays a part of how the learning environment creates collaboration opportunities naturally and efficiently [16]. Related to this, Gutwin et al identified the following types of awareness [9]:

1) Social: provides information on social relationships within the group selected to carry out the task. For example: What should I expect from other members of this group? How will I interact with this group? What role will I take in this group? What roles will the other members of the group assume?

2) Task: shows how the learners accomplish the task. For example: What do I know about this topic and the structure of the task? What do others know about this topic and task? What tools are needed to complete task? How much time is required? How much time is available?

3) Concept: is the awareness of how a particular activity or knowledge fits into the learner's existing knowledge or completes the task. For example: How does this task fit into what I already know about concept? What else do I need to find out about this topic? Do I need to review any of my current ideas in light of this new information? Can I create a hypothesis from my current knowledge to predict the task outcome?

4) Workspace: is the up-to-the-minute knowledge about other learners' interactions within shared knowledge space. For example: What are the other members of the group doing to complete the task? Where are they? What are they doing? What have they already done?

5) Knowledge: is the information about other learner’s activities used to enhance collaboration opportunities in a shared knowledge space. For example: Who is discussing, looking at the same knowledge that I am looking at now? Who has changed the knowledge since I have last looked at it? What knowledge are they discussing now? What knowledge of my input did they change?

By considering the awareness definition previously given, this paper presents a learning model for CDEs that takes into account the experience of awareness collaborations occurring in the environment for achieving the most appropriate future awareness situations. This learning strategy is based on Artificial Neural Network (ANN) technology. The paper also presents the way in which ANN technology is used to learn different types of awareness and therefore different types of collaborations.

The rest of the paper is organized as follows. Section 2 presents the learning model proposed in this paper, starting from the way in which awareness in CDE is represented, continuing with the collaboration process and ending with the ANN-based learning strategy. Some implementation and evaluation aspects are showed in section 3. Finally, section 4 exposes the paper conclusions as well as the future work related with this research.

2 Learning Model to Improve Cooperation in CDEs

2.1 Awareness Representation

Awareness concepts used for the learning model proposed in this paper are defined by extending and reinterpreting the Spatial Model of Interaction (SMI) [2] key concepts, initially proposed by Herrero et al in [10, 11]. There is a collaborative distributed environment E containing a set of n nodes N_i ($1 \le i \le n$). There are r items that can be shared as a mechanism of collaboration between the nodes (such as power, disk space, data and/or applications). Call these items resources R_j ($1 \le j \le r$). The distinction between a resource and the other, and the particular association of these awareness concepts for each resource is the key to distinguish the types of collaboration that exist in the CDE. We have:

1) $N_i.Focus(R_j)$: It can be interpreted as the subset of the space (environment/ medium) on which the user (agent) in N_i has focused his attention aiming of interaction/collaboration with, according to the resource (collaboration item) R_j.

2) $N_i.NimbusState(R_j)$: Indicates the current grade of collaboration N_i can give over R_i. It could have three possible values: *Null*, *Medium* or *Maximum*. If the current grade of collaboration N_i that is given about R_j is not high, and this node could collaborate more over this resource, then $N_i.NimbusState(R_j)$ will get the *Maximum* value. If the current grade of collaboration N_i that is given about R_j is high but N_i could collaborate a little more over this resource, then $N_i.NimbusState(R_j)$ would be *Medium*. Finally, if N_i is overload, its $NimbusState(R_j)$ would be *Null*.

3) $N_i.NimbusSpace(R_j)$: The subset of the space where N_i is currently collaborating with R_j. It will determine those nodes that could be taken into account for new collaborative necessities.

4) $R_j.AwareInt(N_a, N_b)$: This concept quantifies the degree of collaboration over R_j between a pair of nodes N_a and N_b. It is manipulated via *Focus* and *Nimbus*, requiring a negotiation process. Following the awareness classification introduced by Greenhalgh in [8], it could be *Full*, *Peripheral* or *Null* according to (1).

$$R_j.AwareInt(N_a,N_b) = \begin{cases} N_b \in N_a.Focus(R_j) \wedge N_a \in N_b.Nimbus(R_j),\ Full \\ (N_b \in N_a.Focus(R_j) \wedge N_a \notin N_b.Nimbus(R_j)) \vee \\ (N_b \notin N_a.Focus(R_j) \wedge N_a \in N_b.Nimbus(R_j)),\ Peripheral \\ \text{Otherwise},\ Null \end{cases} \quad (1)$$

5) $N_i.RequieredTask(R_j)$: Indicates that N_i requires collaboration with R_j.

6) $N_i.TaskResolution(R_j)$: Determines the score to collaborate R_j in N_i. It is represented with a value within [0, 1]. The closer the value is to 0 the hardest it will be for

N_i to collaborate the R_j necessity. The higher the value (closer to 1) is the complete willingness to collaborate.

Based on the previous concepts, the basic idea is to solve any collaboration necessity given by N_i.*RequieredTask*(R_j) according to the current state of the CDE given by N_i.*Focus*(R_j), N_i.*NimbusState*(R_j) and N_i.*NimbusSpace*(R_j), and therefore following the awareness conditions given by R_j.*AwareInt*(N_a, N_b). To do this it is necessary to obtain the N_i.*TaskResolution*(R_j) score for each node associated to a particular resource, and therefore to identify the better node (usually associated with higher value) for collaborating with who is sending the requirement.

Also, it is important to emphasize that to learn the association (current CDE conditions, *TaskResolution*) for any different resource, represents the apprenticeship of different types of awareness, and therefore collaborations occurring in the CDE. The learning strategy presented in this paper takes into account this aspect. Section 2.2 below explains the way and architecture in which this collaboration process is done.

2.2 Architecture and Collaboration Process

Any node N_a in the CDE is endowed with an agent, who has the corresponding information about *E*, i.e.: N_a.*Focus*(R_j), N_a.*NimbusState*(R_j) and N_a.*NimbusSpace*(R_j) for each R_j. All the nodes (agents) are communicated among each other in the CDE. When there is a requirement from any node N_b about the collaboration on any resource R_j, the agent in N_b sends a N_b.*RequieredTask*(R_j) necessity which is received by the others agents (nodes) in the environment. This is done by sending a message including the N_b.*Focus*(R_j), and N_b.*NimbusSpace*(R_j) as parameters in it.

Any agent is also endowed with two services: S_{TR} and S_{NC}. Once N_b.*RequieredTask*(R_j) is received, S_{TR} is used for N_a to obtain the N_a.*TaskResolution*(R_j) by using the corresponding R_j.*AwareInt*(N_a, N_b) and N_a.*NimbusState*(R_j). After that, N_a replies to N_b the corresponding N_a.*TaskResolution*(R_j) obtained, and, based on this information, S_{NC} is used for N_b to decide the better node to collaborate with it. Fig. 1 shows a general outline of this collaboration process.

S_{TR} is based on ANN technology. There is a different ANN-R_j for each resource R_j. Once the agent receives the N_b.*RequieredTask*(R_j) stimulus, S_{TR} decides first the corresponding ANN-R_j, and then uses it to obtain the N_a.*TaskResolution*(R_j) score. S_{NC} is based on a decision algorithm. Finding the node with the higher value is an example.

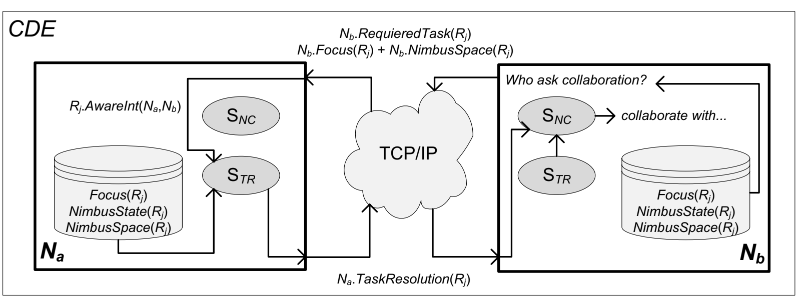


Fig. 1. General outline of the collaboration process

Architecture used to implement this multi-agent system was SOFIA (SOA-based Framework for IA) [18]. SOFIA focuses on the design of a common framework for IAs with the following characteristics: 1) it merges interdisciplinary theories, methods and approaches, 2) it is extensible and open as to be completed with new requirements and necessities, and 3) it highlights the agent´s learning process within the environment. SOFIA general architecture contains four main components (see Fig. 2):

1) The Embodied Agent (IA-EA) or the “body”: It is a FIPA based structure [7] because it has a Service Directory element which provides a location where specific and correspondent services descriptions can be registered. The IA-EA encloses the set of services related to the abilities of sensing stimuli from the environment and interacting with it.

2) The Rational Agent (IA-RA) or the “brain”: This component represents the agent’s intelligent part and therefore, it encloses the set of services used by the agent to implement the process associated with these abilities. It is also a FIPA based structure. S_{TR} and S_{NC} are part of this agent.

3) The Integrative/Facilitator Agent (IA-FA) or the “facilitator”: It plays the role of simplifying the inclusion of new services into the system as well as the execution of each of them when it is needed. The basic function of the IA-FA is to coordinate the integration between the IA-SV and the rest of the IA components. This integration is needed when a new service is integrated with the IA and therefore registered into the corresponding Service Directory or even when an existing service is executed.

4) The IA Services or “abilities” (IA-SV): It is a collection of individuals and independent software components integrated to the system (the IA) which implements any specific ability either to the IA-EA or the IA-RA.

By using SOFIA in this proposal the following should be considered:

1) The IA-SV agent manages *Focus* and *Nimbus* of each resource (as “abilities”).

2) The IA-EA agent (“body”) manages the CDE communication issues.

3) The collaboration decision process is performed by the IA-RA agent (“brain”) by using S_{TR} and S_{NC}.

On the other hand, in overloaded conditions (for example when *AwareInt* is *Peripheral* or *Null*) it is necessary to try to extend the *Focus* or *Nimbus* of one of the nodes so that the *AwareInt* could change to *Full*. This role is also performed by IA-RA through a negotiation process or dialogue that takes place between the IA-RA and the IA-SV. In this regard, the IA-FA agent (“facilitator”) is responsible to manage this negotiation process (see Section 2.4 for details).

2.3 Learning Awareness in CDE

Learning collaborations in this context, by following the awareness conditions, means to learn what the association between the situation environment *E* is in a given moment, and the collaboration scores given to that specific situation ($N_i.TaskResolution(R_j)$ for each node). The strategy to achieve this goal is based on training an ANN performed by the IA-RA agent. Previous results of this research area can be reviewed in [17].

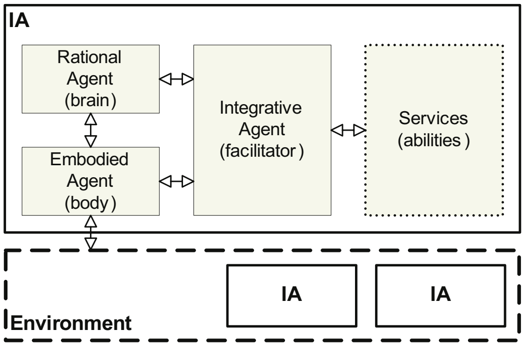


Fig. 2. The SOFIA general architecture.

The ANN used in this proposal consists on a supervised model. There are 2 inputs and 1 output. Being N_b the node who requires collaboration on R_j and therefore who sends the N_b.*RequieredTask*(R_j), the output represents the N_a.*TaskResolution*(R_j) score related to each R_j. The inputs are the following:

- A value *Nst* ∈ [0,1] representing N_a.*NimbusState*(R_j); *Nst* = 1 ⇔ N_a.*NimbusState*(R_j) = *Maximum*, *Nst* = 0.5 ⇔ N_a.*NimbusState*(R_j) = *Medium*, or *Nst* = 0 ⇔ N_a.*NimbusState*(R_j) = *Null*.
- A value *AwI* ∈ [0,1] representing R_j.*AwareInt*(N_a, N_b); *AwI* = 1 ⇔ R_j.*AwareInt*(N_a, N_b) = *Full*, *AwI* = 0.5 ⇔ R_j.*AwareInt*(N_a, N_b) = *Peripheral*, or *AwI* = 0 ⇔ R_j.*AwareInt*(N_a, N_b) = *Null*.

As was mentioned previously, there is a different ANN-R_j for resource. This is an important aspect because:

1) Each collaboration issue can be trained separately from the others. Therefore, each awareness type (social, task, knowledge, and others) can be treated separately.
2) The ANN learning process is less complex so that, the quality of the response given by each ANN-R_j is better.
3) The model is extensible by adding new ANNs when a new collaboration issue (resource) should be incorporated to the CDE.
4) The ANN-R_j do not depends on the nodes. The same trained ANN-R_j is used by all the nodes in the entire CDE. When a new node should be incorporated to the CDE, the only step it needs to required for having and participating in this collaboration model is to have a copy of each ANN-R_j.

Patterns for learning are obtained either by actual scenarios that have been stored during the dynamics of the multi-agent system, or by an automatic generation, in case a deterministic expression for calculating N_i.*TaskResolution*(R_j) exists. Since each ANN-R_j has only 2 inputs and each input has 3 possible values, there are 9 different patterns. Therefore patterns for training any ANN-R_j are the 9 possible combinations of *Nst* = {0, 0.5, 1} and *AwI* = {0, 0.5, 1}.

2.4 Negotiation Process

Sometimes the CDE current conditions are overloaded (N_a.*TaskResolution*(R_j) scores are 0 or closer to 0 $\forall a$, $1 \le a \le n$). In this regard IA-RA initiates a negotiation process aiming to change the environment current conditions in order to obtain a new acceptable answer. The negotiation process is performed by using a protocol defined as part of this model. This protocol is used by IA-RA, IA-FA and IA-SV and consists of the following dialogue:

- REQUEST from IA-RA to IA-FA: IA-RA is aware of an overload in any of the collaborations suggested by S_{NC} (indicated by the *TaskResolution*(R_j) scores) and it decides, with this message, to negotiate with any node an option relief. The information related with the resource overloaded is part of the message.
- REQUEST from IA-FA to IA-SV: Once IA-FA receives the request from IA-RA, and as IA-FA knows what the current "abilities" (*Focus* and *Nimbus*) are, it asks for help aiming to find some node (IA-SV) that could change the abilities mentioned above.
- CONFIRM from IA-SV to IA-FA: A node is confirming that it has changed its abilities (it is available to collaborate) and it is informing its new *Focus* and *Nimbus*.
- DISCONFIRM from IA-SV to IA-FA: A node is confirming that it cannot or it is not interested in changing its abilities.
- INFORM from IA-FA to IA-RA: Once the IA-FA receives the confirmations/disconfirmations from the nodes, and it upgrades all the information related with the *Nimbus* and *Focus* of these nodes, IA-FA sends to IA-RA this updated information in order to execute the process explained in Section 2.2.

Next section presents some aspects related with the implementation and evaluation of the model proposed in this paper and previously defined.

3 Implementation and Evaluation

SOFIA was implemented using JADE [1] framework as it is FIPA-compliant as well as an open-source (based on Java). JADE behaviour model associated to IA-RA, IA-EA, IA-FA, and IA-SV agents were implemented. Fig 3 shows the corresponding implementation class diagram.

The evaluation of the model was carried out by considering the following aspects or steps:

1) The ANN was trained with the 9 possible different patterns by using the deterministic expression *Nst* * *AwI* to calculate the output (N_i.*TaskResolution*(R_j)). ANN was implemented with both the Multilayer Perceptron (MLP) model and the Radial Based Function Network (RBFN) model aiming to compare learning capacities. Results of this training process are showed below.

2) It used different configurations in the CDE, from 5 nodes to 25 nodes varying in 5 for each test block ($n \in \{5, 10, 15, 20, 25\}$) aiming to evaluate the model capability for managing growth in the environment conditions. Simulation for this CDE was carried out in the same computer. Each node was represented by a different SOFIA-based agent.

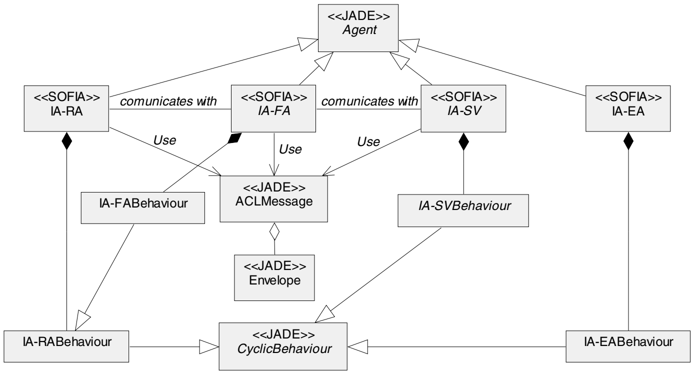


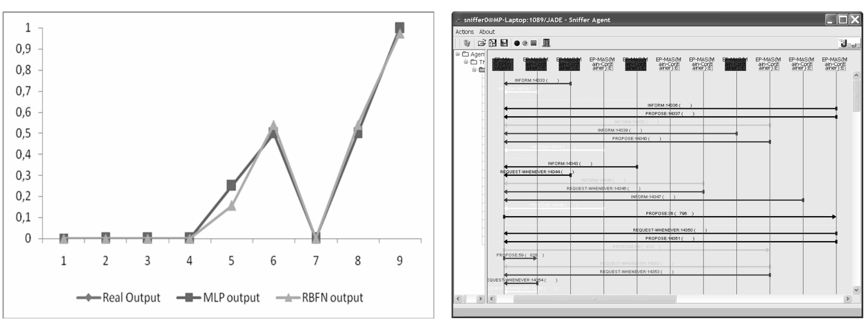
Fig. 3. SOFIA/JADE-based implementation class diagram

3) It used 8 different resources ($r = 8$), aiming to evaluate the model capability for managing different types of awareness for collaboration/cooperation in the CDE.

4) Starting with an initial CDE randomly generated, 100 N_b.*RequieredTask*(R_j) were originated at random (by selecting $N_b \in [1, n]$ and $R_j \in [1, 8]$). Approximately 15% of the scenarios were under overload conditions in order to test the negotiation process aiming to quantify the ability of the model to solve all these specific situations.

Fig. 4-a shows the results of the learning process. In both ANN models (RBFN and MLP) the error is closest to 0. RBFN was configured with 1 hidden unit and was trained in 5500 epochs. MLP instead was configured with 3 hidden units and needed 260.000 epochs for the training process.

The results of the simulation indicate that the proposed model has 92% success in negotiating the way to resolve overload conditions. In those cases where the negotiation was not successful (8%) any node (IA-SV) would or could modify its *Focus/Nimbus* (abilities) based on the current environment configurations. Depending on the number of nodes in the system, the negotiation process requires different time periods for execution (from less than 1 second to about 82 seconds), some of which may not be acceptable because the dynamic of the collaboration process. Fig. 4-b



(a) (b)

Fig. 4. a) ANN training results by using RBFN and MLP, b) Screenshot of the JADE sniffer

shows a screenshot of the JADE Sniffer agent window in which is possible to note the exchange messages during the negotiation process.

4 Conclusions and Ongoing Work

In this paper we use the awareness theoretical aspects to define a multi-agent system and learning model to improve cooperation/collaboration in distributed environments. The proposal model has the capability to learn different types of awareness from previous collaborations that were carried out in the environment to foresee future scenarios. The model has also the capability to negotiate a new CDE configuration in overloaded conditions. In this matter, the results obtained show that the proposed model has 84% success in negotiating the way to resolve overload conditions. However, with more than 15 nodes in the system, the negotiation process requires high time periods for execution.

Even though this method has not yet been tested neither in a real CDE nor a distributed environment simulation, the manner in which the current state of a CDE is represented, and communication between nodes is done, is suitable for real environments. Therefore it is expected that the results of preliminary experiments are also achieved in real scenarios.

We are working on testing this method in real CDE scenarios to obtain more accurate results in experiments. We are also working in reducing the time needed for the negotiation process by properly selecting the nodes with which to establish the dialogue and thus reducing the number of nodes to negotiate.

References

1. Bellifemine, F., Poggi, A., Rimassa, G.: JADE – A FIPA-compliant agent framework, Telecom Italia internal technical report. In: Proc. International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAM 1999), pp. 97–108 (1999)
2. Benford, S.D., Fahlén, L.E.: A Spatial Model of Interaction in Large Virtual Environments. In: Proc. of the Third European Conference on Computer Supported Cooperative Work, Milano, Italy, pp. 109–124. Kluwer Academic Publishers, Dordrecht (1993)
3. Buyya, R., Abramson, D., Giddy, J., Stockinger, H.: Economic Models for Resource Management and Scheduling in Grid Computing. Journal of Concurrency and Computation: Practice and Experience 14(13-15), 1507–1542 (2002)
4. Dourish, P., Bly, S.: Portholes: Supporting awareness in a Distributed Work Group. In: Proc. Conference on Human Factors in Computing Systems (CHI 1992), pp. 541–548 (1992)
5. Fish, R., Kraut, R., Chalfonte, B.: The Video Window System in Informal Communications. In: Proc. Computer Supported Cooperative Work (CSCW 1990), pp. 1–12 (1990)
6. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer applications and High Performance Computing 15(3), 200–222 (2001)
7. Foundation for Intelligent Physical Agents, FIPA Abstract Architecture Specification, SC00001, Geneva, Switzerland (2002), http://www.fipa.org/specs/fipa00001/index.html

8. Greenhalgh, C.: Large Scale Collaborative Virtual Environments, Doctoral Thesis, University of Nottingham (1997)
9. Gutwin, C., Stark, G., Greenberg, S.: Support for Workspace Awareness in Educational Groupware. In: Proc. Computer Supported Collaborative Learning (CSCL 1995), pp. 147–156 (1995)
10. Herrero, P., Bosque, J.L., Pérez, M.S.: An Agents-Based Cooperative Awareness Model to Cover Load Balancing Delivery in Grid Environments. In: Parashar, M. (ed.) GRID 2002. LNCS, vol. 2536, Springer, Heidelberg (2002); Meersman, R., Tari, Z., Herrero, P. (eds.): OTM-WS 2007, Part I. LNCS, vol. 4805. Springer, Heidelberg (2007)
11. Herrero, P., Bosque, J.L., Pérez, M.S.: Managing dynamic virtual organizations to get effective cooperation in collaborative grid environments. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part II. LNCS, vol. 4804, pp. 1435–1452. Springer, Heidelberg (2007)
12. Kuwana, E., Horikawa, K.: Coordination Process Model - based Design for Synchronous Group Task Support System, Technical Reports of Information Processing Society of Japan, Groupware, No. 13, 1–6 (1995)
13. Malone, T.W., Crowston, K.: The interdisciplinary study of coordination. ACM Computing Surveys 26(1), 87–119 (1994)
14. Matsushita, Y., Okada, K. (eds.): Collaboration and Communication. Distributed collaborative media series, vol. 3. Kyoritsu Press (1995)
15. Matsuura, N., Hidaka, T., Okada, K., Matsushita, Y.: VENUS: An informal Communication Environment supporting Interest Awareness. Trans. Information Processing Society of Japan 36(6), 1332–1341 (1995)
16. Ogata, H., Yano, Y.: Knowledge Awareness: Bridging Learners in a Collaborative Learning Environment. International Journal of Educational Telecommunications 4(2), 219–236 (1998)
17. Paletta, M., Herrero, P.: Learning Cooperation in Collaborative Grid Environments to Improve Cover Load Balancing Delivery. In: Proc. IEEE/WIC/ACM Joint Conferences on Web Intelligence and Intelligent Agent Technology, vol. E3496, pp. 399–402. IEEE Computer Society, Los Alamitos (2008)
18. Paletta, M., Herrero, P.: Towards Fraud Detection Support using Grid Technology. Accepted for publication in a Special Issue at Multiagent and Grid Systems - An International Journal (to be published) (2008)

Game Theory and Cognitive Radio Based Wireless Networks

Saurabh Mehta and K.S. Kwak

Wireless Communications Research Center, Inha University, Korea
suryanand.m@gmail.com

Abstract. The ability to model individual, independent decision makers whose actions potentially affect all other decision makers renders game theory particularly attractive to apply to various fields of Information technology, especially, to analyze the performance of wireless networks. In this paper, we discuss how various interactions in cognitive radio based wireless networks can be modeled as a game at different levels of protocol stack. This allows the analysis of existing protocols and resource management schemes, as well as the design of equilibrium-inducing mechanisms that provide incentives for individual users to behave in socially-constructive ways. In nutshell, this paper serves two main objectives; first, to model some of the fundamental questions on cognitive radio based wireless networks as interactive games between the nodes and second, to gain our understanding on inter-discipline research issues.

Keywords: Cognitive Radio, Wireless Networks, Game Theory, Cross-layer Design, Mechanism Design.

1 Introduction

The modern information society will continue to emerge, and demand for wireless communication services will grow. Future generation wireless networks are considered necessary for the support of emerging services with their increasing requirements. Significant amounts of radio spectrum will be needed at dynamically changing times and locations. However, this need for radio spectrum is a problem, that is, the spectrum scarcity. The electromagnetic radio spectrum is a precious natural resource. Despite the recent advancements in communication technology such as Multiple Input Multiple Output (MIMO) antennas, third generation cellular networks and their integration with wireless Local Area Network (LAN) IEEE 802.11, it is difficult to foresee how a truly connected information society can be established, given today's regulation of radio spectrum.

Cognitive radio (CR) is a revolutionary technology that aims for remarkable improvements in efficiency of spectrum usage. Basically, cognitive radios are radio systems that continuously perform spectrum sensing, dynamically identify unused ("white") spectrum, and then operate in this spectrum at times when it is not used by incumbent radio systems. It will change the way the radio spectrum is regulated, but also requires new enabling techniques such as improved spectrum sensing, dynamic spectrum assignment, and cross layer design[1].

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 803–812, 2009.

Table 1. Cognitive radio based wireless networking game

Components of a game	Elements of a wireless network
Players	Nodes in the wireless network
A set of actions	A modulation scheme, Coding rate, transmit power level, etc.
A set of preferences	Performance metrics (e.g. Throughput, Delay, SNR, etc.)

We want to propose a concept of cognitive radio based network (CRN) that can be intelligent about spectrum usage and network performance and thus have great flexibility and security. In order to develop the CRN, we shall use some analyzing tools that can control the CRN and optimize the performance of the whole protocol stack. We believe that techniques originated from game theory can be used to create a toolset for the analysis of CRNs.

Game theory provides a mathematical basis for the analysis of interactive decision-making process between the rational players. It provides tools for predicting what might happen when nodes (players) with conflicting interests interact. Game theory is a collection of modeling tools that aid in the understanding of interactive decision problems. In the past sevral years, the application of game theory to problems in communication and networking has become popular and productive.Specifiaclly, game theoratic models have been developed to better understand congestion control, routing, power control, topology control, trust mangement, and other issues in wired and wireless communication systems.

A game is made up of three basic components: a set of players, a set of actions, and a set of preferences. The palyers are the decision makers in the modeled scenario. The actions are the alternatives available to each palyer. In a wirless system, actions may include the choice of a modulation scheme, coding rate, protocol, flow control parameter, transmit power level, or any other factor that is under the control of the node. When each player chooses an action, the resulting "action profile" determines the outcome of the game. Finally, a preference relationship for each palyer represents that player's evaluation of all possible outcomes. Table 1. shows typical componets of a coginitive radio based wireless networking game.

Appropriately modeling these prefernces realtionships is one of the most challenging aspects of the application of game theory. A clear distinction should be drawn, however, between a game, which must involve multiple decision makers, and an optimization problem, which involves only a single decision maker. A model is usually appropriate only in scenarios where we can resonably expect the decision of each node to impact the outcomes relevant to other nodes. For example, a single car navigating a road way in an attempt to reach a destination as quickly as possible performs an optimization. When the roadway is shared, though, the drivers are engaged in a game in which each attempts to reach his/her destination as quickly as possible without getting in an accident or receiving a traffic ticket.

A CRN is a self-configuring, multihop network in which there is no central authority. Thus, every aspect of the configuration and operation of a CRN must be completely distributed. Furthermore, nodes in a CRN often severely energy and power constrained. Game theory, as we have discussed, offers a suite of tools that may be used effectively in modeling the interaction among independent nodes in a CRN.

2 Necessity

In the CRN, it is essential for CR devices to perform spectrum sensing / sharing tasks in order to co-exist with other types of networks. However, this device-level design would inevitably have high complexities and costs. Also, only CR devices themselves could not deal with the distributed spectrum allocation effectively, therefore additional aids from other aspects should be necessary. So, a new spectrum-sharing strategy with the aid from network level is to be designed and developed. This strategy can solve the coexistence problem of CRNs and simplify the design of CR devices. In fact, it is a cross-layer design strategy that integrates useful information obtained from the physical, MAC and network layers, in order to fully exploit available spectrum resources without introducing interference to existing users.

In the CRN, each node running a distributed protocol must make its own decisions, possibly relaying on information from other nodes. Theses decisions may be constarined by the rules or algorithms of a protocol, but ultimately each node will have some freeway in setting parameters or changing the mode of operation. These nodes, then, are indpendent agents, making decisions about spectrum usage, interference level, radio parameters, node position, network status, transmit power, packet forwarding, backoff time, and so on. Now, important question here is "what does the node seek to optimize?" while making these decisions. In some cases, nodes may seek the "greater good" of the network as a whole. In other cases, nodes may behave selfishly, looking out only their own user's interests. In a final case, nodes may behave maliciously, seeking to ruin network performance for other users. In second and third cases, the application of game theory may be straightforward, as game theory traditionally analyzes situations in which player objectives are in conflict. In the first case, node objectives may be aligned with the network main objective, but still game theory may offer useful insights. Even when nodes have shared objectives, they will each have a unique perspective on the current network state, and leading to possible conflicts regarding the best course of action. Game theory allows us to model CRNs in which there is no centralized entity with full information of network conditions.

3 CRN Challenges

The main problem of a CRN is the coexistence with the Licensed Incumbent Users (LIU), which are the owners of licensed spectrum and have higher priorities. A practical CR device, or in other words, Customer Premise Equipment (CPE), shall not interfere with the existing LIUs. The definitions and requirements are listed below[2].

Licensed Incumbents

- TV Broadcasters: NTSC (US/Canada/Japan analog TV), ATSC (US, Canada DTV), PAL (Europe analog TV), SECAM (France analog TV), DVB-T (Europe/worldwide DTV).
- Land Mobile Radio Networks:
- Public Safety (police, fire, etc.), Commercial (cabs, towing services, etc.)
- Wireless Microphones and other TV broadcast related devices (in the US, FCC rules, Part 74 Low Power Auxiliary Station devices).

Customer Premise Equipments

- Avoid operating on Land Mobile Radio channels.
- Use planning, GPS, databases for Base Station planning, licensing.
- Use Base Station authorization/control and distributed sensing, in BS and CPE's, to avoid unintentional operation on TV channels.
- Use sensing to mitigate interference to Part 74 devices.

It is a challenge to construct an efficient spectrum allocation scheme, which can dynamically manage radio resources in a real-time pattern according the environment. The channel allocation scheme should at least satisfy two basic requirements: 1) it can exploit as many as possible available frequency bands; 2) it must NOT produce more interference to LIUs than before. However, there are some tradeoffs between these two requirements.

Beyond the traditional spectrum sensing method, there would be chances for CPEs to exploit more frequency bands, if more information concerned with spectrum usage of LIU, interference level, positions etc. can be obtained. It is also implied that in the CR network asymmetric links among CPEs could be set up, depending on their specific surroundings. Therefore, the channel allocation scheme should take into account various factors of different layers in order to satisfy the requirements of CPEs. As shown in the figure 1 game theory can be applied to the modeling of a CRN at the physical layer, link layer, and network layer. Applications at the transport layer and above exist also, but we restrict our research up to network layer. A research issue of interest in all those cases is that of how to provide the appropriate incentives to discourage selfish behavior. Selfishness is generally dangerous to overall network performance; examples include a node's increasing its power without regard for interference it may cause on its neighbors (layer 1), a node's immediately retransmitting a frame in case of collisions without going through a backoff phase (layer 2), or a node's refusing to forward packets for its neighbors (layer 3).

3.1 Physical Layer (PHY)

In the PHY case, it needs to offer high performance while keeping the complexity low. In addition, it needs to exploit the available frequency in an efficient manner to provide adequate performance, coverage and data rate requirements of the service. The PHY has also to provide high flexibility in terms of modulation and coding.

The physical layer can perform basic frequency detection and spectrum sensing tasks, in order to find available empty frequency bands. In our considerations, not only

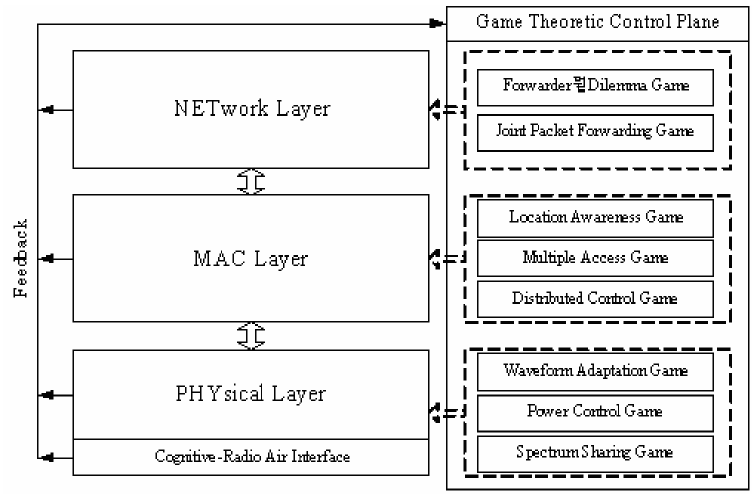


Fig. 1. The classification of the games according to protocol layers

can the empty bands be utilized, but also the on-the-air frequency bands can be exploited under some conditions. This will be a great improvement in the PHY design.

Power control, though closely associated with cellular networks, could be implemented in CRNs due to the potentially significant performance gains achieved when nodes limit their power level. In wireless communication systems, mobile terminals respond to the time varying nature of the channel (short term and long term fading) by regulating their transmitter powers. Specifically, in a CDMA system, where signals of other terminals can be modeled as interfering noise signals, the major goal of this regulation is to achieve a certain signal to interference (SIR) ratio regardless of the channel conditions while minimizing the interference due to terminal transmit power level. Hence, there are two major reasons for a terminal to perform power control; the first one is the limit on the energy available to the mobile node, and the second reason is the increase in quality of service (QoS) by minimizing the interference. In the CRN, a kind of distributed power control can be performed, that means, not only can the node adjust its power according to its own status, but also the distributed networking function can determine the proper power limit to optimize the performance of the whole network.

Finally, Waveform adaptation in CRNs involves the selection of a waveform by a node such that the interference at its receiver is reduced. The interference at the receiver is a function of the correlation of a user's waveform with the waveforms of the other users in the network. Also, in general, the individual nodes involved in transmission have no or very little information about the receiver's interference environment. Hence to minimize the adaptation overhead, distributed waveform adaptation algorithms that require a minimal amount of feedback between receivers and transmitters need to be developed for these networks. Game theory can provide useful insights to this scenario.

3.2 MAC Layer

Beacon Enhancement: If the CPE has the knowledge of which channel the neighboring LIU (typically, TV set) is currently being tuned on, it could have chance to use the bandwidth of the rest on-the-air TV channels at the same time. The local beacon is a good method of identifying the LIU, that is, the LIU itself or a special beacon transmitter periodically broadcasts signals in order to actively announce the existence of the LIU to the surroundings. It becomes easier for the CPE to find neighboring LIUs and act correspondingly. For example, the CPE can estimate the distance from the LIU according to the received beacon signal strength. In addition, if we add more information elements (IE) in the beacon frame, the CPE will benefit from them. For example, in the beacon frame, the position of LIU, the beacon transmit power, the total received interference level can be included. After decoding these IEs, the CPE can calculate the path loss, estimate its possible interference level to the LIU and then decide which potential bands can be available. All beacon transmitters should use the same frequency band to broadcast beacon frames. This common beacon band could be an unlicensed frequency band, or a dedicated channel that is not occupied by LIUs, or just a spare gap between TV channels.

Positioning: The location information of neighboring nodes is important for CPEs to estimate interference to and from the existing LIU devices. Also, the location information can help the CPE decide if a link can be set up with the other CPE. The Global Position System is a mature positioning method suitable for CR networks, since the distance between nodes would be relative large. If the CPE uses some specific radio techniques, such as impulse-radio ultra wideband, the CPE can measure distances and positions by itself.

The position of the LIU could be obtained by reading the beacon frames. Then the location information could be exchanged among CPEs with an ad hoc manner. In the central-controlled structure, the base station of the CR network could broadcast the position information to all CPEs. In this case, the feedback mechanism from the CPE to the base station should be defined.

Interference Estimation: The purpose of interference estimation is to tell if the potential channels of a CPE can be really used or not. At first, based on the position information, the path loss model, its maximum transmit power etc., the CPE could estimate its interference to the LIUs nearby if its potential channels were deployed. For example, if the estimated interference level does not exceed a predefined threshold, the corresponding potential channel for transmission can be regarded as "available".

The interference estimation is also necessary for a receiving CPE that is shadowed by the TV transmitter. Besides some possible interference mitigation technique, the shadowed CPE can estimate its signal-to-interference-and-noise ratio (SINR), according to the total interference level from TV signals, the distance to the transmitting CPE and the transmit power of CPE etc. If the estimated SINR exceeds a predefined threshold, the corresponding potential channel for reception can be regarded as "available".

Medium Access Games-The Slotted Aloha and DCF Games: The medium access control problem, with many users contending for access to a shared communications medium, lends itself naturally to a game theoretic formulation. In these medium access control games, selfish users seek to maximize their utility by obtaining an unfair share of access to the channel. This action, though, decreases the ability of other users to access the channel.

3.3 Network Layer

Update and Exchange Band Information: Each CPE in a CRN shall be continuously in charge of sensing and measuring its surroundings, determining its available transmission and reception channels, and sending relevant information to other CPEs. If there are some events occurred, for example, a neighboring LIU is powered on or off, a LIU is tuning its working channels, or the position of a LIU is changed, the CPE should be able to detect these changes quickly, adjust its available channels, update its frequency usage database and let other CPEs know about it.

In a CRN, all CPEs shall have the consistent consensus about the common set of empty channels (not occupied by TV transmitters or other transceivers), which should be a minimal intersection of all spectrum sensing results from CPEs. This common set of empty channels can be adopted to realize some network control functions and provide seamless data delivery throughout the network. On the other hand, the potential channels of a CPE can in addition be utilized to increase the frequency usage factor and improve the data throughput locally.

Channel Allocation Negotiation: Because CPEs at different locations would have different perspectives of their own "available" channels, a channel allocation negotiation procedure should be needed before a data link can be set up between two adjacent CPEs. Such a negotiation procedure can be categorized as one of functions in the control plane of a CRN. A dedicated control channel can be set up by utilizing a common frequency band that is not occupied by any TV channel throughout a CRN. The control packets can be exchanged in an ad hoc manner, or a base station controls the channel allocation among CPEs while CPEs should send some feedback about their "available" channels to the base station. For example, the channel allocation negotiation procedure for unidirectional transmission could be performed between the communicating pair as follows: If one CPE has data to transmit, it should at first send the "Allocation Request" control packet that contains a list of its available transmission channels to the intended receiver. After having received this "Allocation Request", the receiver should check its own available reception channels to decide which channels can be allowed for receiving data. Then the receiver sends back "Allocation Confirm" that indicates allowed transmission channels to the transmitter. If this negotiation is completed successfully, both transmitting CPE and receiving CPE will tune on the same frequency bands and start data transmission. Similarly, the bi-directional link can be set up by exchanging available transmission channel sets and reception channel sets between two communicating CPEs.

Routing Method: The location information can facilitate the routing selection in the CR network. Since, CPEs at different places would have different available channels and thus different throughput; a routing method can be developed to find a shortest path, or a path with maximum throughput from a source CPE to a destination. If energy-related factors, such as the path loss model, the transmit power, the interference level, the transmission distance, the circuit power consumption and so on, are considered, a kind of energy-efficient routing method can be derived especially for a CRN. After all, the cross-layer design strategy can gather information from almost all aspects, so we can reach optimal solutions for various requirements.

Network Layer Games: Issues such as the presence of selfish nodes in a network, convergence of different routing techniques as the network changes, and the effects of different node behavior on routing, can be analyzed using game theory. Routing is modeled as a zero sum game between two players – the set of routers and the network itself. In a zero-sum game the utility function of one player (minimizing player) is the negative of the other's (maximizing player). The game has an equilibrium when the minmax value of any player's payoff is equal to its maxmin value. In a zero sum game, the maxmin value is defined as the maximum value that the maximizing player can get under the assumption that the minimizing player's objective is to minimize the payoff to the maximizing player. In other words, the maxmin value represents the maximum among the lowest possible payoffs that the maximizing player can get; this is also called the safe or secure payoff.

In the routing game the payoff to each player consists of two cost components, one being the amount of network overhead and the other varying with the performance metric under consideration. For example, for evaluating soundness the cost to the routers is 0 if all routers have a correct view of the topology when the game ends and 1 if any one router does not. The objective of the routers is to minimize the cost function. The action for the routers involved is to send routing control messages as dictated by the routing technique and update their routing information, and for the network to change the state of existing links from up to down and vice versa.

Table 2. Current related works

Subject	The Proposed work/solution	References
Ad-hoc Networks	- cooperation without incentives - incentives for cooperation: currency - incentives for cooperation: reputation system	[5,7,15] [3,6] [13,17,18]
Hybrid ad hoc Networks	-incentives for cooperation	[4,16]
Sensor Networks	-cooperative packet forwarding	[11]
Cognitive radio	- multi-radio channel allocation -IEEE 802.22 Working Group	[21] [23]
Cellular and Wi-Fi Networks (WWANs and WLANs)	-spectrum sharing in cellular Networks - selfish behavior in CSMA/CA -reputation-based Wi-Fi development	[19,22] [9,12] [8,10]
General Networks	- reputation in social Networks	[14]

The game is solved to determine the minmax value of the cost function. It serves to compare the different routing techniques in terms of the amount of routing control traffic required to achieve convergence and the soundness of the routing protocol to network changes. Another issue related to routing involves studying the effect of selfish nodes on the forwarding operation. We summarize the current related works/trends to above mentioned challenges as shown in the table 2.

As describe in the above mentioned games selfish behavior by nodes in a CRN may lead to a suboptimal equilibrium where nodes, through their actions, reach an undesirable steady state from a network point of view. Hence, incentive mechanisms are needed to steer nodes towards constructive behavior (i.e., towards a desirable equilibrium). Even though the bulk of work done in the past few years to answer above mentioned games still they are at a nascent stage.

4 Conclusions

Game theory allows us to model wireless networks in which there is no centralized entity with full information of network conditions. Game theoretic models, like other mathematical models, can abstract away some important assumptions and un-mask critical unanswered questions. We hope this paper will motivate students and researchers to peep at this fascinating analytical tool, and encourage them in modeling problems of a CRN.

References

1. Akyildiz, I.F., et al.: NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. Computer Networks 50, 2127–2159 (2006)
2. Mitran, P.: Cognitive Radio: A Survey. University of Waterloo (Febuary 2008)
3. Buttyan, L., Hubaux, J.-P.: Enforcing Service Availability in Mobile Ad-Hoc WANsL. In: Proceedings of MobiHoc 2000, Boston, MA, USA (August 2000)
4. Ben Salem, N., Buttyan, L., Hubaux, J.-P., Jakobsson, M.: A Charging and Rewarding Scheme for Packet Forwarding. In: Proceedings of MobiHoc 2003, Annapolis, Maryland, USA (June 2003)
5. Felegyhazi, M., Buttyan, L., Hubaux, J.-P.: Equilibrium Analysis of Packet Forwarding Strategies in Wireless Ad Hoc Networks - the Static Case. In: Conti, M., Giordano, S., Gregori, E., Olariu, S. (eds.) PWC 2003. LNCS, vol. 2775, pp. 776–789. Springer, Heidelberg (2003)
6. Buttyan, L., Hubaux, J.-P.: Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. In: ACM/Kluwer Mobile Networks and Applications, vol. 8(5) (2003)
7. Felegyhazi, M., Hubaux, J.-P., Buttyan, L.: The Effect of Mobility on Cooperation in Ad Hoc Networks. In: Proceedings of WiOpt 2004, Cambridge, UK (March 2004)
8. Mundinger, J., Le Boudec, J.-Y.: Analysis of a Robust Reputation System for Self-Organised Networks. European Transactions on Telecommunications, Special Issue on Self-Organisation in Mobile Networking 16(5), 375–384 (2005)
9. Raya, M., Hubaux, J.P., Aad, I.: DOMINO: A System to Detect Greedy Behavior in IEEE 802.11Hotspots. In: Proceedings of MobiSys 2004 (June 2004)

10. Ben Salem, N., Hubaux, J.-P., Jakobsson, M.: Reputation-based Wi-Fi Deployment - Protocols and Security Analysis. In: Proceedings of WMASH 2004, Philadelphia, USA (October 2004)
11. Felegyhazi, M., Buttyan, L., Hubaux, J.-P.: Cooperative Packet Forwarding in Multi-Domain Sensor Networks. In: Proceedings of PerSeNS 2005, in conjunction with PERCOM 2005, Hawaii, USA, March 12 (2005)
12. Cagalj, M., Ganeriwal, S., Aad, I., Hubaux, J.-P.: On selfish behavior in CSMA/CA networks. In: Proceedings of IEEE Infocom 2005, Miami - FL, USA, March 13-17 (2005)
13. Mundinger, J., Le Boudec, J.-Y.: Analysis of a Reputation System for Mobile Ad-Hoc Networks with Liars. In: Proceedings of The 3rd International Symposium on Modeling and Optimization, Trento, Italy (April 2005)
14. Mundinger, J., Le Boudec, J.-Y.: The Impact of Liars on Reputation in Social Networks. In: Mundinger, J. (ed.) Proceedings of Social Network Analysis: Advances and Empirical Applications Forum, Oxford, UK (July 2005)
15. Felegyhazi, M., Hubaux, J.-P., Buttyan, L.: Nash Equilibria of Packet Forwarding Strategies in Wireless Ad Hoc Networks. IEEE Transactions on Mobile Computing (TMC) 5(5) (May 2006)
16. Ben Salem, N., Buttyan, L., Hubaux, J.-P., Jakobsson, M.: Node Cooperation in Hybrid Ad Hoc Networks. IEEE Transactions on Mobile Computing (TMC) 5(4) (April 2006)
17. Buchegger, S., Le Boudec, J.-Y.: Self-Policing Mobile Ad-Hoc Networks by Reputation. IEEE Communication Magazine 43(7), 101 (2005)
18. Mundinger, J., Le Boudec, J.-Y.: Analysis of a Robust Reputation System for Self-Organized Networks. European Transactions on Telecommunications, Special Issue on Self-Organisation in Mobile Networking 16(5), 375–384 (2005)
19. Felegyhazi, M., Hubaux, J.-P.: Wireless Operators in a Shared Spectrum. In: Proceedings of Infocom 2006, Barcelona, Spain, April 23-29 (2006)
20. Felegyhazi, M., Hubaux, J.-P.: Game Theory in Wireless Networks: A Tutorial. In: EPFL technical report, LCA-REPORT-2006-002 (February 2006)
21. Felegyhazi, M., Cagalj, M., Saeedi Bidokhti, S., Hubaux, J.-P.: Non-cooperative Multi-radio Channel Allocation in Wireless Networks. In: Proceedings of Infocom 2007, Anchorage, USA, May 6-12 (2007)
22. Felegyhazi, M., Cagalj, M., Dufour, D., Hubaux, J.-P.: Border Games in Cellular Networks. In: Proceedings of Infocom 2007, Anchorage, USA, May 6-12 (2007)
23. Federal Communications Commission.: Notice for Proposed Rulemaking (NPRM 03 322), Facilitating Opportunities for Flexible, Efficient, and Reliable Spectrum Use Employing Cognitive Radio Technologies

Scalable Grouping Based on Neuro-Fuzzy Clustering for P2P Networks*

Romeo Mark A. Mateo, Hyunho Yang, and Jaewan Lee

School of Electronic and Information Engineering, Kunsan National University
68 Miryong-dong, Kunsan, Chonbuk 573-701, South Korea
{rmmateo, hyyang, jwlee}@kunsan.ac.kr

Abstract. In this paper, we present a scalable grouping based on the proposed neuro-fuzzy clustering (NFC) which is a combination of fuzzy clustering and neuro-fuzzy classification for P2P networks. Fuzzy clustering is used to construct the classifier structure for the grouping technique and neuro-fuzzy algorithm is used to adjust the fuzzy system in creating the accurate classifier. After the adjustment, each peer is processed in multi-ring algorithm using the fuzzy membership function of NFC. The scalable peer grouping contributes on load distribution by forwarding incoming request to the least loaded peer. Performance result showed that the proposed scalable peer grouping classifies more peers in a group for availability of resources and accurately classifies peer group.

1 Introduction

Peer-to-peer (P2P) systems have become popularly used in variety of applications that shares resources because of its efficiency in providing the availability of data by means of replication and ease of resource sharing within the computer networks. Previous technology of P2P is based on centralized architecture like Napster [1] where the system consists of file directory servers that store information of files associated with peer location which shares the resources. This method increases traffic flow connecting to the server caused by peer queries. Another method is the decentralized P2P approach which is designed to distribute the queries to selected peers to overcome the problem of network congestion from centralized technique. Decentralized scheme has two types which are the structured and unstructured P2P. The unstructured P2P does not have specific rules to control the routing of peers like in Gnutella [2] where peers are formed by joining nodes with loose rules. Structured P2P systems like Pastry [3] and Chord [4] uses ring topology to provide scalable P2P system, however, the major shortcoming of the ring topologies are the high dependency on large number of nodes which produces high latency because latency is proportional to the number of nodes. Searching resource in a ring topology is guaranteed in O (log N) where N is the number of peers. Ring topology is more scalable and mostly used by

* This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 813–822, 2009.

P2P designers and so the short coming was solved by many researches or even improved the technique. HRing [5] improves the ring topology where the method constructs routing tables based on the distance between node positions instead of node IDs in order to eliminate the effect of node ID distribution on the long link distribution and efficiently load balance the system. Other than ring schemes there are also hierarchy and distributed trees [6] used for efficient searching of resources but more vulnerable to failures where most of the connection relies on its root nodes. Middleware for P2P was also introduced which provides a transparent operations of services and failure handling in P2P environments. Reference [7] proposes a decentralized event-based object middleware built on top of a P2P system. Distributed interception, high performance synchronous or asynchronous, one-to-one or one-to-many notifications and decentralized object location services are provided by DERMI. Most of these researches provide an efficient resource location and self-organizing of peers.

In this paper, we present a scalable peer grouping for multi-ring P2P networks. The grouping scheme is based on the proposed neuro-fuzzy clustering where the fuzzy membership function is used to have partial memberships to other groups in providing scalable peer classification. The fuzzy system is trained by neuro-fuzzy algorithm for accurate resource search. Dynamic join method processes inner nodes and stores the information of sub nodes using the classification method of fuzzy system. In searching resources, peer queries to inner nodes which perform the P2P multi-ring topology used in resource finding. Moreover, the proposed scalable grouping provides more peers to select the least loaded peer for load balanced P2P system.

2 Interest and Group-Based Peer-to-Peer Systems

Most practices in applications of the Web and P2P networks use the interest of client as an input in processing the appropriate network connections and efficient sharing resources. In the interest-based P2P structure, a peer has higher possibility to answer users' query if that peer has similar interest with the user other than other peers. The technique is to provide main links or grouped the peers that have similar interests. In the previous researches [8, 9], peers depends on both the types of shared files and the interest profile which is explicitly configured by users. An unstructured P2P interest-based grouping is proposed [10] where not using defined methods to process the queries on P2P network based on interest of peers is the challenge. Group-based P2P network model [11] also based on interest of peers to construct the peer groups. These researches mainly use the concept where each peer attempts to select the best neighbors by measuring other peers' interests.

Multi-ring topology is proposed [12] to provide high performance group communication in large-scale P2P networks. The multi-ring topology is based on ring topology like in Pastry where each ring is connected to two neighbor's nodes, left and right, in order to form a ring and the system selects inner nodes are selected to provide a short-cuts to all ring sections. When a node wants to broadcast data to other nodes, it sends both directions to its neighbor. The throughput is no longer limited by less powerful nodes, as powerful nodes communicate directly. The formation of inner ring is also important and a system parameter q defines the ratio of inner ring nodes to outer rings. This decentralized structured P2P topology balances the advantage of a ring and a centralized topology and is used by our research which extends the multi-ring to scalable multi-ring by using the neuro-fuzzy clustering.

3 Scalable Grouping for Multi-ring Topology Based on Neuro-Fuzzy Clustering

This study implements the proposed scalable peer grouping to the multi-ring topology for efficient resource searching and distribution of loads in structured P2P networks. A dynamic peer grouping is used in our proposed method where nodes initially provide data profile to the coordinator node as inputs to process the scalable peer groupings. Unlike the classical publish and subscribe method, the proposed procedure automatically classifies and forms the peer groups. The number of groups depends on k which is analyzed by performing the objective function with different k value and chooses the smallest J with index k. The interest data profile (P_x) which is sent to the coordinator node is based on the resources it shares. An example, if a peer shares numerous music files then its music data profile will have a high value. This paper presents a knowledge-based scalable grouping method for P2P which connects the most relevant peers and the operation is managed by coordinator node to re-organize and group the peers based on the scalable peer groupings.

Figure 1 shows the illustration overlapping of the peer groups and is the result of the scalable peer grouping. Some peers in the group still can be classified to other groups. Peer with index 5 is classified in group A but because of the fuzzy overlaying, it is also classified in group B and C. In this study, a set of profiles $P_x = \{p_1, p_2, \dots, p_n\}$ where each p_n with numerical value is used as measure of a node interest. A peer hosts several files that are used for resource sharing and the peers are grouped based on knowledge based clustering. The values from P_x are sent to the coordinator node to assign the group of a peer. The following subsection explains the procedure of the proposed neuro-fuzzy clustering and selects the best peer node to process or download files.

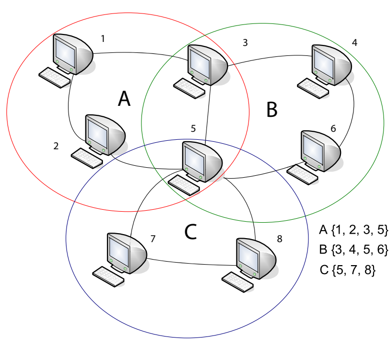


Fig. 1. Scalable grouping of P2P system based on neuro-fuzzy clustering where a partial membership is used to overlap the grouping of a node

3.1 Neuro-Fuzzy Clustering

Each P_x from all peers is collected and transformed into vector values p to be processed to neuro-fuzzy clustering. The procedure starts on partitioning the collection of P data points specified by m-dimensional vectors p (p = 1, 2… P), into c fuzzy clusters, and finds a cluster center in each cluster, minimizing an objective function. The initialization of centers is critical to have the minimum iteration from the objective function. In this study the fuzzy sets are initialized by using the eigen decomposition method. After calculating the values of p in eigen decomposition, p is sorted from lowest to highest value based on *eM*. The values are arranged to a one dimensional array and prepare for calculating the fuzzy system for the fuzzy clustering. *d(A,B)* is the function to determine the distance from set A to B and calculated by subtracting p_{max}, which is the maximum value of p and $p_{\min}$ is the minimum value of p divided by the number of clusters c. The overlapping of the fuzzy set is determined by o and added to the length of the set shown in Equation 1.

$$fuzzylength = d(A,B) = \frac{p_{\max} - p_{\min}}{c} + o \tag{1}$$

The length of the fuzzy set is used to calculate the initial value of the minimum ($Cmin_i$), maximum ($Cmax_i$) and center ($Ccen_i$) value of each fuzzy set. Initial values are presented by the following: $Cmin_i = p_{\min}$, $Cmax_i = p_{\min} + fuzzylength$, and $Ccen_i = p_{\min} + (fuzzylength\,/\,2)$. In next iterations, the values of each fuzzy set are the addition of previous value j ($j=i-1$) and $fuzzylength\,/\,2$. Also, every center of the group is initially set to $c_i = Ccen_i$. In fuzzy partitioning, the membership matrix M is allowed to have elements in the range of 0 to 1. A point's total membership of all clusters must always be equal to unity to maintain the properties of the M matrix. The fuzzy system is used other than the classical fuzzy membership function because the extracted fuzzy rule and fuzzy sets are trained and will be used for classifying peer groups. Every time an update occurs in c_i, the center of the fuzzy sets in the fuzzy system also adjusts where, $Ccen_i = c_i$. These procedures are used to determine the center point of the cluster and center from the fuzzy system. The algorithm for learning the fuzzy rule is based on NEFCLASS [13]. Rules from the training are determined by computing performance values for each rule. If a rule correctly classifies a pattern, the degree of fulfillment value is added to its performance value, if not, the degree of fulfillment is subtracted. The best rules from the procedure are stored and used for fuzzy set learning and classification. After generating the best rules, fuzzy set learning is processed based on rules extracted. A maximum iteration of building the classification model is used for the proposed algorithm indicated by *max*. The learning algorithm on adapting the fuzzy set runs cyclically until *max* is achieved. The calculation of the delta value is given by subtracting the target value to the value from the activation function in Equation 2. The value from Equation 3 is used for calculating the error in Equation 4 where e_R is the error factor for calculating the new fuzzy sets. The fuzzy sets are adjusted using the parameters from Equation 4.

$$\delta_c = t_i - activation(c_i) \tag{2}$$

$$e_R = o_R(1-o_R)\sum_{c\in U_3} W(R,y)\delta_c \tag{3}$$

$$\begin{aligned}
\delta_b &= \sigma \cdot e_R \cdot (C\max_i - C\max_i)\cdot \mathrm{sgn}(p_i - Ccen_\mu);\\
\delta_a &= -\sigma \cdot e_R \cdot (C\max_i - C\max_i) + \delta_b;\\
\delta_c &= \sigma \cdot e_R \cdot (C\max_i - C\max_i) + \delta_b;
\end{aligned} \tag{4}$$

$$\gamma = \frac{1}{C}\sum_{n=1}^{N}\frac{1}{N}\sum_{i=1}^{p} f(x_i) \tag{5}$$

The adaptation of the fuzzy sets is carried out by changing the parameters of its membership function in a way it increase or decrease respectively. Then apply the changes to *W(x',R)* if this does not violate against a given constraint. All the procedures in Equation 1 to 4 are executed by the coordinator node whenever the scalable group mechanism is initialized or after determining a certain threshold of classification error in classifying a node. Every time a new node joins the group, a calculation of classification error is done in Equation 5. This procedure is done in a separate module of the coordinator node to avoid execution delays.

3.2 Peer Classification in Multi-ring Topology

The peer classification is processed after training the fuzzy system. A threshold value controls the membership of a peer presented by Φ where Φ < 1. The configuration of Φ depends on the application domain of the system where the need of more peers to classify in a group varies. Equation 6 shows the computation of classifying peers in a group. Because of the fuzzy membership, a peer can be classified to other group and adds more peers compared to a classical knowledge based method which also has the same goal of grouping the related peers but has a crisp separation method.

$$peerGroup_i(x_k) = \begin{cases} 1 & if\ m_{ik}(u_i) > \Phi, \\ 0 & otherwise. \end{cases} \tag{6}$$

The construction of the scalable peer grouping starts from peer sending its P_x to the coordinator node by *send_reg*(P_x) interface. The coordinator node classifies each peer through the fuzzy system processed in Equation 6. The assigned inner node in a group becomes a leader and obtains its sub nodes. In determining the inner nodes, there is a possibility that peer can have more than one leadership or it could be an inner node to other group which will provide bottleneck to the network by overloading of queries. In our method, the inner node that is already assigned is skipped to the procedure of determining the inner node to avoid network congestion in inner nodes. After determining the inner nodes, the sub nodes are assigned based on the ratio of the inner nodes (*q*) with the outer nodes. All inner nodes have a routing list of each sub nodes as well as all sub nodes have routing list to its inner node or leader. The coordinator node classifies peers using interest properties by the proposed algorithm. After determining the class, it automatically joins the peer. The calculation of classification error is executing at this point and if the value of error reaches the assigned error threshold then it process again the profiles of each node to recalculate the groupings in the

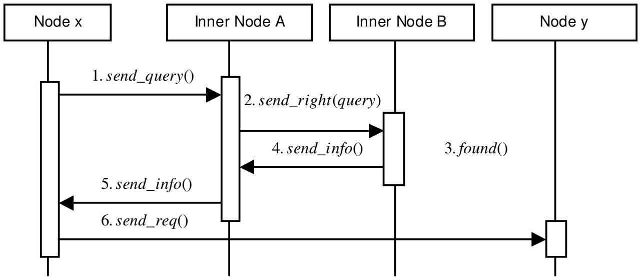


Fig. 2. Search procedure using the inter-inner nodes based on ring topology

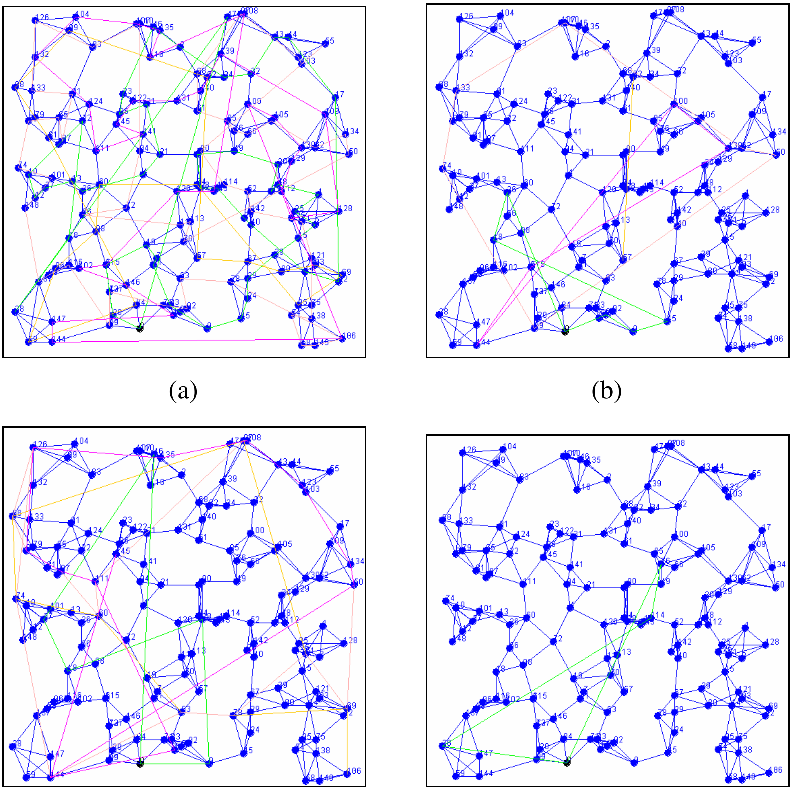
neuro-fuzzy clustering. This procedure is transparently executing within the grouped nodes. Figure 2 presents the searching method within the inner nodes. This assumes that the groups are already constructed. First, a peer *x* request a file where peer *y* obtains it but they were assigned to different inner nodes. The peer executes *send_query()* to the inner node and then the inner node send the query to the right inner node. After determining where the file is, it returns the address of the peer.

Because of the distributed characteristic of the P2P systems specially the unstructured P2P, it is more complex to implement the load balancing. Without the prior knowledge of peer loads, peers could have large load variations through the network system and could cause delays on serving clients. Our approach is simple where we mechanized a least load selection to minimized variation of load distribution. The inner node determines which appropriate peer by comparing its load and selects the least loaded peer that contains the resources. The collected peers which are the peers have the file requested in a query. The σ_n indicate a load of n peer that was selected in collection of S. In Equation 7, the minimum σ_n value is selected from peers $S = \{\sigma_1, \sigma_2,\ldots, \sigma_n\}$ where each σ_n is the load status of a peer. The inner node forwards the ID where the file is available to the peer.

$$CandidatePeer = getindex(\min\{\sigma_1, \sigma_2, ..., \sigma_i\}) \tag{7}$$

4 Experimental Evaluation

This section presents our performance result of the proposed scalable peer grouping used in multi-ring topology. First, number of nodes and distribution of files were defined where 150 peer nodes were configured in the network and 20,800 files with four classes which are classified by music, movies, documents and installers were randomly distributed. A mean of 140 files with 58 music, 32 movies, 38 documents and 12 installers in each peer are being shared within the peer network. The network topology is constructed by nodes consisting four nearest neighbor links. The number of inner node in a group is set to 8 and all queries of sub nodes are always directed to its inner node. The inner nodes flow of query is from right to left peer and the query continues until it reaches to the inner node that is the source of query.



(a) (b) (c) (d)

Fig. 3. Clustering result from scalable multi-ring (a), multi-ring (b), ring (c) and server based grouping (d)

4.1 Performance Measure

In measuring the performance of classification method, cross-validation of classified instances was used. In our experiment, classification method is measured by two constraints; changing the number of clusters with the specified class and determining number of epochs or iteration where a classifier optimizes the model to classify data accurately. The peer grouping schemes were evaluated by adding all peer members in each group. The knowledge-based clustering provides a divide and conquer technique that groups most relevant peer to prevent long routes in searching resources. The crisp grouping of *k*-means is easily determined by comparing the Euclidean distance of peer interest properties to each cluster centers represented by $d_i=|c_i - x|$. The class i that have smallest d_i is the chosen group. In the proposed clustering, because of the trained fuzzy system, peers can also be classified to other class using the threshold (Φ) specified while in classical clustering, peers are only grouped in one class which indicates crisp grouping. Setting up a small threshold will classifies more peers.

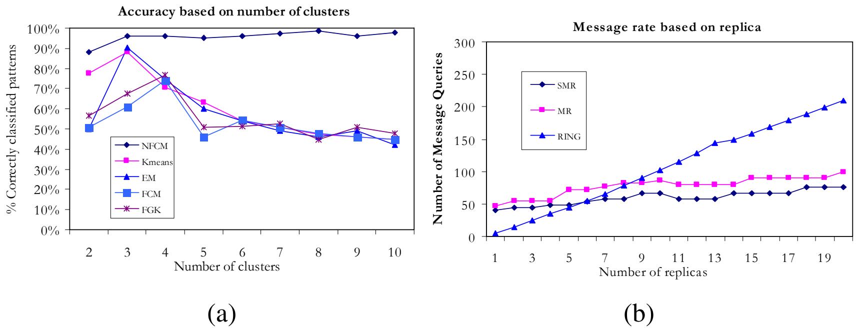


Fig. 4. Comparison of classification accuracy based on number of clusters (a) and message rate based on numbers of replica (b)

4.2 Experiment Result

Figure 3 shows the clustering result of the proposed neuro-fuzzy clustering (NFC) and other methods. It show 4 groups in each method where Fig. (a), Fig. (b) and Fig. (c) are distributed P2P while Fig. (d) is centralized method. The neighbor connections of each node are represented by blue and virtual overlays are presented by other colors. In processing the topology construction, the fuzzy system of NFC is already trained. NFC provides more relative peers in a group and scalability to be used in load balancing scheme to the system while in classical knowledge-based clustering peers are classified only in a single group.

Moreover, the scalability that is shown in Figure 3(a) provides more peers to select the least loaded peer to load balance the system. The trained fuzzy system of NFC is used to configure the classification structure. NEFCLASS-J for neuro-fuzzy clustering and Weka 3.5 [14] for other algorithms were used to compare the performance of NFC to *k*-means, EM, fuzzy clustering (FCM) and fuzzy GK (FGK) for clustering accuracy and multilayer perceptron (MLP) and radial basis function (RBF) for training based on number of iteration or epoch. In determining the clustering accuracy, the constraints of classification model use different number of clusters starting from 2 was set. Variables from the proposed NFC contain 3 fuzzy sets and use the fuzzy system to classify the data. Each class from data was evaluated for accuracy and errors from the classification were subtracted. The classification performance results are shown in Figure 4(a) and overhead performance in Figure 4(b).

In Figure 4(a), from 2 to 3 clusters, an increasing behavior is observed while from 3 to 10 clusters, there is a decrease of classification accuracy from other algorithms. The NFC was constant to its classification accuracy where the mean differences to other algorithm, starting from 4 clusters to 10 clusters, have 35% (*k*-means), 38% (EM), 43% (FCM), and 40% (FGK) better accuracy. Also, the accuracy based on epoch where NFC was compared to MLP and RBF which are accurate classifiers. The learning rate of classifiers was set to 0.05 and the epoch count was set from 10 to 100 to classify data. NFC has a constant result of accuracy from 10 to 100. The goal of the learning schemes is to learn the patterns and use the errors as a factor of learning which is expected to have a less accuracy at the start. However, in lesser epoch, NFC is already accurate on classifying compared to other methods. In Figure 4(b) shows the efficiency of the algorithms based on the number of replicas where the SMR is better in selecting more peers that has the same files. Because of the distributed nature of P2P, the process of load balancing is complex compared to a centralized system and so in the result in Figure 4(b) is necessary in determining the replica to have quick response on selecting the least loaded and avoids delay time on processing the decision.

5 Conclusion

The integration of the intelligent systems mostly contributes to the efficiency and reliability of the system in providing quality of service (QoS) to users. In this paper, we presented a knowledge-based peer grouping for P2P based on the proposed neuro-fuzzy clustering (NFC) for efficient multi-ring P2P topology. To have an accurate classification in the scalable grouping, the fuzzy system of FCM is trained and adjusted by using the neuro-fuzzy algorithm. Simulation shows the proposed peer grouping has more peers classified in each group. Result of scalability of NFC compared to *k*-means and shows that an averaging of 29% more peers in all groups. Moreover, other methods were compared for accuracy of classification and result shows that the proposed NFC outperformed other algorithm in terms of changing the clusters by having 35%, 38%, 43% and 40% better accuracy compared to *k*-means, EM, FCM and FGK, respectively and provides a quick response in selecting more peers that has same files to avoid delays.

References

1. Napster's website, `http://www.napster.com`
2. Gnutella community website, `http://gnutella.wego.com`
3. Rowstron, A., Druschel, P.: Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems. In: Guerraoui, R. (ed.) Middleware 2001. LNCS, vol. 2218, pp. 329–350. Springer, Heidelberg (2001)
4. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. IEEE/ACM Transactions on Networking 11(1), 17–32 (2003)
5. Zhuge, H., Chen, X., Sun, X., Yao, E.: HRing: A Structured P2P Overlay based on Harmonic Series. IEEE Transactions on Parallel and Distributed Systems 19(2), 145–158 (2008)
6. Zhang, C., Krishnamurthy, A., Wang, R.Y.: Brushwood: Distributed Trees in Peer-to-Peer Systems. In: Castro, M., van Renesse, R. (eds.) IPTPS 2005. LNCS, vol. 3640, pp. 47–57. Springer, Heidelberg (2005)
7. Pairot, C., Garcia, P., Skarmeta, S.: DERMI: A Decentralized Peer-to-Peer Event-based Object Middleware. In: Proc. of International Conference on Distributed Computing Systems, pp. 236–243 (2005)
8. Chen, W.T.C., Chao, H., Chiang, J.L.: An Interest-based Architecture for Peer-to-Peer Network Systems. In: Proc. of International Conference on Advanced Information Networking and Applications, pp. 707–712 (2006)

9. Chiou, H., Su, A., Yang, S.: Interest-based Peer Selection in P2P Network. In: Proc. of IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, pp. 549–554 (2008)
10. Yang, J., Zhong, Y., Zhang, S.: An Efficient Interest-Group based Search Mechanism in Unstructured Peer-to-Peer Network Systems. In: Proc. of International Conference on Computer Networks and Mobile Computing, p. 247 (2003)
11. Wu, W., Hu, W., Huang, Y., Qian, D.: Group-based Peer-to-Peer Network Routing and Searching Rules. In: Proc. of International Conference on High Performance Computing and Applications, pp. 510–514 (2005)
12. Junginger, M.O., Lee, Y.: The Multi-ring Topology-High Performance Group Communication in Peer-to-Peer Networks. In: Proc. of International Conference on Peer-to-Peer Computing (2002)
13. Nauck, D., Kruse, R.: NEFCLASS - A Neuro-fuzzy Approach for the Classification of Data. In: Proc. of ACM Symposium on Applied Computing ((1995)
14. Han, J., Kamber, M.: Data Mining Concepts and Techniques, 2nd edn., pp. 1–38. Morgan Kaufman, San Francisco (2006)

Supporting Medical Emergencies by MAS*

Roberto Centeno, Moser Fagundes, Holger Billhardt, and Sascha Ossowski

Centre for Intelligent Information Technologies (CETINIA)
University Rey Juan Carlos
Madrid, Spain
{roberto.centeno, moser.fagundes, holger.billhardt,
sascha.ossowski}@urjc.es

Abstract. In the emerging field of m-Health, advanced applications provide healthcare to people anywhere, anytime using broadband and wireless communications, as well as mobile computing devices. The notion of Service-Oriented Multi-Agent Systems (SOMAS) that has recently been proposed appears to adequately capture the requirements of applications in this field. The THOMAS abstract architecture and software platform supports the construction of SOMAS around an organisational metaphor. In this paper we present an application prototype built on top of THOMAS for a real-world mHealth scenario: mobile medical emergency management in an urban area.

1 Introduction

The use of information and communication technologies to support the efficient provision of healthcare services is an area of huge economic and social potential. In recent years, the notion of m-Health has become prominent, focusing on applications that provide healthcare to people anywhere, anytime using broadband and wireless communications, as well as mobile computing devices [1].

To successfully apply these technologies to this new scenario, a service-oriented approach is gaining popularity [2]. In this context, services are software entities that can be described, published, discovered, orchestrated, and invoked by other software entities. A service-oriented approach to m-health must consider semantics, because in healthcare, every description must have a unique, clear meaning. So, defining and maintaining expressive ontologies for m-health are crucial. Furthermore, healthcare applications are usually based on flexible and complex interactions between people playing different roles in diverse physical and social contexts. Agent technology provides the means to capture this structure because it proposes an interaction-oriented way of designing open software systems [3].

Although these research fields have diverse backgrounds and motivations, there is a growing interest in bridging the different worlds: on the one hand, software agents can be viewed as potential users and providers of semantic services,

* The present work has been partially funded by the Spanish Ministry of Science and Innovation, grants TIN2006-14630-C03-02 (FPI grants program) and CSD2007-00022 (CONSOLIDER-INGENIO 2010).

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 823–833, 2009.

on the other, web service technology can be used to support the interactions in multiagent systems. To this respect, the notion of Service-Oriented Multi-Agent Systems (SOMAS) has recently been put forward [4,5]. However, very few approaches go beyond a shallow integration, that not only refers to the conceptual level, but also affects the proper mechanisms used in the system [6].

The approach taken by the THOMAS project (MeTHods, Techniques and Tools for Open Multi-Agent Systems) uses the notion of *organisations* as a key abstraction to build SOMAS in open environments. It defines an abstract architecture, as well as a software platform, where agents and services interact according to norms and rules put forward by the organisational structure [7].

In this paper, we describe the application of THOMAS to a case study for a next-generation m-health scenario: the coordination of the different actors (patient, emergency centre, ambulances, hospitals, etc) involved in urban medical emergency management based on wireless mobile technology. Our application is build on top of the THOMAS abstract architecture, and is based on real-world data provided by the Madrid Medical Emergency Service (SUMMA112).

The paper is organised as follows. Section 2 describes our problem domain, the management of medical emergencies in the Autonomous Region of Madrid, and outlines our model of it. Section 3 presents the application prototype designed to deal with these problems. A particular scenario of a medical emergency which has been implemented is presented in section 4. Section 5 concludes our work and points to future lines of work.

2 Semantic Domain Model

Medical emergencies have a high priority given the potential life risk to the patients. These extreme circumstances demand the usage of appropriate resources within an acceptable response time.

In the Autonomous Region of Madrid in Spain, medical emergencies are handled by the SUMMA112. The services provided by SUMMA112 include: reception and management of medical calls, management of beds in hospitals, coordination and assistance of medical emergencies "in situ" and inter-hospital transportation of patients. Human resources consist of approximately 100 call operators, 36 medical doctors and 36 coordination technicians. Material resources include four classes of vehicles for medical assistance: Emergency Mobile Units (Mobile UVI) for advanced life support "in situ", occupied by one medical doctor, one DUE and two technicians; Rapid Intervention Vehicles (VIR) equipped with technology and medical instruments similar to Mobile UVIs (its purpose is to reach the patient as fast as possible); Home Assistance Units; and helicopters. The SUMMA112 coordination centre receives more than 1.200.000 emergency calls per year – almost one call each 30 seconds. Around 60.000 of these calls are classified as situations of life risk and receive assistance of Mobile UVIs, VIR or helicopter. The centre's capability of receiving calls is 3.000.000 per year (8.000 per day).

Out of the various services provided by the SUMMA112, we are particularly interested in the different processes that take place in medical emergency assistance, including the transportation of patients to hospitals. This is one of the most active tasks of the SUMMA112 and certainly one of the tasks with a high social impact. The assistance in medical emergencies is managed by SUMMA112's coordination centre. The human resources of the centre play the following roles: *Supervisor* is the highest responsible for the assistance operation inside and outside the Coordination Centre of SUMMA112; *Medical Doctor Regulators* regulate the incoming calls, decide which resource to assign given the circumstances, and provide medical aid to the patients; *Technicians* mobilise the resources assigned by the Medical Doctors Regulators; *Operators* are the professionals that receive the calls and catch the first data; *Nursery staff* manage the inter-hospital transportations and bed allocation in the hospitals.

A typical medical emergency assistance starts when a patient calls SUMMA112, asking for assistance. The call is received by an Operator, who gathers the initial data from the patient. Then she forwards the call to a Medical Doctor, who assigns the resources to attend the request according to its evaluation. In some cases the Operator forwards the call directly to the Technician. The Medical Doctor assigns the resource, however it does not mobilise the resource. Technicians do this task by selecting the units taking into account their availability, distance and time to reach the patient, type of resource (ambulances with different capabilities and outsider's resources, such as *Red Cross*, *Civil Protection*). Finally, according to the patient condition, she is transported to a hospital.

We have modelled the structure of the application domain as a semantic model. The main classes in the model are: *Hospital*, *Patient*, *Vehicle* and *SUMMA*. Other classes, some of them inherited, were modelled in order to specify these main classes. Figure 1 presents a *Protege* screenshot highlighting the Patient class which has the object properties *has-disease* and *has-medical-record*. The remaining properties of a patient are specified through data-type properties.

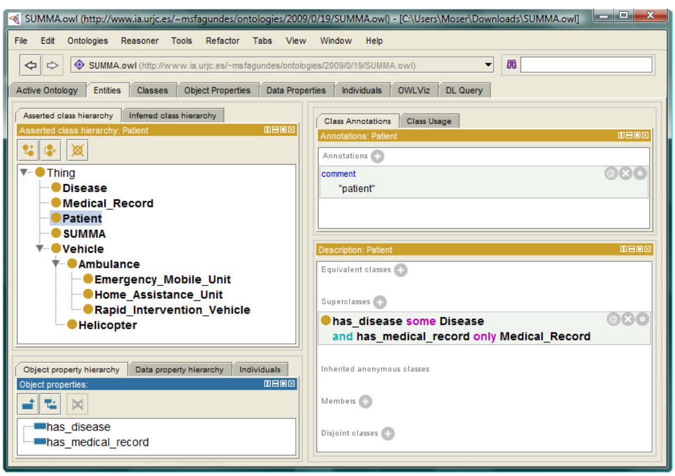


Fig. 1. Classes modelled using the Protege tool

3 System Architecture

The aim of our system is to provide decision support and value added services to the actors involved in medical emergency assistance procedures, basically, to physicians, patients and coordination staff. There are two factors that have determined the basic decision regarding the system architecture. The first factor consists of the complexity of the underlying application domain and its natural structure – namely a set of autonomous entities that collaborate with each other in order to obtain their goals. The quality of the whole system depends strongly on the efficiency and effectiveness of this collaboration, that is, on the mechanisms used to organise and coordinate the interactions among the active entities. With this in mind, we have chosen an organisation-based multiagent system as the primary architecture. Organisation-based multiagent systems allow for the inclusion of organisational mechanisms or elements (e.g., norms, incentive and punishing mechanisms). Such elements assure the functioning of the system and can improve its efficiency by controlling and manipulating the possible interactions that may take place, that is, without considering the actual agents that will participate in the system. The entities that participate in the domain are represented by agents playing predefined roles in an organisation (mHealth organisation).

The second factor we have considered is the fact that identified entities in the application domain can be classified into two classes: *i) active entities* whose actions are driven from objectives and that possess a certain degree of freedom in their actions, and *ii)* entities that actually act as providers of certain services (e.g. a medical record store, emergency centre finder, etc.). We propose an architecture where active entities are implemented as agents and non-active entities as web services, and outline its implementation on top of the *THOMAS* platform [8].

3.1 THOMAS Platform

The THOMAS platform implements an abstract architecture for open multiagent systems based on the organisational paradigm. The agents have access to the functionality offered by THOMAS through a range of services included in several modules. The main components of THOMAS are the following:

- *Service Facilitator (SF)*: this component provides simple and complex services to the active agents and organisations. Basically, its functionality is like a yellow page service and a service descriptor in charge of providing a green page service.
- *Organisation Manager Service (OMS)*: it is mainly responsible of the management of the organisations and their entities. It allows to create and to manage organisations.
- *Platform Kernel (PK)*: it maintains basic management services for an agent platform; it is in charge of communication between agents, etc.

Besides the possibility to structure an application through the concept of organisation, THOMAS allows for a seamless integration of multiagent systems with

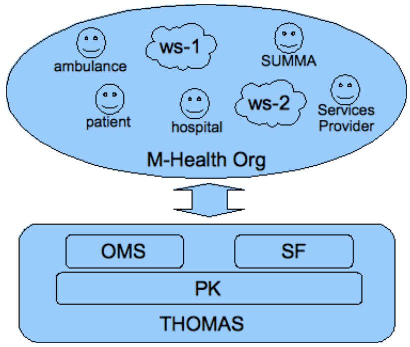


Fig. 2. System architecture

web services. Web services can be easily included and can be registered in the service facilitator.

Figure 2 presents the system architecture using the THOMAS platform. Our mHealth application prototype makes use of the SF, OMS and PK modules of the platform.

3.2 Domain System Architecture

The whole system has been implemented as an mHealth organisation within the THOMAS platform. We have mapped the active entities of the semantic domain model to roles in that organisation. We defined the roles: patient, hospital, ambulance, emergency coordinator, and "service provider". The organisation requires exactly one agent to play the role "emergency coordinator". The other roles can be played by any number of agents. The last role is used to define agents that provide access to web services. The non-active elements of the semantic model have been mapped to the following web services: "medical record store", "emergency centre finder", "ambulance finder", "hospital finder". All these web services are accessible through "Service Provider Agents" (agents playing the role "Service Provider").

In our current prototype, the organisation comprises one type of agent for each possible role in the organisation:

- **Patient:** Agents playing role patient represent potential users of medical emergency assistance. Each agent playing the role patient allows the user to contact an emergency centre. For this, the agent uses web services (if available) in order to find the closest emergency centre. If a centre is found, the agent automatically engages an interaction with the agent representing the centre to perform an emergency call. In this call it informs about the current location of the patient and a description of the symptoms (obtained through an user interface).
- **Hospital:** Agents playing this role represent a hospital. Each hospital has different characteristics such as the number of available beds, the set of

diseases the hospital is able to treat, location, etc. In the scope of the mHealth organisation, hospital agents decide whether a patient should be admitted, taking into account the number of free beds, the dangerousness of the illness, etc.

– **Ambulance:** agents which are playing this role in the organisation, represent an ambulance vehicle together with the human resources assigned to it. In our current prototype, we have not taken into account the different characteristics of the ambulances: all emergency vehicles are assumed to offer the same services. Ambulance agents can receive missions to take patients and to find the "best" hospital for each patient. When a new mission is received, the ambulance has to pick up the patient, to decide upon the "most adequate" emergency centre for her treatment (usually the closest hospital that provides the medical services needed), and to transport her to this destination. The ambulance agent has to inform the emergency coordinator when a mission is accepted and at the same time it can obtain medical history data of the patient from appropriate web services (if they are available). This functionality may be very useful because it allows the physician travelling in the ambulance to analyse the patient's medical history before they actually arrive at the patient's location.
– **SUMMA:** This agent plays the emergency coordinator role which is a main piece in the organisation. An agent playing this role is able to receive emergency calls from patients, and to find the "best" ambulance for each case, thus performing the high-level management of the emergency assistance procedure.
– **Services Provider:** Service provider agents provide access to web services.

Regarding web services, we have implemented simple web service instances for each identified type of service:

– *Emergency Centre Finder:* this service finds the responsible emergency centres for a given location. The result of the invocation of this service will be the identifier of an agent which is playing the emergency coordinator role, in our particular case, the SUMMA agent. This service could be external to the organisation.
– *Medical record store:* this web service allows agents to store and retrieve the patients' medical history data providing user name and password. This service could be external to the organisation.
– *Ambulance finder*: this service can be considered as internal to an emergency coordinator centre. The service may be implemented with different strategies to decide which is the best ambulance to assign to a given patient. In the current stage, we return the ambulance that is closest to the patient's position.
– *Hospital finder*: similar to the ambulance finder, this service is considered to belong to the organisation. It is used by an agent playing the role ambulance to obtain the most appropriate destination hospital for a given patient. Currently we assign the closest hospital to the patients location that is able to treat the identified disease.

As mentioned before, within the organisation, all web services are made accessible through service provider agents. The web services have been specified using the standards OWL-S [9] and WSDL [10]. This allows for using standard techniques for service discovery, composition, etc. One advantage of implementing the last two tasks as web services is that we can easily compare different strategies to assign ambulances and hospital to patients.

4 Case Study

In order to evaluate our system we have created a spatial environment simulation tool in which the mHealth organisation can be embedded.

4.1 Spatial Environment Simulator

The environment simulator is an independent module that captures key features of our problem domain. It has two fundamental functions: *i)* it allows to set up agents (and thus, organisations), and *ii)* it recompiles information about the actions that take place in the organisation. The simulation tool is composed of a control layer and a graphical interface.

The control layer receives the actions/interactions and updates the scenario according to these actions. The agents interact with the simulator using the FIPA-REQUEST protocol. The message parameters vary according to the intended action and the role played by the agent. The actions are executed asynchronously – in the exact time step the simulator receives the requests.

Before the simulation starts, the participant agents have to register in the simulator. The registration also follows the FIPA-REQUEST protocol, where the agent specifies its name, role and initial location in the message content. Once the simulation begins, the environment is constantly updated. Basically the simulator controls two parameters: time and location. It represents a spatial world where each agent is positioned at a particular location and at a particular point in time.

According to the role played by an agent, it has a set of available actions. Three roles are represented in the physical environment: patients, hospitals and ambulances. Figure 3 illustrates the set of available actions for each role as well as the simulator design. Patients have only one available action which consists of informing their position in the map in order to receive assistance. The hospitals are able to accept or refuse patients brought by the ambulances, as well as to release the patients. Finally, the ambulances are able to catch patients that ask for help, release patients in the hospitals, and move around the physical environment by specifying a destination position. Complex routes can be composed by specifying a sequence of positions. Figure 3 illustrates the simulator design and the set of available actions for each role.

Figure 4 illustrates the graphical interface by showing an execution where five hospitals (*Escorial*, *Cantoblanco*, *CarlosIII*, *DoceOctubre* and *Fuenlabrada*), one ambulance (*Ambulance1*) and one patient (*Patient1*) joined the mHealth organisation. On the left side of the figure it is shown the *checkbox* for hospital

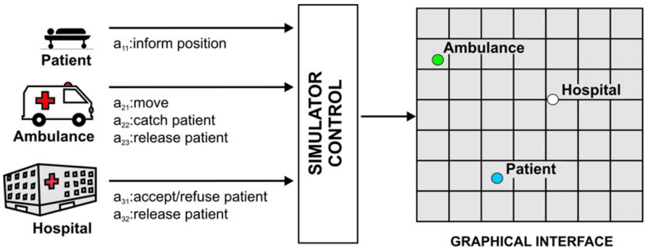


Fig. 3. Simulator design and available agent actions

selection. This *checkbox* window is not part of the simulator, but part of an agent launcher.

While the simulator implements real geographical coordinates, at the moment the simulation of travel times is still somewhat simple, as it currently does not take into account the specific topology of the Madrid road network.

4.2 Medical Emergency Assistance Example

We have used the environment simulation tool to carry out several example traces. In order to do so we have set up the mHealth organisation with 31 hospitals and 60 ambulances. The data of the hospitals regarding bed capacities, and ability to treat certain diseases correspond to the hospitals existing in the Autonomous Region of Madrid. In the following, we describe one trace as an example.

Suppose that Bob, a British tourist visiting Madrid, suddenly feels a strong pain in his chest. As this is his first time in Madrid, he does not know what he should do. However, he carries his PDA with the mHealth software suite installed. In this emergency situation, Bob's personal agent joins the *mHealth organisation* playing the *patient* role. Immediately, the patient agent contacts a *services provider agent* looking for an emergency centre finder service.

When this service is found it returns the SUMMA112's identifier which is the agent playing the *emergencies coordinator role* in the mHealth organisation.

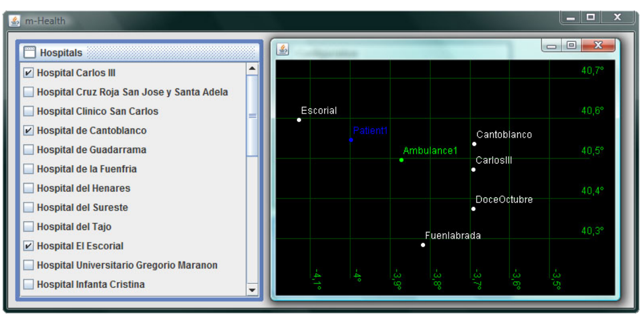


Fig. 4. Simulator screenshot

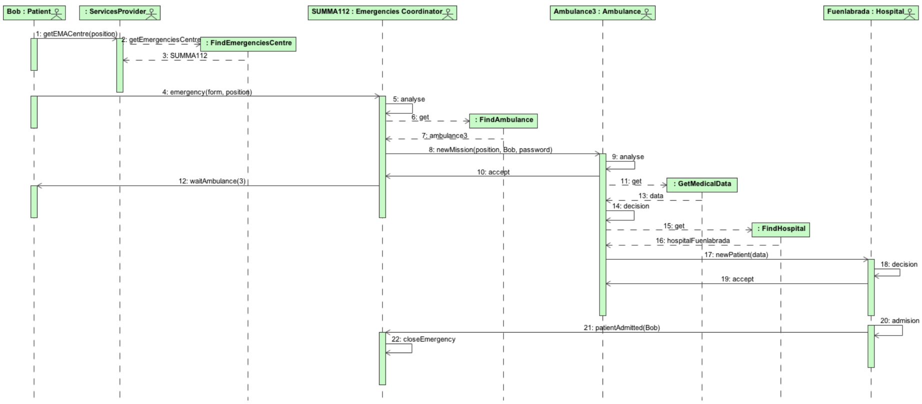


Fig. 5. Case study sequence diagram

Together with the SUMMA112's identifier the patient agent receives the form for the patient's data and symptoms. This form is shown to Bob in order to fill it out.

Once Bob selects his symptoms the form is sent to the agent SUMMA112 by the patient agent. When the SUMMA112 receives the new emergency call, it uses its classification method to classify the emergency and to perform a preliminary diagnosis. SUMMA112 detects that Bob is most likely suffering a hearth attack. Therefore, he is classified as an urgent patient that needs an ambulance urgently.

The SUMMA112 agent will use its *ambulance finder service* to select the most adequate ambulance to take Bob. Ambulances number 3, 6 and 28 are available closed to Bob's current location. The ambulance finder service will select *ambulance* 3 because it is the closest among all available ambulances in that area.

After an ambulance is selected, the SUMMA112 agent sends a message with the new mission to the *ambulance* 3 agent, which is the agent representing *ambulance* 3 in the mHealth organisation. When *ambulance* 3 receives the new mission it confirms to accept to take Bob immediately.

While the vehicle is going to Bob's position, the *ambulance 3* agent tries to find possible providers of medical data of patients. It finds the service provider agent that implements the *medical record store service* and obtains all available medical history data for Bob. Once the medical history is obtained it is shown to the physician in *ambulance* 3. The physician analyses the data and can determine the required assistance.

When *ambulance* 3 arrives to Bob's location, the physicians records his symptoms and perform an "in situ" diagnosis and decide to what hospital he should be taken. In order to do so, the *ambulance* 3 agent uses the *hospital finder service* sending Bob's location, symptoms and diagnosis. Taking into account these information and the set of available units in the medical centres, the Fuenlabrada hospital is selected because is the nearest hospital which is able to treat heart attacks.

After that, the *ambulance* 3 agent sends a message to the agent representing the *Fuenlabrada hospital* requesting the admission of the new patient. At this moment, the hospital decides to admit Bob because it has free beds and Bob is transferred to the hospital as fast as possible.

Finally, when Bob is admitted at the hospital, the *Fuenlabrada hospital* agent informs SUMMA112 agent about the admission of Bob. At this point, the SUMMA112 agent closes the case successfully.

5 Conclusions

In this paper we have presented an application prototype in the m-health domain. The aim of the application is to provide decision support and value added services to the actors involved in medical emergencies - physicians, patients, hospitals and coordination staff. According to the characteristics of the domain, an organisation-based multiagent system has been chosen where a set of autonomous entities collaborate with each other in order to assist medical emergencies successfully. Along the medical emergency assistance process there exists some entities which just provide certain services without any notion of objectives (e.g. medical record store, emergency centre finder, etc.). In order to best incorporate such services the organisation-based multiagent system has been combined with service oriented computing. With this in mind, our application has been built as an instantiation of the THOMAS abstract architecture – an architecture that combines both technologies, allowing agents to interact with services according to norms, roles, etc., in the scope of an organisation. We have illustrated the dynamics of our prototype by a case study based on real-world data provided by the Madrid Medical Emergency Service (SUMMA112).

Other works, such as [11], have been developed to support medical emergencies processes. In this paper, the authors deal with the problem of selecting the best ambulance to be assigned to an emergency call. They also propose a multiagent approach where each actor involved in a medical emergency event is represented through an agent in an agent society. In addition, the selection of the ambulance is carried out following an auction based mechanism and a trust model based on the past experience with the ambulances' drivers. The main difference with our work is that they focus on the problem of selecting the best ambulance supposing that the hospital is known in advance. Whereas our system intents to provide support the the actors involved in the whole process from its start (the call from a patient) to the end (the admission of the patient in a hospital). Some other work related to disaster management in general has been proposed for instance in [12] dealing with complex scenario where different teams such as polices, fire brigades, ambulances, etc. have to coordinate each other to assist emergencies. In our work we focus on medical emergencies, in a similar way to the work presented in [13], taking into account just hospitals and ambulances.

As future work we pretend to study more complex mechanisms to select the best ambulance and hospital which could optimize the global utility of the system regarding certain desirable parameters (e.g., response time and cost). These mechanisms may take into account aspects like uncovered regions, past experiences with hospitals and ambulances, etc. We would like to compare different mechanisms within different situations (from "normal" functioning to high-load situations such as multiple accidents or other large-scale disasters). In addition, we plan to extend to application in order to provide support also in other typical cases of medical emergency assistance. For instance, in the case a patient is not classified as urgent, the SUMMA112 centre could recommend to visit a hospital soon. The system could provide additional help to that patient, e.g., proposing a route to reach the closest hospital.

References

1. Cáceres, C., Fernández, A., Ossowski, S.: Cascom - context-aware health-care service coordination in mobile computing environments. ERCIM News (60) , 77–78 (2005)
2. Huhns, M.N., Singh, M.P.: Service-oriented computing: Key concepts and principles. IEEE Internet Computing 9(1), 75–81 (2005)
3. Cáceres, C., Fernández, A., Ossowski, S., Vasirani, M.: Agent-based semantic service discovery for healthcare: An organizational approach. IEEE Intelligent Systems 21(6), 11–20 (2006)
4. Huhns, M.N.: A research agenda for agent-based service-oriented architectures. In: Klusch, M., Rovatsos, M., Payne, T.R. (eds.) CIA 2006. LNCS, vol. 4149, pp. 8–22. Springer, Heidelberg (2006)
5. Huhns, M.N., Singh, M.P., Burstein, M.H., Decker, K.S., Durfee, E.H., Finin, T.W., Gasser, L., Goradia, H.J., Jennings, N.R., Lakkaraju, K., Nakashima, H., Parunak, H.V.D., Rosenschein, J.S., Ruvinsky, A., Sukthankar, G., Swarup, S., Sycara, K.P., Tambe, M., Wagner, T., Gutierrez, R.L.Z.: Research directions for service-oriented multiagent systems. IEEE Internet Computing 9(6), 65–70 (2005)
6. Fernández, A., Ossowski, S.: Exploiting organisational information for service coordination in multiagent systems. In: Padgham, L., Parkes, D.C., Müller, J., Parsons, S. (eds.) AAMAS (1), IFAAMAS, pp. 257–264 (2008)
7. Ossowski, S., Julián, V., Bajo, J., Billhardt, H., Botti, V., Corchado, J.: Open mas for real world applications: An abstract architecture proposal. In: Proc. XII Conference of the Spanish Association for Artificial Intelligence (CAEPIA), vol. 2(1), pp. 151–160 (2007)
8. Argente, E., Julian, V., Botti, V.: Mas modelling based on organizations. In: 9th Int. Workshop on Agent Oriented Software Engineering (AOSE 2008), pp. 1–12 (2008)
9. Web based service ontology, http://www.daml.org/services/owl-s-/1.0
10. Language, W.S.D., http://www.w3.org/tr/wsdl
11. López, B., Innocenti, B., Busquets, D.: A multiagent system to support ambulance coordination of urgent medical transportation. IEEE Intelligent Systems 23(5), 50–57 (2008)
12. Rescue, R.: http://www.robocuprescue
13. Ciampolini, A., Mello, P., Storari, S.: A multi-agent system for medical services synergy and coordination. In: Nealon, J., Cortes, U., Fox, J., Moreno, A. (eds.) International ECAI 2004 workshop on Agents applied in health care, pp. 38–46 (2004)

Ontology-Based Query Interface in a System for Semantic Integration of XML Data

Tadeusz Pankowski

Institute of Control and Information Engineering,
Poznań University of Technology, Poland
tadeusz.pankowski@put.poznan.pl

Abstract. We present an ontology-oriented approach to formulate queries in XML data integrating system in P2P environment. In this approach, XML schema trees (tree-pattern schemas) are annotated in a global domain ontology by terms of some OWL categories. The annotations are then used to two purposes: (1) to infer mappings between local XML schemas, and (2) to create local "ontology views" which can be queried by users. We propose a set of rules for inferring mappings between XML schemas based on semantic relationships between terms used for annotating these schemas. Queries are issued against the local ontology using a user-friendly GUI. The query is answered in the local data source as well as in semantically connected data sources in a distributed P2P connected system. In this way the following objectives are accomplished: (1) queries can be formulated in a natural way conforming to the user's understanding of the application domain; (2) queries are executed in a dynamic P2P system, where answers are obtained from many data sources semantically connected by means of mappings between their schemas; (3) queries are efficiently executed by translating them into XQuery programs.

1 Introduction

Semantic information integration plays the central role in building of large scale systems such as P2P databases or e-commerce systems, where data comes from many different sources with different schemas [7]. In peer-to-peer (P2P) data management systems, the autonomous computing nodes (the *peers* or *partners*) cooperate to share resources and services. The peers are connected to some other peers they know or discover. In such systems the user issues queries against an arbitrarily chosen peer and expects that the answer will include relevant data stored in all P2P connected data sources. The data sources are related by means of schema mappings, which are used to specify how data structured under one schema (the source schema) can be transformed into data structured under another schema (the target schema) [10,12]. A query must be propagated to all peers in the system along semantic paths of mappings and reformulated accordingly. The partial answers must be merged and sent back to the user.

In such systems the query interface should be independent on any particular data source and should be based on a domain ontology of the application domain.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 834–843, 2009.

Usually, the mediated schema presented to the user is based on a non-semantic data model, i.e. it is a relational schema defined in SQL or an XML schema specified in XML Schema or DTD [18]. Then, the data can be retrieved using a non-semantic query language such as SQL or XQuery. Such a solution can be accepted by application developers but could not be acceptable for normal users, who usually do not have enough database knowledge but are well acquaintanted with domain knowledge described by a domain ontology. Thus, the challenging issue is creating ontologies based on XML schemas and using them in both query formulation and query answering.

Related Work. There is a number of papers addressing the problem of semantic data integration among XML sources. In [17] a global ontology is constructed by merging individual local ontologies, which represent XML source schemas. Ontologies (both global and local) are expressed in RDF Schema (RDFS). Mappings between the global ontology and local XML schemas are established manually. Representation of XML source schemas in RDFS and OWL ontologies was also shown in [5]. In some approaches (e.g. in the MOMIS system [4]) semantic metadata is embedded into XML by means of annotation (e.g. by WordNet annotating). Klein [13] proposes a procedure to transform XML data into RDF by annotating the XML document via external RDFS specifications. Correspondences between relational schemas are usually expressed by source-to-target dependencies over relational formulas [1,10]. Arenas and Libkin adopted this approach to XML schemas using pattern-formulas instead of relational ones [2]. Frameworks for retrieval and browsing ontological data are proposed in [9] and [11]. In [18] retrieval operations on ontology are mapped into queries over data sources of COTS products.

Contributions. We make the following contributions in this paper:

– We propose a way for creating ontology views over XML schemas (called xs-ontologies) by annotating these schemas with terms of some OWL categories (classes, object properties and datatype properties) taken from a global domain ontology [8]. We propose an RDF-oriented representation of xs-ontologies preserving the nesting structure of the XML schemas.
– The main novelty of the paper are rules for inferring mappings between XML schemas. We start from semantic relationships between (extended) RDFS-triples constituting xs-ontologies. We propose recursive rules which shows how from these relationships and from already inferred mappings, new mappings between underlying XML schemas can be derived.
– We sketch an ontology-based query language OnBE for querying a local ontology. Expressions in OnBE consists of two parts: *navigational expression* part corresponding to *conjunctive XQuery* [17], and *query qualifier* part, where restrictions over variables defined in the navigational part are formulated. Restrictions can involve existential and universal quantifications.
– We developed algorithms for translating OnBE queries into XQuery. In this process ontology-to-schema and schema-to-schema mappings are used.

XQuery programs are propagated and executed over all P2P-accessible data sources [16,15].

The structure of the paper is as follows. In Section 2 we discuss the problem of acquiring a schema ontology by annotating a local XML schema. Rules for inferring ontology and schema mappings are proposed in Section 3. The ontology-based language OnBE is presented in Section 4. We discuss its constructs and propose a method for translating and executing OnBE queries. Section 5 concludes the paper.

2 Deriving XML Schema Ontology

Annotation of XML schema is commonly used as a method for enriching XML schema with some semantic information [5,4,9,14,17]. This information can be used to support many data integration activities. In our approach, applied in SixP2P system [16,15,6], the annotation consists in assigning OWL elements of categories *classes*, *objectProperties*, and *datatypeProperties*, to elements and edges of XML schema trees. Terms used in annotation of a schema S are taken from a global domain ontology O_{Dom}, and we obey the following rules:

- non-terminal nodes are annotated with OWL *class* names;
- edges between two non-terminal nodes are OWL *object property* names, where the predecessor is the domain and the successor – the range of the property;
- edges between non-terminal and terminal nodes are annotated by OWL *datatype property* names, where the domain is the non-terminal node and the range – the terminal one.

We assume that XML schemas will be specified by *tree-pattern formulas* [16]. For example, schema S_1 from Figure 1 has the following specification:

$$S_1(x_1, x_2, x_3, x_4, x_5, x_6) = /authors[author[name = \$x_1 \wedge univ = \$x_2 \wedge paper[title = \$x_3 \wedge year = \$x_4 \wedge conf[name = \$x_5 \wedge org = \$x_6]]]]$$

Annotations are stored in `<appinfo>` elements within XML schema, where syntax of `<appinfo>` element is:

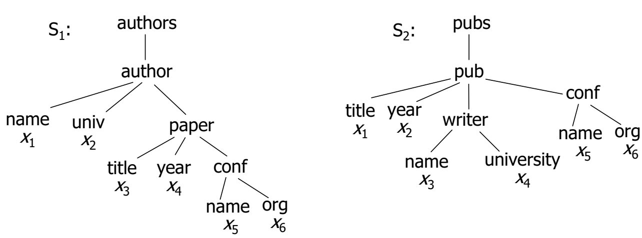


Fig. 1. Sample XML schema trees S_1 and S_2

Table 1. Annotation of schema S_1

Annotated schema element	*OWL element category*	*OWL element name*	*Range (schema element)*
authors	*class*	*authors*	
	objectProperty	*contains*	*author*
author	*class*	*author*	
	objectProperty	*authorOf*	*paper*
	datatypeProperty	*name*	*name*
	datatypeProperty	*university*	*univ*
paper	*class*	*paper*	
	objectProperty	*presentedAt*	*conf*
	datatypeProperty	*title*	*title*
	datatypeProperty	*year*	*year*
conf	*class*	*conference*	
	datatypeProperty	*name*	*name*
	datatypeProperty	*organizer*	*org*

```
annotation        ::= <xs:annotation> appinfo </xs:annotation>
appinfo           ::= <xs:appInfo source="SixP2P">
                          class
                          objectProperty*
                          datatypeProperty*
                      </xs:appInfo>
class             ::= <class name=string/>
objectProperty    ::= <objectProperty name=string range=string/>
datatypeProperty  ::= <datatypeProperty name=string range=string/>
```

Table 2. xs-ontology $O_{S_1 \cup S_2}$ corresponding to annotations of S_1 and S_2

O_{S_1}: Triples from xs-ontology of S_1	O_{S_2}: Triples from xs-ontology of S_2
$(authors, contains, author)$	$(publications, contains, publication)$
$(author, name, \$x_1)$	$(publication, title, \$x_1)$
$(author, university, \$x_2)$	$(publication, title, \$x_2)$
$(author, authorOf, paper)$	$(publication, writtenBy, writer)$
$(paper, title, \$x_3)$	$(writer, name, \$x_3)$
$(paper, year, \$x_4)$	$(writer, university, \$x_4)$
$(paper, presentedAt, conference)$	$(publication, presentedAt, conference)$
$(conference, name, \$x_5)$	$(conference, name, \$x_5)$
$(conference, organizer, \$x_6)$	$(conference, organizer, \$x_6)$

Relations in xs-ontology $O_{S_1 \cup O_2}$:
$author \sqsubseteq writer$
$paper \sqsubseteq publications$
$authorOf \sqsubseteq inverseOf(writtenBy)$
$isRoot(authors)$
$isRoot(publications)$

An annotation of XML schema tree S_1 from Figure 1 is given in Table 2. S_2 can be annotated in a similar way.

The annotation of S in a global domain ontology O_{Dom} determines an ontology representing S. We will call it *xs-ontology* (*XML schema ontology*) and denote by O_S. That ontology consists of terms from O_{Dom}, used in the annotation, variables (denoting places for datatypeProperty values), and subsumptions [3] defined over O_{Dom} involving annotating terms. Additionally, $isRoot()$ denotes the class annotating the root of the schema. The xs-ontology O_S will be represented by a set of triples and a set of relations, as is shown in Table 2.

3 Inferring Schema Mappings

A crucial issue in P2P data integration is to define *mappings* between different schemas. Schema mapping is a specification defining how data structured under one schema (the *source schema*) is to be transformed into data structured under another schema (the *target schema*). A schema mapping specifies the semantic relationship between a source schema and a target schema.

A mapping from a source schema S to a target schema T is an expression of the form (a source-to-target formula [10]) [16]

$$m_{ST} := \forall \mathbf{x}(S(\mathbf{x}) \Rightarrow \exists \mathbf{y} T(\mathbf{x}', \mathbf{y})), \quad (1)$$

where $\mathbf{x}' \subseteq \mathbf{x}$, and $\mathbf{y} \cap \mathbf{x} = \emptyset$. Variable names are used to indicate correspondences between text values of paths bound to variables. In practice, a correspondence also involves a function that transforms values of source and target variables. These functions are irrelevant to our discussion, so they will be omitted. A mapping is a special case of a query, where the query qualifier is $TRUE$.

In this section, we will show how a mapping from S to T can be inferred automatically using xs-ontology $O_{T \cup S}$. First, we will define tree formulas over xs-ontologies and mappings between two xs-ontologies. Next we define rules for inferring ontology mappings. Mappings between xs-ontologies can be easily translated into mappings between underlying XML schemas.

Definition 1. *A tree formula (TF) E (or $E(\mathbf{x})$ with explicit indication of variables) over an xs-ontology O_S is an expression conforming to the syntax:*

$$\begin{array}{l} E ::= a = x \mid P[E] \mid E \wedge \cdots \wedge E, \\ P ::= \epsilon \mid c \mid c/P, \end{array} \quad (2)$$

where a is a datatypeProperty name, c is a class name and $x \in \mathbf{x}$ is a variable. P is a path (possibly empty, ϵ) consisting of class names.

The TF over O_S is closely connected with the schema S, e.g. for O_{S_1} its TF E_1 differs from S_1 only in names: schema names in S_1, and annotating names in E_1, i.e.

$$\begin{array}{l} E_1 = authors[author[name = \$x_1 \wedge university = \$x_2 \wedge paper[title = \$x_3 \wedge \\ \quad year = \$x_4 \wedge conference[name = \$x_5 \wedge organizer = \$x_6]]]] \end{array}$$

Definition 2. *The following recursive rules define a class of TFs over an xs-ontology O_S:*

– *if $(c, a, \$x') \in O_S$, then both $a = x$ and $c[a = x]$ are TFs (over O_S); (names of variables are immaterial, we only require that they are different in TF);*
– *if $E_1, ..., E_k$ are TFs, then $E_1 \wedge \cdots \wedge E_k$ is a TF;*
– *if $(c, r, c') \in O_S$ and $c'[E]$ is a TF, then also $c[c'[E]]$ is a TF.*
– *if $(c, r, c') \in O_S$ and $c'/P[E]$ is a TF, then $c/c'/P[E]$ is a TF.*
– *if $E_1 \wedge \cdots \wedge E_m$, and $c[E_1]$,..., $c[E_m]$ are TFs, then $c[E_1 \wedge \cdots \wedge E_k]$ is a TF.*

Definition 3. *Let $E_1, ..., E_m$, $m \geq 1$, be TF (over O_S). $P(P_1, ..., P_m)$ is a prefix of $(E_1, ..., E_m)$, denoted $P(P_1, ..., P_m) = prefix(E_1, ..., E_m)$, if $P[P_1[E_1] \wedge \cdots \wedge P_m[E_m]]$ is a TF. The prefix is maximal if $isRoot(first(P))$ holds in O_S.*

Definition 4. *A mapping from a source ontology O_S to a target ontology O_T is a formula of the form*

$$E_S(\mathbf{x}) \Rightarrow E_T(\mathbf{x}), \tag{3}$$

where $E_S(\mathbf{x})$ and $E_T(\mathbf{x})$ are TFs over O_S and O_T, respectively.

The following rules specify the way of inferring mappings between two xs-ontologies O_S and O'_S.

$$(R1)\ \frac{(c, a, \$x), (c', a', \$x'), c \sqsubseteq c', a \sqsubseteq a'}{c[a = \$x] \Rightarrow c'[a' = \$x]}$$

$$(R2)\ \frac{c[E_1] \Rightarrow c'[E'_1], c[E_2] \Rightarrow c'[E'_2]}{c[E_1 \wedge E_2] \Rightarrow c'[E'_1 \wedge E'_2]}$$

$$(R3)\ \frac{c_1[E_1] \Rightarrow c'_1[E'_1], c_2[E_2] \Rightarrow c'_2[E'_2], (c_1, r, c_2), (c'_1, r', c'_2), r \sqsubseteq r'}{c_1[E_1 \wedge c_2[E_2]] \Rightarrow c'_1[E'_1 \wedge c'_2[E'_2]]}$$

$$(R4)\ \frac{c_1[E_1] \Rightarrow c'_1[E'_1], c_2[E_2] \Rightarrow c'_2[E'_2], (c_1, r, c_2), (c'_2, r', c'_1), r \sqsubseteq inverseOf(r')}{c_1[E_1 \wedge c_2[E_2]] \Rightarrow c'_2[E'_2 \wedge c'_1[E'_1]]}$$

$$(R5)\ \frac{\begin{array}{c} E_1 \Rightarrow E'_1, ..., E_n \Rightarrow E'_n, \\ P(P_1, ..., P_n) = prefix(E_1, ..., E_n), P'(P'_1, ..., P'_n) = prefix(E'_1, ..., E'_n) \end{array}}{P[P_1[E_1] \wedge \cdots \wedge P_n[E_n]] \Rightarrow P'[P'_1[E'_1] \wedge \cdots \wedge P'_n[E'_n]]}$$

The graphical interpretation of above rules is given in Figure 2. The promise of a rule is depicted on the left-hand side, and the conclusion – on the right-hand side. Labeled edges on the left-hand side denote RDF triples from the ontology.

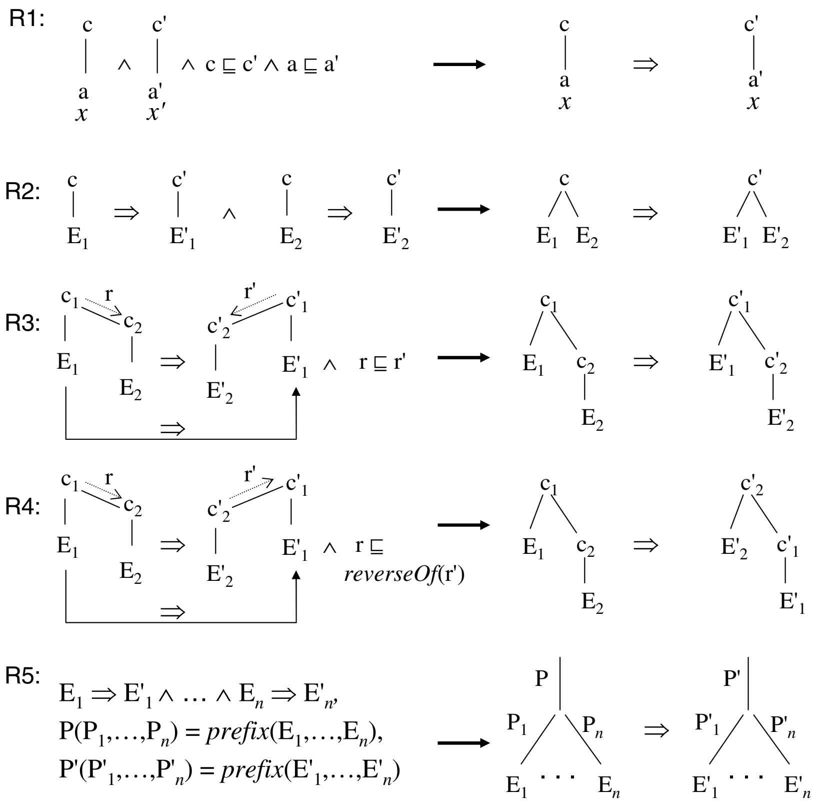


Fig. 2. Graphical interpretation of rules $R1$–$R5$. Fat arrows separate promises and conclusions of the rules. Double arrow shows that the left-hand side can be mapped (transformed) into the right-hand side.

Using these rules, the following mapping can be inferred:

$$\begin{aligned}
M_{1,2} = {} & authors[author[name = \$x_1 \wedge university = \$x_2 \wedge \\
& \quad paper[title = \$x_3 \wedge year = \$x_4 \wedge \\
& \quad conference[name = \$x_5 \wedge organizer = \$x_6]]]] \Rightarrow \\
& publications[publication[title = \$x_3 \wedge year = \$x_4 \wedge \\
& \quad conference[name = \$x_5 \wedge organizer = \$x_6] \wedge \\
& \quad writer[name = \$x_1 \wedge university = \$x_2]]]
\end{aligned}$$

Considering annotation of XML schema elements with class names, the mapping can be easily translated into the mapping over XML schemas:

$$\begin{aligned}
m_{1,2} = {} & S_1(x_1, x_2, x_3, x_4, x_5, x_6) \Rightarrow S_2(x_3, x_4, x_5, x_6, x_1, x_2) \Leftrightarrow \\
& /authors[author[name = \$x_1 \wedge univ = \$x_2 \wedge \\
& \quad paper[title = \$x_3 \wedge year = \$x_4 \wedge conf[name = \$x_5 \wedge org = \$x_6]]]] \Rightarrow \\
& /pubs[pub[title = \$x_3 \wedge year = \$x_4 \wedge \\
& \quad conf[name = \$x_5 \wedge org = \$x_6] \wedge writer[name = \$x_1 \wedge university = \$x_2]]]
\end{aligned}$$

4 Ontology-Based Query Language – OnBE

In Figure 3 there is an example of query against the ontology *Bib* shown in the left-hand side window.

> Q : *Give names of authors from the Warsaw University who are authors of publications in 2004 and 2005, presented at some conferences. Give titles and years of publications and organizers of these conferences.*

The upper right-hand side window displays a graphical form of so-called *navigational expression*. Such a graph is created automatically while the user choose semantically connected ontological concepts. There are variables associated to datatype properties. The user can write conjunction of conditions in the bottom right-hand window. The conjunct $\$x_2 = "Warsaw"$ restricts authors to those who are from $"Warsaw"$, while the conjunct $\$x_4 = EVERY("2004", "2005")$ says that we are interested in authors for whom there exists at least one path for every value ("2004" and "2005") leading from the author to this value. Semantics of these conditions conform to a subset of `WHERE` condition in XQuery. In fact, the navigational expression together with the conditional part, are translated into XQuery program.

The translation of OnBE expression is carried out in two steps: (1) translation into a *tree-pattern query*, i.e. conjunction of the schema and the query qualifier obtained from the specified conditions; (2) translating of tree-pattern query into XQuery program (using the algorithm proposed in [16]).

The query in Fig 3 is formulated against the ontology of schema S_1 and the answer is expected to conform also to this ontology (schema). So, its form on the schema level is as follows ($x \mapsto v$ denotes replacement of variable name x with v):

$$
\begin{aligned}
&Q_{11} := S_1(x_1, x_2, x_3, x_4, x_5, x_6) \wedge \Phi(x_2 \mapsto v_2, x_4 \mapsto v_4) = S_1(x_1, x_2, x_3, x_4, x_5, x_6) \wedge \\
&\exists v_2 (v_2 \in \{"Warsow"\} \wedge /authors/author[name = x_1][univ = v_2]) \wedge \\
&\forall v_4 (v_4 \in \{"2004", "2005"\} \rightarrow /authors/author[name = x_1]/paper[year = v_4]) \\
&\Rightarrow /authors[author[name = \$x_1 \wedge paper[title = \$x_3 \wedge year = \$x_4 \wedge conf[org = \$x_6]]]]
\end{aligned}
$$

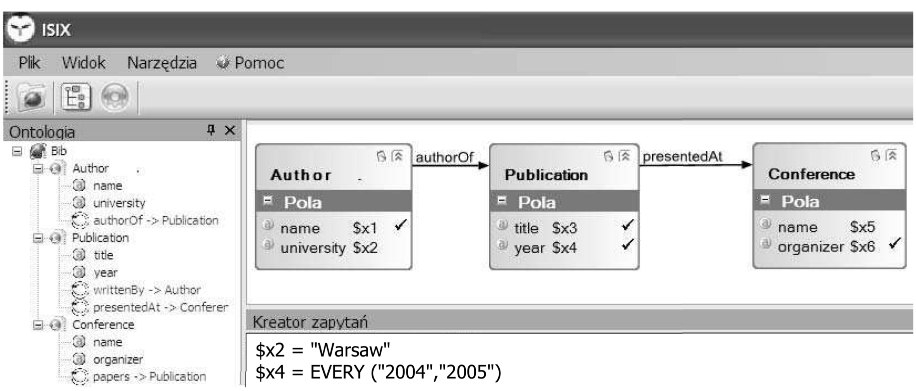


Fig. 3. Querying ontology using OnBE query interface

Query qualifier is a condition constraining the set of paths identified by values of the key of the first object occurring in the navigational expression. In our case *name* is the key of *Author*, so expressions defining paths in our example begin with $/authors/author[name = x_1]$.

The considered query Q_{11} is translated into the following XQuery program:

```
<authors>{
  for $a in /authors/author, $x1 in $a/name,$x2 in $a/univ,$p in $a/paper,
      $x3 in $p/title,$x4 in $p/year, $c in $p/conf, $x6 in $c/org
  where (some $v2 in ("Warsaw") satisfies $a[name=$x1][univ=$v2]) and
     (every $v4 in ("2004","2005") satisfies $a[name=$x1]/paper[year=$v4])
  return
   <author>{$x1}
      <paper>{$x3}{$x4}
        <conf>{$x6}</conf>
      </paper>
   </author>
}</authors>
```

Fig. 4. XQuery program corresponding to the OnBE query in Figure 3

After propagating Q_{11} to a peer with schema S_2, we take into account the mapping m_{12} and use it to obtaining the following reformulated form of Q_{11}, i.e. we obtain the query Q_{21} selecting data from an instance of S_2 and transforming it to an instance of S_1:

$$Q_{21} = S_2(x_1, x_2, x_3, x_4, x_5, x_6) \wedge \Phi(x_6 \mapsto v_6, x_2 \mapsto v_2) \Rightarrow /authors[auhor[\\ name = \$x_3 \wedge paper[title = \$x_1 \wedge year = \$x_2 \wedge conf[org = \$x_6]]]]$$

The reformulation and propagation procedures were deeply discussed in [15].

5 Conclusion

In this paper we proposed an approach to querying data in a system for semantic integration of XML data. In this approach we distinguish the following two main steps: (1) A knowledge engineer annotates locally available XML schemas. The ontology mapped to the schema (xs-ontology) consists of three ontological notions (according to the OWL approach): classes, object properties, and datatype properties. (2) We propose rules for inferring mappings between xs-ontologies and their underlying schemas. (3) The end user formulates queries against the ontology in the ontology-based query language OnBE (*Ontology By Example*). Queries in OnBE are formulated using a graphical interface. (4) OnBE queries are translated into queries in XQuery using mappings stored in the system metadata repository. The basics of OnBE language has been implemented in SixP2P (*Semantic Integration of XML data in P2P*) system [6,16,15].

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading (1995)
2. Arenas, M., Libkin, L.: XML Data Exchange: Consistency and Query Answering. In: PODS Conference, pp. 13–24 (2005)
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Petel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge (2003)
4. Beneventano, D., Bergamaschi, S.: The MOMIS methodology for integrating heterogeneous data sources. In: IFIP Congress Topical Sessions, pp. 19–24 (2004)
5. Benslimane, S.M., Benslimane, D., Malki, M., Amghar, Y., Saliah-Hassane, H.: Acquiring OWL ontologies from data-intensive Web sites. In: ICWE, pp. 361–368 (2006)
6. Brzykcy, G., Bartoszek, J., Pankowski, T.: Schema Mappings and Agents' Actions in P2P Data Integration System. Journal of Universal Computer Science 14(7), 1048–1060 (2008)
7. Calvanese, D., Giacomo, G.D., Lenzerini, M., Rosati, R.: Logical Foundations of Peer-To-Peer Data Integration. In: Proc. of the 23rd ACM SIGMOD Symposium on Principles of Database Systems (PODS 2004), pp. 241–251 (2004)
8. Cybulka, J., Meissner, A., Pankowski, T.: Semantic-driven XML Data Exchange within Web-serviced Business Applications. In: Abramowicz, W., Mayr, H.C. (eds.) Technologies for Business Information Systems, ch. 26, pp. 299–311. Springer, Heidelberg (2007)
9. Eklund, P., Cole, R.: Retrieving and Exploring Ontology-based Information. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 405–414. Springer, Heidelberg (2004)
10. Fagin, R., Kolaitis, P.G., Popa, L., Tan, W.C.: Composing Schema Mappings: Second-Order Dependencies to the Rescue. In: PODS, pp. 83–94 (2004)
11. Fluit, C., Sabou, M., van Harmelen, F.: Supporting User Tasks through Visualisation of Light-weight Ontologies. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 415–434. Springer, Heidelberg (2004)
12. Fuxman, A., Kolaitis, P.G., Miller, R.J., Tan, W.C.: Peer data exchange. ACM Trans. Database Syst. 31(4), 1454–1498 (2006)
13. Klein, M.C.A.: Interpreting XML Documents via an RDF Schema Ontology. In: 13th International Workshop on Database and Expert Systems Applications (DEXA 2002), pp. 889–894. IEEE Computer Society, Los Alamitos (2002)
14. Klein, M.C.A., Broekstra, J., Fensel, D., van Harmelen, F., Horrocks, I.: Ontologies and Schema Languages on the. In: Spinning the Semantic Web, pp. 95–139 (2003)
15. Pankowski, T.: Query propagation in a P2P data integration system in the presence of schema constraints. In: Hameurlain, A. (ed.) Globe 2008. LNCS, vol. 5187, pp. 46–57. Springer, Heidelberg (2008)
16. Pankowski, T.: XML data integration in SixP2P – a theoretical framework. In: EDBT Workshop Data Management in P2P Systems (DAMAP 2008), ACM Digital Library, pp. 11–18 (2008)
17. Xiao, H., Cruz, I.F.: Integrating and Exchanging XML Data Using Ontologies. In: Spaccapietra, S., Aberer, K., Cudré-Mauroux, P. (eds.) Journal on Data Semantics VI. LNCS, vol. 4090, pp. 67–89. Springer, Heidelberg (2006)
18. Yuan, J., Bahrami, A., Wang, C., Murray, M.O., Hunt, A.: A Semantic Information Integration Tool Suite. In: VLDB, pp. 1171–1174 (2006)

Action Generator Based on Primitive Actions and Individual Properties

Jun-Hyuk Lee, Geun-Jae Jung, and Jong-Hee Park

E10-506, Electrical Engineering and Computer and Science Dept., Kyoung-pook Nat'l University, Sankyuk-Dong, Puk-gu, Daegu, South Korea
{yakult4jo, cooky8884, jhpark}@ee.knu.ac.kr

Abstract. This paper presents a method of how to simulate various human actions. Actions are formed in the Action Hierarchy, and are reified in terms of expanding the hierarchy. While inheritance, super classes provide overall information of actions, and sub classes are used as the actual functions of actions. Primitive actions as leaves of the action hierarchy are proposed for an atomic action to be shared in several actions, and used as key frame. The Action Generator guarantees to return accurate primitive actions to an agent based on the Action Library and the Action Hierarchy, and also provides individual patterns of actions.

Keywords: Action Generator, Action Hierarchy, Primitive Action.

1 Introduction

Improving the realism of human actions has become important factors in computer animation. Although the realistic actions in three-dimensional simulation have been developed with techniques in kinematics, dynamics, biomechanics, and robotics[1], most techniques focus only on a single action, do not pursue the diversity of actions[2]. They apply the same formulations to all agents in spite of different individual patterns and a diversity of actions. Moreover, it takes much time and searching space to produce numerous actions, for example, motion capture[3].

To provide a diversity of actions for each agent we analyse human properties and build the Action Hierarchy. For this purpose we propose the system to modulate actions in terms of the Action Generator. The action class is classified into three parts, the precondition, the procedure and the effect[4], and instantiated by various types of agent. The generator is to provide accurate actions from the Action Library or the Action Hierarchy. Primitive actions[5][6], the atomic actions, are also defined in the hierarchy, and shared amongst actions as reusable resources.

In section 2 the Human Class is defined as the foundation of human properties. In section 3 the general formulation of action is described, and how the action class communicates with the human class is introduced. In section 4 the Action Generator as the important part to return an accurate action to an agent is explained. Finally we draw a conclusion on modeling human actions in section 5.

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 844–852, 2009.

2 Human Class

An agent is in general modelled as a synthetic actor composed of parts, e.g. head, trunk, etc, and joints, e.g. neck, etc as below.

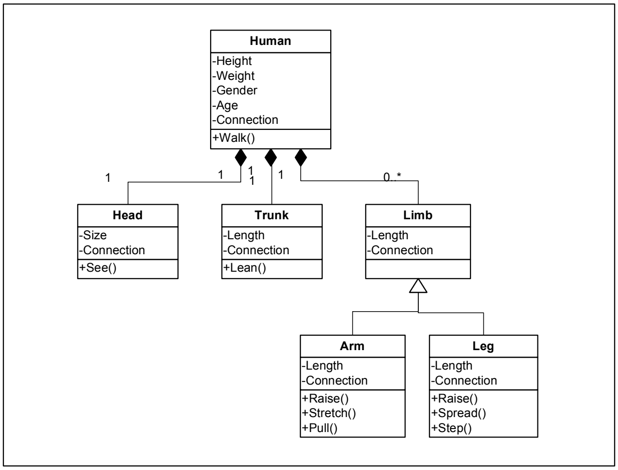


Fig. 1. Class Diagram of Human Class

Each part has its associated actions, for example, a leg alone cannot perform *Walk()*, but can *Raise()* while human as a composite class can perform *Walk()*. The connection describes joints of associated parts, and is linked to the Joint Tag that has the information about joints. Especially, the Joint Tag can express abnormality, i.e. disabilities[7]. Actions in the function slot are formulated in the Action Hierarchy and assigning a proper action is delegated to the Action Generator.

3 Action

Since there are generally many actions that an agent is able to perform, the system should select the best action for a situation. Also actions generated from the system should present individual patterns according to the properties in human class, and diversity as well. An agent is stimulated through sensors, and the sensors collects the information about the environment through the action of *Perceive()*, e.g. *See()* with eyes. In reality the environment is partially observable and stochastically processed because of incomplete information and noise[8]. The natural environment is unpredictable, so, an agent can never predict exactly. However, we assume that an agent can receive the complete information of the environment. The following diagram represents the overall structure of how human, action and environment communicate with each other.

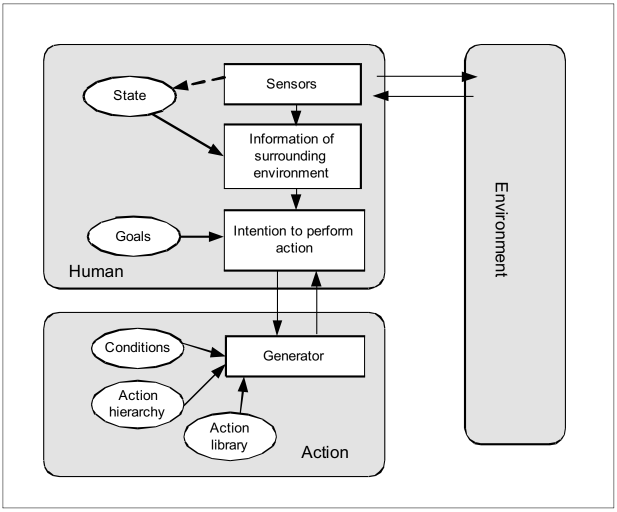


Fig. 2. Schematic Diagram of System

When an agent has a need for an action[9][10], the Action Generator searches for an appropriating action without violation of constraints to produce the needed result. Notice that an action can be initiated only if an agent requests, which is called ' occurrence of action', and the action affects the current environment.

3.1 Action Class

An action is able to be defined as a function with parameters from an agent, and is specified in terms of the roles and <precondition, procedure, and effect>[4][10].

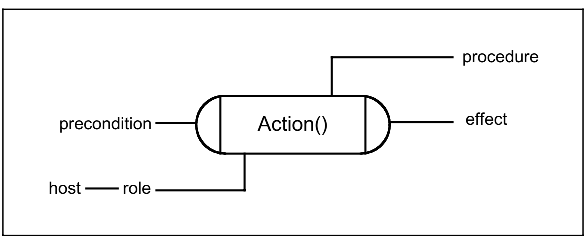


Fig. 3. Structure of Action

The precondition and the effect each could be specified in terms of a comprehensive state as an accumulated result of changes in the situation. The precondition also encompasses multiple roles that assign both hosts and props, e.g. food in *Eat().* The effect could be either intentional effect or side effect[10]. The procedure involves a sequence of changes and optionally motion. It defines corresponding primitive actions and their concatenations, that is, it is accomplished in terms of the decomposition of an action into primitive actions. The decomposition should guarantee not only obtaining primitive actions but also finding out proper sub-actions in composite actions. The table below shows an example of *Walk()* concerned with legs[5].

Table 1. Structure of Walk()

PRECONDITION
Existence of legs
The power to move
Psychological factors
PROCEDURE
Raise leg
Spread leg
Step leg
EFFECT
Movement
Energy consumption

3.2 Primitive Action

An action is specialised in terms of primitive actions, an atomic action[5][6], as leaves in the Action Hierarchy. Each part of human body defines its relevant primitive actions in functions slot[7], for example, *Raise()* in leg as exemplified table 1. Primitive actions are defined as functions, and can be calculated using associated kinematics and dynamics. They are typically simulated as key frames while interpolation techniques are used to generate in-between positions[7]. All primitive actions are calculated by the velocity and the angular velocity using the Jacobian Matrix[5]. An example of parameters of a leg for calculation is as following.

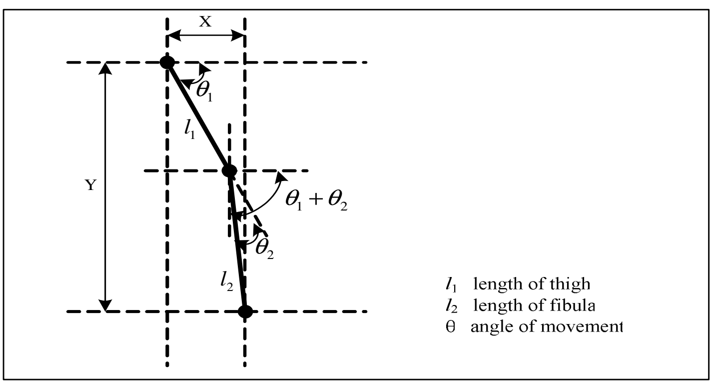


Fig. 4. Parameters of Leg

The most important advantage of using primitive actions is that they are reusable resource. Examples of primitive actions are below.

Table 2. Examples of Primitive Actions

Walk()		*Climb()*		*Jump()*	
Arm	Leg	Arm	Leg	Arm	Leg
Raise()	*Raise()*	*Raise()*	*Raise()*	*Raise()*	*Sit()*
Stretch()	*Spread()*	*Stretch()*	*Spread()*	*Pull()*	*Rise()*
Pull()	*Step()*	*Pull()*	*Step()*		

Walk() and *Climb()* use the same primitive action, *Raise()*, that is, one primitive action can occur across different actions. Conversely, the primitive action, *Raise()* in *Walk()* appears in different parts, arm and leg, however, the primitive action is executed in a different manner. Moreover, primitive actions have priority constraint against others, if they appears in same parts, for example, *Raise(), Spread()* and *Step()* in *Walk()* so that the Action Generator can plan the concatenation of primitive actions. Hence, one primitive action can be involved in various actions, and executed in a different manner.

4 Action Generator

Providing an accurate action to an agent, the Action Generator searches for an appropriate action through the Action Library or the Action Hierarchy. The generator is mainly divided into two parts, Action Library and Action Hierarchy.

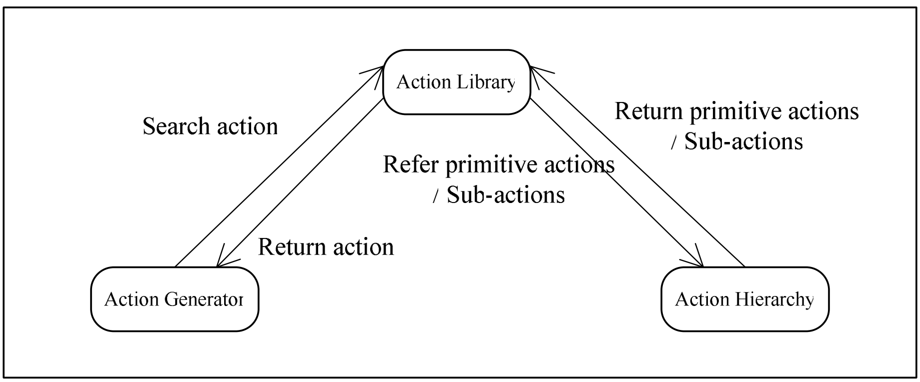


Fig. 5. State Diagram of Action Generator

4.1 Action Library

The general descriptions of actions are stored in the Action Library, from which they are extracted an appropriate action while the Action Hierarchy instantiates all information of actions. The library is structured as

$$\mathbf{T}_i = \langle \text{effect, rank, precondition, } l_i \rangle$$

Where T and l denote a tuple of the library and the location of action i in the Action Hierarchy respectively.

The effect is given from the need of an agent, and the rank shows how many times to be used. For example, *Walk()* is very frequently executed in daily routine rather then *Ski()*, and thus, *Walk()* can get higher rank. After finding a matched action, the library concerns about precondition, that is, the result action should satisfy the triggering condition of the action, for example, existence of leg in table 1. The location of action, *l*, has a pointer to where the action is placed in the Action Hierarchy.

The library monitors the hierarchy so that it maintains updated information in the hierarchy. In case that either a novel action appears or an action requested is not in the library, the library refers to the hierarchy, and such information is updated in the library. Thereupon, the library can maintain the up-to-date information.

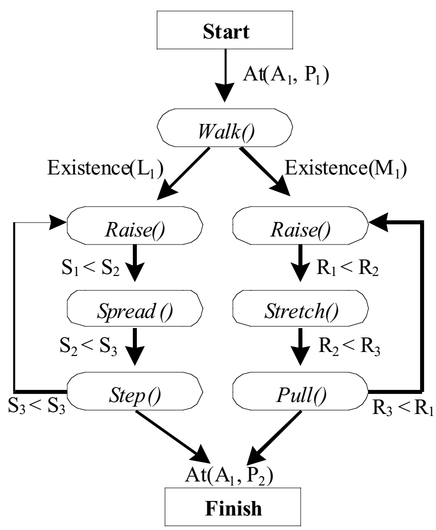
The library is composed of three works;

- Search
- Planning
- Synchronisation

Search

The library attempts to search for an action based on the effect part. If satisfactory, it evaluates other conditions, i.e. rank and precondition. If there is high rank in T_i, the library returns the high ranked action, which means the high ranked action has the

Init (At (Agent(A_1), Place(P_1))
Goal (At (A_1, Place(P_2))
Action (*Raise*,
PRECONDTION: Existence(Leg(L_1))
ORDERING: Raise < Spread),
Action (*Spread*,
PRECONDTION: Existence(Leg(L_1))
ORDERING: Spread < Step),
Action (*Step*,
PRECONDTION: Existence(Leg(L_1))
ORDERING: Step < Raise),
Action (*Raise*,
PRECONDTION: Existence(Arm(M_1))
ORDERING: Raise < Stretch),
Action (*Stretch*,
PRECONDTION: Existence(Arm(M_1))
ORDERING: Stretch < Pull),
Action (*Pull*,
PRECONDTION: Existence(Arm(M_1))
ORDERING: Pull < Raise),
Action (*Move*(Path),
EFFECT: ¬At(A_1, P_1) ^ At(A_1, P_2)
PRECONDITION: At(A_1, P_1))
Action (*Walk*(Path),
EFFECT: ¬At(A_1, P_1) ^ At(A_1, P_2)
PRECONDITION: At(A_1, P_1))



S_1:Raise(), S_2:Spread(), S_3:Step(), R_1:Raise(), R_2:Strech(), R_3:Pull()

Fig. 6. Searching and Planning of Walk()

better chance to be right. In contrast, if the attempt failed or a novel action is required, the hierarchy is invoked to search an accurate action and its result is influenced in the library.

Figure 6 shows how to search for *Walk()* in ADL[8]. The library searches according to the intended effect, A_1 at P_2, and it reaches *Move(),* then evaluates other constraints, A_1 at P_1, and role. If satisfactory, *Walk()* is finally chosen and returns its primitive actions. Further explanation of role is given in section 4.2. Now the library sets the rank to '1' in order for future use and exports the primitive action to synchronisation.

Planning

Planning in general carries out decomposition. If there is a composite action, planning searches for sub actions referring to the Action Hierarchy, then lists the result. For example, if *Tennis()* is requested, planning decomposes the action into sub-actions, *Hit(), Run()* and *Wait()* referring to the Action Hierarchy, then organises them. However, if the required action is not found in the library, planning searches the hierarchy until an appropriate action is reached, then the result is projected on in the library.

Panning also adjusts primitive actions to individual properties, e.g. age, joints, etc. Especially joints are important in case of disabilities[7], and their information influences the whole process of actions, and hence, can provide identical patterns of actions.

Synchronisation

Synchronation is to oragnise primitive actions according to a right timing. The information about timing is in 'ORDERING' of each primitive action. For example, *Raise()* should be executed in advance of *Spread(),* and *Spread()* is performed ahead of *Step()* and so on. This is expressed as

$Start < S_1$

$S_1 < S_2$

$S_2 < S_3$

$S_3 < S_1 \mid End$

However, the primitive actions in legs and the primitive actions in arms can be executed in parallel because they are independent parts.

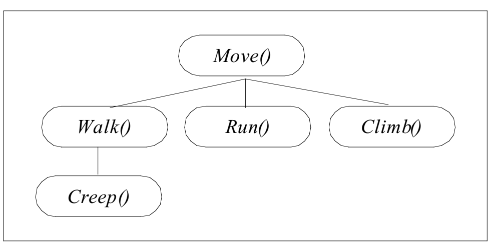


Fig. 7. A Partial Action Hierarchy

4.2 Action Hierarchy

All actions are represented in the ontology, the Action Hierarchy[4], and recursively specialised along with the hierarchy. Every descendent class inherits all the properties of its ancestors, but descendents may override ancestor's actions to specify the changes in the preconditions, procedures and effects of their associated action[10].

In particular roles can expand the hierarchy according to diverse criteria. *Walk(), Run()* and *Climb()* inherits the same properties from *Move()* as the topmost action. However, the role of *Walk()*, path, is the normal terrain while path of *Move()* covers all types of terrains. Therefore, an agent can obtain a proper action as shown in figure 6. If an agent passes the hill, *Climb()* may be chosen because path of *Climb()* is for hill. In addition, *Walk()* can be more specialised in terms of a host, for example, a host of *Creep()* is for baby who cannot toddle yet in contrast to *Walk()*'s covers adults. The advantages of the action hierarch are summarised as below[10].

1. The changes of preconditions, procedures and effects caused by the change of roles can be easily represented through the generalization hierarchy of actions.
2. Actions with same intended effects but with different abstraction level can be organized into the hierarchy.

5 Conclusion

The human class defines the human properties that characterize individual patterns of actions, and communicates with the Action Generator if the need occurs. For providing an accurate action the Action Generator employs the Action Library and the Action Hierarchy. The former contains the general information of the Action hierarchy, and the latter encompasses all information of actions.

Actions are specialised along with the hierarchy and diversified according to role. They can be decomposed into sub-actions by the library. Primitive actions are defined as functions of actions with human properties, and can be reused in various actions. Especially primitive actions have the priority order due to synchronisation.

References

1. Maurel, W., Thalmann, D.: A Case Study on Human Upper Limb Modelling for Dynamic Simulation. Computer Methods in Biomechanics and Biomechanical Engineering 24(2), 203–1158 (1999)
2. Luh, G.-C., Cheng, C.W.C.: Behavior-based intelligent mobile robot using an immunized reinforcement adaptive learning mechanism. Advanced Engineering Informatics 16, 85–98 (2001)
3. Pettre, J., Simeon, T., Laumond, J.P.: Planning human walk in virtual environments, IEEE/RSJ International Conference on, vol. 3, pp. 3048–3053 (2002)
4. Park, J.H.: Activities as the Dynamic Component of Situations, Technical Report, AIMM Lab (2007)
5. Lee, I.K., Park, J.H.: Representation of 'Walk' for Quadruped Animal Based on Primitive Action and Class Inherit. Journal of the Korea Contents Society 6(5), 85–95 (2006)

6. Zheng, G., Li, W., Ogunbona, P., Dong, L., Kharitonenko, I.: Simulation of Human Motion for Learning and Recognition. In: Sattar, A., Kang, B.-h. (eds.) AI 2006. LNCS, vol. 4304, pp. 1168–1172. Springer, Heidelberg (2006)
7. Lee, J.H., Park, J.H.: Modelling Different Patterns of Human Walking based on Individual Properties. In: Proceeding of Applied Simulation and Modelling. ACAT Press (2008)
8. Russell, S., Norvig, P.: Artificial Intelligence, A Modern Approach. Prentice-Hall, Englewood Cliffs (2003)
9. de Sevin, E., Thalmann, D.: A Motivational Model of Action Selection for Virtual Humans, Computer Graphics International (2005)
10. Ji, S.J.: A Knowledge Model for Simulating Human-like Behavior of virtual Inhabitant, The Graduate School of Kyungpook National University (2007)

Control of a Bipedal Walking Robot Using a Fuzzy Precompensator

Soraya Bououden, Foudil Abdessemed, and Benali Abderraouf

Université Hadj Lakhdar,
Rue chahid Boukhlouf, Batna 05000, Algeria
soriabouden@yahoo.fr

Abstract. The Computed Torque Control (CTC) is an effective motion control strategy for a biped walking robot, which can ensure globally asymptotic stability. However, CTC scheme requires precise dynamical models of biped robot. To handle this impossibility, we proposed an approach combing CTC and fuzzy logic controller to regulate the dynamic walking of a planar 7 degrees-of-freedom under-actuated biped robot to follow a specified trajectory. A computed torque is used to achieve high speed and high precision tracking while the fuzzy controller behaves remedies for any parameter deviation compensating thus for unknown uncertainties and disturbances. Finally, computer simulation are carried on and results are presented to show tracking capability and effectiveness of the proposed scheme.

Keywords: Biped Robot, Dynamic Walking, Computed Torque, Fuzzy Precompensator.

1 Introduction

Because a biped's dynamics are multivariable, high order, nonlinear, and time variant, it is difficult to design a controller using traditional techniques. However, many novel approaches have been explored. A survey of the literature review on biped robots control, enables us to emphasize many control approaches, namely those approaches based on predictive control [1], intuitive control [3], optimal Control [21], neural network control [4], observer based control [22], Hybrid zero dynamic [9], Computed torque control [16], and tracking control [5], [17].

Many researchers have shown that fuzzy set theory can solve dynamic biped locomotion problems. [20] proposed a fuzzy logic structure that implemented the walk coordinator as a rule-based system with the ZMP method. In his subsequent works, [18],[19] an evolution learning scheme (genetic algorithm) is used to modify the fuzzy rules and membership functions to generate different walking patterns. In previous efforts [13],[14], a three layered neural network controller with the back-propagation through time (BTT) algorithm [24] can generate robotic walking gaits successfully. [15] proposed a fuzzy controller combined with a linearized inverse biped model. The controller provides the control signals at each control time instant. The algorithm used to train the controller is backprobagation through time. In

A. Håkansson et al. (Eds.): KES-AMSTA 2009, LNAI 5559, pp. 853–862, 2009.

addition, other researchers have made good progresses in the control of biped robots by means of learning techniques such as fuzzy logic control by using fuzzy networks [23].

The objective of this paper is to control a five-link d.o.f under-actuated biped robot using a fuzzy logic controller to follow a specified trajectory. This present interest is largely due to the successful applications of fuzzy logic controller to a variety of industrial systems. Its main components are an inference engine and a set of linguistic *if-then* rules that encode the behavior of the biped.

However, the main difficulty in designing a fuzzy logic controller is the efficient formulation of the fuzzy *if-then* rules. In fact, it is easy to produce the antecedent parts of a fuzzy control rules, but it is very difficult to produce the consequent parts without expert knowledge. Here, however, the genetic algorithm play the role of an expert system and obtain a solution that gives the optimum rule base for the precompensator controller.

Our control approach borrows from previous results for robot manipulator control [10]. The approach reveals a new scheme of robot control using a fuzzy precompensator. This module is inserted within a PD regulator in order to make each joint track a desired trajectory. In this paper, we want to exploit the advantages given by the computed torque method by introducing the fuzzy precompensator in order to achieve high speed and high precision tracking while the mechanical structure is subject to structured and unstructured uncertainties.

The rest of the paper is organized as follows: In the next section the biped robot prototype is described, then the dynamic model is presented in section 3, singling out the appropriate dynamic model for each phase of the walking cycle. In section 4 the control approach is presented. Finally simulation results are given in section 5 to attest the efficiency of the proposed scheme. The paper ends with concluding remarks.

2 The Rabbit Prototype Description

Rabbit is a prototype robot [6],[7] (see Fig. 1) presenting a five-links under-actuated biped robot with 7 d.o.f, and 4 actuators, namely only the femurs and the tibias are actuated. Rabbit is aimed to experiment walking as well as running gaits without feet nor elastic actuators, furthermore it enables easily transitions between gaits.

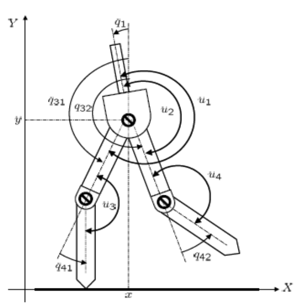


Fig. 1. Mechanical structure of the biped

3 The Dynamic Model

The Lagrange formalism [21] enables the mathematical model describing the biped moving in the sagittal plan to be written as Follows:

$$M(q)\ddot{q} + N(q,\dot{q})\dot{q} + G(q) = Su \tag{1}$$

Where $M(q) \in \Re^{7\times 7}$ is the inertia matrix, $N(q,\dot{q}) \in \Re^{7\times 7}$ contains the centrifugal and Coriolis forces terms, $G(q) \in \Re^{7}$ is the vector of gravitational forces, $u = [u_1, u_2, u_3, u_4]^T \in \Re^4$ is the vector of control inputs, $S \in \Re^{7\times 4}$ is a torque distribution matrix, $q = [q_{31}\ q_{41}\ q_{32}\ q_{42}\ q_1\ x\ y]^T \in \Re^7$ is the vector of generalized coordinates. It is assumed that the walking movements take place in the sagittal plane, and on a horizontal surface without obstacles. The dynamic model of the bipedal robot thus consists of two parts: A differential equation describing the biped in the swing phase, and an algebraic equation describing the impact phase.

3.1 The Single Support Phase

In this phase the biped is modelled by the following differential equation:

$$M(q)\ddot{q} + N(q,\dot{q})\dot{q} + G(q) = Su + J_1^T \lambda \tag{2}$$

where $J_1(q)$ represents the jaccobian matrix of the holonomic contact constraints, and λ the Lagrange multipliers of contact forces. The contact constraints may be expressed by

$$y_{p_1} = \dot{y}_{p_1} = \ddot{y}_{p_1} = 0 \quad ; \; x_{p_1} = \dot{x}_{p_1} = \ddot{x}_{p_1} = 0 \tag{3}$$

where (x_{p_1}, y_{p_1}) denote the cartesian coordinates of the stance leg's foot, namely

$$\begin{aligned} y_{p_1}(q) &= y + l_3\cos(q_{31}) + l_4\cos(q_{31}+q_{41}) \\ x_{p_1}(q) &= x - l_3\sin(q_{31}) - l_4\sin(q_{31}+q_{41}) \end{aligned} \tag{4}$$

Using (4) in (3) leads to

$$J_1(q)\ddot{q} + \pi_2(q,\dot{q}) = 0 \tag{5}$$

Where $\pi_2(q,q)$ is defined as

$$\pi_2(q,q) = \begin{pmatrix} -l_3\dot{q}_{31}^2\cos(q_{31}) - l_4(\dot{q}_{31}+\dot{q}_{41})^2\cos(q_{31}+q_{41}) \\ l_3\dot{q}_{31}^2\sin(q_{31}) + l_4(\dot{q}_{31}+\dot{q}_{31})^2\sin(q_{31}+q_{41}) \end{pmatrix}$$

The constrained dynamic model in the Single Support phase is then given by:

$$\begin{aligned} &M(q)\ddot{q} + N(q,\dot{q})\dot{q} + G(q) = Su + J_1^T(q)\lambda \\ &J_1(q)\ddot{q} + \pi_2(q,\dot{q}) = 0 \end{aligned} \tag{6}$$

3.2 The Impact Model

According to [12]. The impact between the swing leg and the ground is considered as a rigid collision [2], it occurs when the swing leg hits the ground. Our objective is to derive the velocities just after the impact in terms of positions and velocities just before the impact. The impact model proposed in [12] is used here. The expressions of the velocities $\dot{q}^+$ juste after impact, and the contact forces are given by

$$\begin{cases} \dot{q}^+ = [I - M^{-1}J_2^T(J_2M^{-1}J_2^T)^{-1}J_2]\dot{q}^- \\ \lambda = [(J_2M^{-1}J_2^T)^{-1}J_2]\dot{q}^- \end{cases} \tag{7}$$

On the other hand, the impact model must account for the relabelling of the robot coordinates (i.e the swing leg becomes the new stance leg and vice versa) this can be expressed by $q^+ = Rq^-$. To summarise, the global impact model that includes both the jumps in velocities and the permutation of the coordinates shortly writes

$$\begin{pmatrix} q^+ \\ \dot{q}^+ \end{pmatrix} = \Delta(q)\begin{pmatrix} q^- \\ \dot{q}^- \end{pmatrix} \tag{8}$$

4 The Fuzzy Modeling Scheme

The controller build around the system is shown in Fig. 2 where we can see the computed torque controller, fuzzy precompensator [11] and the robot mechanical structure. The output of the precompensator is considered now as a new reference input to the computed torque-plant system. The design of the computed torque involves decomposing the control design problem into an inner-loop design and an outer-loop design. It is important to realise that the computed torque method depends on the inversion of the biped robot dynamics, and is sometimes called inverse dynamics control. The block diagram is shown in Fig. 2.

The purpose of the nonlinear inner-loop is to perform feedback linearization of the robotic system. Let

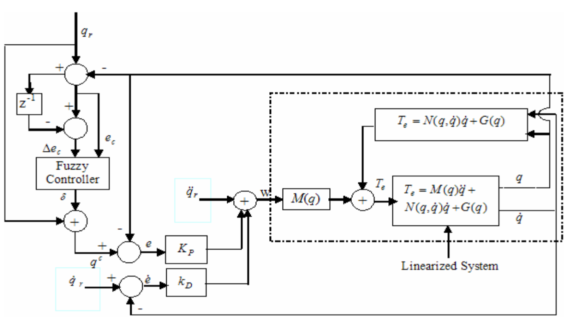


Fig. 2. Diagram simulation of the controlled system

$$M(q)w + N(q,\dot{q})\dot{q} + G(q) = T_e \tag{9}$$

This control eliminates the nonlinearities in the system. Therefore, by substituting (9) into (1), and assuming that $M(q)$ is non singular, we obtain

$$\ddot{q} = w \tag{10}$$

where w is the new input control vector.

The dynamics of the precompensator, computed torque controller is explained as follows: the two inputs to the PD controller are $e_i(k)$ and $\dot{e}_i(k)$; where:

$$e_i(k) = q_i^c(k) - q_i(k) \tag{11}$$

is the trajectory tracking error vector and its first-order derivative is

$$\dot{e}_i(k) = \dot{q}_{ri}(k) - \dot{q}_i(k), \quad i=1,\ldots,7 \text{ refer to the ith link.} \tag{12}$$

Note that the desired angular position value is not directly compared to the measured one, but passes first through the precompensator to be transformed after that to a new reference angular value to the computed torque-plant system. In fact we can write:

$$q_i^c(k) = q_{ri}(k) + \delta_i(k) \quad i=1,\ldots,7 \tag{13}$$

where: the controller output $\delta_i(k)$ is generated by a nonlinear mapping function F, implemented using a fuzzy logic such that:

$$\delta(k) = \delta(k-1) + F[e(k), \Delta e(k)] \tag{14}$$

where

$$\begin{aligned} e_{ci}(k) &= q_{ri}(k) - q_i(k) \\ \Delta e_{ci}(k) &= e_{ci}(k) - e_{ci}(k-1) \end{aligned} \quad i=1,\ldots,7 \tag{15}$$

The model (11) represents a set of n=7 decoupled double integration systems, each one of which can be controlled by a suitable linear control law.

A useful decoupled control law is the proportional plus derivative control which has the form

$$w_i(k) = \ddot{q}_{ri} + k_{P_i} e_i(k) + k_{D_i} \dot{e}_i(k), \quad i=1,\ldots,7 \tag{16}$$

where k_p, k_D are the controller gains. To obtain a critically damped closed loop performance, one must select:

$$\begin{aligned} k_D &= diag\,[k_{Di}] = diag\,[2\lambda], \\ k_p &= diag\,[k_{pj}] = diag\,[\lambda^2], \end{aligned}$$

where λ is the desired bandwidth.

The precompensator is a fuzzy controller, which uses a rule-based expert in the form of if-then statements [8]. There exist many approach to the derivation of the rules. In our case we have used the rules given in [10], since it was proved their generality to control any mechanical system. As it is shown in Table 1.

The following linguistic variables and their associated memberships functions are used (see Fig. (3));

Table 1. Fuzzy logic rules for precompensator

Δe \ e	NL	NM	NS	ZE	PS	PM	PL
NLd	0.614	-0.70	0	0	0	-8.94	-1.16
NMd	0	-0.54	0	-0.14	0	-1.54	0
NSd	0	0.812	-0.46	0	0.847	0.349	0
ZEd	0	-0.55	0	0	0.308	0	0.34
PSd	0	0	-0.60	0	0.605	0	0
PMd	0	-0.91	0.828	0	0.487	0.555	0
PLd	0.717	0	0	0	0	0	0.61

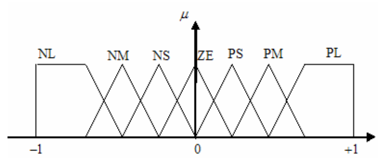


Fig. 3. Membership functions

There are also many ways for performing defuzzification. The strategy adopted here is the height defuzzification method given by:

$$\delta(k) = C_F \frac{\sum_{i=1}^{49} \mu(\gamma_i)\gamma_i}{\sum_{i=1}^{49} \mu(\gamma_i)} \tag{17}$$

where γ_i is the support value at which the membership function reaches the maximum value. C_F is the scaling factor, defined for the three universes of discourse as follows:

$$C_{Fe} = 3.15, \quad C_{F\Delta e} = 5, \quad C_{Fu} = 10$$

4.1 Reference Trajectory

There are several methods to determine the reference trajectory for the walking robot. The human's walking pattern is most often used. In the walking phase, the aim is to let the robot to make a step forward. We define the generalized coordinates $q = [q_{31} \ q_{41} \ q_{32} \ q_{42} \ q_1 \ x \ y]^T$ as cubic spline interpolation in order to specify initial, final and intermediary configurations for the biped as well as its initial and final speeds. These reference trajectories allow the walking robot to start from some initial configuration of the legs defined by:

$$q_{31}(0) = q_{310}; \ q_{41}(0) = q_{410}; \ q_{32}(0) = q_{320}; \ q_{42}(0) = q_{420}; q_1(0) = q_{10}\ ; x(0); y(0).$$

to a final symmetrical configuration defined by:

$$q_{31}(tf) = q_{320}; \ q_{41}(tf) = q_{420}; \ q_{32}(tf) = q_{310}; \ q_{42}(tf) = q_{410}; q_1(tf); x(tf); y(tf).$$

Let *tf* denote the time of one walking step.

To reproduce a human gait we impose a trajectory of inverted pendulum to the support leg while the swing leg follows an opposite movement. To generate a periodic motion we choose the initial conditions of each phase as the final conditions of the previous one.

5 Illustrative Simulations

Consider the biped robot model (1) with the parameters summarized in table 2

Table 2. The model parameters

Parameter	mass	length	inertia
torso	*20 Kg*	*0.625 m*	*2.22 Kg.m²*
femur	*6.8 Kg*	*0.4 m*	*1.08 Kg.m²*
tibia	*3.2 Kg*	*0.4 m*	*0.93 Kg.m²*

The position vector $q=[q_{31}\ q_{41}\ q_{32}\ q_{42}]^T$ is defined by three simple parameters, namely y, d and ρ Indeed, simple computations give:

$$\begin{cases} q_{31} = \pi - \arctan(\rho d / y) - \varphi_3 \\ q_{41} = \pi - \varphi_{34} \\ q_{32} = \pi + \arctan(\dfrac{d(1-\rho)}{y}) - \varphi_3 \\ q_{42} = \pi - \varphi_{34} \end{cases} , \varphi_{34} = \arccos(\frac{-((1-\rho)^2 d^2 + y^2) + l_3^2 + l_4^2}{2 l_3 l_4}) \quad \varphi_3 = \arctan(\frac{y}{d(1-\rho)}) \tag{18}$$

5.1 Simulation: Cyclic Walking Starting from Rest

The aim of this simulation is to take the robot from a rest position to a constant speed periodic walking. For a mean walking velocity of ($0.4 ms^{-1}$) the control design parameters are chosen as follows: $t_f = 0.5(s)$, $y = 0.755$, $d = 0.2(m)$, $\rho = 0.45$, $\lambda = 200$. The sampling time $Tau = 10^{-3}(s)$, the initial conditions on the torso are: $q_5(0) = 9^\circ$, $q_5(tf) = 0$, $\dot{q}_5(t_f) = 0.$, the chosen of the angle of the stance leg tibia at the middle time of the SS phase is $q_{42}(tf/2) = 0.70(rad)$. the chosen configuration of the

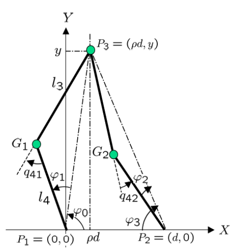


Fig. 4. Computation scheme for the position's reduced parameterization

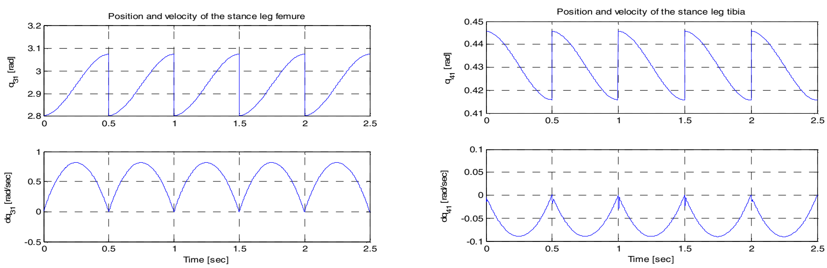


Fig. 5. The Positions, velocities of the stance leg femure and tibia

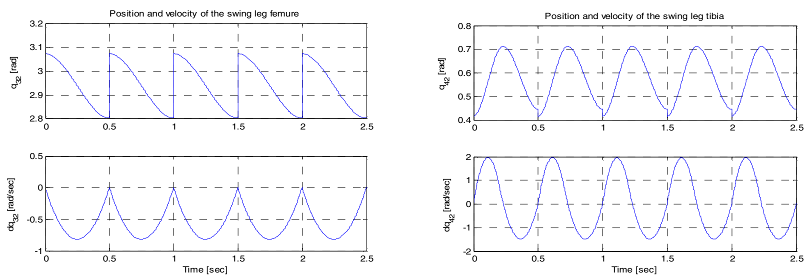


Fig. 6. The Position, velocities of the swing leg femure and tibia

biped is depicted in figure 1. We present some simulation results obtained for 7 degrees of freedom biped. The aim is to make the biped walking for 5 steps by alternating the two legs at a fixed mean velocity of (0.4m\s) The obtained results in this case are summarized by the following: The positions, velocities of the stance leg limbs are illustrated in figure 5, whereas those of the swing leg are depicted in figure 6. The input torques are plotted in figure 7 for the femurs, and figure 8 for tibias. Figure 9 illustrates the movement of the robot by means of a set of walking stick figures. The errors between measured and desired position values obtained by the controlled system are illustrated in figure 10. We remark that the controller succeeds in correcting the errors by driving each joint variable to its corresponding reference. Notice that the

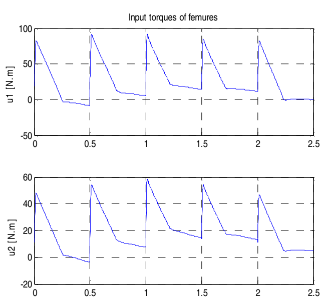


Fig. 7. The femures torques

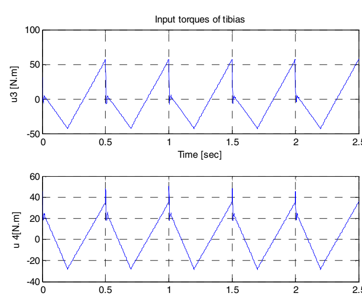


Fig. 8. The tbias torques

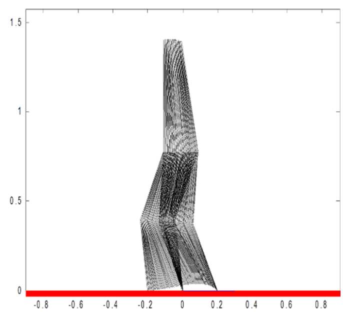


Fig. 9. The stick figures of the robot walking

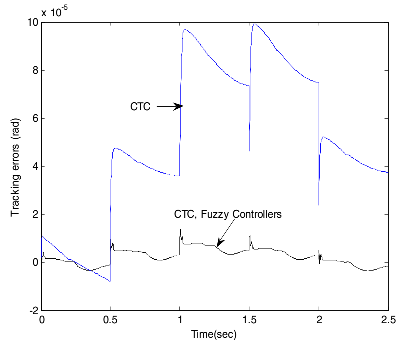


Fig. 10. Tracking errors

torso started with a non zero position and ended with a zero one, making the gait as normal as possible.

6 Conclusion

In this paper, we proposed an approach combining CTC and fuzzy precompensator controller to regulate the dynamic walking of a planar 7 degrees-of-freedom under-actuated biped robot to follow a specified trajectory. A computed torque is used to achieve high speed and high precision tracking while the fuzzy controller intervenes to compensate for uncertainties. The results obtained show that fuzzy theory is a good tool to system control when it is well handled. We have used it here to build a fuzzy precompensator in order to provide the right tracking trajectory of a robot.

References

1. Chemori, A., Amir, M.: Generation of Multi-steps limit cycles for Rabbit using a low dimensional nonlinear predictive control scheme. In: International Conference on Intelligent Robots and Systems, IEEE/RSJ IROS, Sendai, Japan (2004)
2. Brogliato, B.: Nonsmooth mechanics. Models dynamics and control. LNCIS, vol. 220. Springer, Heidelberg (1996)
3. Sabourin, C., Bruneau, O., Fontain, J.G.: Control strategy for the dynamic walking of a bipedal robot based on passive and active stages. In: 12th Inter. Symp. On Measur. and Control in Robotics, ISMCR (2002)
4. Sabourin, C., Bruneau, O.: Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks. Robotics and Autonomous Systems 51, 81–99 (2005)
5. Chevallereau, C., Formal'sky, A., Perrin, B.: Low energy cost reference trajectories for a biped robot. In: Proc. IEEE Conf. Robotics Automat. Leuven, Belgium, pp. 1398–1404 (1998)
6. Chevallereau, C., Sardin, P.: Design and actuation optimisation of a 4 axes biped robot for walking and running. In: Proc. IEEE Conf. Robotics Automat., Leuven, Belgum, pp. 3365–3370 (2000)

7. Chevallereau, C., Abba, G., Aoustin, Y., Plestan, F., Westervelt, E.R., Canudasde, C., Wit Grizzle, J.W.: Rabbit: A testbed for advanced control theory. IEEE Control System Magazine 23(5), 57–79 (2003)
8. Mandani, E.H., Assilian, S.: Application of fuzzy algorithms for control of simple dynamic plant. In: Proc. of the Institute of Electrical Engineers, vol. 121, pp. 585–1588 (1974)
9. Westervelt, E.R., Grizzle, J.W., Koditschek, D.E.: Hybrid zero dynamics of planar biped walkers. IEEE Trans. On Automat. Contr. 48(1), 42–56 (2003)
10. Abdessemed, A., Benmahammed, K.: A two layer robot controller design using evolutionary algorithm. Journal of intelligent and Robotic Systems 30, 73–94 (2001)
11. Abdessemed, A., Benmahammed, K.: Soft Computing based Mobile Manipulator Controller Design. Industrial Robotics Therory Modelling Control, ARS/PVI, Germany, December 2006, p. 964 (2006) ISBN 3-86611-284-x
12. Hurmuzlu, Marhitu, D.B.: Rigid body collisions of planar kinematics chains with multiple contact points. Int. J. Rob Research 13(1), 82–92 (1994)
13. Juang, J.G., Lin, C.S.: Gait synthesis of a biped robot using back-propagation through time algorithm. In: Proc. IEEE Int. Conf. Neural Networks, vol. 3, pp. 1710–1715 (1996)
14. Juang, J.G.: Intelligent path training of a five-link walking robot. In: Proc. IEEE Int. Symp. Intelligent Control, pp. 1–6 (1996)
15. Juang, J.G.: Fuzzy Neural Network approaches for Robotic Gait Synthetics. IEEE Transactions on systems. MNA, and cybernetics-part B cybernetics 30(4) (August 2000)
16. Park, J.H., Kim, K.D.: Biped robot walking using gravity compensated inverted pendulum mode and computed torque control. In: Proc. IEEE International Conference on Robotics Automat., Leuven Belgium (1998)
17. Roussel, L., Canudas, C., Goswami, A.: Generation of energy optimal complete gait cycles for biped robots. In: Proc. IEEE Conf. Robotics Automat., Leuven, Belgium, pp. 2036–2041 (1998)
18. Magdalena, L., Monasterio, F.: Evolutionary-based learning applied to fuzzy controllers. In: Proc. IEEE Int. Conf. Fuzzy Systems, vol. 3(4), pp. 1111–1118 (1995)
19. Magdalena, L., Monasterio, F.: Learning gait patterns for the fuzzy synthesis of biped walking. In: Proc. Int. Joint Conf. NAFIPS, IFIS and NASA, pp. 248–250 (1994)
20. Magdalena, L., Monasterio, F.: Fuzzy controlled gait synthesis of a biped walking machines. In: Proc. IEEE Int. Conf Fuzzy Systems, vol. 2, pp. 1334–1339 (1993)
21. Hardt, M., Kreutz-Delgado, K., Helton, J.: Minimal energy control of a biped robot with numerical methods and recursive symbolic model. In: Proc. 37th IEEE Conf. Decision Contr., Florida, USA, pp. 413–416 (1998)
22. Lebastard, V., Aoustin, Y., Plestan, F.: Observer-Based Control of a biped robot. In: Fourth International Workshopon Robot Motion and Control, June 17-20 (2004)
23. Murakami, S., Yamamoto, E., Fujimoto, K.: Fuzzy control of Dynamic biped walking robot. In: IEEE International Conference on Fuzzy Systems (1995)
24. Werbos, P.: Backpropagation through time: What it does and how to do it. Proc. IEEE 78, 1550–1560 (1990)

Author Index