LNAI 4476

Vladimir Gorodetsky Chengqi Zhang
Victor A. Skormin Longbing Cao (Eds.)

Autonomous Intelligent Systems: Agents and Data Mining

Second International Workshop, AIS-ADM 2007
St. Petersburg, Russia, June 2007
Proceedings

Springer

Lecture Notes in Artificial Intelligence 4476

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Vladimir Gorodetsky   Chengqi Zhang
Victor A. Skormin   Longbing Cao (Eds.)

Autonomous Intelligent Systems: Agents and Data Mining

Second International Workshop, AIS-ADM 2007
St. Petersburg, Russia, June 3-5, 2007
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Vladimir Gorodetsky
St. Petersburg Institute for Informatics and Automation
39, 14-th Liniya, St. Petersburg, 199178, Russia
E-mail: gor@mail.iias.spb.su

Chengqi Zhang
Longbing Cao
University of Technology
Sydney Broadway, Sydney NSW 2007, Australia
E-mail: {chengqi,lbcao}@it.uts.edu.au

Victor A. Skormin
US Air Force, Binghamton University (SUNY)
Binghamton, NY 13902, USA
E-mail: vskormin@binghamton.edu

Library of Congress Control Number: 2007927313

CR Subject Classification (1998): I.2, H.2.8, H.4, H.3, C.2.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-72838-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-72838-2 Springer Berlin Heidelberg New York

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

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2007
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12071050 06/3180 5 4 3 2 1 0

Preface

Since early 1990, multi-agent systems (MAS), data mining, and knowledge discovery (KDD) have remained areas of high interest in the research and development of intelligent information technologies. Indeed, MAS offers powerful metaphors for information system conceptualization, a range of new techniques, and technologies specifically focused on the design and implementation of large-scale open distributed intelligent systems. KDD also provides intelligent information technology with powerful ideas, algorithms, and software means to help cope with the main problem of artificial intelligence, formulated in the well-known question "Where does the knowledge come from?", thus actually making modern applications intelligent and adaptive.

The evident recent trend in both science and industry is to integrate and take advantage of both technologies. The existing experience with combined application of multi-agent technology to design architectures of distributed (hierarchical and peer-to-peer) data mining and KDD systems, as well as the utilization of data mining and KDD achievements to provide enhanced intelligence of MAS, confirms the fact that both technologies are capable of mutual enrichment and their integrated use may result in intelligent information systems with new emergent properties. The 1st International Workshop "Autonomous Intelligent Systems: Agents and Data Mining" (AIS-ADM 2005, June 6–8, 2005, St. Petersburg, Russia) was a response to the aforementioned trend. It confirmed the interest of academic and industry communities in advancing the efforts to integrate achievements in MAS and KDD, thus resulting in a new dimension and further progress in intelligent information technology.

This volume contains the papers presented at the 2nd International Workshop "Autonomous Intelligent Systems: Agents and Data Mining" (AIS-ADM 2007, June 3–5, 2007, St. Petersburg, Russia). Like the first workshop, it was organized by the St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS) in cooperation with the St. Petersburg Scientific Center of the Russian Academy of Sciences and Binghamton University (SUNY, USA).

The workshop objective was to discuss current state of the art in this integrated area of research, new problems, challenges, future trends, as well as the mutual impact and contribution of both technologies, KDD and MAS, in the cross-boundary integrated research area. Workshop organizers have also realized the importance of gathering together not only specialists working in the above-mentioned cross-boundary, but also specialists working in either data mining or in MAS in order to contribute to the creation of a productive research community.

AIS-ADM 2007 provided an international forum of multi-agent and data mining researchers. A total of 39 papers from 18 countries related to various aspects of both theory and applications of MAS, data mining, and their cross-boundary area were submitted to AIS-ADM 2007. Out of these 17 were selected for regular presentations and 6 papers were selected for short presentations. Three technical sessions were organized, namely: 1. Agent and Data Mining; 2. Agent Competition and Data Mining, and 3. Text Mining, Semantic Web, and Agents. The AIS-ADM 2007 program was enriched by four distinguished invited speakers: Hillol Kargupta, Vladimir Khoroshevsky, Sandip Sen, and Ning Zhong.

The success of the workshop was assured by team efforts from the sponsors, organizers, reviewers, and participants. We would like to acknowledge the contribution of the individual Program Committee members and thank the paper reviewers. Our sincere gratitude goes to the participants and all authors of the submitted papers.

We are grateful to our sponsors: The European Office of Aerospace Research and Development (EOARD) of the US Air Force, USA Office of Naval Research Global (ONRGlobal), US Army International Technology Center-Atlantic Research Division, and the Russian Foundation for Basic Research (RFBR) for their generous support.

We wish to express our gratitude to the Springer team managed by Alfred Hofmann for their help and cooperation.

June 2007

Vladimir Gorodetsky
Chengqi Zhang
Victor Skormin
Longbing Cao

Organization

General Chairs

Rafael M. Yusupov (SPIIRAS, Russia)
Victor Skormin (AFRL/IF, Binghamton University, USA)

Program Committee Chairs

Vladimir Gorodetsky (SPIIRAS, Russia)
Chengqi Zhang (University of Technology, Sydney, Australia)
Victor Skormin (AFRL/IF, Binghamton University, USA)
Longbing Cao (University of Technology, Sydney, Australia)

Program Committee

Shlomo Berkovsky (University of Haifa, Israel)
Sviatoslav Braynov (University of Illinois at Springfield, USA)
Cory Butz (University of Regina, Canada)
Wei Dai (Victoria University, Australia)
Nathan Griffiths (University of Warwick, UK)
Heikki Helin (TeliaSonera, Finland)
Henry Hexmoor (Southern Illinois University, USA)
Xiaohua (Tony) Hu (Drexel University, Philadelphia, USA)
Seunghyun Im (University of Pittsburgh at Johnstown, USA)
Xiaolong Jin (University of Bradford, UK)
Oleg Karsaev (SPIIRAS, Russia)
Kristian Kersting (Massachusetts Institute of Technology, USA)
Vladimir Khoroshevsky (Computer Center of the Russian Academy of Sciences, Russia)
Matthias Klusch (DFKI, Germany)
Daniel Kudenko (University of York, UK)
Mikko Laukkanen (TeliaSonera, Finland)
Jiming Liu (University of Windsor, Canada)
Michael Luck (King's College London, UK)
Simon Miles (King's College London, UK)
Vladimir Marik (Czech Technical University in Prague, Czech Republic)
Pericles Mitkas (Aristotle University of Thessaloniki, Greece)
Joerg Mueller (Technische Universität Clausthal, Germany)
Stephen Munroe (University of Southampton, UK)
Mircea Negoita (Pertronic Industries Ltd., New Zealand)
Hung Son Nguyen (Warsaw University, Poland)

Ngoc Thanh Nguyen	(Wroclaw University of Technology, Poland)
Eugenio Oliveira	(University of Porto, Portugal)
Leonid Perlovsky	(Harvard University, Air Force Research Laboratory, USA)
Zbigniew Ras	(University of North Carolina, USA)
Michele Sebag	(CNRS — Universite Paris-Sud Orsay, France)
Andrzej Skowron	(Institute of Mathematics, Warsaw University, Poland)
Zhong Zhi Shi	(Institute for Computer Technology, China)
Alexander Smirnov	(SPIIRAS, Russia)
Katia Sycara	(Carnegie Mellon University, USA)
Andreas Symeonidis	(Aristotle University of Thessaloniki, Greece)
Boleslaw Szymanski	(Rensselaer Polytechnic Institute, NY, USA)
Huaglory Tianfield	(Glasgow Caledonian University, UK)
Santtu Toivonen	(VTT Technical Research Centre, Finland)
Ran Wolff	(Haifa University, Israel)
Philipp Yu	(IBM T.J. Watson Research Center, USA)
Nikolay Zagoruiko	(Sobolev Institute of Mathematics SD RAS, Russia)
Yanqing Zhang	(Georgia State University, USA)
Wen-Ran Zhang	(Georgia Southern University, USA)
Zili Zhang	(Deakin University, Australia)
Ning Zhong	(Maebashi Institute of Technology, Japan)
Hai Zhuge	(Chinese Academy of Sciences, China)

Reviewers

LShlomo Berkovsky	(University of Haifa, Israel)
Sviatoslav Braynov	(University of Illinois at Springfield, USA)
Cory Butz	(University of Regina, Canada)
Wei Dai	(Victoria University, Australia)
Vladimir Gorodetsky	(SPIIRAS, Russia)
Nathan Griffiths	(University of Warwick, UK)
Heikki Helin	(TeliaSonera, Finland)
Alexander Hornung	(Clausthal University of Technology, Germany)
Seunghyun Im	(University of Pittsburgh at Johnstown, USA)
Xiaolong Jin	(University of Bradford, UK)
Oleg Karsaev	(SPIIRAS, Russia)
Kristian Kersting	(Massachusetts Institute of Technology, USA)
Vladimir Khoroshevsky	(Computing Center of the Russian Academy of Sciences, Russia)
Daniel Kudenko	(University of York, UK)
Mikko Laukkanen	(TeliaSonera, Finland)
Jiming Liu	(University of Windsor, Canada)
Michael Luck	(King's College London, UK)
Simon Miles	(King's College London, UK)
Joerg Mueller	(Technische Universität Clausthal, Germany)
Stephen Munroe	(University of Southampton, UK)

Mircea Negoita	(Pertronic Industries Ltd., New Zealand)
Hung Son Nguyen	(Warsaw University, Poland)
Eugenio Oliveira	(University of Porto, Portugal)
Zbigniew Ras	(University of North Carolina, USA)
Andrzej Skowron	(Institute of Mathematics Warsaw University, Poland)
Zhong Zhi Shi	(Institute for Computer Technology, China)
Alexander Smirnov	(SPIIRAS, Russia)
Andreas Symeonidis	(Aristotle University of Thessaloniki, Greece)
Boleslaw Szymanski	(Rensselaer Polytechnic Institute, NY, USA)
Huaglory Tianfield	(Glasgow Caledonian University, UK)
Santtu Toivonen	(VTT Technical Research Centre, Finland)
Ran Wolff	(Haifa University, Israel)
Philipp Yu	(IBM T.J. Watson Research Center, USA)
Nikolay Zagoruiko	(Sobolev Institute of Mathematics SD RAS, Russia)
Yanqing Zhang	(Georgia State University, USA)
Wen-Ran Zhang	(Georgia Southern University, USA)
Zili Zhang	(Deakin University, Australia)
Hai Zhuge	(Chinese Academy of Sciences, China)

Sponsors

European Office of Aerospace Research & Development (EOARD) of the U.S. Air Force
U.S. Office of Naval Research Global (ONRGlobal)
U.S. Army International Technology Center-Atlantic Research Division
Russian Foundation for Basic Research (RFBR)

Table of Contents

Invited Talks

Agent and Data Mining

Peer-to-Peer Data Mining, Privacy Issues, and Games

Kanishka Bhaduri, Kamalika Das, and Hillol Kargupta*

Computer Science and Electrical Engineering Department
University of Maryland, Baltimore County
Baltimore, MD-21250, USA
{kanishk1,kdas1,hillol}@cs.umbc.edu

Abstract. Peer-to-Peer (P2P) networks are gaining increasing popularity in many distributed applications such as file-sharing, network storage, web caching, searching and indexing of relevant documents and P2P network-threat analysis. Many of these applications require scalable analysis of data over a P2P network. This paper starts by offering a brief overview of distributed data mining applications and algorithms for P2P environments. Next it discusses some of the privacy concerns with P2P data mining and points out the problems of existing privacy-preserving multi-party data mining techniques. It further points out that most of the nice assumptions of these existing privacy preserving techniques fall apart in real-life applications of privacy-preserving distributed data mining (PPDM). The paper offers a more realistic formulation of the PPDM problem as a multi-party game and points out some recent results.

1 Introduction

Peer-to-peer (P2P) systems such as Gnutella, Napster, e-Mule, Kazaa, and Freenet are increasingly becoming popular for many applications that go beyond downloading music without paying for it. P2P file sharing, P2P electronic commerce, and P2P monitoring based on a network of sensors are some examples. Novel data integration applications such as P2P web mining from the data stored in the browser cache of different machines connected via a peer-to-peer network may revolutionize the business of Internet search engines. A peer-to-peer clustering algorithm that clusters the URL-s visited by each user (with due privacy-protection) in to different subjects by exchanging information with other peers can be very useful for discovering web-usage patterns of users. This may help characterizing each user based on their browsing pattern, and forming clique of peers having similar interest. Also, this may help routing query about a particular topic to the most appropriate peer in a P2P network. There can be many other similar interesting information integration and knowledge discovery applications involving data distributed in a P2P network.

Privacy is an important issue in many of these P2P data mining applications. Privacy-preserving data mining offers many challenges in this domain. The algorithms must scale up to very large networks and must be asynchronous. Moreover, many of the assumptions (e.g. semi-honest, abides by the protocol) that existing privacy-preserving

* The author is also affiliated with Agnik LLC, MD, USA.

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 1–10, 2007.
© Springer-Verlag Berlin Heidelberg 2007

data mining algorithms make may not be valid. We may have some peers trying to sabotage the computation. This paper presents a high-level overview of an effort to address some of these problem using game theoretic framework for privacy-preserving data mining.

The rest of the paper is organized as follows. In the next section (Section 2), we present the related work on P2P computing and PPDM. A web-mining application that motivates the need for game theoretic PPDM algorithms is presented next in Section 3. The next few sections are devoted on describing the game theoretic approach for PPDM and the preliminary results. We conclude the paper in Section 6.

2 Related Work

This section presents a very brief related work both on data mining and privacy preserving techniques in P2P networks.

2.1 Data Mining in P2P Networks

Knowledge discovery and data mining from P2P network is a relatively new field with little related literature. Some researchers have developed several different approaches for computing basic operations (*e.g.* average, sum, max, random sampling) on P2P networks, *e.g.* Kempe *et al.* [8], Boyd *et al.* [4], Jelasity and Eiben [9] and Bawa et al. [3]. Mehyar *et al.* [15] proposed a new approach for averaging on a P2P network using the Laplacian of a graph.

All the approaches mentioned so far require resources that scale directly with the size of the system. For more scalable approaches, researchers explored the paradigm of *local* algorithms for doing data mining in P2P network. *Local* algorithms [12,16,2,11,10] are ones in which the result is usually computed using information from just a handful of nearby neighbors. Still, it is possible to make definite claims about the correctness of the result. These algorithms are very scalable as resource requirements are independent of the size of the system and a good fit for P2P networks spanning millions of peers. Lately, simple thresholding based *local* algorithms have been used for complicated data mining tasks in P2P systems: majority voting [19], L2 norm thresholding [18] and possibly more. For a detailed survey interested readers are urged to look into [5].

2.2 Privacy Preserving Data Mining

Privacy-preserving data mining can be roughly divided into two groups: data hiding and rule hiding. The main objective of data hiding is to transform the data or to design new computation protocols so that the private data still remains private during and/or after data mining operations; while the underlying data patterns or models can still be discovered. Techniques like additive perturbation [1], multiplicative perturbation [14], secure multi-party computation [20] all fall into this category. On the other hand, rule hiding tries to transform the database such that the sensitive rules are masked, and all the other underlying patterns can still be discovered [17]. For a detailed review of PPDM and game theory please refer to [7].

3 Motivational Application: Peer-to-Peer Client-Side Web-Usage Mining

Before we address the privacy issues in P2P data mining, we present a motivational application in which preserving privacy of a peer's data is important. We discuss an exciting application of web usage mining using the concepts of client-side web cache and P2P technology based on [13] and emphasize the need for privacy in such mining operation.

Traditional web mining has spent a considerable amount of effort on analyzing the server logs. However, since the results of these analysis are not accessible to the users, the later are deprived of their own generated knowledge which can potentially be used for better searching, routing, forming trust-based communities etc. Dynamically aggregating peers with similar interests could greatly enhance the capability of each individual, could facilitate knowledge sharing, and reduce the network load. In order to solve this problem, we present a framework where the users themselves can form implicit communities by sharing their own browsing behavior. Throughout the remainder of this discussion, we use the term 'peer' and 'user' interchangeably, to refer to the same physical entity – a user (peer) browsing the Internet and connected to other users (peers) in the network.

This application uses the frequency of the web domains a user has visited during a period of time as the user's profile vector. Each user maintains a profile vector that keeps the frequency of visit of common web-domains. To measure the similarity between two users' browsing patterns, we use inner product between their profile vectors. To do that, the application uses *order* statistics-based *local* algorithm to measure inner product between different users' profile vector and that information is used to form communities such that users with high similarity in profile vectors are placed in the same community. One of the big advantages of this framework is, any meta-level technique that can measure similarity in metric space (vectors, trees and the like) can be plugged into this framework , and help to form similarity based communities which will share common interest between each other to enhance browsing/online experience.

It is obvious that user privacy is a big concern in such a P2P applications. Since formation of these communities involves sharing the actual browsing data, it may violate the privacy of the users. For any user it is imperative that its browsing data is not revealed in its raw form while forming these communities – otherwise it is almost impossible to convince web users to take part in such P2P computation where every user shares some data, does some computation and finally gets benefited from the aggregate result (by being part of a similar minded community). In order to safeguard each user's private data, Liu et al. [13] have used cryptographic secure inner product protocols to compute the inner product between two users' profile vectors. However, such secure multi-party protocols are based on honest/semi-honest third party assumptions, which assumes that a user or a group of users will follow the protocol as specified and will not form a malicious liaison or do anything to extract private information from other users. In real-life, however, such ideal assumptions fall apart since very little control exists on each user's behavior and there is no centralized coordinator or administrator to monitor and govern all user activity. Besides, experimental results reported by Liu et al. [13] show that this secure protocol is (1) computationally very intensive (2) expensive from communication point of view and (3) not scalable at all. It is evident from the results that privacy preserving techniques designed for standard data mining is not going to work well in a P2P setup. A completely different approach is necessary to ensure privacy in a P2P setup. That motivates us to introduce a game theoretic approach to privacy preserving data mining which does not suffer from above-mentioned issues and relaxes some of the assumptions regarding user behavior.

4 Game Formulation

In this section we present a high level overview of PPDM algorithms designed as games.

We model the large-scale multi-party data mining applications as games where each participant tries to maximize its benefit or utility score by optimally choosing the strategies during the entire PPDM process. Let $D = \{d_1, d_2, \cdots d_n\}$ be a collection of n different nodes where each node represents a party with some privacy-sensitive data. The goal is to compute certain functions of this multi-party data using some PPDM algorithm. Most existing PPDM algorithms assume that every party cooperates and behaves nicely.

For example, consider a well-understood algorithm for computing sum based on the secure multi-party computation framework (details to be described in Section 5). Upon receipt of a message, a node performs some local computation, changes its states, and sends out some messages to other nodes. Most privacy-preserving data mining algorithms for multi-party distributed environments work in a similar fashion. During the execution of such a PPDM algorithm, each node may have the following objectives, intentions, responsibilities: (1) perform or do not perform the local computation, (2) communicate or do not communicate with the necessary parties, (3) protect the local private data, (4) attack the messages received from other parties for divulging privacy-sensitive information regarding other parties, and (5) collude with others to change the protocols for achieving any of the above tasks. Our goal is to view multi-party privacy-preserving data mining in a realistic scenario where the participating nodes are not necessarily assumed to be well-behaved; rather we consider them as real-life entities with their own goals and objectives which control their own strategies for dealing with each of the above listed dimensions.

The nodes in the system can adopt different strategies for communication, computation, collusion or launching of a privacy breach attack. Every such decision is motivated by the utility associated with the choice. The utility value represents the benefit that a node gets by performing (not performing) a necessary communication or computation step that is part of the protocol or by colluding (not colluding) with other nodes in the network. The actions change the local state of the party. The entire play of the game by player i can therefore be viewed as a process of traversing through a game tree where each tree-node represents the local state described by player i's initial state and messages communicated with other nodes. Each run r represents a path through the tree ending at a leaf node. The leaf node for path (run) r is associated with a utility function value $u_i(r)$. A strategy σ_i for player i prescribes the action for this player at every node along a path in the game tree. In the current scenario, the strategy prescribes the actions for computing, communication, privacy protection, privacy-breaching attack, and collusion with other parties. A strategy σ_i for player i essentially generates the tuple $(I^{(M)}, I^{(R)}, I^{(S)}, I^{(A)}, I^{(G)})$ where the Is are indicator variables for a node's computation, communication (receive and send), privacy attack and collusion strategies. Now we can put together the overall objective function for the game of multi-party secure sum computation.

$$u_i(\overline{\sigma}) = w_{m,i} c_m U(I_i^{(M)}) + w_{r,i} c_r U(I_{i,t}^{(R)}) + w_{s,i} c_s U(I_i^{(S)}) + w_{g,i} \sum_{j \in D-G} g(v_j) \quad (1)$$

Here $c_m U(I_i^{(M)})$ denotes the overall utility of performing a set of operation $M_{i,t}$, indicated by I_i^M (similarly for other notations), w's are the weights of the corresponding computation, computation etc., D denotes the set of all nodes, G denotes the set of colluding nodes and $g(v_j)$ is the benefit of node j, due to its local value v_j.

5 Illustration: Multi-party Secure Sum Computation

Suppose there are s individual sites, each with a value $v_j, j = 1, 2, \ldots, s$. It is known that the sum $v = \sum_{j=1}^{s} v_j$ (to be computed) takes an integer value in the range $0, 1, \ldots, N-1$. The basic idea of secure sum is as follows. Assuming sites do not collude, site 1 generates a random number R uniformly distributed in the range $[0, N-1]$, which is independent of its local value v_1. Then site 1 adds R to its local value v_1 and transmits $(R + v_1)$ mod N to site 2. In general, for $i = 2, \ldots, s$, site i performs the following operation: receive a value z_{i-1} from previous site $i-1$, add it to its own local value v_i and compute its modulus N. Then site 1, which knows R, can subtract R from z_s to obtain the actual sum. This sum is further broadcast to all other sites. For this secure sum protocol one may construct different utility functions based on different parameters such as cost of communication, computation or the cost of launching a privacy attack. It can be shown that a privacy-breach attack on a secure sum protocol by a single node might not be very successful. Similarly it can be shown that the utility of collusion in secure sum protocol depends on the size of the network, the number of colluding nodes, and the range of values at the different nodes. [7].

5.1 Game Equilibrium

Let us consider this simple unconstrained version of the objective function given in 1. In order to better understand the nature of the landscape let us consider a special instance of the objective function where the node performs all the communication related activities as required by the protocol resulting in the following objective function (neglecting the constant term contributed by the communication-related factors):

$$u_i(\overline{\sigma}) = w_{m,i} c_m U(I_i^{(M)}) + w_{g,i} \sum_{j \in D-G} g(v_j)$$

Figure 1 shows a plot of this function as a function of $c_m U(I_i^{(M)})$ and k, the number of colluding parties. It shows that the optimal solution takes a value of $k > 1$. This implies that in a realistic scenario for multi-party secure sum computation, parties will have a

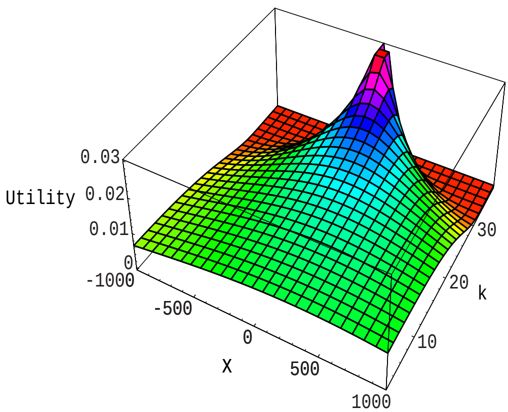


Fig. 1. Plot of the overall objective function. The optimal strategy takes a value of $k > 1$.

tendency to collude. Therefore the non-collusion ($k = 1$) assumption that the classical SMC-algorithm for secure sum makes is sub-optimal.

One way to deal with the problem is to penalize by increasing the cost of computation and communication. For example, if a party suspects a colluding group of size k' (an estimate of k) then it may split every number used in a secure sum among k' different parts and demand k rounds of secure sum computation one for each of these k' parts. This increases the computation and communication cost by k-fold. This linear increase in cost with respect to k, the suspected size of colluding group, may be used to counteract any possible benefit that one may receive by joining a team of colluders. The modified objective function with the penalty term is

$$u_i(\overline{\sigma}) = w_{m,i} c_m U(I_i^{(M)}) + w_{g,i} \sum_{j \in D-G} g(v_j) - w_p * k'$$

Here w_p refers to the weight associated with the penalty. Figure 2 shows a plot of the modified objective function. It shows that the globally optimal strategies are all for $k = 1$. The strategies that adopt collusion always offer a sub-optimal solutions. An appropriate amount of penalty for violation of the policy may reshape the objective function in such a way that the optimal strategies correspond to the prescribed policy. Our plan is to borrow the concept of Cheap Talk from game theory and economics [6] in order to develop a distributed mechanism for penalizing policy violations. Cheap Talk is simply a pre-play communication which carries no cost. Before the game starts, each player engages in a discourse with each other in order to influence the outcome of the game. For example, in the well known Prisoner's Dilemma game one might add a round of pre-play communication where each player announces the action they intend to take. Although cheap talk may not effect the outcome of Prisoner's Dilemma game, in many other games the outcome may be significantly influenced by such pre-game communication. We would like to use Cheap Talk to communicate the threat of penalty. Cheap talk works when the parties depend on each other, their preferences are not opposite to each other, and the threat is real. The algorithm in the following section

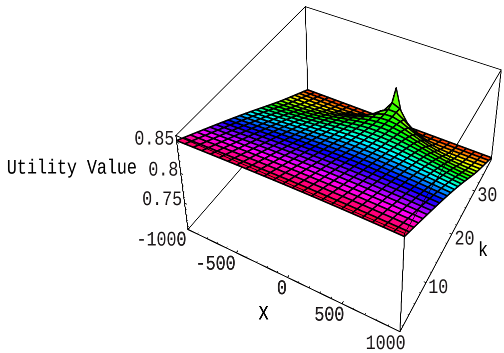


Fig. 2. Plot of the modified objective function. The globally optimal strategies are all for $k = 1$.

describes a variant of the secure sum computation technique that offers a distributed mechanism for penalizing policy violations using a cheap talk-like mechanism.

5.2 Secure Sum with Penalty: Distributed Control

Consider a network of s nodes where a node can either be *good* or *bad*. *Bad* nodes collude to reveal other nodes' information; while *good* nodes follow the correct secure sum protocol. Before the secure sum protocol starts, the colluding (*bad*) nodes send invitations for collusions randomly to nodes in the network. If such a message is received by a *good* node, then it knows that there are colluding groups in the network. To penalize nodes that collude, this *good* node splits its local data into k' random shares where k' is an estimate of the size of the largest colluding groups. One possible way to estimate this could be based on the number of collusion-invitations a good node receives. On the other side, the *bad* nodes, on receiving such invitation messages, form a fully connected networks of colluding groups. After this the secure sum protocol starts, as in the traditional secure sum protocol, nodes forward their own data (after doing the modulus operation and random number addition). However, good nodes do not send all the data at one go – rather they send random shares at each round of the secure sum. Hence, it takes several rounds for the secure sum to complete.

5.3 Results

We have implemented the above cheap talk-based solution without and performed multi-agent simulations in order to study the behavior of the agents. This experiment assumes that the agents are rational in the sense that they choose actions that maximize their utility function. The details of the experimental setup are described in [7]. The results obtained from our simulation are represented by the following figures.

Initially we start with a fixed percentage (say 30%) of the nodes to be *bad*. After every round each node measures the cost (or penalty) it incurred due to collusion. If the

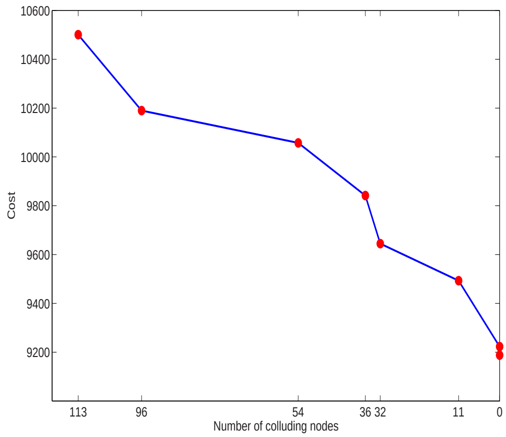


Fig. 3. Variation of cost with changes in number of colluding nodes

penalty sustained is too high (a dynamic threshold currently set by the user), some of the bad nodes decide not to collude again. Once these *bad* nodes turn into *good* ones, they send deallocate messages to their colluding groups and also set their estimates of collusion size k' same as the size of the collusion to which they belonged.

We observe in Figure 3 that for subsequent rounds of the secure sum computation the cost or overall penalty assigned decreases as the number of *bad* nodes decreases. When the ratio of *bad* to *good* nodes is significantly low, we can observe that the cost almost reaches an equilibrium. This is because the contribution of the penalty function becomes negligible and the total cost is governed mainly by the computation and communication costs that remain almost constant over successive rounds of secure sum with hardly any collusion. Further details about the experiments can be found elsewhere [7].

6 Conclusion

This paper presented an overview of privacy issues in P2P data mining from a game theoretic perspective. It first motivated the discussion using a P2P client-side web mining application. The paper pointed out some the emerging issues for protecting privacy in P2P data mining. It also noted that many of the existing privacy-preserving data mining algorithms often assume that the parties are well-behaved and they abide by the protocols as expected. The paper offered an overview of a recent effort to formulate the PPDM problem as a multi-party game where each party tries to maximize its own objective or utility.

The paper opens up many new possibilities. It offers a new approach to study the behavior of existing PPDM algorithms. It also offers many challenges. The paper illustrated the concept using the multi-party secure sum computation algorithm. We need to explore other PPDM algorithms. We need to design protocols that offer more robust performance. Analysis of Nash and other kinds of game theoretic equilibrium states will be crucial for further advancing this line of research.

Acknowledgments

The authors acknowledge support from NSF CAREER award IIS-0093353. The authors would also like to thank Souptik Datta for his valuable contributions.

References

1. R. Agrawal and R. Srikant. Privacy preserving data mining. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 439–450, Dallas, TX, May 2000.
2. B. Awerbuch, A. Bar-Noy, N. Linial, and D. Peleg. Compact distributed data structures for adaptive network routing. *Proceedings of the 21st ACM Symposium on the Theory of Computing (STOC)*, 1989.
3. M. Bawa, A. Gionis, H. Garcia-Molina, and R. Motwani. The price of validity in dynamic networks. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 515–526, 2004.
4. S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: Design, analysis and applications. In *Proc. IEEE Infocom'05*, volume 3, pages 1653–1664, Miami, March 2005.
5. S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta. Distributed data mining in peer-to-peer networks. *IEEE Internet Computing special issue on Distributed Data Mining*, 10(4):18–26, 2006.
6. Joseph Farrell and Matthew Rabin. Cheap talk. *The Journal of Economic Perspectives*, 10(3):103–118, 1996.
7. H. Kargupta, K. Das, and K. Liu. A game theoretic approach toward multi-party privacy-preserving distributed data mining. In *Communication*, 2007.
8. D. Kempe, A. Dobra, and J. Gehrke. Computing aggregate information using gossip. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FoCS)*, pages 482–491, 2003.
9. Kowalczyk W., Jelasity M., and Eiben A. Towards Data Mining in Large and Fully Distributed Peer-To-Peer Overlay Networks. In *Proceedings of BNAIC'03*, pages 203–210,2003.
10. Kuhn F., Moscibroda T., and Wattenhofer R. What Cannot be Computed Locally! In *Proceedings of the 23rd Symposium on Principles of Distributed Computing (PODC)*, 2004.
11. Kutten S. and Peleg D. Fault-Local Distributed Mending. *Proceedings of the 14th Annual ACM Symposium on Principles of Distributed Computing*, 1995.
12. Linial N. Locality in Distributed Graph Algorithms. *SIAM J. Comp.*, 21:193–201, 1992.
13. K. Liu, K. Bhaduri, K. Das, P. Nguyen, and H. Kargupta. Client-side web mining for community formation in peer-to-peer environments. *SIGKDD Explorations*, 8(2):11–20, 2006.
14. K. Liu, H. Kargupta, and J. Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 18(1):92–106, January 2006.
15. M. Mehyar, D. Spanos, J. Pongsajapan, S. Low, and R. Murray. Distributed averaging on a peer-to-peer network. In *Proceedings of IEEE Conference on Decision and Control*, 2005.
16. Naor M. and Stockmeyer L. What Can Be Computed Locally? *Proceedings of the 25th ACM Symposium on Theory of Computing (SToC)*, pages 184–193, 1993.
17. V. S. Verykios, A. K. Elmagarmid, B. Elisa, Y. Saygin, and D. Elena. Association rule hiding. In *IEEE Transactions on Knowledge and Data Engineering*, 2003.

18. R. Wolff, K. Bhaduri, and H. Kargupta. Local L2 thresholding based data mining in peer-to-peer systems. In *Proceedings of SIAM International Conference in Data Mining (SDM)*, Bethesda, Maryland, 2006.
19. Wolff R. and Schuster A. Association Rule Mining in Peer-to-Peer Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 34(6):2426–2438, December 2004.
20. A. C. Yao. How to generate and exchange secrets. In *Proceedings 27th IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.

Ontos Solutions for Semantic Web: Text Mining, Navigation and Analytics

Vladimir Khoroshevsky[2], Irina Efimenko[1], Grigory Drobyazko[1],
Polina Kananykina[1], Victor Klintsov[1],
Dmitry Lisitsin[1], Viacheslav Seledkin[1], Anatoli Starostin[1],
and Vyacheslav Vorobyov[1]

[1] Ontos AG, 84/2 Vernadskogo Av.,
119606 Moscow, Russia
{Irina.Efimenko, Grigory.Drobyazko, Polina.Kananykina, Victor.Klintsov, Dmitry.Lisitsin, Viacheslav.Seledkin, Anatoli.Starostin, Vyacheslav.Vorobyov}@avicomp.com
[2] Computer Center RAS, 40 Vavilov str,
GSP-1 Moscow, Russia
khor@ccas.ru

Abstract. This paper deals with the problem of development and implementation of semantic navigation through Web-content. Multi-agent architecture of a solution for Semantic Web and innovative services are presented. In the context of the proposed solution Web mining is carried out by special OntosMiner agents, which provide the ontology-driven processing of multilingual text collections on the basis of the special kind of content extraction technologies. First evaluation results of the presented solution are discussed as well.

Keywords: Semantic Web, Semantic navigation, Multi-Agent system, Ontology-Driven Content Extraction, Natural Language Processing.

1 Introduction

It is generally known that, at the present stage of social development, the worldwide process of accumulating and distributing information has a snowballing nature. Thereby the existing "traditional" information retrieval, storage and analysis methods are continuously loosing their efficiency, because of their inaccuracy and/or redundancy.

According to a common expert estimation, about 80 % of all reasonably accessible information is contained in natural-language texts. Meanwhile, the main shortcoming of "traditional" document processing methods consists in their inadequacy becoming apparent when attempts are made to solve a problem of understanding the meaning of texts and performing, on this basis, semantic search and analysis of information.

It is clear that such a situation could not be passed over by the world IT community and several years ago the World Wide Web Consortium (W3C), which deals with the development and implementation of technological standards for the Internet, offered the "Semantic Web" concept [1]. The gist of this concept is the transformation of

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 11–27, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Web content into a form that allows its efficient processing by both humans and computers based on additional descriptions expressing the meaning of documents. In order to receive such a benefit, the Resource Description Framework (RDF) was adopted in 2004 as a W3C specification. In the same year, the Web Ontology Language (OWL) got the status of a recommended technology [2]. The OWL-like knowledge representation tools also ensure a technological baseline for generalizing data, retrieving logical links between data elements, drawing inferences, and even making solutions based on these inferences [3, 4, 5].

It is evident that the creation and implementation of technologies and systems capable of understanding unstructured and semi-structured texts “in a man-like manner” and allowing mankind to efficiently handle huge volumes of accumulated information is one of the main problems for information community nowadays.

Ontos AG is one of the companies that develop and implement knowledge management technologies for building up next generation information systems based on the Semantic Web concept. Main innovations of Ontos AG are in the field of development and implementation of:

- special multi-agent system for information-to-knowledge transformation based on the newest linguistic technologies allowing to extract information from texts and text collections in various natural languages under the control of domain ontologies;
- methods of concentrating knowledge retrieved from various documents and/or collections of documents in an integrated semantic storage which is implemented as a basic RDF-store and supports all main operations related to the aggregation of semantically meaningful entities and relations between them, efficient pattern-matching search of desirable text fragments, and exchange with “outer space” in the XML and/or OWL languages;
- innovative methods of knowledge handling that employ visual representation of the meaning of documents and/or document collections in the form of cognitive maps (special-purpose semantic networks); methods of dynamic generation of textual digests from document collections; methods of automatic generation of textual summaries of documents and/or document collections in a specified target language;
- semantic navigation through Web-content and the user-friendly services for digesting and/or summarization.

Main goal of this presentation is to discuss the Ontos solutions for semantic navigation through Web-content and innovative services supported by special multi-agent platform.

The rest of the paper is organized as follows. Discussion of the Ontos solution architecture and software platform is given in the next section. Section 3 of the paper presents users’ casual activities supported by Ontos SemanticWeb system and future trends of Ontos AG R&D. Conclusions are presented in the last section of the paper.

2 Ontos Solution Architecture and Software Platform

2.1 General Remarks

The Ontos technologies and product line were designed for carrying out tasks pertaining to the processing of information from unstructured or semi-structured sources. These technologies are ultimately a synthesis of methods and techniques in the fields of structural linguistics and expert systems, data/knowledge base management systems, cross-platform programming tools, standards issued by the WWW-consortium, industrial standards. An important innovative solution within Ontos is an interface that is simple, easy to integrate and intuitively clear to users of various platforms. It ensures the attractiveness of science-intensive Ontos information technologies for a wide range of the customers.

2.2 Ontos Solution Architecture

In general, the technological cycle of information processing, being a part of solutions offered by Ontos AG, can be presented by the diagram depicted on Fig. 1.

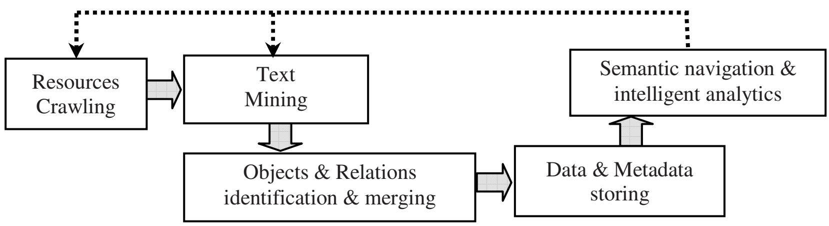


Fig. 1. Information processing technological cycle

An important feature of the above depicted process organization is employing the distributed grid computing technology [6] which provides the system scalability by the means of involving additional computational power whenever the flow of documents being processed requires that, as well as the higher reliability and fault tolerance of the system as a whole. Taking into account the above mentioned items, it is possible to illustrate the Ontos SOA (Service Oriented) Architecture as follows (Fig. 2).

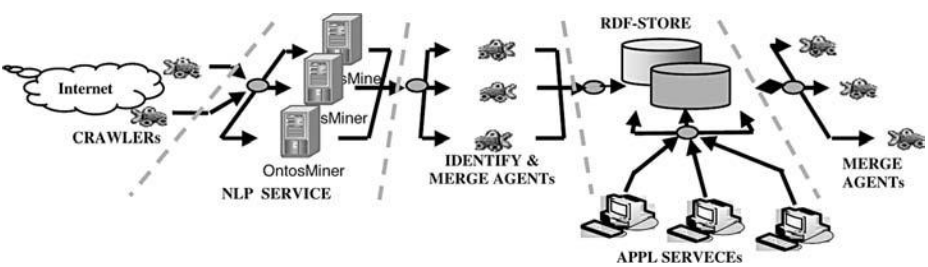


Fig. 2. Ontos SOA Architecture general view

2.2.1 Resources Crawling

According to Fig. 1 and Fig. 2 “Resources Crawling” means the monitoring of certain Internet pages (in accordance with specified criteria), as well as acquisition via Internet of text documents presented on these pages for processing. As a rule, the resource crawling is performed automatically. In some cases, however, when the information extraction accuracy is of higher importance, the involvement of the system administrator in this process is possible. In such cases the administrator specifies URL addresses of documents to be processed, and conditions of extracting a text component from the pages of those documents.

The system may be configured by several types of agents responsible for the acquisition of input documents. Such agents scan data sources and retrieve documents by themselves. To the moment the following functionality is included:

- Crawling pages of Internet sites;
- Crawling the File System folders;
- Processing of RSS feeds.

In addition to that, Ontos technologies provide external applications with the possibility of calling the Ontos Annotation Server (NLP Service) via special API and transferring documents for processing. Crawlers are implemented as deliberative agents communicating with other components of the system using RMI.

2.2.2 Text Mining

NLP services use special Broker that transfers text components extracted from documents to an appropriate OntosMiner processor, depending on a domain model and language of the text.

OntosMiner is an intellectual core of the Ontos technologies. The tasks of this multi agent service include the automatic linguistic analysis of texts in accordance with specified domain ontology and on the basis of special linguistic and expert rules. This analysis results in the extraction of semantically meaningful, in terms of the specified ontology, concepts and relations between them from the text, which are represented for output as XMLs. Detailed discussion of Ontos solution for “Text Mining” component is presented in [7, 8]. So, below we’ll concentrate on the multi agent architecture of NLP Services and on the discussion of knowledge used by agents involved into text mining.

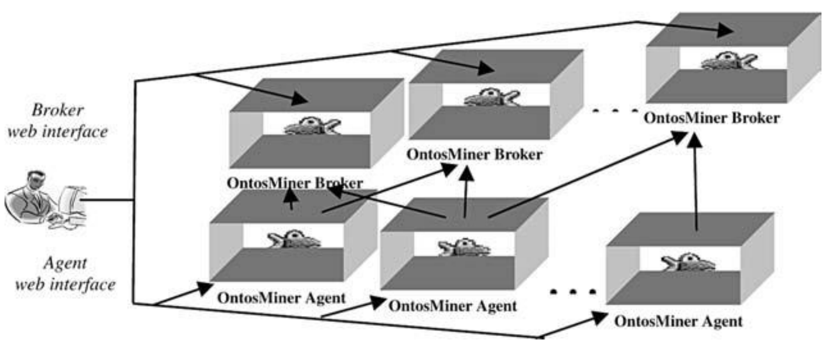


Fig. 3. Common architecture of OntosMiner based multi agent system

NLP services are implemented as multi agent system with architecture presented in Fig. 3. There are three main groups of the components that can be outlined within this architecture: Broker & Agent Web Interfaces, OntosMiner Brokers and OntosMiner Agents.

Interface group supports communications between the user and multi agent system itself. Main tasks of the components involved into such a communication are the following:

- for each OntosMiner Broker, to depict those Crawlers that provide input queue for this broker;
- for each OntosMiner Agent, to depict those OntosMiner Brokers that will manage texts processing;
- to provide the possibilities of control of OntosMiner Brokers and Agents "on-line" statuses.

After the registration each OntosMiner Broker "knows" its OntosMiner Agent and vice versa. In addition to that, each OntosMiner Broker "knows" language and domain appropriate for each OntosMiner Agent. So, OntosMiner Brokers are simple deliberative agents with the main goal to pass the text received from Crawlers to appropriate OntosMiner Agent and, after the end of text processing, to pass the results to agents responsible for merging new cognitive map with knowledge that already exists within RDF-store.

The behavior of each OntosMiner Agent is much more complicated. First of all, it should "know" domain ontology that is going to be used for text processing. It should also "know" how to process input text. And finally, it should "know" how to present the results received as an output of text processing.

To the moment, the knowledge about the domain ontology of concrete OntosMiner Agent is its internal knowledge. For example, during installation procedure, OntosMiner Agents designed for English "Business Duties" texts processing catches ontology depicted in Fig. 4 and OntosMiner Agents created for the processing of Russian texts in "Medicine" domain catches ontology in Fig. 5.

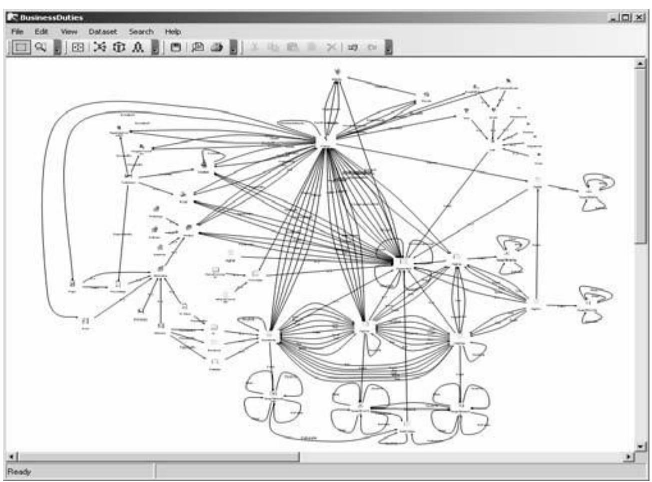


Fig. 4. Domain ontology "Business Duties"

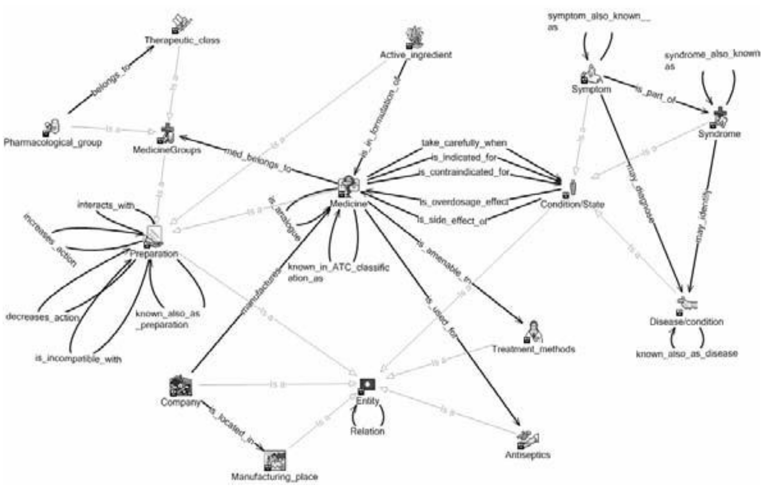


Fig. 5. Domain ontology "Medicine"

Main concepts and semantic relations between them specified within these ontologies are presented in Tables 1, 2.

Table 1. Entities and relations specified in domain ontology "Business Duties"

Entities	**Attributes**
Person;	FamilyName; FirstName; FullName; Gender; Birthday
Organization (State, Commercial, Government, Educational, Military, MassMedia)	Title; hasType; NumberOfClients; NumberOfEmployees; Revenues
Location	Type
JobTitle	Obligations
Title	
Date	dd; mm; yy
Period	StartPoint; EndPoint; Length
Time	hh; mm; ss;
PostAddress	country; province; code; city; postOffice; street; floor; block; building; corpus; flat
Email	netCountry; netType
Url	netCountry; netType
Phone	phoneCountry
Money	CurrencyUnit; Amount
Percent	Amount
Degree	Speciality; SubSpeciality
Semantic Relations	**Attributes**
Buy	Price; Seller; When
Sell	Price; Buyer; When

Table 1. {*continued*}

Found	When
Own	Quota; When
InvestInto	Investment; When
EarnDegreeFrom	Degree; Specialty; SubSpecialty; When
GraduatedFrom	Honor; Specialty; SubSpecialty; When
BeEmployeeOf	JobTitle; When
BeLocatedIn	When
RivalWith	Domain; When
BePartners	Domain; When
BePartOf	
ParticipateIn	When
MergeWith	When
ShareWith	When
The Same?	

Table 2. Entities and relations specified in domain ontology “Medicine”

Entities	**Relations**
Treatment methods, Medication, Preparation, Company (Producer), Medicine, Medicine Groups, Active Ingredient, Condition/State, Symptom, Syndrome, etc.	Interacts with, Is overdosage effect of, Is analogue, Is indicated for, Is contraindicated for, Is side effect of, Take carefully when, Produce, LocatedIn, etc.

Knowledge about text processing technology is represented to the moment as special XML-file where each text processing resource is indicated by its own XML-tag (with appropriate parameters) and resource chain that is the sequence of references to the needed resources. XML-specifications of MorphTagger resource and the chain of OntosMiner/English 3.0 resources are depicted below as an example.

```
<RESOURCE NAME="MorphTagger" TYPE="JPlusTransducer">
  <PARAM NAME="grammarURL">
     <VALUE>./resources/morph/main.jape</VALUE>
  </PARAM>
</RESOURCE>

<CHAIN NAME="MainChain">
 <REF RESOURCE="Tokeniser"/><REF RESOURCE="MorphTagger"/>
 <REF RESOURCE="Gazetteer"/>..........................................................................
 <REF RESOURCE="VP_Chunker"/><REF RESOURCE="NP_Chunker"/>
 <REF RESOURCE="DeepNETransducer"/>
 <REF RESOURCE="SemanticTagger"/>
 <REF RESOURCE="XMLGenerator"/>
</CHAIN>
```

Mapping of internal representation of text processing results to an appropriate ontology is specified in separate XML-definition file.

OntosMiner Brokers and Agents communicate via RMI.

After being processed by identification and merging components (Merge Agents) the results received by NLP service are converted to the RDF format and are placed into the Expert Knowledge Base (EKB) implemented as RDF-store [9].

2.2.3 Scalable Early Objects Identification

Early objects identification is fulfilled by module EOI (Early Objects Identification) that retrieves objects and relations extracted from the newly processed documents before they are put in the EKB.

The term “early identification” implies that any object (relation) revealed during the semantic processing of a text is checked for the presence of similar objects in the RDF-store. It is achieved by comparing the object types as well as the structure and values of their attributes and their relations to other objects. If the Identifier agent that is, to the moment, the part of Merger agent finds out a “similar” object in the EKB, the procedure of merging these two objects into one is executed by the Merger agent.

This procedure includes the integration of attributes and their values of the objects along with the attributes and their values of the corresponding relations.

The user can set up criteria for determining the identity of similar objects and adjust the identity threshold values according to the chosen criteria.

Information that does not satisfy the listed conditions is placed into a database, parallel to the EKB. This parallel database stores information that has not yet got sufficient corroboration to be moved to the Expert Knowledge Base.

2.2.4 Expert Knowledge Base and RDF-Store

Expert Knowledge Base (EKB) is built up automatically. It ensures the storage of entities (objects) and relationships between them extracted from documents, i.e., meta information relevant to a specified domain model, as well as of external links (URL) to the processed documents. General flow of information processing at EKB level is illustrated by Fig. 6.

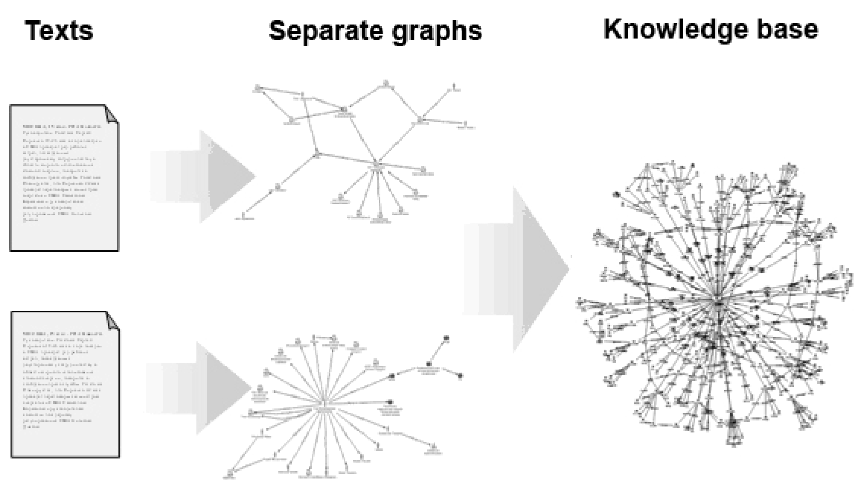


Fig. 6. General flow of information processing at EKB level

At the heart of the EKB there is the RDF-store implemented in various structured data storage environments (currently, in the SQL Server 2003 [10] and Oracle 10g R2 [11]).

3 Ontos Solution for Semantic Web

3.1 Semantic Services – General View

To the moment semantic services implemented within Ontos solution for Semantic Web are the following ones:

- Ontos Navigation Server (ONS).
- Ontos Inference Server (OIS).
- Ontos Semantic Portal (OSP).

The Ontos Navigation Server is presented by Semantic navigation and Semantic ranking & relevance modules and services.

The OIS functionality is presented by such services as Semantic Digesting and Semantic Summarizing (multilingual).

Goal of the Ontos solutions within the Ontos Semantic Portal is to semantically integrate the best and most often visited thematic Internet resources.

All the above depicted services and components are discussed in the next sections of the paper.

3.2 Ontos Navigation Server (ONS)

As it was mentioned above, main ONS modules and services are connected to semantic navigation and semantic ranking & relevance.

3.2.1 Semantic Navigation

Semantic navigation allows the user to surf between the knowledge base objects through the relations, acquire information on attributes of these objects, and pass to the source documents.

From the user's point of view, the service operation looks like the appearance of additional hyperlinks on a Web page which is open in a browser (Fig. 7).

After having been extracted by the system during the process of semantic analysis of the current page content, the entities and relations are "superimposed" on the document pages displayed in the browser window, thus establishing the correspondence between a given text fragment on the page and an object in the EKB.

When a mouse pointer is moved over a semantic link, the user can activate an object card, this card being a starting point for the semantic navigation. The navigation card contains:

- Object attributes.
- Object relations (navigational routes in the cyberspace).
- Documents the object was found in (with relevance measure).

The navigation card allows navigating through semantic links between the objects. During this process, all links to the pages having occurrences of the objects are

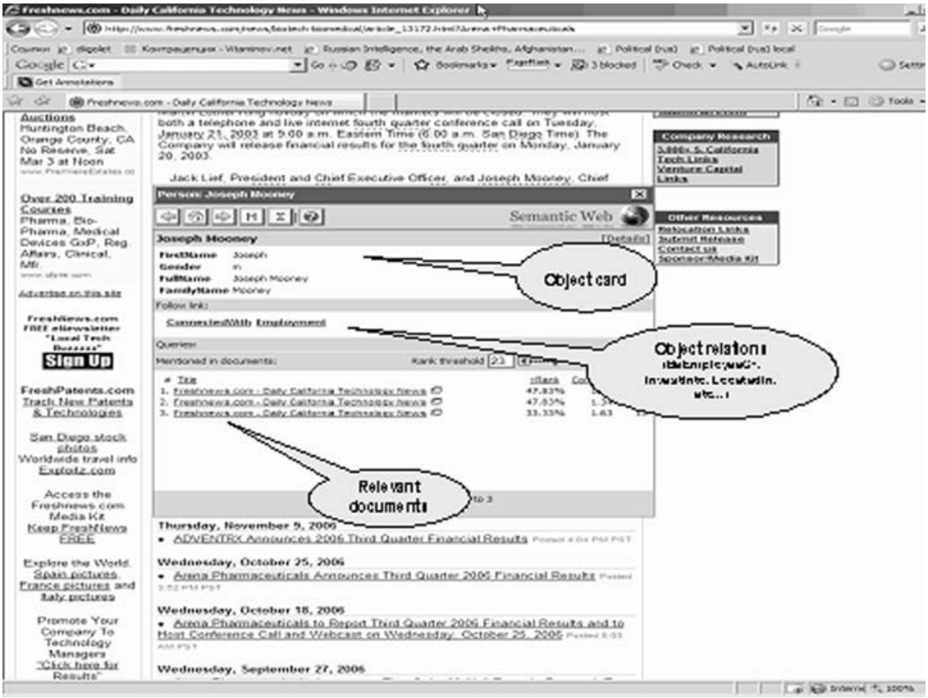


Fig. 7. Annotated page with object's card

"drawn up". The links to the pages are ranked in the card according to their semantic relevance. Therefore, the semantic navigation system integrates many Web sites from similar or neighboring subject areas into one thematic mega portal.

Visitors to the sites having been indexed with the Ontos technologies do not need any more to look through heaps of pages produced by the common search engines, in order to find out the required information. Now it is enough just to choose a corresponding object on a page of any indexed site and open its card. The card will have generalized information on the object collected from all pages of annotated information resources.

3.2.2 Semantic Indexing

Meta information extracted from a document includes specifications of multiple entities of different types (people, organizations, addresses, etc.), their attributes and semantically significant relations between them. The positions of entities and relations in the document text are also included, thus ensuring the feedback from the meta information to the text.

The entities and relations together form a semantic network (cognitive map). Some of the entities have more relations in comparison with others. These are the so-called "focuses" of the document, i.e., its main topics in terms of the domain ontology being used. The more relations an entity (an object) has, the more significant is the role it plays in the document semantics. It is a basis for computing the semantic relevance (Fig. 8).

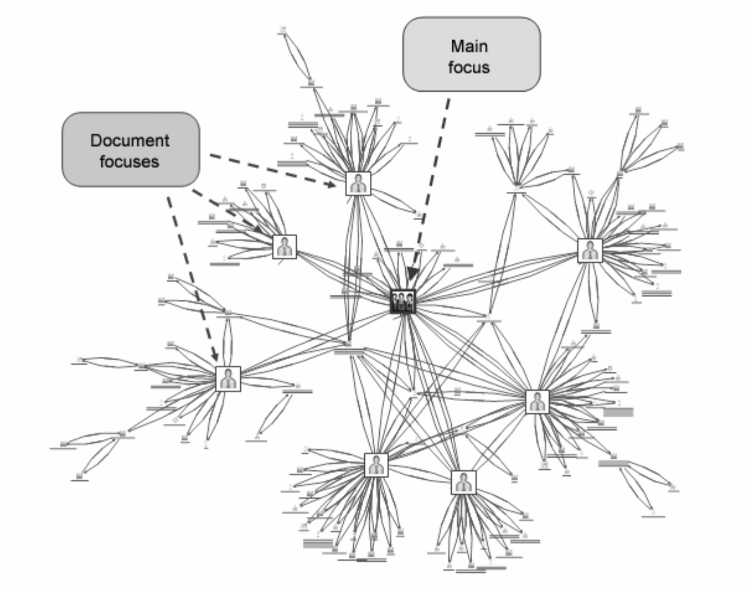


Fig. 8. Semantic indexing of cognitive map

For example, a query “Google” sent to the Ontos knowledge base is not going to return, as the most relevant, those documents in which the word “Google” was met most often, but those ones where the “Google” focus had the biggest number of relations, such as “BeEmployeeOf” (the company’s employees), “Found” (the company’s founders and foundations), “Merge&Acquired” and “Buy/Sell” (merged and acquired companies).

A query “Eric Schmidt is EmployeeOf...” would return documents where Dr. Schmidt’s work for different companies was mentioned.

Generally all the entities with a number of relations exceeding an averaged index for a given document can be considered as the document “topics”. After having got semantic “weights” of the entities and relations in accordance with their significance (thus, the weight of the “Organization” entity exceeds the weight of the “Date” entity), system can calculate a “semantic index” of a document and its “focuses”. Basing on such an index, it is possible to determine the semantic relevance of the document to a specific subject, and ultimately perform the semantic clusterization of documents. The system sorts out the links to documents returned in the navigation card according to the following criteria:

- Number of objects identified in a document.
- Normalized focus number of an object of interest in a document.
- Contrast ratio of an object of interest in a document.

As it was already mentioned, the ranking of documents in accordance with the Ontos technologies is performed per object focuses. The focus of an object having maximum weight is considered to be the main focus of the document, and a value of 100 % is assigned to it. The weights of other objects are normalized relative to the main focus. If this parameter has a value in the range of 70 to 90 percent, then the object defined by this focus is semantically "close" to "the main object" of the document.

Moreover, an average number of semantic links connected to an object is calculated in a document. Then the ratio of a number of semantic relations of a particular object to the average number for the whole document determines the "contrast ratio" of this object in the document. This additional index is important to the analysts who evaluate the semantic intensity of documents.

The system allows adjusting the relevance threshold value of documents in an navigation card, thus changing a number of documents returned by the system. The threshold value equal to 100 % means that a given object has a maximum number of revealed semantic links, and therefore is a candidate for the main topic of the document.

3.3 Ontos Inference Server (OIS)

3.3.1 Semantic Digesting

The semantic digesting service makes it possible to create an on-demand report (digest) consisting of text fragments taken from one document or collection of documents that contain information relevant to a given query. A digest is generated by request when navigating across the EKB. It includes document fragments (sentences) with entities and relations collected en route (Fig. 9).

Digests are based on the special-purpose algorithms applied to collections of cognitive maps and/or their fragments, as well as to the corresponding document

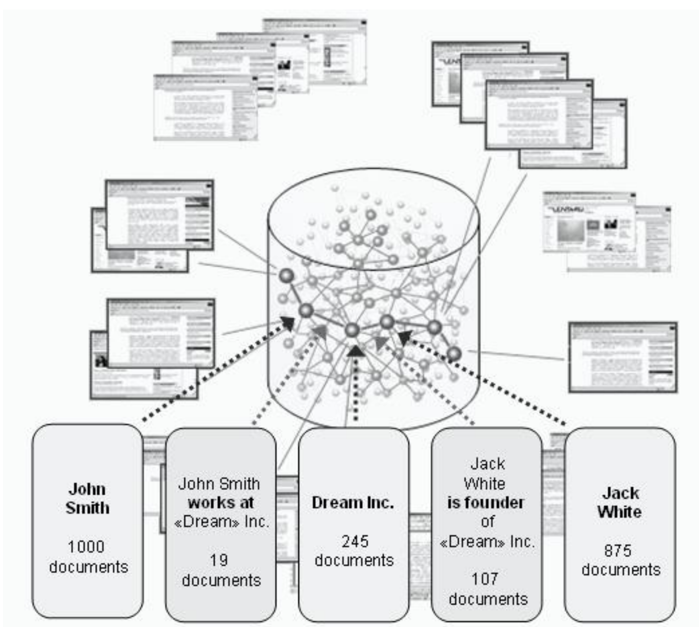


Fig. 9. Surf for digesting

arrays. This complex process looks rather simple and logical for a user. Moving from one object of interest to another one through the semantic links is performed by the user in a manner that is very much similar to the traditional and thoroughly familiar Web surfing. It results in a list of documents which can be later used for generating a digest based on the user's objects of interest. The system stores navigating routes from their starting points up to the current ones, so a "navigation report", or semantic digest, can be generated at any moment.

When a digest is being generated, combinations of sentences with the objects of interest and relations between them from the selected documents are used. Documents publication dates and hyperlinks to the source are indicated for each fragment of the digest (Fig. 10).

This service can be useful both for the end users and information/news portals as a tool for the automatic generation of digests for news feeds.

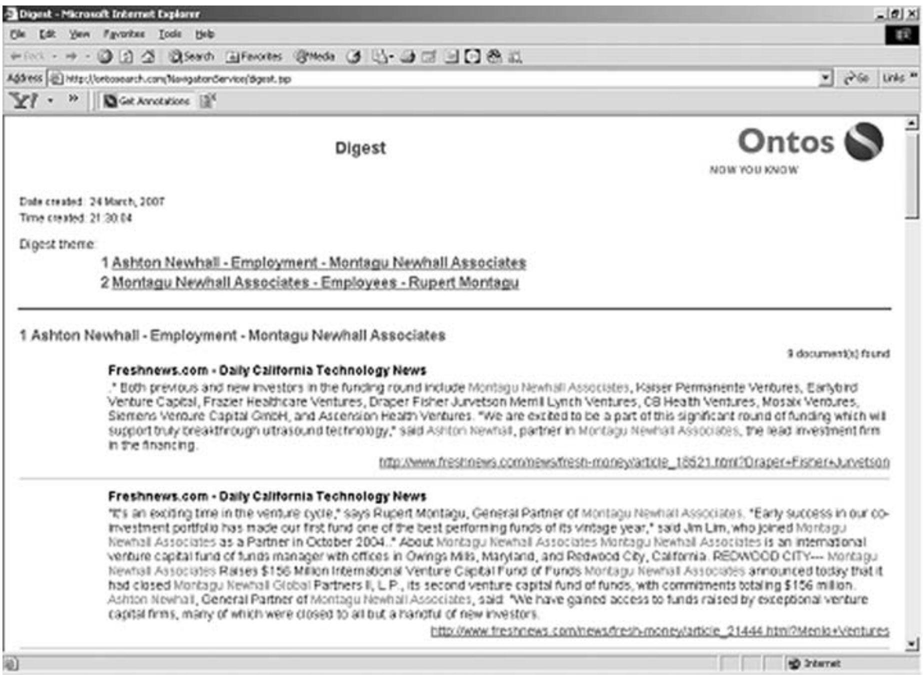


Fig. 10. Digest created on demand

3.3.2 Semantic Summarization

The semantic summarization is a process of creating a textual summary of some amount of information stored in a knowledge base.

The semantic summarization service is designed for automating the routine work on the initial acquisition and analysis of objects of interest represented in natural language texts.

There are special entities in the system called Summary Agents. They are able to get some information from the knowledge base (some sub-graph of the whole

RDF-graph), process this information and generate text in various languages (Russian or English at the moment).

The summary is created using one of the preset scenarios. Scenario determines the layout of a summary and an angle of view according to which the summary is generated.

At the moment Summary Agents support the following scenarios:

Retelling. In this mode agent is trying to sort the information according to the logic of the source documents and uses the dates to bind facts to the timescale where it is possible.

CV. This mode is used to create persons' biographies. Therefore, it is necessary to select a person the system is going to focus on. The structure of the summary in this case is similar to the structure of real CVs (Fig. 11).

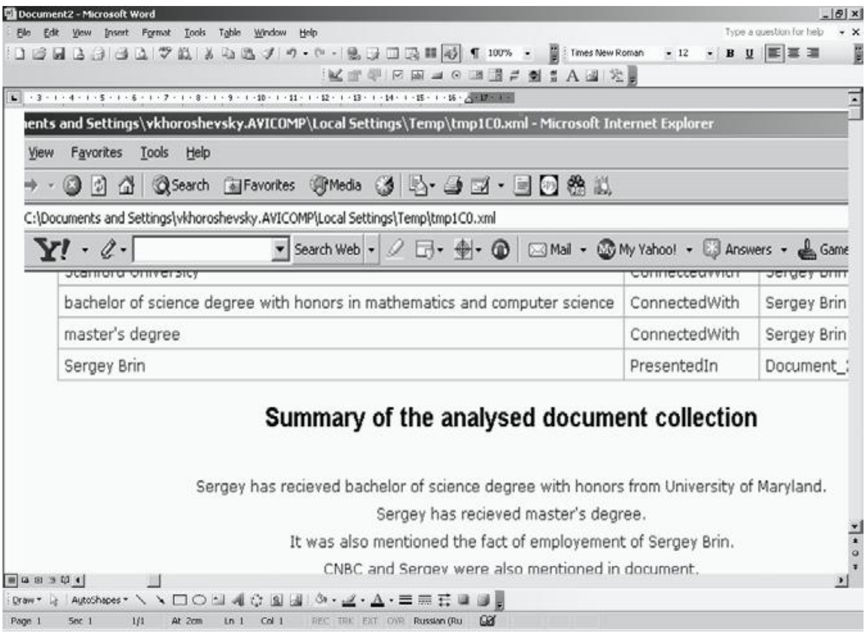


Fig. 11. CV example

Report. While working in this mode agent takes the information from the knowledge base as a neighborhood of the entity the user is interested in. The depth of the neighborhood varies from 1 to 3 (Fig. 12).

Then agent converts extracted relations into the list of tables according to the following principles:

1. Each table corresponds to some entity from the extracted area. First row of a table contains name of the entity. Other rows of a table correspond to certain relation types. First cell of each row contains the relation name ("BeEmployeeOf","Connected With" etc.). Second cell contains the list of names of the entities that are linked with the focus entity using corresponding relation.

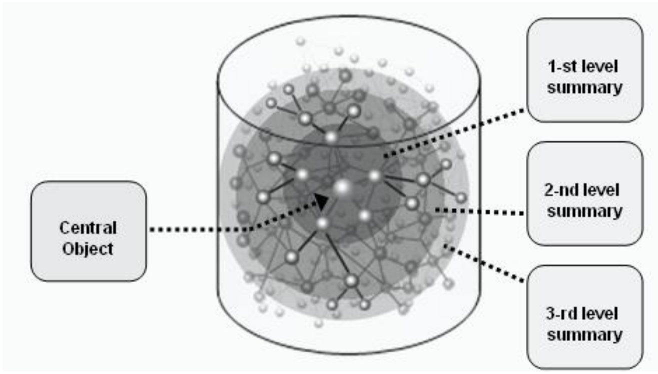


Fig. 12. “Report” summarization mode

2. There are no similar tables in the output (each entity is described once).
3. The closer an entity is to the entity of interest, the closer the corresponding table is to the beginning of the summary. Thus, the table describing the focus entity is always first.

Summarizer Agent itself consists of three parts: OntoMapper, LayoutGenerator and SummaryGenerator. OntoMapper converts the input cognitive map from the domain ontology to another generic ontology that we will call “summarization ontology”. The summarization ontology is a model that only contains such entities and relations that summarizer can “verbalize”, i.e. can find corresponding text generation rules for the constructions with these entities and relations. During this conversion OntoMapper can leave some information, since, when summarizing texts, one does not always need concrete information about the entities themselves being more interested in the relations between them. For instance, “Commercial organization” and “University” can be mapped into common concept “Organization”. After the mapping is done, the LayoutGenerator starts. This module considers the input cognitive map as a set of triples <*subject,predicate,object*>. It sorts these triples according to the scenario and passes the derived list of triples to the SummaryGenerator. In its turn, the SummaryGenerator uses a set of text-generation rules and converts the input triple-list to the plain text. There is a specific rule set for each output language. Each rule has left side and right side. Left side is a pattern that must be matched with a part of the input triple-list. Right side is directive that says what to generate if the left side is matched with some input. More then one triple may be presented in a pattern. This allows generating language constructions using conjunction.

Thus, the summarization service has its own generic ontology, and it is able to generate text only from cognitive maps represented in this ontology. Such technique allows to write the summarization rules only once per language, but not per model. Moreover, the left sides of rules are quite the same for all languages. Only right sides differ. The only thing one needs to do when he needs to summarize data represented in some new ontology that is conceptually included in the summarization ontology is to teach OntoMapper to map this ontology to the summarization ontology. Certainly, sometimes there is a need to expand the generic ontology and text-generation rule set. For instance, if summarization ontology doesn’t contain medical concepts, Summary

Agent will never be able to summarize medical data. This means, that the summarization ontology is growing along with the projects developed. It is not generic in common. It is generic regarding the concrete domains and projects.

3.4 Ontos Semantic Portal

As it was already mentioned before, Ontos solutions within the Semantic Portal deal with the integration of the most interesting thematic Internet resources.

Every resource usually has its own wide and stable audience having got accustomed to the structure of a given site and the style of information it presents. That is why the main idea of the Ontos solutions consists in providing the visitors of such sites with an additional semantic markup (color highlighting) of the site pages without altering their general structure and the proper content. Such markup is "active" allowing the users to switch on the mode of semantic navigation across the pages of all sites united by the Portal.

The technology of using the semantic markup is quite simple and unlaboured. By selecting any of the objects highlighted by the markup, e.g. a person or an organization, the user can move to the navigation card of this object and choose all the documents where the object is mentioned. It is important that this procedure is based not on the keywords, but on the semantics of the selected object and its relations with other objects. Moreover, when adjusting the relevance index towards the main focuses, the user can select those documents where a given object is the main topic. Then the user can "see" all the objects somehow linked to the current one (for example, other employees or co-owners of a particular company; other indications for the medicine, etc.), as well as look through the corresponding documents. After such a "promenade" across the Web pages, it will be possible to get a Digest of selected documents, or automatically generate a Summary on an object the user is interested in.

The following thematic portals are currently available:

- News semantic portal (politics, economics) – documents in Russian.
- Medical semantic portal (descriptions of medicines and illnesses, news) – documents in Russian.
- News semantic portal (economics) – documents in English.

4 Conclusion and Future Trends

Problems of the development of Ontos AG solutions for Semantic Web based on Web mining systems, which process multilingual text collections in context of domain models represented by ontologies, and appropriate intelligent services were discussed in the paper. To the moment all the presented solutions and services are implemented and are in use.

Ontos AG future trends in this domain are connected to the development and implementation of the new Web mining agents based on OntosMiner family systems for various domains with extended domain ontologies and the new intelligent services for Internet community and corporative customers.

Acknowledgements. Authors would also like to say many thanks to Artem Bozukov, Andrej Kondabarov, Peter Zhalybin and OntosMiner Group specialists, since it is impossible to overestimate their contribution into development and implementation of the presented solutions.

References

1. Tim Berners-Lee, James Hendler, Ora Lassila: The Semantic Web, Scientific American, May (2001)
2. Deborah L. McGuinness, Frank van Harmelen, (eds.): OWL Web Ontology Language Overview, W3C Recommendation, February 10, 2004, http://www.w3.org/TR/2004/REC-owl-features-20040210/
3. Nigel Shadbolt, Tim Berners-Lee and Wendy Hall: The Semantic Web Revisited, IEEE Intelligent Systems 21(3), May/June (2006) 96–101
4. Youyong Zou, Tim Finin and Harry Chen: F-OWL: an Inference Engine for the Semantic Web, http:// ebiquity.umbc.edu/get/a/publication/128.pdf
5. Youyong Zou: Agent-Based Services for the Semantic Web, Ph.D. Dissertation, University of Maryland, Baltimore County, August (2004)
6. Savas Parastatidis, Jim Webber, Paul Watson, Thomas Rischbeck: A Grid Application Framework based on Web Services Specifications and Practices, North East Regional e-Science Centre School of Computing Science, University of Newcastle, Newcastle-upon-Tyne, NE1 7RU, United Kingdom (2003)
7. Khoroshevsky V.: OntosMiner: Multilingual Information Extraction Systems Family. Proceedings of the CAI'2004, Tver, Russia, 2004 (In Russian)
8. Efimenko, I.V., Khoroshevsky, V.F., Klintsov, V.P.: Multilingual Content Extraction Systems: Ontology-Driven Semantic Analysis, In the Proceedings of International Conference SPECOM-2005, Greece (2005)
9. Bob DuCharme: RDF: Store Metadata About Anything, Anywhere, Dr. Dobbs Magazine, April (2005)
10. Hardware and Software Requirements (SQL Server Express), http://msdn2.microsoft.com/en-us/library/ms143680.aspx
11. Oracle Database Online Documentation 10g Release 2 (10.2), http://www.oracle.com/pls/db102/homepage

Robust Agent Communities

Sandip Sen, Sabyasachi Saha, Stéphane Airiau, Teddy Candale,
Dipyaman Banerjee, Doran Chakraborty, Partha Mukherjee, and Anil Gursel

Department of Mathematical & Computer Sciences
The University of Tulsa, USA
sandip@utulsa.edu

Abstract. We believe that intelligent information agents will represent their users interest in electronic marketplaces and other forums to trade, exchange, share, identify, and locate goods and services. Such information worlds will present unforeseen opportunities as well as challenges that can be best addressed by robust, self-sustaining agent communities. An agent community is a stable, adaptive group of self-interested agents that share common resources and must coordinate their efforts to effectively develop, utilize and nurture group resources and organization. More specifically, agents will need mechanisms to benefit from complementary expertise in the group, pool together resources to meet new demands and exploit transient opportunities, negotiate fair settlements, develop norms to facilitate coordination, exchange help and transfer knowledge between peers, secure the community against intruders, and learn to collaborate effectively. In this talk, I will summarize some of our research results on trust-based computing, negotiation, and learning that will enable intelligent agents to develop and sustain robust, adaptive, and successful agent communities.

1 Introduction

Humans are often referred to as social animals. What this implies is that societal relations and interactions are important and essential in our lives. Various modalities and temporal horizons of social interactions enrich our life and supports our material and spiritual pursuits. As agent researchers, we envision autonomous, intelligent agents as augmentations of our natural selves that can relieve us of some of our chores and responsibilities. While some of these tasks can be achieved individually, without input from others, a significant proportion of such tasks would require our agents to interact with agents of our peers or other members of our society. Such interactions may involve sharing of resources, give-and-take of information and expertise, collaborative efforts for a common cause, etc. Just as we humans thrive in our societies, these intelligent agents will contribute and benefit from artificial agent societies. Hence, *agents must be social entities*. Until that happens, our agents will necessarily be limited in potential, bereft of the opportunities provide by the social scaffolding that we enjoy, and constrained by the limited local resources, knowledge, and capabilities.

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 28–45, 2007.
© Springer-Verlag Berlin Heidelberg 2007

But before the vision of a vibrant, productive agent society becomes a reality, significant research investments in codifying and formalizing social protocols, norms, and mechanisms must be made. Our MultiAgent SysTEms ReSearch (MASTERS) research group at the University of Tulsa have been studying, for more than a decade, mechanisms, algorithms, and techniques to foster social collaboration and coordination mechanisms ranging from negotiation schemes to trust-based reasoning, predictive and proactive coordination protocols to multiagent learning algorithms. In particular, we emphasize the value of a stable environment for agents where they choose relationships to enter in, resources to share, commitments to make and keep, agents to interact with, etc. While monetary transactions are necessary for one-time interactions or in unstable societies with significant flux in agent compositions, our work focuses on environments where agents are resident for significant time periods and hence interact repeatedly with the same agents. Thus we can utilize rich interaction modalities and principles including trust, belief, context, history, future expectations, shared values, etc. that allow for some of the same richness of interaction that we enjoy in human societies. We are primarily interested in leveraging history of interactions to identify trustworthy partners to engage in long-term relationships. We are actively working on developing principles and mechanisms that allow agents to build and sustain communities. Our vision is that of robust agent communities that form, grow, and flourish from proactive collaboration, that benefit from each other's resources and expertise, that encourages active cooperation to develop, sustain, secure, and enrich its inhabitants.

In this invited talk, I present representative results from some of these investigations. As it is not possible to do justice to all of these research, I have chosen to highlight the primary motivation, technical approach and summary contribution from a select few recent research efforts. I refer the reader to our website (`http://www.mcs.utulsa.edu/~sandip`) for a list of our papers which would provide a more thorough and representative overview of our research.

We will overview the following subset of our research areas:

- In Section 2 we present results from our work on trusted, reciprocal relationship maintenance in agent communities and its applications in P2P networks.
- In Section 3 we present a multiagent learning scheme that solve well-known social dilemma problems like the Prisoner's dilemma,
- In Section 4 we present a framework by which one agent in the community can teach classification knowledge to another agent without knowledge of the latter's knowledge representation or learning algorithms.
- In Section 5 we present results on emergence of social norms when populations of agents interact repeatedly,
- In Section 6 we outline a study that involves the use of simple learning strategies to identify optimal partnerships in a large population.
- In Section 7 we overview a protocol for negotiating fair and efficient allocation of multiple, indivisible resources.
- In Section 8 we present strategies that allow agents to bid for bundles of items from concurrent auctions selling individual items.

- In Section 9 we present techniques that allow agents to share referrals about service providers that allow the agent community to settle in states where all agents are satisfied with their current service provider.
- In Section 10 we outline techniques for improving security and data integrity in sensor networks by detecting malfunctioning nodes.
- In Section 11 we summarize a probabilistic reasoning mechanism for detecting deviations from team plans by a team member.

2 Reciprocal Relationships

Our work on reciprocal relationships allow self-interested agents to leverage complementary expertise in stable agent communities. An agent can ask for help for a task from an expert in that task type. The help from the expert saves the requesting agent significant cost which is greater than the cost incurred by the expert agent while helping. In another situation, the role of the helping and the helped agents may be reversed. Thus, if the environment has sufficient cooperation possibilities, self-interested agent will find it prudent to enter into reciprocal relationships. Our work allows agents to develop, nurture, and sustain such relationships while avoiding exploitative agents who receive help but do not reciprocate.

In our early work on reciprocity [1,2], we had a restrictive assumption that agents have fixed strategies for deciding whether or not to help other agents. More specifically, agents were either reciprocative, selfish (never returned help), philanthropic (always helped) or individual (never gave or received help). Agents with specified strategies interacted repeatedly over a sustained period of time and their effectiveness was calculated as a function of the total cost incurred to complete all assigned tasks and the agents never changed their expertise.

A more realistic scenario would be to give an agent the freedom of choosing from one of several help-giving strategies and to change its strategy as dictated by the environmental conditions. An agent may be inclined to adopt a strategy if agents using that strategy is observed to be performing better than others. Such a strategy adoption method leads to an evolutionary process with a dynamically changing group composition of agent strategies [3]. In [4], we present a mathematical model to capture the dynamics of the agent population. Using this analytical model, one can predict the population distribution of future given the initial environmental settings. Given the initial strategy profile in the population and the assigned task load, we have constructed decision surfaces using which a rational agent can choose the most beneficial strategy for the long run.

2.1 Resisting Free Riding and Collusion in P2P Networks

Peer-to-peer (P2P) systems enable users to share resources in a networked environment. P2P systems are vulnerable to problems including free-riding users who utilize resources without contributing in turn, collusion between groups of users to falsely promote or malign other users, and zero-cost identity problem that allows nodes to obliterate unfavorable history without incurring any expenditure. We used a reciprocity-based mechanism to tackle these problems [5]. We assume each node in the P2P network is managed by a self-interested agent. Our mechanism rewards an agent by enhancing its probability to receive resources it requested only when the node itself shares its own resources with other nodes in the system. This motivates nodes to contribute resources instead of free-riding.

In a P2P network, the tasks can be mapped into resources that an agent requires at a particular instance of time. Every agent has expertise in resource type $T \in \Upsilon$ where Υ is the set of all such resource types. Agents request resources of types in which they are not experts from other agents. The probability that an agent has a particular resource of a given type is much higher if an agent is an expert in that resource type than when it is not. An agent helps another agent if it provides a resource that is requested from it. Reciprocative agents return help, selfish agents try to free-ride. The expected utility of agent m for interacting with agent o requesting a resource type τ at time T is

$$E_T(m,o,\tau) = \sum_{t=T}^{\infty} \gamma^{t-T} [\sum_{x \in \Upsilon} (D_m^t(x) \mathrm{Pr}_{m,o}^t(x) cost_m(x)) - \\ - \sum_{x \in \Upsilon} (D_{o,m}^t(x) \mathrm{Pr}_{o,m}^t(x) cost_m(x))],$$

where $cost_i(x)$ is the expected cost that i incurs to procure a resource of type x by itself, γ is the time discount, and Υ is the set of different area of expertise. The evaluation of the expected utility of agent m helping agent o considers all possible interactions in future and for all types of resources. In the above equation, $D_m^t(x)$ is the expected future distribution of resource types that agent m may require at time instance t, and $D_{o,m}^t(x)$ is the expected future distribution of resource types that agent o may ask from m at time instance t. We define $\mathrm{Pr}_{i,j}^t(x)$ as the probability that agent j will share a resource of type x, when requested by agent i at time step t.

For a sufficiently large agent population, interaction between any two given agent may be infrequent, and it can take a long time to ensure enough interaction among agents to build up informative interaction histories. To alleviate this problem, we propose to use a reputation mechanism. In this reputation framework, when an agent m is asked for help by another agent o, m requests other agents, $\mathcal{C}$, who have interacted with o before to share their experiences about o. Upon request, the $\mathcal{C}$ agents report their complete interaction history with o to m. The helping agent, m, then uses this information to compute a more accurate probability of o's help-offering behavior for different resource types by weighing its personal experience with o and the average of the probabilities reported by the $\mathcal{C}$ agents. Therefore, the $\mathrm{Pr}_{m,o}^T(x)$ term in the previous equation is replaced by $\mathrm{Pr}_o^T(x)$, the reputation of o for providing help for task type x. $\mathrm{Pr}_o^T(x)$, is calculated as $\mathrm{Pr}_o^T(x) = (1-\alpha)\mathrm{Pr}_{m,o}^T(x) + \alpha \frac{\sum_{a \in \mathcal{A}-\{m,o\}} \mathrm{Pr}_{a,o}^T(x)}{|\mathcal{A}|-2}$ where $\mathrm{Pr}_{a,o}^T(x)$ is the opinion about o reported by a. These opinions are averaged from all agents except the interacting parties and the weight α on others opinion is an inverse function on the number of times m has asked o for help.

Agents, however, may collude to disrupt this mechanism by reporting good opinions about other colluders to third parties. We propose a Bayesian update scheme to discriminate between truthful and lying agent. In this approach, reciprocative agents assumes every one to be truthful and then uses a Bayesian update technique to judge the truthfulness of each agent based on its interaction experience with those about whom reputation was reported. Subsequently, the opinions reported by an agent is weighted by its estimated truthfulness. Experimental results showed that our mechanism effectively removes the free-riding, zero-cost identity and collusion problem in a P2P network.

2.2 Reciprocity Between Super-Peers

Super-peer networks have been proposed to address the issue of search latency and scalability in traditional peer-to-peer (P2P) networks. In a super-peer network, instead of having a fully distributed systems of peer nodes with similar or comparable capabilities, some nodes that possess considerable computing power and resources are designated as super-peers. We address the problem of mutual selection by super-peers and client peers. In particular, we evaluate alternative decision functions used by super-peers to allow new client peers to join the cluster of clients under it. By formally representing and reasoning with capability and query distributions, we develop peer-selection functions that either promote concentration or diversification of capabilities within a cluster, and evaluate their effectiveness of different peer-selection functions for different environments where peer capabilities are aligned or are independent of their queries. Super-peers are responsible to find other peers which can provide an answer to a query, either by using peers from its pool of clients, or by requesting help from other super-peers. Our goal is to dynamically build the network of super-peers from a fully distributed network and ensure that peers are contributing to the community. Super-peers use a reciprocity mechanism to ensure that there are no free-riders in the system [6]. Each super-peer also ensures that all its client peers are contributing by enforcing load balancing within its cluster of client peers.

3 Learning to Solve Social Dilemmas

Agents in a society are often confronted by social dilemmas. We formulate such social dilemma as general-sum game repeatedly played by self-interested agents and use a learning approach to solve such problems. Most exist learning mechanisms developed for game playing assume complete and perfect information, i.e., players can observe the payoffs received by all the players. This may not be possible in a large number of real environments and we assume players can observe the actions of all other players but not their payoffs. Rather than convergence to any Nash equilibrium strategy profile, we prefer Pareto-Optimal outcomes that also generate a Nash Equilibrium payoff (POSNE) for repeated two-player, n-action general-sum games. We introduce the Conditional Joint Action Learner (CJAL) which learns the conditional probability of an action taken by the opponent given its own actions and uses it to decide its next course of action [7].

We assume repeated play of a stage game by a set S of 2 agents where each agent $i \in S$ has a set of actions A_i. We use the following notations: $E_t^i(a_i)$ is the expected utility of an agent i at iteration t for an action a_i, $Pr_t^i(a_i)$ is the probability that agent i plays action a_i at iteration t, and $Pr_t^i(a_j|a_i)$ is the conditional probability that the other agent, j, will play a_j given that the i^{th} agent plays a_i at iteration t. The joint probability of an action pair (a_i, a_j) at iteration t is given by $Pr_t(a_i, a_j)$.

A *CJAL* learner is an agent i who at any iteration t chooses an action $a_i \in A_i$ with a probability proportional to $E_t^i(a_i) = \sum_{a_j \in A_j} U_i(a_i, a_j) Pr_t^i(a_j|a_i)$, where a_j is the action taken by the other agent. These expectations are learned by using Q-learning:$E_t^i(a_i) = \sum_{a_j \in A_j} Q_t^i(a_i, a_j) * \frac{n_{t-1}^i(a_i,a_j)}{n_{t-1}^i(a_i)}$, where $n_t^i(a_i) = \sum_{a_j \in A_j} n_t^i(a_i, a_j)$ is the number of times agent i has played action a_i until iteration t and $Q_t^i(a_i, a_j)$, the estimated payoff from joint action (a_i, a_j) is updated after the $(t-1)th$ interaction as

$$Q_t^i(a_i, a_j) \leftarrow Q_{t-1}^i(a_i, a_j) + \alpha(U_i(a_i, a_j) - Q_{t-1}^i(a_i, a_j)),$$

where $0 < \alpha \leq 1$ is the learning rate.

We empirically show that under self-play and if the payoff structure of the Prisoner's Dilemma game satisfies certain conditions, a CJAL learner, using a random exploration strategy followed by a completely greedy exploitation technique, will successfully resolve the Prisoner's dilemma and produce cooperation. We also experimentally demonstrated the convergence of CJAL using limited exploration in self-play to POSNE outcomes on a representative testbed containing all structurally distinct two-player conflict games with ordinal payoffs. Though CJAL was not explicitly designed to optimize measures like social welfare, fairness (measured by the product of player payoffs) and success in converging to POSNE outcomes, it outperforms well-known existing multiagent learning algorithms like JAL and WOLF-PHC on these metrics.

4 Agent-Teaching-Agent (ATA)

Few researchers have addressed the problems of one, knowledgeable, agent teaching another agent. We investigate how an agent can use its learned knowledge to train a peer agent in its community with a possibly different internal knowledge representation. The knowledge being transferred is a concept description, a boolean-valued function that classifies input examples as members or non-members of the target concept. We assume that the trainer agent does not have access to the internal knowledge representation of the trainee agent, but can evaluate its concept recognition abilities by asking it to categorize selected exemplars and non-exemplars of the target concept. The trainer agent also do not have access to the original training set from which it learned its concept description. We have developed an Agent Teaching Agent (ATA) framework which focuses on incremental selection of training examples by the trainer to expedite the learning of the trainee agent [8,9].

We envisage an iterative training procedure in which alternatively the trainer selects a set of training and testing exemplars, the trainee trains using the training set and then classifies the testing set, the trainer observes errors made by the trainee in classifying the instances in the last testing set and accordingly generates the next training and testing sets. This iterative process converges when error of the trainee falls below a given threshold. We now present these iterative training steps in an algorithmic form:

Procedure Train-Agent(Trainer,Trainee,Domain-Info){
Select initial training set N_0 and initial testing set T_0 from Domain-Info
$i \leftarrow 0$
repeat
  Train trainee agent on training set N_i
  Let M_i be the instances in T_i misclassified by trainee after training on N_i
  $T_{i+1} \leftarrow T_i \cup newTestInstances(M_i)$ and Domain-Info
  $N_{i+1} \leftarrow N_i \cup newTrainingInstaces(M_i)$
  $i \leftarrow i + 1$
until $|M_i| < threshold$}

This procedure needs to be fleshed out to realize an actual implementation. In particular, we have to specify procedures for selection of the initial training and testing sets, N_0 and T_0, and the generation of the next test set T_{i+1} based on the mistakes, M_i, made by the trainee on the current test set. We have developed these procedures for instance based and decision tree learners to work on problems with real-valued attributes. When selecting the initial training and testing instances, the goal is to select the most discriminating examples that help identify regions of the input space that do and do not belong to the target concept. For example, if a hyperplane separates instances of the target concept from non-instances, then points close to and on both sides of that hyperplane should be selected as initial training and testing set members. When selecting the next set of training and testing instances, the goal is to first isolate the mistakes made on the previous test set, and for each of these instances, find a few neighboring points, use some of them as part of the training data and the rest as part of the test data in the following iteration. The true classification of these points will not be known in general, and only their estimated classification, based on the concept description knowledge previously acquired by the trainer, can be used.

In our initial experiments with instance-based and decision tree learners as training and trainee agents we found that incremental training results in rapid improvement in classification performance of the trainee agent over the entire test set. The final accuracy is comparable to the accuracy of the trainer's knowledge or the trainee's knowledge if it had access to the original training set. Particularly interesting results were obtained with the Haberman data set obtained from the UCI repository: IB2, an instance-based learner, acting as a trainer can train C4.5, a decision-tree learner, to have better testing accuracy than itself. The trainer can, via the iterative training process, produce a more competent trainee! We have used this framework to train new agents joining a team of experts [8].

5 Social Learning of Norms

Behavioral norms are key ingredients that allow agent coordination where societal laws do not sufficiently constrain agent behaviors. Whereas social laws need to be enforced in a top-down manner, norms evolve in a bottom-up manner and are typically more self-enforcing. While effective norms and social conventions can significantly enhance performance of individual agents and agent societies and have merited in-depth studies in the social sciencethere has been little work in multiagent systems on the formation of social norms.

We have recently used a model that supports the emergence of social norms via learning from interaction experiences. In our model, individual agents repeatedly interact with other agents in the society over instances of a given scenario [10]. Each interaction is framed as a stage game. An agent learns its policy to play the game over repeated interactions with multiple agents. We term this mode of learning *social learning*, which is distinct from an agent learning from repeated interactions against the same player. We are particularly interested in situations where multiple action combinations yield the same optimal payoff. The key research question is to find out if the entire population learns to converge to a consistent norm.

The specific social learning situation for norm evolution that we consider is that of learning "rules of the road". In particular, we have considered the problem of which side of the road to drive in and who yields if two drivers arrive at an interaction at the same time from neighboring roads [10]. When two cars arrive at an intersection, a driver will sometimes have another car on its left and sometimes on its right. These two experiences can be mapped to two different roles an agent can assume in this social dilemma scenario and corresponds to an agent playing as the row and column player respectively. Consequently, an agent has a private bimatrix: a matrix when it is the row player, one matrix when it is the column player. For this problem, we use a bimatrix where both players get a high value (4) if they choose the same action and a low payoff (-1) otherwise. Note that either action combinations (0,0) or (1,1) would be equally desirable. Each agent has a learning algorithm to play as a row player and as a column player and learns independently to play as a row and a column player. An agent can observe opponent action but not their payoff. The goal is then for all agent to develop a norm of choosing the same action consistently.

Each agent is paired in each time period for interaction with a randomly selected agent from a subset of the population. An agent is randomly assigned to be the row or column player in any interaction. We assume that the stage game payoff matrix is known to both players, but agents cannot distinguish between other players in the population. Hence, each agent can only develop a single pair of policies, one as a row player and the other as a column player, to play against any other player from the agent population.

In our initial experiments, any two agents in the population had an equal probability of interaction. We observed that social learning was successful in generating consistent norms in the population. The main conclusions of this study was as follows:

• The number of interactions required to evolve a consistent norm increases with the population size and the number of actions, m, available to each agent.
• The number of interactions required to evolve a consistent norm varies with the learning algorithm used by the agents. If different agents used different learning algorithms (heterogeneous learning population), the convergence rate is in between the rates for homogeneous populations using the constituent learning algorithms.
• Different norms producing equal payoffs emerged equally often over different runs. However, when we introduced non-learners, i.e., fixed-strategy agents who always chose a given action (for example, always driving on the right), only a handful of additional non-learners following a given norm compared to others led to the corresponding norm emerging significantly more often in the population.
• If the population was segregated with very infrequent interactions between agents belonging to different sub-populations, different norms could emerge in different sub-population. It was surprising to see divergent norms emerging even when 25% of the interactions were across sub-populations.

In a more recent paper [11], we have enhanced the interaction model to study spatial interaction effects on norm emergence. In this enhanced model, the agents are distributed over space where each agent is located at a grid point. An agent is allowed to interact only with agents located within its neighborhood composed of all agents within a distance D of its grid location (we have used the Manhattan distance metric, i.e., $|x_1 - x_2| + |y_1 - y_2|$ is the distance between grid locations (x_1, y_1) and (x_2, y_2)). We vary D to allow for different neighborhood sizes. We have experimented with a society of 225 agents placed on a 15 by 15 grid and using the WoLF-PHC learning scheme.

We present in Figure 1 the dynamics of the average payoff of the population over a run when all agents are learning concurrently. We conclude that a norm has emerged in the population when the average payoff of the population reaches 3.5. From Figure 1 we observe that the smaller the neighborhood distance, the faster the emergence of a norm. This is because, for a given number of iterations, the agents interact more often with a particular neighbors for smaller neighborhoods. This means that the impact an agent has on another agent is larger when the neighborhood size is small. In addition, an agent with few neighbors will encounter few different behaviors from its neighbors, and it is *a priori* easier to coordinate with a small set of agents rather than a larger one. the decreasing interaction frequency between pairs of learners increases the time for exploration of the behavior space and thereby influences the learning patterns of the agents in the network.

6 Finding Partners

Human and artificial agents routinely make critical choices about interaction partners. The decision about which of several possible candidates to interact

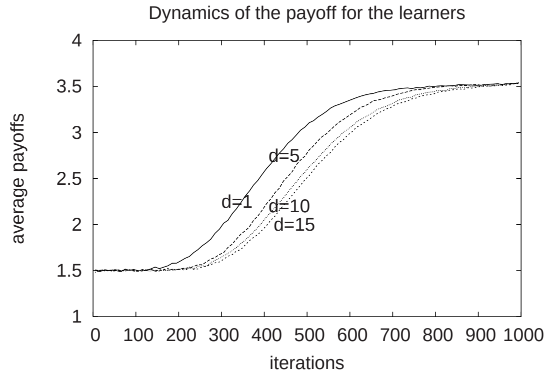


Fig. 1. Influence of neighborhood size on learning rate. All agents are learning.

with has significant importance on the competitiveness, survivability, and overall utility of an agent. We assume that an agent has time and resource constraints that limit its participation to only a fixed number, k, of relationships or interactions with other agents in a particular time period. Therefore, in a given time period, an agent is free to choose to interact with any k other agents from a society of N agents. A bilateral relationship is established in a time period, however, if both agents choose to do so. The goal of this research is to investigate the extent to which well-known, simple learning schemes can identify and sustain mutually beneficial relationships in these conditions [12].

A number of multiagent learning algorithms have been developed recently that converge to equilibrium in repeated play. We, however, believe that it is much more likely that simple, single-agent reinforcement learning techniques will be used by a large majority of the agents that in open real-world environments. Therefore, we use Q-learning as the learning algorithm used by our population of agents. We do not know of any other research that has attempted such massively concurrent learning by a large number of utility maximizing agents using single-agent reinforcement learning techniques where the agent utilities are closely coupled. Not only is the likelihood of convergence of such interlinked learning to effective selections unclear *a priori*, no weak guarantees about performance can also be provided. Our experiments, however, show that independent Q-learning by concurrent learners with sufficient exploration is surprisingly robust in identifying most of the mutually beneficial relationships in the society.

7 Negotiating Fair Allocation of Resources

We study the problem of autonomous agents negotiating the allocation of multiple indivisible resources. It is difficult to reach optimal outcomes in bilateral or multi-lateral negotiations over multiple resources when the agents' preferences for the resources are not common knowledge. Self-interested agents often end up negotiating inefficient agreements in such situations. We have developed a protocol for negotiation over multiple indivisible resources which can be used by rational agents to reach efficient outcomes [13]. Our proposed protocol enables the negotiating agents to identify efficient solutions using systematic distributed search that visits only a subspace of the whole solution space.

We represent a negotiation scenario for allocation of multiple indivisible resources as a 3-tuple $\langle \mathcal{A}, R, \mathcal{U} \rangle$, where $\mathcal{A} = \{1, 2\}$ is the set of agents, $R = \{r_1, r_2, \ldots, r_H\}$, $H \geq 2$, is the set of H indivisible resources whose allocation are being negotiated, and $\mathcal{U} = \{U_1, U_2\}$ is the set of utility functions, where U_i is the utility function of agent i. Each resource is considered as a negotiation issue. The negotiating agents must agree on the allocation of the resources.

We assume that the issues or resources are ordered, e.g., lexicographically. We conceptualize the allocations of the resources as a binary *negotiation tree*. The root node represents a null allocation to the agents and then each successive level represents allocation of the next resource in the order chosen. The left and right branches at the l^{th} level imply that the l^{th} resource will be allocated to agent 1 and 2 respectively. Each leaf node at level H represents one possible allocation of the resources and the path from the root node to that leaf node lists the allocation of all the resources. A negotiation tree is created by the negotiating agents in a distributed, top-down process starting at the root node. At any level, agent 1 can only create the right child of a node in the previous level of the tree. Similarly, agent 2 can only create the left child nodes. Each agent, however, may choose not to create any of the nodes it can create, and such a node will be marked as *black* node and it will be pruned from the negotiation tree.

At each node of the negotiation tree, each agent has a *best possible agreement (BPA)* which is the allocation where the resources until the current level are allocated according to the path from the tree root to this node and the remaining resources are allocated to this agent. An individually rational agent will prune a node whose BPA utility is less than the utility it can receive otherwise.

The *Protocol to reach Optimal agreement in Negotiation Over Multiple Indivisible Resources (PONOMIR)* consists of three phases. The first phase consists of a primary allocation procedure using any one of *strict alteration* or *balanced alteration* protocol to produce a default allocation L. The second phase consists of distributed formation of the negotiation tree by the negotiating agents where agents prune nodes as mentioned above. If no nodes are created at level $l < H$, the final allocation is L. Otherwise L and the nodes at level H make up the probable agreements Q at the end of the second phase. In the third phase, agents reach the final Pareto optimal solution by exchanging offers from Q. Agents take turn in making offers from Q, and the recipient removes agreements from Q that are dominated by the received offer. When Q cannot be reduced further, an agreement is picked randomly from it.

Our goal was to develop protocols that lead rational agents to Pareto optimal agreements and to increase fairness as much as possible. As a measure of fairness, we use *egalitarian social welfare*. PONOMIR is not strategy-proof and does not guarantee Pareto optimal agreements if agents are arbitrarily risk seeking. The rational behavior of the agents, who have no prior knowledge of the preferences of the other agents, depends on their risk attitudes. We assume that such agents will be *cooperative-individually rational* which means that i) an agent will not take any risky action that can lead to an agreement which produces less utility than what it is already assured of, and, ii) if there exists two agreements which produces same utility to it but different utility to the opponent, the agent will agree to accept any of the agreement proposed by the opponent. PONOMIR guarantees Pareto optimal agreements if the participating agents are cooperative-individually rational. The agreements reached also guarantees at least as much egalitarian social welfare as the agreements reached by the existing protocols.

8 Bidding for Bundles in Auctions

Agents with preferences for bundles of items are faced with a difficult computational problem when each of the several electronic auctions sell only one item. While an optimal bidding strategy is known when bidding for item bundles in sequential auctions, only suboptimal strategies are known for simultaneous auctions. We investigate a multi-dimensional bid improvement scheme, motivated by optimization techniques, to derive optimal bids for item bundles in simultaneous auctions [14].

We consider multiple sealed-bid auctions offering items from the set $\mathcal{I}$. A valuation function ϑ expresses the bidder's preferences for bundles or subsets of items from the set $\mathcal{I}$, i.e., the bidder is willing to pay up to $\vartheta(I)$ for a bundle of items $I \subseteq \mathcal{I}$. Each item i is available only in the single-item single-unit auction a_i. We do not specify the particular auction type but make the *exogenous price* assumption: the bids of our bidder do not influence the auction closing prices. An auction is modeled by the probability distribution F_i of the closing prices of the item being offered in that particular auction. We assume these distributions to be continuous, independent, and known by the bidder. In practice approximate price distributions can be learned from observing electronic markets. When an auction closes, a closing price $p_i \in [\underline{p_i}, \overline{p_i}]$ is drawn from the distribution F_i. The bidder gets the item if it has placed a bid b_i greater than or equal to the closing price, i.e., if $p_i \leq b_i$, and the winning payment is equal to the closing price p_i. All auctions run in parallel and their closing times are not known by the bidder. The bidder place bids represented by $B = (b_1, \ldots, b_N) \in \mathcal{B}$ where $\mathcal{B}$ is the bid domain for all auctions. Replacing a bid is not allowed in this model.

Once all the auctions close, the bidder can compute its utility $\alpha(B, P)$ where $P = (p_1, \ldots, p_N)$ represents the closing prices of all auctions. The set of acquired items $\mathcal{I}_{ac}(B, P)$ is calculated as $\mathcal{I}_{ac}(B, P) = \{i \in \mathcal{I} \ s.t. \ p_i \leq b_i\}$ and the corresponding utility received by the bidder is $\alpha(B, P) = \vartheta(\mathcal{I}_{ac}(B, P)) - \sum_{i \in \mathcal{I}_{ac}(B, P)} p_i$.
The expected utility is then $\bar{\alpha}(B) = E_P[\alpha(B, P)]$ which can be calculated as specified in Proposition 1:

Proposition 1 (Expected utility)

$$\bar{\alpha}(B) = \sum_{I \subseteq \mathcal{I}} \left\{ \left(\prod_{i \in I} F_i(b_i) \right) \left(\prod_{j \notin I} (1 - F_j(b_j)) \right) \vartheta(I) \right\} - \sum_{i=1}^{N} \int_{\underline{p_i}}^{b_i} p_i\, f_i(p_i)\ dp_i,$$

where $F_i(b_i) = Pr\{p_i \leq b_i\} = \int_{\underline{p_i}}^{b_i} f_i(p_i)\, \mathrm{d}p_i$ and f_i is the pdf of F_i. Our research objective is to find a bid vector B^* which maximizes the expected utility $\bar{\alpha}$: $B^* = \underset{B \in \mathcal{B}}{\operatorname{argmax}}\ \bar{\alpha}(B)$. Assume that the bidder has decided by some means to bid the vector B. If B is sub-optimal, there is at least one item whose bid can be improved, i.e., there exist i and δ_i such that $\bar{\alpha}(B) < \bar{\alpha}((b_i + \delta_i) \vee B_{-i})$ where $B_{-i} = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_N)$ and $b'_i \vee B_{-i} = (b_1, \ldots, b_{i-1}, b'_i, b_{i+1}, \ldots, b_N)$. By repeating this process, we can realize the best improvement possible for the item i, which is equivalent to maximizing the function $b_i \mapsto \bar{\alpha}(b_i \vee B_{-i})$.

Definition 1 (Optimal bid for item i). *$\beta_i(B_{-i})$ is the optimal bid for item i given bids for item $j \neq i$ is fixed: $\beta_i(B_{-i}) = \underset{b_i \in [\underline{p_i}, \overline{p_i}]}{\operatorname{argmax}}\ \bar{\alpha}(b_i \vee B_{-i})$.*

Proposition 2 (Optimal bid for item i)

$$\beta_i(B_{-i}) = \sum_{\substack{I \subseteq \mathcal{I} \\ i \in I}} \prod_{\substack{j \in I \\ j \neq i}} F_j(b_j) \prod_{l \notin I} (1 - F_l(b_l)) \vartheta(I) - \sum_{\substack{J \subseteq \mathcal{I} \\ i \notin J}} \prod_{j \in J} F_j(b_j) \prod_{\substack{l \notin J \\ l \neq i}} (1 - F_l(b_l)) \vartheta(J).$$

To implement this solution approach, an initial bid vector B is chosen and N components of this bid vector are repeatedly improved in any predetermined order. Improving the bid for item i involves replacing b_i by $\beta_i(B_{-i})$: $B \leftarrow \beta_i(B_{-i}) \vee B_{-i}$. We refer to this improvement as *single improvement* and the sequence of N improvements as *N-sequential improvement*. In the bid domain $\mathcal{B}$, the single improvement can be regarded as going from B to the hyper-surface $b_i = \beta(B_{-i})$ by moving parallel to the b_i-axes. The process is stopped when no further improvement can be made. We refer to this sequential bid-improvement process as a Multi-Dimensional Bid Improvement (MDBI) scheme.

The MDBI algorithm produces optimal bid vectors with infinite random restarts. As this is infeasible in practice, we evaluated variants of the scheme with finite number of restarts and carefully chosen starting bid vectors. A surprising result from these experiments is that good solutions can be achieved without restarts when items are substitutable, complementary, or are non-related. In general, small number of restarts can be used to approximate optimal expected utility in all cases. A desirable property of our algorithm, in contrast to existing schemes, is its approximately linear time complexity. Experimental results show that the MDBI scheme scales up effectively with larger number of items.

9 Selecting Service Providers Based on Referrals

Agents searching for high-quality services can use either their own interaction experience or referrals from peer agents. We assume that agents want a quality of service that exceeds an acceptable performance threshold. The performance of a resource depends on its intrinsic characteristics and is inversely correlated to the current workload it is handling. Individual agent satisfaction depends both on the resource selected and choices made by the other agents. We study efficient decentralized protocols for finding satisfying resources [15]. Locally optimal actions can increase the number of conflicts of interests where resources are shared. Referrals from other agents can help agents find more satisfying service providers. However, such referrals may cost the referring agent since the load of the referred provider may increase, with corresponding performance deterioration.

Framework: Let $\mathcal{E} =< \mathcal{A}, \mathcal{R}, perf, L, S, \Gamma >$ where: (i) $\mathcal{A} = \{a_k\}_{k=1..K}$ is the set of agents, (ii) $\mathcal{R} = \{r_n\}_{n=1..N}$ is the set of resources, (iii) $f: \mathcal{R} \times \mathbb{R} \rightarrow [0,\ 1]$, intrinsic performance function of a provider, (iv) $L = \mathcal{A} \rightarrow \mathbb{R}_+$, daily load assigned to agents, (v) $S:\ \mathcal{A} \times [0,1] \rightarrow [0,\ 1]$, satisfaction function for agents, (vi) $\Gamma = \{\gamma_1,\ \ldots,\ \gamma_K\}$, set of satisfaction thresholds of agents. If a set $\mathcal{A}_n^d$ of agents use the provider r_n on day d then the feedback received by every agent in $\mathcal{A}_n^d$ at the end of the day d is $perf = f\left(r_n,\ \sum_{a \in \mathcal{A}_n^d} L(a)\right)$. An agent $a_k \in \mathcal{A}_n^d$ evaluates the performance of r_n by the the satisfaction it obtained and is given by $s = S(a_k,\ perf)$. An agent is satisfied if $s > \gamma_k$.

Definition 2 (Distribution of agents over providers). *We call* **distribution of agents over providers** *the set* $D = \{\mathcal{A}_n\}_{n=1..N}$ *where* $\mathcal{A}_n$ *is the set of agents using resource* r_n. *The set of distributions is denoted by* $\mathcal{D}$.

Definition 3 (Γ-acceptable distribution). *A distribution* D *is said to be* **Γ-acceptable** *if each agent is satisfied by the resource they use in* D. *The set of* Γ*-acceptable distributions is denoted by* $\mathcal{D}_\Gamma$.

A Γ-acceptable distribution is an equilibrium concept and our goal is to develop mechanisms that enable a group of agents to converge to such a distribution. We present alternative strategies for selecting service providers.We evaluate three kinds of agents: agents who find providers on their own without using information from other agents (*No Referral* or *NR*), agents who use referral to locate providers and are trustful of the referrals received(*Referral (Truthful)* or *RT*).

Definition 4 (Entropy). *Given an environment where agents are identical and resource* r_n *has capacity* C_n, *we call entropy of a distribution* D: $\mathcal{E}(D) = \sum_{n=1}^{N} \max(0,\ |\mathcal{A}_n| - C_n)$.

Each Γ-acceptable distribution has zero entropy. The lower the entropy the better the distribution since less agents are unsatisfied.

We claim that when agents choose their actions based on local perspective only, the system is likely to move from a distribution with a low entropy to one with a higher entropy and vice and versa. Such oscillations can be controlled by limiting the number of agents moving simultaneously, K_{move}. We experimentally show K_{move} has a critical influence on the convergence speed. A high value for K_{move} leads to system oscillations and hence is undesirable.

Experimental results:
1. There exists a lower bound of the number of resources N^* for effective system performance. For any agent type, convergence speed is optimal when $N = N^*$.
2. For any agent type, performance in Zone I, i.e., for $N < N^*$ is worse compared to performance in Zone II, i.e., for $N \geq N^*$.
3. When $N \geq N^*$ for all strategies, i.e., in the range $N \geq 100$: *RT* converges faster than *NR*.
4. *NR* is more robust than other algorithms as it produces convergence for a much larger range of environments, e.g., only *NR* leads to convergence within the iteration limit for $N \leq 20$.

Interestingly, systems without referrals appear to be more robust in the sense they have satisfactory or reasonable performance even for extremely small number of providers, i.e., for more challenging environments. Referrals, however, do facilitate convergence when there are a significant number of providers.

10 Detecting Malfunctioning Nodes in Sensor Networks

Sensor network applications often require remote, distributed monitoring of inaccessible and hostile locations. These networks are vulnerable to both physical and electronic security breaches. Nodes in a sensor network can be hierarchically organized, where the non-leaf nodes serve as the aggregators of the data sensed at the leaf level and data collected from the entire network is available at the root node to users for monitoring the environment. Erroneous data, either from compromised or defective nodes, can influence the aggregated result and can undermine the effectiveness of the network for effectively reporting data about the environment. Current research on sensor networks use outlier detection mechanisms by a parent node to detect the erroneous nodes among its children. It is assumed that the data reported by the children of a node is sampled from the same distribution and they are of almost equal values. We believe that effective, distributed attack mechanisms will eschew egregious deviations and report smaller errors by a colluding group of compromised nodes. We have developed a distributed reputation framework that can learn to recognize such distributed attacks over repeated data aggregation periods [16].

The above approaches, however, will be ineffective for networks where the data sensed vary widely from one portion to another as the data sensed by different nodes are not from the same distribution. In such a scenario, we have used neural network based learning technique to predict data reported by different nodes. We train the nets offline from sufficient data collected after initial network deployment [17]. Subsequently, parent nodes monitor their children by calculating the differences between the value reported by a child node and that predicted by the net based on the data reported by that node's siblings. Each node incrementally updates the reputations of its child nodes based on those calculated differences. We have used robust schemes like Q_learning and a Beta-reputation scheme to detect the faulty/malicious nodes. We have incorporated different degrees of node's physical and geometrical features (e.g varying coordinates of anomalous nodes, the number of malicious nodes, the errors imparted, data pattern etc) into our experiments and demonstrated robust system performance under varying environmental conditions.

11 Detecting Deviation from Team Plans

Effective decentralized control mechanism are required for multiple agents cooperating to achieve a common goal while limited by computing and communication limitations and possible security breaches. Multiagent planning techniques computing near-optimal joint-strategies that can handle intrinsic domain uncertainties. Uncertainties related to agents deviating from the recommended joint-policy, however, is typically not taken into consideration. We focus on hostile domains, where teams must quickly identify deviations from team plans by compromised agents. There is a growing need to develop techniques that enable the system to recognize and recover from such deviations. We have developed a distributed probabilistic intrusion detection scheme for detecting a particular type of deviations by team members [18].

The problem of decentralized control can be effectively modeled as a decentralized partially observable Markov Decision Process (DEC-POMDP). A DEC-POMDP is given by a tuple $< I, S, \{A_i\}, \{\Omega_i\}, O, P, R, b_0 >$ where I is the finite set of agents indexed by $1 \ldots n$, S is a finite set of states, A_i is a finite set of actions available to agent i and $A = \times_{i \in I} A_i$ is the set of joint actions where a $=< a_1, \ldots, a_n >$ denotes a joint action, Ω_i is a finite set of observations available to agent i and $\Omega = \times_{i \in I} \Omega_i$ is the set of joint observations where $o =< o_1, \ldots, o_n >$ denotes a joint observation, O is the observation function given by $O(s, a_1, \ldots, a_n, o_1, \ldots, o_n)$, the probability of observing the joint observation $(o_1, \ldots, o_n)$ when transitioning to state s after taking joint action $(a_1, \ldots, a_n)$, P is the set of Markovian state transition probabilities where $P(s, a, s')$ denotes the probability of taking action a in state s and reaching state s', R: $S \times A \rightarrow \Re$ is the common reward function, and b_0 is the initial belief state for all agents. We assume that the agent's observations are independent. Thus the observation function can be represented as $O = \times_{i \in I} O_i$ where $O_i(s, a_1, \ldots, a_n, o_i)$ is the probability that agent i observes o_i given the joint-action $< a_1, \ldots, a_n >$ resulted in state s. The decision problem spans over a finite horizon T. The policy for agent i, π_i is represented by a policy tree. Each node corresponds to an action and each edge corresponds to an observation that the agent makes at that time interval. We assume that a centralized planner computes the policy tree for each agent. The running belief state of agent i at time interval t is its estimate of the physical states and the observation histories of the other agents and is given by $RB_i^t : S \times \boldsymbol{o^t_{-i}} \rightarrow [0, 1]$ where $\boldsymbol{o^t_{-i}}$ are the *t'th* observation histories of other agents. We define Bel_i^t as the set of all such possible combinations of physical states and observation histories that have a positive probability in RB_i^t: $Bel_i^t = \{b | RB_i^t(b) > 0\}$. The agents update RB_i^t and Bel_i^t with each execution step.

Each agent i maintains a set $V_i = \{R_i^j\}$ where R_i^j is the reputation of the jth agent as computed by agent i, $\forall j \neq i$, and is updated in each iteration by:

$$R_i^j \leftarrow R_i^j - \kappa(R_i^j) \times \sum_{\forall <s, \boldsymbol{o_{-i}^t}> \in Bel_i^{t^{max}}} (max_{o_j \in \Omega_j} O_j(s, <\pi_i(\boldsymbol{o_i^{t-1}}), \pi_{-i}(\boldsymbol{o_{-i}^{t-1}})>, o_j)) - O_j(s, <\pi_i(\boldsymbol{o_i^{t-1}}), \pi_{-i}(\boldsymbol{o_{-i}^{t-1}})>, o_j^t>))/|Bel_i^{t^{max}}|$$

where $Bel_i^{t^{max}} = \{b | RB_i^t(b) = max_{b' \in Bel_i^t} RB_i^t(b')\}$. $Bel_i^{t^{max}}$ is a subset of beliefs most convincing to i. Based on $Bel_i^{t^{max}}$, i reasons about the last observational transition that each of the other agents have made. Note, a simple malicious agent k would often fake observations and this inconsistency would gradually reflect in the Bel_i^t of i and result in a higher value for the numerator. This would result in a sharp drop of R_i^k. The function κ is monotonically decreasing with R_i^j and thus facilitates faster detection. We have shown the effectiveness of this scheme on the Tiger problem [18].

12 Conclusions

While the brief summaries presented here provide only coarse outlines of the research results, my website (`http://www.mcs.utulsa.edu/~sandip`) can be perused both to obtain complete papers with extended discussion on these topics as well as to obtain details on related research of key relevance to the topic of developing sustainable agent communities. In addition, our research group has worked on related areas on multiagent learning, trust-based computing, peer-level negotiation schemes, proactive information dissemination, cooperative security envelops, etc. that are key components of the set of technologies required to design, develop and implement self-interested social agents. Such agents must balance local needs with societal constraints to maximize long-term utility. In particular, they have to leverage complementary expertise, proactively seek out collaboration opportunities, and cooperatively avoid unforeseen problems and inefficiencies. Our ongoing work is focused on techniques and methods to make our vision of a vibrant, self-sufficient intelligent agent community a reality in the foreseeable future.

Acknowledgments. This work has been supported in part by an NSF award IIS-0209208, a DOD-ARO Grant #W911NF-05-1-0285, and a NASA Research Initiation Grant NCC5-586. We also thank our collaborator Prithviraj Dasgupta.

References

1. Sen, S.: Reciprocity: a foundational principle for promoting cooperative behavior among self-interested agents. In: Proceedings of the Second International Conference on Multiagent Systems, Menlo Park, CA, AAAI Press (1996) 315–321
2. Sen, S.: Believing others: Pros and cons. Artificial Intelligence **142**(2) (2002) 179–203

3. Sen, S., Dutta, P.S.: The evolution and stability of cooperative traits. In: Proceedings of the First Intenational Joint Conference on Autonomous Agents and Multiagent Systems, New York, NY, ACM Pres (2002) 1114–1120
4. Saha, S., Sen, S.: Predicting agent strategy mix of evolving populations. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, New York, NY, ACM Pres (2005) 1075–1082
5. Banerjee, D., Saha, S., Dasgupta, P., Sen, S.: Reciprocal resource sharing in p2p environments. In: Proceedings of the Fourth Intenational Joint Conference on Autonomous Agents and Multiagent Systems, New York, NY, ACM Pres (2005) 853–859
6. Airiau, S., Sen, S., Dasgupta, P.: Effect of joining decisions on peer clusters. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents & Multi Agent Systems, New York, NY, ACM Pres (2006) 609–615
7. Banerjee, D., Sen, S.: Reaching pareto optimality in prisoner's dilemma using conditional joint action learning,. In: Working Notes of the AAAI-05 Workshop on Multiagent Learning (2005)
8. Chakraborty, D., Sen, S.: Teaching new teammates. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents & Multi Agent Systems, New York, NY, ACM Pres (2006) 691–693
9. Kar, P.P., Sen, S.: The agent-teaching-agent framework. In: Proceedings of the Second International Joint Conference on Autonomous Agents & Multi Agent Systems, New York, NY, ACM Pres (2003) 1028–1029
10. Sen, S., Airiau, S.: Emergence of norms through social learning. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence(IJCAI'07) (January 2007) 1507–1512
11. Mukherjee, P., Sen, S., Airiau, S.: Norm emergence in spatially constrained interactions. In: Working Notes of the Adaptive and Learning Agents Workshop at AAMAS'07 (2007)
12. Sen, S., Gursel, A., Airiau, S.: Learning to identify beneficial partners. In: Working Notes of the Adaptive and Learning Agents Workshop at AAMAS'07 (2007)
13. Saha, S., Sen, S.: An efficient protocol for negotiation over multiple indivisible resources. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence(IJCAI'07) (January 2007)
14. Candale, T., Sen, S.: Multi-dimensional bid improvement algorithm for simultaneous auctions,. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (2007)
15. Candale, T., Sen, S.: Effect of referrals on convergence to satisficing distributions. In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M.P., Wooldridge, M., eds.: Proc. 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), ACM (2005) 347–354
16. Basak, S., Sen, S.: Using distributed reputation management to preserve data integrity in sensor networks,. Journal of Autonomic and Trusted Computing (to appear)
17. Mukherjee, P., Sen, S.: Detecting malicious sensor nodes from learned data patterns. In: Working Notes of the Agent Technology for Sensor Networks workshop at AAMAS'07 (2007)
18. Chakraborty, D., Sen, S.: Distributed intrusion detection in partially observable markov decision processes. In: Proceedings of the Sixth International Joint Conference on Autonomous Agents & Multi Agent Systems (2007) (to appear)

WI Based Multi-aspect Data Analysis in a Brain Informatics Portal

Ning Zhong[1,2] and Shinichi Motomura[2]

[1] The International WIC Institute/BJUT
[2] Department of Life Science and Informatics
Maebashi Institute of Technology
460-1 Kamisadori-cho, Maebashi 371-0816, Japan
zhong@maebashi-it.ac.jp

Abstract. In order to investigate human information processing mechanism systematically, Web intelligence (WI) based portal techniques are required for brain data measurement, management and analysis. Building a brain informatics portal is, in fact, to develop a data mining grid centric multi-layer grid system on the Wisdom Web, on which various data mining agents are deployed, for multi-aspect data analysis. We propose an approach for collecting, modeling, transforming, managing, and mining multiple human brain data obtained from systematic fMRI/EEG experiments. The proposed approach provides a new way in Brain Informatics (BI) for automatic analysis and understanding of human brain data to replace human-expert centric visualization. We attempt to change the perspective of cognitive scientists from a single type of experimental data analysis towards a holistic view at a long-term, global field of vision to understand the principle, models and mechanisms of human information processing system.

1 Introduction

Human intelligence related research studies the nature of intelligence towards our understanding of intelligence. The capabilities of human intelligence can be broadly divided into two main aspects: perception and thinking. So far, the main disciplines with respect to human intelligence are cognitive science that mainly focuses on studying mind and behavior based cognitive architecture/models of intelligence, as well as neuroscience that mainly focuses on studying biological basis of human brain. In cognitive neuroscience, although many advanced results with respect to "perception oriented" study have been obtained, only a few of preliminary, separated studies with respect to "thinking oriented" and/or a more whole information process have been reported [9].

Brain Informatics (BI) can be regarded as brain sciences in Web intelligence (WI) centric IT age [36,38]. Although brain sciences have been studied from different disciplines such as cognitive science and neuroscience, BI represents a potentially revolutionary shift in the way that research is undertaken. BI is proposing to study human brain from the viewpoint of informatics (i.e. human

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 46–59, 2007.
© Springer-Verlag Berlin Heidelberg 2007

brain is an information processing system) and use informatics (i.e. WI centric information technology) to support brain science study, in particular, WI provides urgent research needs.

As a crucial step in understanding human intelligence in depth, we must first fully master the mechanisms in which human brain operates. Ignoring what goes on in human brain and focusing instead on behavior has been a large impediment to understand how human being does complex adaptive, distributed problem solving and reasoning. These results reported, over the last decade, about studying human information processing mechanism, are greatly related to progress of measurement and analysis technologies. Various noninvasive brain functional measurements are possible recently, such as fMRI and EEG. If these measurement data are analyzed systematically, the relationship between a state and an activity part will become clear. Furthermore, it is useful to discover more advanced human cognitive models based on such measurement and analysis. Hence, new instrumentation and new data analysis methods are causing a revolution in both AI and Brain Sciences. The synergy between AI and Brain Sciences will yield profound advances in our understanding of intelligence over the coming decade [18,22].

In order to investigate human information processing mechanism systematically, Web intelligence (WI) based portal techniques are required for brain data measurement, management and analysis [36]. Building a brain informatics portal is, in fact, to develop a data mining grid centric multi-layer grid system on the Wisdom Web, on which various data mining agents are deployed, for multi-aspect data analysis [12,39]. In recent papers [16,35,36,37], we have reported an approach for collecting, modeling, transforming, managing, and mining multiple human brain data obtained from visual and auditory psychological experiments by using fMRI and EEG. We observed that each method (fMRI and EEG) has its own strength and weakness from the aspects of time and space resolution. fMRI provides images of functional brain activity to observe dynamic activity patterns within different parts of the brain for a given task. It is excellent in the space resolution, but inferior time resolution. On the other hand, EEG provides information about the electrical fluctuations between neurons that also characterize brain activity, and measurements of brain activity at resolutions approaching real time. Hence, in order to discover new knowledge and models with respect to human information processing system, not only individual data source obtained from only single measuring method, but multiple data sources from various practical measuring methods are required.

It is also clear that the future of Brain Informatics will be affected by the ability to do large-scale mining of fMRI and EEG brain activations. The key issues are how to design the psychological and physiological experiments for obtaining various data from human information processing mechanism, as well as how to analyze and manage such data from multiple aspects for discovering new models of human information processing system. Although several human-expert centric tools such as SPM (MEDx) have been developed for cleaning, normalizing and visualizing the fMRI images, researchers have also been studying how the fMRI images can be automatically analyzed and understood by using data mining and statistical learning techniques [15,18,26]. Furthermore, spectral analysis [1] and wavelet analysis [11] are the main stream as the frequency analysis methods of EEG brain waves.

The rest of the paper is organized as follows. Section 2 describes what is the Data-Brain and how to build it on a data grid by collecting multiple data sources from cognitive fMRI/EEG experiments. Section 3 provides the data mining grid centric multi-layer Grid model for multi-aspect human brain data analysis. Section 4 introduces multi-aspect peculiarity oriented mining approach and its application in multiple human brain data analysis, as a case study. Finally, Section 5 gives concluding remarks.

2 Towards Data-Brain Construction

The Data-Brain is a brain database with all of data related to all major aspects and capabilities of human information processing mechanism for systematic investigation and understanding of human intelligence. The key questions are how to obtain such data by systematic fMRI/EEG experiments, how to manage such huge multimedia data for systematic investigation and understanding of human intelligence, as well as how to analyze such data from multi-aspect and multi-level for discovering new cognitive models. A new conceptual model is needed to build such a Data Brain, in addition to the following supporting capabilities,

– It is a grid-based, simulation and analysis oriented, dynamic, spatial and multimedia database;
– It deals with multiple data sources, multiple data forms, multiple levels of data granulation;
– It provides multiple views and organizations;
– It includes various methods for data analysis, simulation, visualization, as well as corresponding knowledge and models.

At first, agents for data collecting, storing and retrieving are deployed on the Grid platform, like Globus, as a standard Grid service. OGSA-DAI is used to build database access applications [40]. The aim of OGSA-DAI is to provide the middleware glue to interface existing databases, other data resources and tools to each other in a common way based on the Open Grid Services Architecture (OGSA). This middleware is based on the GGF-defined OGSI specification and layered on top of the Globus toolkit 3 OGSI implementation (GT3 Core).

Multiple data sources are collected by various cognitive fMRI/EEG experiments, modeling and transformation, and they are recorded to the corresponding databases through the Grid service on the distributed sites. Furthermore, the data-flow is a collection of descriptions for the dynamic relationship among multiple data sources on the data-grid. In the current study, data sources from cognitive fMRI/EEG experiments, to be collected on the data-grid, include

- human multi-perception mechanism for studying the relevance between auditory and visual information processing;
- human deductive/inductive reasoning mechanism for understanding the principle of human reasoning and problem solving in depth;
- human learning mechanism for acquiring personalized student models in an interactive learning process dynamically and naturally.

Event-related experimental designs have become an important methodology in EEG/fMRI research to evaluate the high level characteristics of human information processing in the central nervous system [17]. There are, at present, two main methods called event-related potential (ERP) and event-related fMRI for event-related experimental designs. ERP is a tiny signal embedded in the ongoing EEG. By averaging the traces, investigators can extract this signal, which reflects neural activity that is specifically related cognitive events [10]. ERPs are best suited for addressing questions about the time course of cognition rather than elucidating the brain structures that produce the electrical events. ERPs also provide physiological indices of when a person decides to response, or when an error is detected. On the other hand, event-related fMRI follows the same logic as used in ERP/EEG studies and provides the spatial resolution. Thus, event-related fMRI will further allow fMRI and EEG to be combined in paradigms that are identical across methods. By using such techniques, it is now becoming possible to study the precise spatiotemporal orchestration of neuronal activity associated with perceptual and cognitive events [20], as well as systematic collection of human brain data for building a Data Brain.

3 Multi-aspect Human Brain Data Analysis on the Grid

In papers [12,36], we have developed a conceptual model with three levels of workflows, corresponding to the three-layer Grid, namely data-grid, mining-grid, and knowledge-grid, respectively, which is utilized to manage, represent, integrate, analyze, and utilize the information coming from multiple, huge human brain data sources. In the multi-tier architecture, lower levels provide middleware support for higher level applications and services, thereby opening the door to developing more complex, flexible, and effective systems. Furthermore, the three-level workflows are generated dynamically, based on the conditions (situations), data quality analysis, and a multi-phase mining process.

Multi-aspect analysis in a multi-phase mining process is an important methodology for knowledge discovery from real-life data [7,27,28,29]. There are two main reasons why a multi-aspect analysis approach needs to be used. The first reason is that we cannot expect to develop a single data mining algorithm for analyzing all main aspects of the real-life data towards a holistic view because of complexity of the real-world applications. Hence, various data mining agents need to be cooperatively used in the multi-phase data mining process for performing multi-aspect analysis as well as multi-level conceptual abstraction and learning. The other reason is that when performing multi-aspect analysis for complex real-world problems such as brain informatics, a data mining task needs to be decomposed into sub-tasks. Thus these sub-tasks can be solved by using one or more data mining agents that are distributed over different computers on the Grid. Thus the decomposition problem leads us to the problem of distributed cooperative system design.

We are concerned to extract significant features from multiple brain data measured by using fMRI and EEG in preparation for multi-aspect data mining that uses various data mining agents, such as the GDT-RS inductive learning system for discovering decision rules [30], the LOI (learning with ordered information) for discovering ordering rules and important features [21,32], as well as the POM (peculiarity oriented mining) for finding peculiarity data/rules [33], for multi-aspect analysis in multiple data sources so that such rules mentioned above can be discovered automatically. Furthermore, by meta learning, the rules discovered by LOI and POM can be used to improve the quality of decision rules discovered by GDT-RS.

We emphasize that both pre-processing/post-processing steps are important before/after using data mining agents. In particular, informed knowledge discovery, in general, uses background knowledge obtained from experts (e.g. brain scientists) about a domain (e.g. cognitive nueroscience) to guide a spiral discovery process with multi-phase such as pre-processing, rule mining, and post-processing, towards finding interesting and novel rules/features hidden in data. Background knowledge may be of several forms including rules already found, ontologies, taxonomic relationships, causal preconditions, ordered information, and semantic categories. Such brain-informatics related knowledge, the generated hypotheses are deployed on the knowledge-grid and the knowledge-flow is utilized to generate, evaluate, refine, and employ knowledge on the knowledge-grid for various knowledge-based inference and reasoning [23,24,25,34]. From the top-down perspective, the knowledge level is also the application level with the support from both the mining level and the data level to serve brain scientists and the portal itself updating. From the bottom-up perspective, the data level supplies the data services for the mining level, and the mining level produces new rules and hypotheses for the knowledge level to generate active knowledge.

In general, several kinds of rules and hypotheses can be mined from different data sources by multi-aspect mining. The results cannot be directly utilized to support brain scientists' research activities until they are combined and refined into more general ones to form *active knowledge*, through an explanation-based inductive reasoning process. On the other hand, from the application viewpoint, distributed Web inference engines under the knowledge-flow management will employ such active knowledge with various related knowledge sources together to implement knowledge services for supporting brain scientists' research activities on the Wisdom Web based brain-informatics portal [23,24,25].

4 Multi-aspect Peculiarity Oriented Mining - A Case Study

At the current stage, our purpose is to investigate the spatiotemporal features and flow of human information processing system to understand

- how a peculiar part of the brain operates,
- how they work cooperatively to implement a whole information processing, and
- how they are linked functionally to individual differences in performance.

As a step in this direction, we observe that fMRI brain imaging data and EEG brain wave data extracted from human information processing mechanism are *peculiar* ones with respect to a specific state or the related part of a stimulus. Based on this point of view, we propose a way of *peculiarity oriented mining* for knowledge discovery in multiple human brain data, without using conventional imaging processing to fMRI brain images and frequency analysis to EEG brain waves [16,35,37]. The proposed approach provides a new way for automatic analysis and understanding of fMRI brain images and EEG brain waves to replace human-expert centric visualization. The mining process is a multi-step one, in which various psychological experiments, physiological measurements, data cleaning, modeling, transforming, managing, and mining techniques are cooperatively employed to investigate human information processing mechanism.

Fig. 1 gives the global picture of an example about how to investigate the spatiotemporal features and flow of human information processing system. In the cognitive process from perception to computation, data are collected in several event-related time points, and transformed into various forms in which Peculiarity Oriented Mining (POM) centric multi-aspect data analysis (MDA) can be carried out efficiently and effectively. Furthermore, the results of separate analysis can be explained and combined into a whole flow.

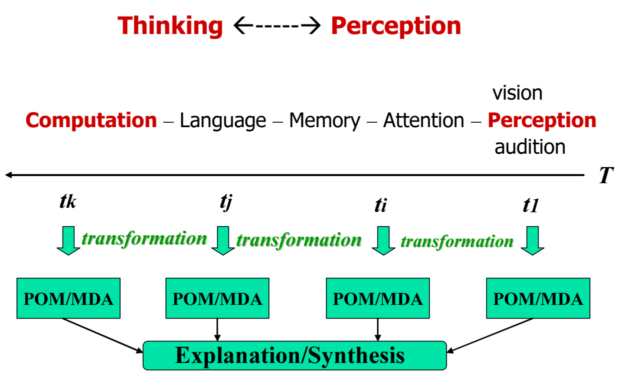


Fig. 1. Investigating the spatiotemporal features and flow of human information processing system

4.1 Peculiarity Oriented Mining (POM)

The main task of peculiarity oriented mining is the identification of peculiar data. An attribute-oriented method, which analyzes data from a new view and is different from traditional statistical methods, is recently proposed by Zhong *et al.* and applied in various real-world problems [33,35].

Peculiar data are a subset of objects in the database and are characterized by two features: (1) very different from other objects in a dataset, and (2) consisting of a relatively low number of objects. The first property is related to the notion of distance or dissimilarity of objects. Intuitively speaking, an object is different from other objects if it is far away from other objects based on certain distance functions. Its attribute values must be different from the values of other objects. One can define distance between objects based on the distance between their values. The second property is related to the notion of support. Peculiar data must have a low support.

At attribute level, the identification of peculiar data can be done by finding attribute values having properties (1) and (2). Let x_{ij} be the value of attribute A_j of the i-th tuple in a relation, and n the number of tuples. Zhong *et al* [33] suggested that the peculiarity of x_{ij} can be evaluated by a *Peculiarity Factor*, $PF(x_{ij})$:

$$PF(x_{ij}) = \sum_{k=1}^{n} N(x_{ij}, x_{kj})^{\alpha} \tag{1}$$

where N denotes the conceptual distance, α is a parameter to denote the importance of the distance between x_{ij} and x_{kj}, which can be adjusted by a user, and $\alpha = 0.5$ as default.

Based on peculiarity factor, the selection of peculiar data is simply carried out by using a threshold value. More specifically, an attribute value is peculiar if its peculiarity factor is above minimum peculiarity p, namely, $PF(x_{ij}) \geq p$. The threshold value p may be computed by the distribution of PF as follows:

$$\begin{aligned} threshold = {} & mean\ of\ PF(x_{ij}) + \\ & \beta \times standard\ deviation\ of\ PF(x_{ij}) \end{aligned} \tag{2}$$

where β can be adjusted by a user, and $\beta = 1$ is used as default. The threshold indicates that a data is a peculiar one if its PF value is much larger than the mean of the PF set. In other words, if $PF(x_{ij})$ is over the threshold value, x_{ij} is a peculiar data. By adjusting parameter β, a user can control and adjust threshold value.

4.2 Applications in Multiple Human Brain Data Analysis

Data Modeling and Transformation. The existing multimedia data such as fMRI brain images and EEG brain waves might not be suitable for data mining. Hence, a key issue is how to transform such data into a unique representation format (i.e. table). For such transformation, we develop a system and cooperatively utilize a software tool called MEDx (SPM) to formalize, clean and conceptualize fMRI brain images, so that such images can be represented in a relational data model and stored in a relational database. Fig. 2 shows examples of brain images transformed by using the MEDx. Furthermore, the ER (Entity-Relationship) model conceptualized fMRI brain images [35].

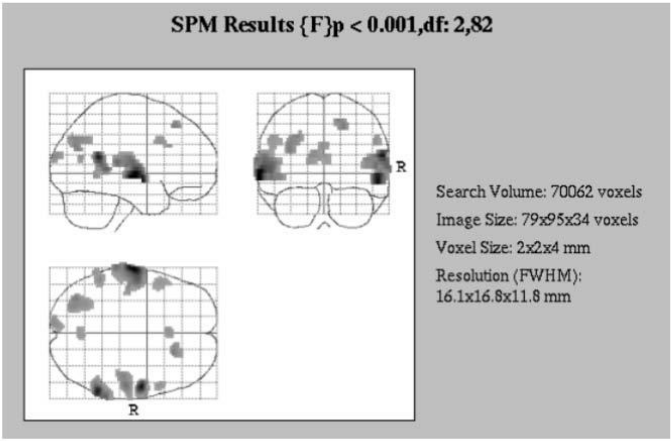


Fig. 2. Examples of brain images transformed by MEDx

Although the brain images transformed by using the MEDx can be represented in a relational data format, a problem is that the number of such data is too big. For instance, the number of the data for each subject is 122880 (i.e. 64 (pixels) $\times$ 64 (pixels) $\times$ 30 (images)). A way to reduce the number of data is to use a software tool called Talairach Daemon that is based on the Brodmann map as prior knowledge. Brodmann assigned numbers to various brain regions by analyzing each area's cellular structure starting from the central sulcus (the boundary between the frontal and parietal lobes). In our experiments, the Brodmann map is used as prior knowledge to obtain the Brodmann area values from the Talairach Daemon, before using our peculiarity oriented mining system to analyze the visual and auditory computation related datasets.

The Mining Processes of fMRI Imaging Data. Two peculiarity oriented mining processes, namely mining (1) and mining (2), are carried out on the fMRI data, respectively. In the process of mining (1), the prior knowledge of Brodmann areas is used before peculiarity oriented analysis. This process can be divided into the following steps [35]:

Step 1. Formalize/transform fMRI data by using the MEDx system.
Step 2. Obtain the Talairach Daemon coordinate and active values from the MEDx data.

Step 3. Get Brodmann area values from the Talairach Daemon.

Step 4. Create the visual and auditory calculation related databases, respectively, by using the Brodmann area values. If there are multiple values in a Brodmann area, use the sum of the values.

Step 5. Carry out peculiarity-oriented mining in the visual and auditory calculation related databases, respectively.

Step 6. Compare/evaluate the results.

On the other hand, in the process of mining (2), our peculiarity oriented mining is carried out on the fMRI data transformed in MEDx, directly (i.e. without prior knowledge of Brodmann areas is used before data mining), so that we will be able to analyze and compare the results of using the Brodmann area or not.

The Mining Process of EEG Brain Waves Data. The mining process of brain-wave data as shown in Fig. 3 is a multi-step one with data modeling and transformation. Brain wave data are typically a kind of time series data with noises. Hence, the removal of noises and data modeling are required. Usually, the low pass filter (LPF) and the band pass filter (BPF) are employed to remove noises by frequency analysis. However, we applied the ERP averaging for noise filtering. Furthermore, various methods of model transformation are carried out for multi-aspect analysis. The methods of model transformation include ERP oriented time-series POM on each channel, presenting the peculiarity distribution of ERP among channels, transforming time-series into frequency by FFT for computing averaging power spectrogram traces, presenting in the distance based matrix for clustering and finding peculiar channels.

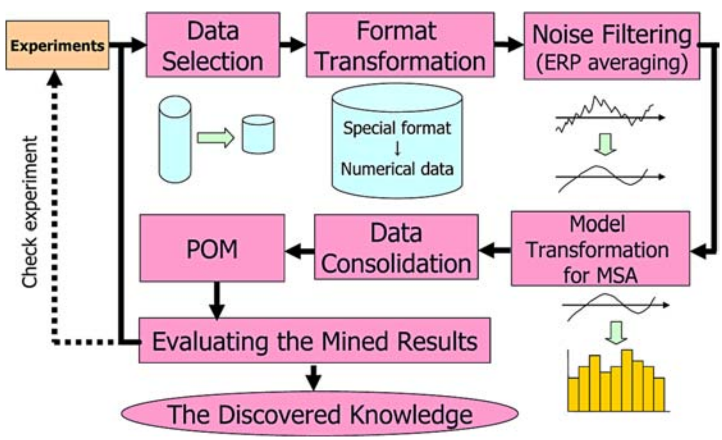


Fig. 3. The POM based EEG data mining process

4.3 Peculiarity Vector Oriented Mining for ERP Data Analysis

Unfortunately, the POM method in the attribute-value level stated in Section 3.1 is not fit for ERP data analysis. The reason is that the useful aspect for ERP data analysis is not amplitude, but the latent time. After smoothing enough by moving average processing, in the time series, we pays the attention to each potential towards N pole or P pole. Furthermore, the channel with the direction different from a lot of channels are considered to be a peculiar channel at that time. Hence, the distance between the attribute-values is expressed at the angle. And this angle can be obtained from the inner product and the norm in the vector. Let inclination of wave i in a certain time t be x_{it}. The extended PF corresponding to ERP can be defined by the following Eq. (3):

$$PF(x_{it}) = \sum_{k=1}^{n} \theta(x_{it}, x_{kt})^{\alpha}. \tag{3}$$

where θ is an angle which the wave in time t makes, and can be computed by the following Eq. (4):

$$cos\theta = \frac{1 + x_{it} \cdot x_{kt}}{\sqrt{1 + x_{it}^2}\sqrt{1 + x_{kt}^2}}. \tag{4}$$

The extended POM method has been used for the ERP data analysis. Fig. 4 shows a result in which the peculiarity in the ERP data with respect to addition between 2 digits with the visual stimulus is presented. Fig. 5 gives the channels shown in Fig. 4 and their presentation on the EEG cap electrode based on the 10-20 system. After 300 milliseconds, the result in which the PF values are high

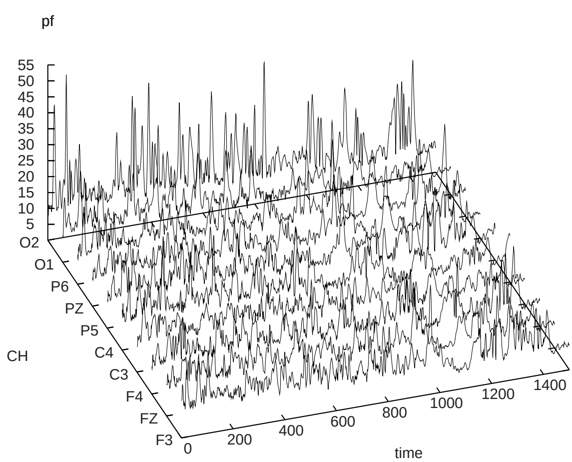


Fig. 4. A result obtained by the extended POM method

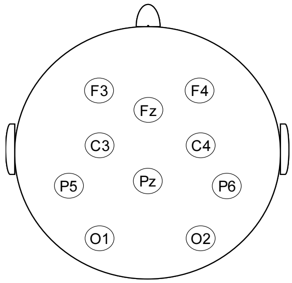


Fig. 5. The channels shown in Fig. 4 and their presentation on the EEG cap electrode based on the 10-20 system

was obtained from the stimulus presentation after the vicinity by the occipital lobe and the frontal lobe near after 700 milliseconds. We can see that a higher value of peculiarity that a different potential changes from a lot of other channels for that time has occurred.

5 Concluding Remarks

A new methodology of combining cognitive neuroscience and WI based multi-aspect analysis is presented to investigate human information processing mechanism, systematically. Multi-aspect analysis in multiple brain data sources is an important data mining methodology in BI. The proposed methodology attempts to change the perspective of cognitive scientists from a single type of experimental data analysis towards a holistic view at a long-term, global field of vision to understand the principle, models and mechanisms of human information processing system.

Since this project is very new, we just had preliminary results [16,35,36,37]. The future work includes studying the neural structures of the activated areas and trying to understand how a peculiar part of the brain operates and how it is linked functionally to individual differences in performance by combining various mining methods with reasoning. Some of lessons in cognitive neuroscience are applicable to novel technological developments in BI, yet others may need to be enhanced or transformed in order to manage and account for the complex and possibly more innovative practices of sharing, analyzing and creating data/knowledge that are made technically possible by the Wisdom Web and Knowledge Grids [14,36].

Acknowledgments

I would like to express my gratitude to Professor Vladimir Gorodetsky and other organizers of AIS-ADM-07 for the kind invitation and the excellent organization. I am grateful to all my research collaborators, assistants, and students who have, over the years, together contributed to the development of Web Intelligence (WI) and Brain Informatics (BI). Special thanks to Jiming Liu, Yiyu Yao, Jinglong Wu, Benjamin Wah, and Shengfu Lu for our joint projects and discussions. I am very grateful to people who have joined or supported the WI and BI communities, members of the WIC advisory board, WIC technical committee, and WIC research centres, as well as keynote/invited speakers of IEEE/WIC/ACM WI-IAT conferences. This work is partially supported by the grant-in-aid for scientific research (No. 18300053) from the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

1. Banquet, J.P.: Spectral Analysis of the EEG in Meditation. *Electroencephalography and Clinical Neurophysiology* **35** (1973) 143–151
2. Banfield, J. and Raftery, A.: Model-based Gaussian and Non-Gaussian Clustering. Biometrics **49** (1993) 803–821
3. Berman, F.: From TeraGrid to Knowledge Grid, *CACM* **44** (2001) 27–28
4. Cannataro, M. and Talia, D.: The Knowledge Grid, *CACM* **46** (2003) 89–93
5. Consularo, L.A., Lotufo, R.A., and Costa, L.F.: Data Mining Based Modeling of Human Visual Perception, K.J. Cios (ed) *Medical Data Mining and Knowledge Discovery*, Physica-Verlag (2001) 403–431
6. Cerutti, S., Chiarenza, G., Liberati, D., Mascellani, P., and Pavesi, G.: A Parametric Method of Identification of Single Trial Event-related Potentials in the Brain. *IEEE Trans. Biomed. Eng.*, 35(9) (1988) 701–711
7. Fayyad, U.M., Piatetsky-Shapiro, G., and Smyth, P.: From Data Mining to Knowledge Discovery: an Overview. *Advances in Knowledge Discovery and Data Mining*, MIT Press (1996) 1–36
8. Foster, I. and Kesselman, C. (eds.): *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann (1999)
9. Gazzaniga, M.S. (ed.): *The Cognitive Neurosciences III*, The MIT Press (2004)
10. Handy, T.C.: *Event-Related Potentials, A Methods Handbook*, The MIT Press (2004)
11. Hornero, R., Martin-Fernandez, M., Alonso, A., Izquierdo, A., and Lopez, M.: A DSP Implementation of Wavelet Transform to Detect Epileptiform Activity in the EEG. *Proc. 8th Annual International Conference on Signal Processing Applications and Technology*, ICAPAT'97 (1997) 692–696
12. Hu, J. and Zhong, N.: Organizing Multiple Data Sources for Developing Intelligent e-Business Portals. *Data Mining and Knowledge Discovery*, Vol. 12, Nos. 2–3, Springer (2006) 127–150
13. Liu, J., Zhong, N., Yao, Y.Y., and Ras, Z.W.: The Wisdom Web: New Challenges for Web Intelligence (WI). *Journal of Intelligent Information Systems*, 20(1) Kluwer (2003) 5–9
14. Liu, J.: Web Intelligence (WI): What Makes Wisdom Web? *Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI'03)* (2003) 1596–1601
15. Megalooikonomou, V. and Herskovits, E.H.: Mining Structure-Function Associations in a Brain Image Database. K.J. Cios (ed.) *Medical Data Mining and Knowledge Discovery*, Physica-Verlag (2001) 153–179
16. Motomura, S., Zhong, N., and Wu, J.L.: Brain Waves Data Mining for Human Multi-perception Activity Analysis. *Proc. Inter. Workshop on Advanced Technologies for e-Learning and e-Science (ATELS'04)* (2004) 65–72

17. Gazzaniga, M.S., Ivry, R., and Mangun, G.R.: *Cognitive Neuroscience: The Biology of the Mind*, W.W. Norton (2002) 2nd Edition
18. Mitchell, T.M., Hutchinson, R., Just, M., and Niculescu, R.S., Pereira, F., and Wang, X.: Classifying Instantaneous Cognitive States from fMRI Data. *Proc. American Medical Informatics Association Annual Symposium* (2003) 465–469
19. Newell, A. and Simon, H.A.: *Human Problem Solving*, Prentice-Hall (1972)
20. Rosen, B.R., Buckner, R.L., and Dale, A.M.: Event-related functional MRI: Past, Present, and Future. *Proceedings of National Academy of Sciences, USA*, 95(3) (1998) 773–780
21. Sai, Y., Yao, Y.Y., and Zhong, N.: Data Analysis and Mining in Ordered Information Tables. *Proc. 2001 IEEE International Conference on Data Mining (ICDM'01)*, IEEE Computer Society Press (2001) 497–504
22. Sternberg, R.J., Lautrey, J., and Lubart, T.I.: *Models of Intelligence*, American Psychological Association (2003)
23. Su, Y., Liu, J., Zhong, N., Zheng, L., and Liu, C.: A Method of Distributed Problem Solving on the Web. *Proc. 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, IEEE Computer Society Press (2005) 42–45
24. Su, Y., Zheng, L., Zhong, N., Liu, C., and Liu, J.: Distributed Reasoning Based on Problem Solver Markup Language (PSML): A Demonstration through Extended OWL. *Proc. 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)*, IEEE Press (2005) 208–213
25. Tomita, K., Zhong, N., and Yamauchi, H.: Coupling Global Semantic Web with Local Information Sources for Problem Solving. *Proc. First International Workshop on Semantic Web Mining and Reasoning* (2004) 66–74
26. Tsukimoto, H., and Morita, C.: The Discovery of Rules from Brain Images. *Proc. First Inter. Conf. on Discovery Science*, LNAI 1532, Springer (1998) 198–209
27. Zhong, N., and Ohsuga, S.: Toward A Multi-Strategy and Cooperative Discovery System. *Proc. First Int. Conf. on Knowledge Discovery and Data Mining (KDD-95)*, AAAI Press (1995) 337–342
28. Zhong, N., and Ohsuga, S.: Automatic Knowledge Discovery in Larger Scale Knowledge-Data Bases. C. Leondes (ed.) *The Handbook of Expert Systems*, Vol.4: Chapter 29, Academic Press (2001) 1015–1070
29. Zhong, N., Liu, C., and Ohsuga, S.: Dynamically Organizing KDD Process. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(3), World Scientific (2001) 451–473
30. Zhong, N., Dong, J.Z., and Ohsuga, S.: Rule Discovery by Soft Induction Techniques. *Neurocomputing*, An International Journal, 36(1–4), Elsevier (2001) 171–204
31. Zhong, N., Liu, J., and Yao, Y.Y.: In Search of the Wisdom Web. *IEEE Computer*, 35(11) (2002) 27–31
32. Zhong, N., Yao, Y.Y., Dong, J.Z., and Ohsuga, S.: Gastric Cancer Data Mining with Ordered Information. J.J. Alpigini et al (eds.) *Rough Sets and Current Trends in Computing*, LNAI 2475, Springer (2002) 467–478
33. Zhong, N., Yao, Y.Y., and Ohshima, M.: Peculiarity Oriented Multi-Database Mining. *IEEE Transaction on Knowlegde and Data Engineering*, 15(4) (2003) 952–960
34. Zhong, N.: Developing Intelligent Portals by Using WI Technologies. J. P. Li et al. (eds.) *Wavelet Analysis and Its Applications, and Active Media Technology.* Vol. 2, World Scientific (2004) 555–567
35. Zhong, N., Wu, J.L., Nakamaru, A., Ohshima, M., and Mizuhara, H.: Peculiarity Oriented fMRI Brain Data Analysis for Studying Human Multi-Perception Mechanism. *Cognitive Systems Research*, 5(3), Elsevier (2004) 241–256

36. Zhong, N., Hu, J., Motomura, S., Wu, J.L., and Liu, C.: Building a Data Mining Grid for Multiple Human Brain Data Analysis. *Computational Intelligence*, 21(2), Blackwell Publishing (2005) 177–196
37. Zhong, N., Motomura, S., and Wu, J.L.: Peculiarity Oriented Multi-Aspect Brain Data Analysis for Studying Human Multi-Perception Mechanism. *Proc. SAINT 2005 Workshops (Workshop 8: Computer Intelligence for Exabyte Scale Data Explosion)*, IEEE Computer Society Press (2005) 306–309
38. Zhong, N.: Impending Brain Informatics (BI) Research from Web Intelligence (WI) Perspective. International Journal of Information Technology and Decision Making, 5(4), World Scientific (2006) 713–727
39. Zhong, N., Liu, J., and Yao, Y.Y.: Envisioning Intelligent Information Technologies (iIT) from the Stand-Point of Web Intelligence (WI). *Communications of the ACM*, 50(3) (2007)
40. The OGSA-DAI project: *http://www.ogsadai.org.uk/*

Agent-Mining Interaction: An Emerging Area*

Longbing Cao, Chao Luo, and Chengqi Zhang

Faculty of Information Technology, University of Technology, Sydney, Australia
{lbcao,chaoluo,chengqi}@it.uts.edu.au

Abstract. In the past twenty years, *agents* (we mean autonomous agent and multi-agent systems) and *data mining* (also knowledge discovery) have emerged separately as two of most prominent, dynamic and exciting research areas. In recent years, an increasingly remarkable trend in both areas is *the agent-mining interaction and integration*. This is driven by not only researcher's interests, but intrinsic challenges and requirements from both sides, as well as benefits and complementarity to both communities through agent-mining interaction. In this paper, we draw a high-level overview of the agent-mining interaction from the perspective of an emerging area in the scientific family. To promote it as a newly emergent scientific field, we summarize key driving forces, originality, major research directions and respective topics, and the progression of research groups, publications and activities of agent-mining interaction. Both theoretical and application-oriented aspects are addressed. The above investigation shows that the agent-mining interaction is attracting everincreasing attention from both agent and data mining communities. Some complicated challenges in either community may be effectively and efficiently tackled through agent-mining interaction. However, as a new open area, there are many issues waiting for research and development from theoretical, technological and practical perspectives.

1 Introduction

In the past twenty years, two of most prominent, dynamic and exciting research areas—autonomous agent and multi-agent systems (AAMAS) [32] and data mining (or knowledge discovery in databases (KDD))[17] have emerged and developed separately. These two independent research streams have been created and originally evolving with separate aims and objectives. They target individual methodologies and techniques coping with domain-specific problems and challenges in respective areas.

AAMAS (or for short agent) is a powerful technology for autonomous intelligent system analysis and design. It is also a thoughtful computing paradigm for dealing with system complexities in tackling open complex problems such as openness, distribution, human involvement, societal characteristics with a new

* This work is sponsored by Australian Research Council Discovery Grant (DP0773412, LP0775041, DP0667060, DP0449535), and UTS internal grants.

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 60–73, 2007.
© Springer-Verlag Berlin Heidelberg 2007

perspective on computing and intelligence. Agent studies theoretical, methodological, experimental and practical issues in developing agent-based computing and agent-oriented intelligent systems. The major topics of interest consist of research on individual agents, multi-agent systems, methodology and techniques, tools and applications. The AAMAS technologies are currently contributing to many diverse domains such as software engineering, user interfaces, e-commerce, information retrieval, robotics, computer games, education and training, ubiquitous computing, and social simulation. The benefits from agent are comprehensive and diverse, from academic disciplines, to the sciences, the social sciences and the humanities.

On the other hand, data mining and KDD (for short mining) research the process of analyzing data to identify hidden but interesting patterns or relationships. KDD applies many existing computational techniques from statistics, information retrieval, machine learning, artificial intelligence, pattern recognition, and database technologies. KDD is increasingly widely deployed into varying applications and fields, for instance, web mining and services, text mining, telecommunications, retail, governmental service, fraud, security, business intelligence studies.

In recent years, an increasingly evident trend has emerged. The trend is the interaction and integration between agent and mining. Its development has reached to the level as a new and promising area, and towards a first-class citizen in the science and technology family[5,6,7,8,9].

As identified in a recent position meeting and related activities[5], there are many research topics and open issues from either part of agent and mining interaction. In particular, issues for agent-driven data mining, and issues for mining-driven agents are attracting research interest mainly. However, there are some mutually fundamental issues that are not paid attention in the emerging research. These issues are significant because of their fundamental and necessary roles in establishing a symbiotic relation between agent and mining.

In this paper, we present a systematic view of the evolution and development of agent-mining interaction and integration through a survey of related activities till today. Through a systematic investigation, we summarize key driving forces, originality, major research directions and respective topics, and the progression of research groups, publications and activities of agent-mining interaction. Both theoretical and application-oriented aspects are addressed.

Through reviewing the related work in the above areas, this survey evidences that

1. The agent-mining interaction is attracting ever-increasing attention from both agent and data mining communities,
2. The interaction and integration between agent and mining can greatly complement and strengthen each side of both communities. Some complicated challenges in either community may be effectively and efficiently tackled through agent-mining interaction,

3. Furthermore, as a newly emergent area, agent and mining interaction and integration has potential to create new interesting symbiosis opportunities in both academic and business worlds,
4. However, as a new open area, there are many issues waiting for research and development from theoretical, technological and practical perspectives.

The remaining of this paper is organized as follows. Section 2 discusses challenges in agent and mining communities. In section 3, we present an evolutionary map of agent-mining interaction. Research directions and topics are summarized in Section 4. Section 5 lists some of useful research resources for the agent-mining interaction. A case study is briefed in Section 6. Conclusions are drawn in Section 7.

2 Challenges in Agent and Mining Communities

As addressed in[5,6,7], agent can enhance data mining through involving agent intelligence into data mining systems, while an agent system can benefit from data mining via extending agent knowledge discovery capability. Nevertheless, the agent-mining interaction symbiosis cannot be established if mutual issues are not solved. These mutual issues involve fundamental challenges hidden in both sides and particularly the interaction and integration. Figure 1 presents a view of issues in agent-mining interaction. It highlights the part of mutual issues.

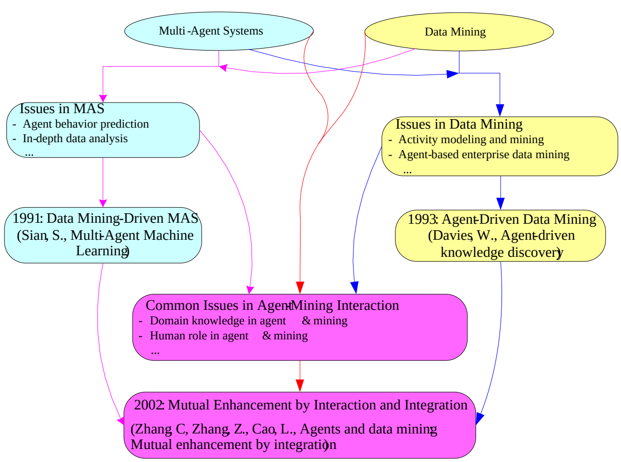


Fig. 1. Issues in agent-mining interaction

Mutual issues constraining agent-mining interaction and integration consist of the following categories.

- Human issues, namely the roles and human intelligence that may be better played or involved by humans to support agent-mining interaction.

- Domain issues, namely the roles and domain intelligence that may be essential in agent-mining interaction.
- Data and knowledge issues, namely the roles and data/knowledge intelligence in building agent-mining symbionts.
- Nonfunctional issues, namely the roles of nonfunctional aspects in building agent-mining symbionts.

Why do these issues matter both sides of agent and mining, and the interaction and integration between agent and mining? There are both explicit reasons and implicit ones. Explicit reasons may include the following system complexities.

– Explicit limitations and challenges in pure agent systems, as addressed in [5,6,7], that can be complemented by data mining, for instance, data mining driving agent learning, user modeling and information analysis.
– Explicit limitations and challenges in pure data mining systems, as discussed in [5,6,7], that can be better serviced by agent technology, for instance, agent-based data mining infrastructure, agents for data management and preparation, agent-based service providing.

Implicit driving forces for including the above mutual issues are equally significant.

– Agent-mining symbionts are substantially essential for dealing with complex intelligence phenomenon and system complexities in complex intelligent systems. Simple intelligent systems and other issues that can be tackled using one side of technologies, for instance, an agent-based data integration system, may not necessarily involve both sides.
– Intelligence emergence in agent-mining interaction may massively strengthen problem-solving capability of an intelligent system, which cannot be played by either part.
– Implicit roles need to be discovered through interdisciplinary studies, which may extremely promote either one side or the whole of an agent-mining integrative system, once the roles are disclosed and properly developed.

3 Evolution of Agent-Mining Interaction

3.1 Fast Progression

We draw a conclusion that agent-mining interaction and integration is emerging as a new member of scientific family due to the following survey findings.

– Ever increasing and a decent number of publications: an initial literature review has disclosed that there are around 150 conference and journal papers, 7 books and proceedings, and some technical reports published by 2006 on topics associated with agent and mining interaction and integration[1] This uptrend is getting increasingly clearer than before in recent three years. Figure 2 illustrates an incremental change of the number of papers published from 1997 to 2005 by major presses including Springer, IEEE and ISI press. It shows a major number rise from 2003. In 2005, the number of publications almost triples that of 2003.

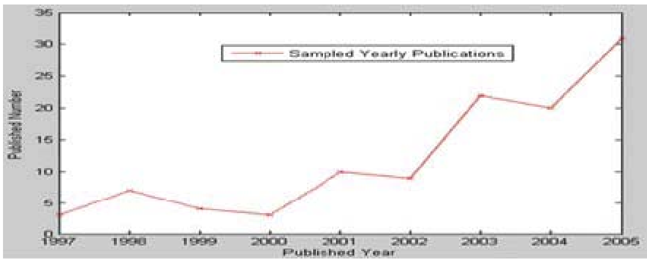


Fig. 2. Incremental change of publication numbers

Table 1 further summarizes the quantity distribution of publications in terms of major search engines. The hit numbers are more or less associated with corresponding publishers that published the papers.

Table 1. Paper publisher distribution

	ACM Portal	IEEExplore	SpringerLink
Hit Number	8	59	48

– Ever increasing level and quality of papers: With the increase of publication numbers, publication quality is also extremely improved. A typical trend is that more and more journal papers and books/proceedings published after 2003. For instance, two specific books have been published [29,31]. In addition, every year can see some papers accepted by top-ranking conferences and journals such as AAMAS [22], PAKDD [26], and IEEE journals [18,19,24,28] in both communities. Table 2 also presents a comparative statistics of two workshops: the standalone AIS-ADM2005 and the IADM2006 co-located with IEEE/ACM/WIC WI-IAT2006. The above indicators reflect the quality progression of publications.

[1] In this paper, the literature survey is undertaken through information retrieval by the following search engines: Google, IEEEXplore, SpringerLink, ACM Portal, Science@Direct, infosci, amazon, ISI, etc. Further action has been on investigating individual research groups and researcher's websites. The survey results are available from the Agent & Mining Interaction and Integration website (www-staff.it.uts.edu.au/˜lbcao/amii/amii.htm).

Table 2. Paper publisher distribution

	Submission number	Accepted papers	Attendees
AISAMD2005	29	17 (58.6%)	30
ADMI2006	45	15 (30%)	40

- Ever increasing number of professional activities: Another typical indicator of whether a research topic is evolving into a new separate area is the number and quality of professional activities, and the involvement of key research groups and researchers from both communities in these activities. Table 3[2] summarizes the numbers of professional activities in this area till 2006, which include workshops, special issues, tutorials, and books. Another landmark is the Open Position Meeting joint with IADM2006. This event presented 7 position talks by active research groups from Australia, Canada, China, Greece, Poland and Russia. The meeting also pointed out the significance of setting up an international steering committee to guide the development of agent-mining interaction.
- Increasing transparent academic voice pursuing a separate area and a first-class citizen in the scientific family: this is evidenced by panel discussions in AIS-ADM2005[3] and an open position meeting joint with IADM2006[4].

Table 3. Decent numbers of professional activities

	Workshops	Special issues	Tutorials	Panel discussions
Number	5	1	7	2

3.2 Evolutionary Characteristics

Further, we investigate some of evolutionary characteristics of this area. In particular, there are following key driving forces.

- From one-way to two-way interaction: The area was originally initiated by involving data mining into agent to enhance agent learning [10,30]. Recently, issues in two-way interaction and integration are broadly studies in different groups.
- Mutual needs: As discussed in [9], people have found many issues in each of the related communities. These issues cannot be tackled by simply developing internal techniques. Rather, techniques from the other discipline can greatly complement the problem-solving when they are combined with the existing techniques and approaches.

[2] Searched in December 2006 by using keywords.
[3] Int. workshop on Autonomous Intelligent Systems: Agent and Data Mining 2005.
[4] Int. Workshop on Agent and Data Mining Interaction, 2006.

– Intrinsic associations and utilities: The interaction and integration between agent and mining is also driven and connected by intrinsic associations and utilities, as discussed in [7,8,9], in both communities.
– Application drives: Application request is one of the key driving forces of this new trend. In Section 4.3, we present some of major application domains and problems that may be better handled by both agent and mining techniques.
– Major research groups and researchers [6] in respective communities tend to undertake both sides of research. Some of them are trying to link them together to solve problems that cannot be tackled by one of them only, for instance, agent-based distributed learning [20,21,22,15,16], agent-based data mining infrastructure [4,5,16], data mining driven agent intelligence enhancement [4,25].
– Broad research from theoretical, technological and practical perspectives: publications and projects have involved not only technological issues, but also theoretical and practical problems. A cross-disciplinary and multi-dimensional study roadmap is becoming clearer.

On the other hand, we also draw an evolutionary tree of this area by combining the emergence of significant landmarks and events in the life of agent-mining interaction (see Figure 3). The survey clearly indicates us that agent-mining interaction and integration has emerged as a prominent, challenging, dynamic and exciting area.

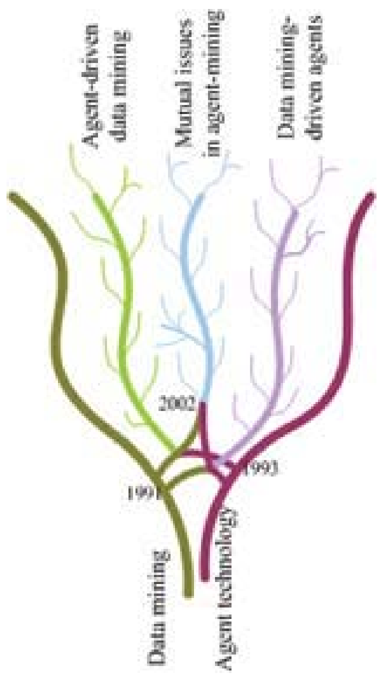


Fig. 3. Evolution of agent-mining interaction as a scientific area

4 Research Directions in Agent-Mining Interaction

In investigating research directions and topics in agent-mining interaction area, we summarize an overview as shown in Figure 3.

4.1 Data Mining Driven Agents

The agent-mining interaction was originally initiated by data mining driven agent learning in 1991 [10,30]. Data mining has potential to enhance agent technology through introducing and improving learning and reasoning capability of agents. For instance, the following illustrates some of popular research topics.

- Collaborative learning in multiagents
- Data mining-driven agent learning, reasoning, adaptation and evolution
- Data mining-driven multiagent communication, planning and dispatching
- Data mining-driven user modeling
- Data mining-driven user servicing
- Data mining-driven network servicing
- Data mining-driven agent recommender
- Data mining-driven trading agents
- Data mining agent assistant
- Data mining enhancing agent intelligence enhancement
- Decentralized clustering in large multi-agent systems
- Distributed learning in agent coordination
- Distributed learning in multi-agent systems
- Emergent agent organization and behavior
- Information gathering agents
- Learning agents
- Web mining agents
- Self-learning agents, etc.

4.2 Agent Driven Data Mining

In sometime around 1993, another effort was started on agent-based data mining [11,12,13]. It is to utilize agent technology to enhance data mining. The enhancement may be embodied in terms of varying aspects, for instance, infrastructure, distributed processing, human involvement. The following lists some of research topics.

- Activity modeling and mining
- Agent-based enterprise data mining
- Agent-based data mining infrastructure
- Agent-based data warehouse
- Agent-based mining process and project management
- Agent-based distributed data mining
- Agent-based distributed learning

- Agent-based grid computing
- Agent-based human mining cooperation
- Agent-based link mining
- Agent-based multi-data source mining
- Agent-based interactive data mining
- Agent-enriched ontology mining
- Agent-based parallel data mining
- Agent-based web mining
- Agent-based text mining
- Agent-based ubiquitous data mining
- Agent knowledge management in distributed data mining
- Agent for data mining data preparation
- Agent-human-cooperated data mining
- Agent networks in distributed knowledge discovery and servicing
- Agent service-based KDD infrastructure
- Agent-supported domain knowledge involvement in KDD
- Agent system providing data mining services
- Automated data mining learning
- Autonomous learning
- Distributed agent-based data preprocessing
- Distributed learning
- Domain intelligence in agent-based data mining
- Mobile agent-based knowledge discovery
- Multi-agent reinforcement learning
- Multi-agent knowledge discovery
- Protocols for agent-based data mining
- Self-organizing data mining learning, etc.

4.3 Mutual Enhancement Issues in Agent-Mining Interaction

After years of unorganized development of the above discussed one-way effect, people further recognize fundamental mutual issues in agent-mining interaction [4,9,33,2], which involve common issues of both parties. The studies on these mutual issues can not only tackle problems towards one-way enhancement as discussed in Sections 4.1 and 4.2, but also two-way strengthening in building a super-intelligent agent-mining symbiont. However, these issues have not attracted sufficient attention in the community.

- Architecture and infrastructure problems
- Actionable capability of agent-mining symbionts
- Constraints in agent and mining
- Data intelligence in agent and mining
- Domain knowledge in agent and mining
- Domain intelligence in agent and mining
- Evaluation issues such as technical significance and business expectation
- Gap filling between technical and business expectations

- Human intelligence and role in agent and mining
- Human-system interaction
- Intelligence metasynthesis in agent and mining
- Knowledge management in agent and mining
- Metadata and meta-knowledge in agent and mining
- Nonfunctional issues such as usability, expendability, openness
- Ontology issues in agent and mining
- Organizational issues such as business factors, process
- Performance issues such as effectiveness, efficiency, scalability
- Social issues such as security, privacy, trust
- Services request and response, service-oriented management
- System management, etc.

4.4 Application Studies

Besides the above technical development, agent-mining interaction is actually driven by broad and increasing applications [7]. At the same time, many researchers focus on the development of agent-mining systems for dealing with specific business problems. In this way, business problems can be better handled compared with using unilateral technology. For instance, we summarize the following application domains.

- Artificial immune systems
- Artificial and electronic markets
- Auction
- Business intelligence
- Customer relationship management
- Distributed data extraction and preparation
- E-commerce
- Finance data mining
- Grid computing
- Healthcare
- Internet and network services, eg., recommendation, personal assistant, searching, retrieval, extraction services
- Knowledge management
- Marketing
- Network intrusion detection
- Parallel computing, eg., parallel GA
- Peer-to-peer
- Semantic web
- Social intelligence & social network analysis
- Supply chain management
- Trading agents
- Text mining
- Web mining, etc.

5 Research Resources on Agent-Mining Interaction

At this point, more and more research resources are coming up on agent-mining interaction. This is evidenced by not only papers but research books. For a systematic and comprehensive understanding of the area of agent-mining interaction, the website AMII[5] collects both original and state-of-the-art information in terms of research topics, groups, projects, activities and open issues. This website presents a systematic one-stop portal of the area. Furthermore, references [5,6,7,8,9,16,22] summarize challenges and prospects of agent-mining interaction and integration from both theoretical and practical perspectives.

On the other hand, there are some resources highlighting one side of the problem. For instance, the book [25] discusses agent intelligence through data mining, while [27] highlights intelligent agents for data mining and information retrieval. The report [9,6] mainly highlights mutual issues in agent-mining interaction. Proceedings [2,14,23] collect papers from the workshops AIS-ADM2005 and IADM2006.

6 F-Trade: An Agent-Mining Symbiont

In this section, we briefly introduce an agent-mining symbionet – F-Trade[6] to illustrate the development and use of agent-mining interaction technology in tackling both research and business issues. Figure 4 shows some screenshots of the F-Trade.

F-Trade [4] is initialism of Financial Trading Rules Automated Development and Evaluation, a web-based automated enterprise infrastructure for trading strategies and data mining on stock/capital markets. The system offers data connection, management and processing services. F-Trade supports online automated plug and play of, and automatic input/output interface construction for trading signals/rules and data mining algorithms, data sources, and system components. It provides powerful and flexible supports for online backtesting, training/test, optimization and evaluation of trading strategies and data mining algorithms. Users can plugin, subscribe, supervise and optimize trading strategies and data mining algorithms in a human-machine cooperated manner.

F-Trade is built in Java agent services on top of Windows/Linux/Unix. XML is used for system configuration and metadata management. A super-server serves for the application server, another one acts as data warehouse. It is constructed with online connectivity to distributed data sources including huge stock data in AC3[7] as well as user-specific data sources.

Major roles played by agent in the F-Trade consist of agent service-based architecture, agent-driven human interaction, agent for data source management, data collection and dispatch agents roaming to remote data sources, agentized trading strategies and data mining algorithms, agent and service recommenders

[5] http://www-staff.it.uts.edu.au/~lbcao/amii/amii.htm.
[6] www.f-trade.info, www.ftrade.info
[7] www.ac3.org.au.

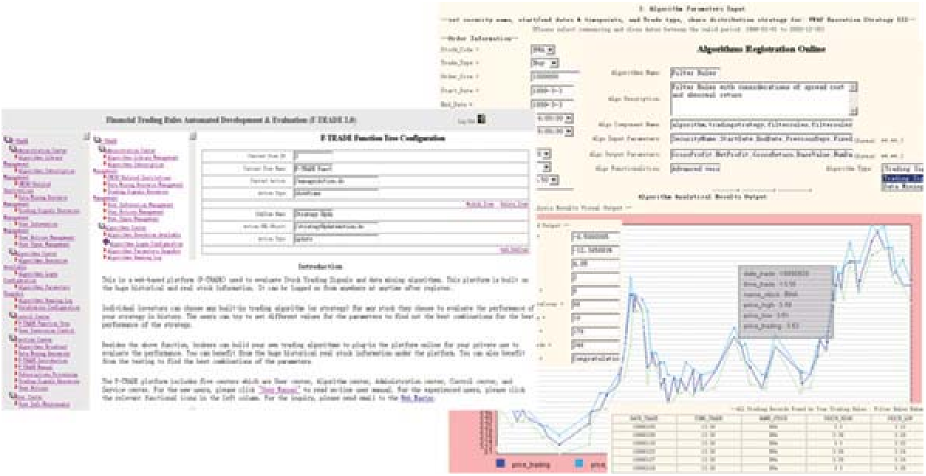


Fig. 4. F-Trade: an agent-mining symbiont

providing optimum algorithms and rules to users, and so on. On the other hand, data mining assists the system in aspects such as data mining-driven trading rule/algorithm recommender agents, data mining-driven user services, data mining-driven trading agent optimizers, mining actionable trading rules in generic trading pattern set, parameter tuning of algorithm agents through data mining, etc. Mutual issues involve ontology-based domain knowledge representation and transformation to problem-solving terminology, human involvement and agent-based human interaction with algorithms and the system for algorithm supervision, optimization and evaluation, etc.

7 Conclusions

Agent and data mining interaction and integration has emerged as a prominent and promising area in recent years. The dialogue between agent technology and data mining can not only handle issues that are hardly coped with in each of the interacted parties, but also create innovative and super-intelligent techniques and symbionts. In this way, both communities and its interactive emergence can be massively enhanced.

In this paper, we present a high-level overview of the area development and major directions. The investigation highlights the following findings: (1) agent-mining interaction is emerging as a new area in the scientific family, (2) the interaction is extremely promoting the progress of agent and mining communities, (3) it is creating ever-increasing development of innovative and significant techniques and systems towards super-intelligent symbionts, however, (4) as a new area, it just starts and has many open issues waiting for a decent involvement of research resources in particular practical and research projects from both communities.

References

1. Aciar, S., Zhang, D., Simoff, S., Debenham, J.: Informed Recommender Agent: Utilizing Consumer Product Reviews through Text Mining. Proceedings of IADM2006. IEEE Computer Society (2006)
2. Baik,S., Cho, J., Bala, J.: Performance Evaluation of an Agent Based Distributed Data Mining System. Advances in Artificial Intelligence, Volume 3501/2005 (2005)
3. Cory, J., Butz, Nguyen, N., Takama, Y., Cheung, W., Cheung, Y.: Proceedings of IADM2006 (Chaired by Longbing Cao, Zili Zhang, Vladimir Samoilov) in WI-IAT2006 Workshop Proceedings. IEEE Computer Society (2006)
4. Cao, L., Wang, J., Lin, l., Zhang, C.: Agent Services-Based Infrastructure for Online Assessment of Trading Strategies. Proceedings of IAT'04 (2004) 345–349
5. Cao, L.: Integration of agent and data mining. Technical report, 25 June 2005. http://www-staff.it.uts.edu.au/~lbcao/publication/publications.htm
6. Cao, L., Gorodetski, V.: AREA OVERVIEW-Agent & data mining interaction (ADMI). WI-IAT 2006 IADM Workshop panel discussion, Hongkong (2006)
7. Cao, L.: Agent-mining interaction: theoretical challenges and prospects. Technical report (2006)
8. Cao, L.: Agent-mining interaction: application challenges and prospects. Technical report (2006)
9. Cao, L.: Mutual issues in agent-mining interaction. Technical report (2006)
10. Brazdil, P., Muggleton, S.: Learning to Relate Terms in a Multiple Agent Environment. EWSL91 (1991)
11. Davies, W.: ANIMALS: A Distributed, Heterogeneous Multi-Agent Learning System. MSc Thesis, University of Aberdeen (1993)
12. Davies, W.: Agent-Based Data-Mining (1994)
13. Edwards, P., Davies, W.: A Heterogeneous Multi-Agent Learning System. In Deen, S.M. (ed) Proceedings of the Special Interest Group on Cooperating Knowledge Based Systems. University of Keele (1993) 163–184
14. Gorodetsky, V., Liu, J., Skormin, V.A.: Autonomous Intelligent Systems: Agents and Data Mining book. Lecture Notes in Computer Science Volume 3505 (2005)
15. Gorodetsky, V., Karsaeyv, O. and Samoilov, V.: Multi-agent technology for distributed data mining and classification Intelligent Agent Technology. IAT 2003 (2003) 438–441
16. Gorodetsky, V., Karsaev, O., Samoilov, V.: Infrastructural Issues for Agent-Based Distributed Learning. Proceedings of IADM2006, IEEE Computer Society Press
17. Han, J., Kamber, M.: Data Mining: Concepts and Techniques (2nd version). Morgan Kaufmann (2006)
18. Kaya, M. and Alhajj, R.: A novel approach to multiagent reinforcement learning: utilizing OLAP mining in the learning process. IEEE Transactions on Systems, Man and Cybernetics, Part C, Volume 35, Issue 4 (2005) 582–590
19. Kaya, M. and Alhajj, R.: Fuzzy OLAP association rules mining-based modular reinforcement learning approach for multiagent systems. IEEE Transactions on Systems, Man and Cybernetics, Part B, Volume 35, Issue 2 (2005) 326–338
20. Klusch, M., Lodi, S. and Gianluca, M.: The role of agents in distributed data mining: issues and benefits. Intelligent Agent Technology (2003) 211–217
21. Klusch, M., Lodi, S. and Moro, G.: Agent-Based Distributed Data Mining: The KDEC Scheme. Intelligent Information Agents: The AgentLink Perspective Volume 2586 (2003) Lecture Notes in Computer Science

22. Klusch, M., Lodi, S. and Moro, G.: Issues of agent-based distributed data mining. Proceedings of AAMAS, ACM Press (2003)
23. Letia, I.A., Craciun, F., et. al.: First Experiments for Mining Sequential Patterns on Distributed Sites with Multi-agents. Intelligent Data Engineering and Automated Learning - IDEAL 2000: Data Mining, Financial Engineering, and Intelligent Agents, 19 Volume 1983 (2000)
24. Liu, J. and You, J.: Smart shopper: an agent-based web-mining approach to Internet shopping. IEEE Transactions on Fuzzy Systems, Volume 11, Issue 2 (2003)
25. Mitkas, P.: Knowledge Discovery for Training Intelligent Agents: Methodology, Tools and Applications. Autonomous Intelligent Systems: Agents and Data Mining, Lecture Notes in Computer Science Volume 3505 (2005)
26. Lu, H., Sterling, L. and Wyatt, A.: Knowledge Discovery in SportsFinder: An Agent to Extract Sports Results from the Web. PAKDD-99, Volume 1574 (1999)
27. Mohammadian, M.: Intelligent Agents for Data Mining and Information Retrieval, Idea Group Publishing (2004)
28. Ong, K., Zhang, Z. and Ng, W., Lim, E.: Agents and stream data mining: a new perspective. IEEE Intelligent Systems, Volume 20, Issue 3 60–67
29. Rea, S.: Building Intelligent .NET Applications: Agents, Data Mining, Rule-Based Systems, and Speech Processing. Addison-Wesley Professional (2004)
30. Sian, S.: Extending Learning to Multiple Agents: Issues and a Model for Multi-Agent Machine Learning (MA-ML). In Proceedings of the European Workshop Sessions on Learning EWSL91 (Kodratroff, Y.) Springer-Verlag (1991) 458–472
31. Symeonidis, A., Mitkas, P.: Agent Intelligence Through Data Mining, Springer (2006)
32. Wooldridge, M.: An Introduction to MultiAgent Systems, Wiley (2002)
33. Zhang, C., Zhang, Z., Cao, L.: Agents and Data Mining: Mutual Enhancement by Integration. Autonomous Intelligent Systems: Agents and Data Mining, Volume 3505 (2005)
34. Zhang, Z., Zhang, C.: Agent-Based Hybrid Intelligent System for Data Mining. Agent-Based Hybrid Intelligent Systems, Volume 2938 (2004)
35. Zhong, N., Liu, J., Sun, R.: Intelligent agents and data mining for cognitive systems? Cognitive Systems Research Volume 5, Issue 3 (2004) 169–170

Evaluating Knowledge Intensive Multi-agent Systems

Christos Dimou[1], Andreas L. Symeonidis[1,2], and Pericles A. Mitkas[1,2]

[1] Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54 124, Thessaloniki, Greece
[2] Intelligent Systems and Software Engineering Laboratory, Informatics and Telematics Institute/CERTH, 57 001, Thessaloniki, Greece

Abstract. As modern applications tend to stretch between large, ever-growing datasets and increasing demand for meaningful content at the user end, more elaborate and sophisticated knowledge extraction technologies are needed. Towards this direction, the inherently contradicting technologies of deductive software agents and inductive data mining have been integrated, in order to address knowledge intensive problems. However, there exists no generalized evaluation methodology for assessing the efficiency of such applications. On the one hand, existing data mining evaluation methods focus only on algorithmic precision, ignoring overall system performance issues. On the other hand, existing systems evaluation techniques are insufficient, as the emergent intelligent behavior of agents introduce unpredictable factors of performance. In this paper, we present a generalized methodology for performance evaluation of intelligent agents that employ knowledge models produced through data mining. The proposed methodology consists of concise steps for selecting appropriate metrics, defining measurement methodologies and aggregating the measured performance indicators into thorough system characterizations. The paper concludes with a demonstration of the proposed methodology to a real world application, in the Supply Chain Management domain.

1 Introduction

During the previous years, the exponential growth of the amount of available data has pushed user needs towards a more knowledge-demanding direction. Today's applications are, therefore, required to extract knowledge from large or distributed repositories of text, multimedia or hybrid content. The nature of this quest renders impossible to use traditional deterministic computing techniques. Instead, the challenge for open and dynamic solutions in knowledge discovery is addressed by various machine learning and other soft computing techniques. Most notably, Data Mining (DM) produces useful patterns and associations from large data repositories that can later be used as *knowledge nuggets*, within the context of any application.

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 74–87, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Individual knowledge discovery activities, introduced by DM techniques are often required to be orchestrated, integrated and presented to end users in a unified way. Moreover, integrated knowledge should be exploited and embodied in autonomous software for learning purposes. Agent Technology (AT) is a promising paradigm that is suitable for modelling and implementing the unification of DM tasks, as well as for providing autonomous entity models that dynamically incorporate and use existing knowledge. Indeed, a plethora of agent-related solutions for knowledge-based systems can be found in the literature, and more specifically in the area of Agent-related data mining.

Despite the numerous related agent development methodologies, that deal with most of the steps of the development lifecycle, there is a remarkable lack of generalized evaluation methodologies for the systems in question. Evaluation of performance, a fundamental step of any development methodology, provides developers with countable, qualitative and verifiable attributes in an effort for better understanding the nature of a system at hand. Additionally, generalized and standardized evaluation procedures allow third parties to safely verify the acclaimed properties of systems or newly discovered scientific findings.

Existing evaluation approaches address either the DM algorithmic issues or the overall system performance. Both approaches come short in the case of AT and DM integration, due to the complex and dynamic nature of the produced systems. In the case of DM evaluation, focus is given on the statistical performance of individual techniques, in terms of precision and recall, ignoring the actual impact of the extracted knowledge to the application level. In the case of overall system evaluation, existing methods fail to deal satisfactorily with emergent agent behaviors that may not be known at design time.

In this paper, we present a generalized methodology for evaluating the performance of DM-enriched Multi-Agent Systems (MAS). A consice set of iterative methodological steps is presented, focusing on three fundamental evaluation aspects, namely the selection of a) metrics, b) measurement method, and c) aggregation methods. The proposed methodology is designed to assist developers as an off-the-shelf tool that can be integrated in the system development methodology. As an example of this incorporation, we briefly present Agent Academy, an open source platform for developing and training agents through data mining.

The remainder of this paper is organized as follows: in Section 2 we review the related bibliography; in Section 3, we present the proposed evaluation methodology in detail; in Section 4, we apply the evaluation guidelines to a real world demonstrator; finally, in Section 5 we summarize and provide future pointers.

2 Related Work

The proposed work draws from and extends related work in the direction of both AT-DM integration and evaluation of intelligent agents. In the field of AT-DM integration, there exist various related efforts that either use AT for enhanced DM or exploit DM advantages for incorporation in MAS. Most notably, in [1], inductive and deductive logic are combined for reasoning purposes in the field of customer care. In this work, deduction is used when complete information is available, whereas induction is employed to forecast behaviors of customers when the available information is incomplete. Inductive and deductive reasoning are also combined in [2], where logic terms of model data, information and knowledge are incorporated and processed by deductive agents. In [3], an integration of deductive database queries and inductive analysis on these queries and their produced knowledge is presented. Finally, [4] presents a unified methodology for transferring DM extracted knowledge into newly created agents. Knowledge models are generated through DM on the various levels of knowledge diffusion and are dynamically incorporated in agents. The iterative process of retraining through DM on newly acquired data is employed, in order to enhance the efficiency of intelligent agent behavior.

In our effort to study crucial performance issues for DM-enriched MAS, we present a literature review on intelligent agent evaluation. Although evaluation is an all encompassing term that may refer to either algorithmic performance or system performance, in this work we focus mainly on the latter. Indeed, within the context of DM and Machine Learning, a plethora of metrics and evaluation tools have been proposed, including precision, recall, F-measure, ROC Curves and fitness functions.

However, there is a remarkable lack of evaluation methodologies for intelligent systems that employ such algorithms for knowledge extraction purposes. Instead, researchers often have to devise their own ad-hoc metrics and experimental procedures. In fact, in some cases, the chosen parameters or input data are chosen so as to produce the best results for the -each time presented- method. Moreover, the findings are often supported by qualitatively arguments only, in favor of the proposed system and no debate with respect to its drawbacks is provided. Consequently, it is impossible for a third party to repeat the evaluation procedure and validate the quality of the proposed solution by concluding to similar results. The need for a generalized evaluation framework is, thus, evident.

In the literature, two general research approaches towards the direction of engineering aspects evaluation exist: a) bottom-up and b) top-down. The first approach represents the strong AI perspective on the problem, indicating that intelligent systems may exhibit any level of intelligence comparable to human abilities. Zadeh [5] argues that evaluating such systems is infeasible today, due to the lack of powerful formal languages for defining intelligence and appropriate intelligent metrics. The second approach represents the weak AI or engineering perspective, according to which intelligent systems are systems of increased complexity that are nevertheless well-defined in specific application domains, designed for solving specific problems. It is suggested that intelligent performance can be effectively evaluated after a concise decomposition of the problem scope and definitions of relative metrics and measurement procedures. Driven by the urging need to evaluate and compare existing or emergent applications, we adopt the top-down approach.

Ongoing domain-specific efforts for generalized metrics and evaluation methodologies exist in application fields, such as robotics and autonomic computing.

In robotics, evaluation efforts span from autonomous vehicle navigation (e.g. [6],[7]) to hybrid human-robot control systems (e.g. [8],[9]). In autonomic computing, emphasis is given to the quality assessment of the selected self-managing techniques [10]. Both fields provide usefull metrics and thorough methodological steps. However, neither of the above approaches are complete and mature nor do they provide us with relevant tools for the case of knowledge infusion in autonomous entities.

3 Evaluation Methodology

The proposed evaluation methodology serves as an off-the-shelf tool for researchers and developers in this field. Composed of theoretical analysis tools, it provides guidelines and techniques that can be used, adopted or extended for the application domain at hand. The methodology follows the top-down approach, mentioned earlier, and is therefore applicable to existing applications or applications that meet current agent oriented engineering concepts and follow the definitions for agent systems and DM terms provided in previous sections. Moreover, we only consider only observable agent behaviors that derive from the applied DM techniques. We therefore need to generalize existing DM metrics and introduce new intelligent performance metrics.

For establishing an evaluation framework that meets the above characteristics, we define:

- *Horizontal aspects*, the essential methodological steps, that if followed sequentially in an iterative manner, will comprise a complete evaluation methodology. The horizontal aspects of our methodology are:
  - *Definitions and theoretical background* on evaluation terms and relevant techniques.
  - *Theoretical representation tools* that can help designers chose what to measure, how to measure and how to integrate specific findings.
- *Vertical aspects* are specific techniques that may be part of any of the above horizontal aspects and deal with the following three terms[11]:
  - *Metrics* that correspond to system features to be measured.
  - *Measurement methods* that define the actual experimental procedure of assigning measurement values to the selected metrics.
  - *Aggregation* of the metric-measurement pairs in single characterizations for the system.

In the remainder of this section, we examine the above mentioned horizontal aspects in turn, analyzing each of their vertical aspects accordingly.

3.1 Definitions and Theoretical Background

The definitions of relevant terms and the corresponding theoretical background is of vital importance, in order to determine the scope and goals of evaluation. Any developer, before actually initiating his/her experiments, must have full grasp of what can and what cannot be evaluated. We, hereinafter, present relevant definitions and background theory with respect to: a) metrics, b) measurement methods, and c) aggregation.

Metrics. Metrics are standards that define measurable attributes of entities, their units and their scopes. Metrics are the essential building blocks of any evaluation process, since they allow the establishment of specific goals for improvement. A specific metric provides an indication of the degree to which a specific system attribute has met its defined goal. Deviation from the desired range of values indicates that improvement is needed in the related parts or modules of the system. With respect to a complete evaluation methodology, a metric is the answer to the question: "*What should I evaluate?*".

Measurement. Measurement is defined as "the process of ascertaining the attributes, dimensions, extend, quantity, degree of capacity of some object of observation and representing these in the qualitative or quantitative terms of a data language"[12]. Having selected the appropriate metrics, measurement is the next fundamental methodological step that systematically assigns specific values to these metrics. Typical measurement methods consists of experimental design and data collection. A measurement method is the answer to the question "*How should I perform the experimental evaluation?*".

Aggregation. Aggregation, or composition, is the process of summarizing multiple measurements into a single measurement is such a manner that the output measurement will be characteristic of the system performance. Aggregation groups and combines the collected measurements, possibly by the use of weights of importance, in order to conclude to atomic characterization for the evaluated system. For example, an evaluated system may perform exceptionally well in terms of response time metrics (timeliness), but these responses may be far from correct (accuracy). An aggregation process must weightedly balance contradicting measures and provide an overall view of parts or the whole of the system, within boundaries of acceptable performance. Aggregation is the answer to the question: "*What is the outcome of the evaluation procedure?*".

3.2 Theoretical Representation Tools

We next present a set of theoretical representation tools that aim to assist users throughout the designing of the evaluation procedure, by providing sets of options and guidelines for intelligent performance assessment.

Metrics. We introduce a metrics representation theoretical tool for metric categorization in the form of an acyclic directed graph. The graph is organized in layers or *views* of granularity from general to specific, as further explained below. A user may traverse the graph in a top-down manner and, depending on the choices made, he/she shall conclude to a set of suitable metrics. This graph is designed to be general, but also provides the option of extensibility for necessary domain specific metrics.

In the proposed approach, we organize a metrics graph into four views, as depicted in Figure 1:

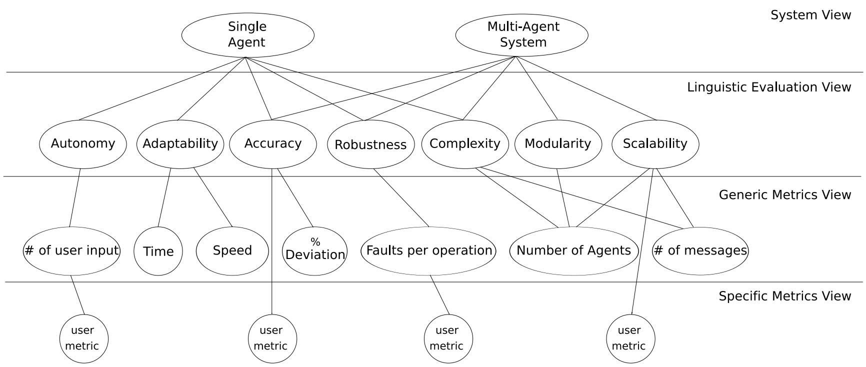


Fig. 1. Metrics graph

These views include:

1. *System view*: At the top-most level, the class of the application is selected. A user may chose between single-agent, multi-agent society and multi-agent competition, depending on the scope and focus of the evaluation effort.
2. *Linguistic evaluation view*: At this level, a user chooses the appropriate verbal characterizations of system aspects, such as accuracy, timeliness, robustness and scalability. These abstract high level characterizations exclude parts of the underlying metrics, while focusing on the aspects of interest to the evaluator.
3. *Generic metrics view*: This level consists of metrics that are general and independent of the application field, such as response time, number of agents and message exchange frequency. The user may either use directly these metrics or refine them by continuing to the next level.
4. *Specific metrics view*: The final level consists of metrics that are specific to the application field. These metrics are only defined by the user, since they are not known *a priori* to a generalized evaluation methodology. Newly defined metrics must conform to the metric definition and parametrization presented in the previous section. Finally, they must be appended to one of the graph nodes of the above levels with directed arcs.

After selecting the metrics from this graph, the user is requested to define a set of parameters for each metric, including the preferred scale of measurement and other attributes, such as frequency of measurement, time intervals etc.

Measurement Methods. Before implementing the actual measurement process, one must define the measurement method. Kitchenham [13] provides a categorization of measurement techniques, with respect to the types of properties employed and the nature of the experimental technique. Inspired by this work, we provide the following categorization:

1. Quantitative experiment: An investigation of the quantitative impact of methods/tools organized as a formal experiment
2. Quantitative case study: An investigation of the quantitative impact of methods/tools organized as a case study
3. Quantitative survey: An investigation of the quantitative impact of methods/tools organized as a survey
4. Qualitative screening: A feature-based evaluation done by a single individual who not only determines the features to be assessed and their rating scale but also does the assessment. For initial screening, the evaluations are usually based on literature describing the software method/tools rather than actual use of the methods/tools
5. Qualitative experiment: A feature-based evaluation done by a group of potential user who are expected to try out the methods/tools on typical tasks before making their evaluations
6. Qualitative case study: A feature-based evaluation performed by someone who has used the method/tool on a real project
7. Qualitative survey: A feature-based evaluation done by people who have had experience of using the method/tool, or have studied the method/tool. The difference between a survey and an experiment is that participation in a survey is at the discretion of the subject
8. Qualitative effects analysis: A subjective assessment of the quantitative effect of methods and tools, based on expert opinion
9. Benchmarking : A process of running a number of standard tests using alternative tools/methods (usually tools) and assessing the relative performance of the tools against those tests

Having selected the measurement method, one must thoroughly provide an experimental design prototype and a data collection procedure. As stated earlier, our methodology can only provide a set of guidelines that any designer may adjust to their specific application. A typical experimental design procedure must describe thoroughly the objectives of the experiments and ensure that these objectives can be reached using the specified techniques.

The last step of the measurement methodology is to carry out the data collection process. Here, the basic guidelines for the designer to follow are to ensure that the data collection process is well defined and monitor the data collection and watch for deviations from the experiment design.

Aggregation. Following the collection of measurement values and the construction of metric-measurement pairs, the problem of aggregation arises. In the evaluation process, aggregation occurs naturally in order to summarize the experimental findings into a single characterization of the performance, either of single modules, or the system as a whole. In the case of the metrics graph of the proposed methodology, after having the measurements collected, the user must traverse the graph in a bottom-up manner. From the *specific metrics view* and the *general metrics view*, he/she must proceed upwards and, at each view, apply aggregation techniques to provide single characterizations for every parent node.

It is apparent that a natural method for combining diverse and heterogeneous measurement information and linguistic characterizations is needed. We argue that *fuzzy aggregation* provides us with the appropriate natural functionality for this purpose. The term *natural* refers to the ability of the evaluator to express the evaluation findings in a manner that is coherent to their natural language. In other words, the fuzzy aggregation process translates the problem of combining numerical, ordinal or other measures into a collection of verbal characterizations for the system performance.

The proposed fuzzy aggregation method consists of four steps:

1. *Define weights in the metrics graph.* This process determines the importance of each node in the metrics graph with respect to the overall system performance. This decision relies heavily on the application domain as well as the requirements of each application. Hence, the determination of the weights may occur either a) semi-automatically, in case historical data on the importance of each node are available, possibly by an expert system, or b) directly by an expert user, the system designers in most cases.
2. *Define corresponding fuzzy scales for each metric.* The next step deals with the definition of fuzzy scales for the selected metrics. Fuzzy scales are defined by ordinal linguistic variables, such as *low*, *moderate*, *high* and membership functions that map numerical values to the above variables. Having the scales defined, one may already have scales for *natural* characterizations of performance, such as *high response time* or *moderate accuracy*, with respect to desired values.
3. *Convert actual measurements to fuzzy scales.* The conversion is a simple import of the selected measurements to the membership functions defined in the previous step.
4. *Apply a corresponding fuzzy aggregation operator at each view of the graph.* A wide variety of fuzzy aggregation operators exists [14], which can be categorized in:
   - Conjunctive operators, that perform aggregation with the logical "and" connection.
   - Disjunctive operators, that perform aggregation with the logical "or" connection.
   - Compensative operators, which are comprised between minimum and maximum, such as mean or median operators.
   - Non-compensative operators, that do not belong to any of the above categories, such as symmetric sums.

Theoretical Tools: Summary. In Table 1, we summarize the required methodological steps with respect to the theoretical tools, which take place at the

Table 1. Summarization of methodological steps

1. Traverse metrics graph and select metrics
2. Provide domain specific metrics (optionally)
3. Determine metrics parameters
4. Specify measurement method and parameters
5. Execute experiments
6. Define weights in the graph
7. Define fuzzy scales and convert measurements accordingly
8. Select and apply aggregation operators on the collected measurements

evaluation process of a development methodology. In section 4, we present a real world case study on which the presented methodology is thoroughly applied.

4 A Real World Demonstrator

For validating the proposed methodology, we have selected Supply Chain Management (SCM) as a representative domain for testing agents that utilize DM techniques. We have implemented an SCM agent under the name Mertacor that has successfully participated in past Trading Agent SCM Competitions. Mertacor combines agent features with DM techniques. In the remainder of this section, we provide an overview of the SCM domain, the competition scenario and Mertacor's architecture. We conclude by applying the proposed evaluation methodology to different implementations of Mertacor.

Supply Chain Management and TAC Competition. SCM tasks comprise the management of materials, information and finance in a network consisting of suppliers, manufacturers, distributors and customers. SCM strategies target at the efficient orchestration of the sequence of tasks, from raw materials to end-user service. Traditional SCM relied heavily on rigid and predefined contracts between participating parties. However, the need for dynamic configuration of the supply chain, as indicated nowaydays by global markets, became imperative. Modern SCM approaches focus on the integration, optimization and management of the entire process of material sourcing, production, inventory management and distribution to customers.

Mertacor Architecture. Mertacor, as introduced in [15], is an agent that has successfully participated in the Trading Agent Competitions (TAC) [16]. The architecture of Mertacor consists of four cooperating modules:

1. the *Inventory Module*(IM). Mertacor introduces an assemble-to-order (ATO) strategy, which is a combination of two popular inventory strategies, namely make-to-order and make-to-stock.

2. the *Procuring Module*(PM). This module predicts future demands and orders affordable components, balancing between cheap procurement and running needs in the assembly line.
3. the *Factory Module*(FM). This module constructs assembly schedules and provides the Bidding Module with information on the factory production capacity, based on simulation of customer demand for the next 15 game days.
4. the *Bidding Module*(BM). This module attempts to predict a winning bid for each order, by performing DM on logs of past games.

Mertacor's core integrates this modules into a transparently robust unit that handles negotiations with both customers and suppliers. This architecture provides flexibility and extensibility, permitting the application of Mertacor's strategy to other real-life SCM environments.

Evaluating Mertacor's Performance. In the remainder of this section, we apply the proposed evaluation methodology to various implementations of Mertacor. In our effort to assess the impact of DM in Mertacor's performance, we require that the experiments are planned is such way that deals with both DM algorithmic-specific efficacy and their impact on overall agent performance. We follow the methodological steps defined in Table 1.

Step 1: Traverse metrics graph and select metrics Starting at the *System view*, we select the *Single Agent* node and corresponding path. This choice is attributed to the nature of auctioning environments; we, being the developers of Mertacor, have complete control only on the agent's executing threat and observe the auctioning world only through Mertacor's perspective. We, therefore, need to focus on performance aspects that exclusively deal with this single agent.

At the *Linguistic Evaluation View*, we select the linguistic metrics of *Accuracy* and *Timeliness*. Indeed, from our experience in SCM auctions, these three characteristics are the most fundamental, as the outcome of each auction is heavily dependent on the deviation of the forecasted bid, the timely deliver of the bid and the ability of the agent to adapt in dynamic environments, respectively.

At the *Generic Metrics View* we only select *Time*, as standard metric for *Timeliness*. The rest of the metrics are domain specific and are, therefore, defined in the next methodological step.

Step 2: Provide domain specific metrics Metrics for *Accuracy* should directly refer to DM related performance, since the outcome of the application of DM is directly related to the selected bid. For this purpose, we have selected the *Correlation Coefficient* (cc), the *Relative Absolute Error* (RAE) and the *Root Mean Square Error* (RMSE) metrics.

An instance of the metrics graph for this evaluation effort is depicted in Figure 2

Step 3: Determine metrics parameters We now continue by defining the scale of each metric. For the two linguistic metrics, *Accuracy* and *Timeliness*, we define

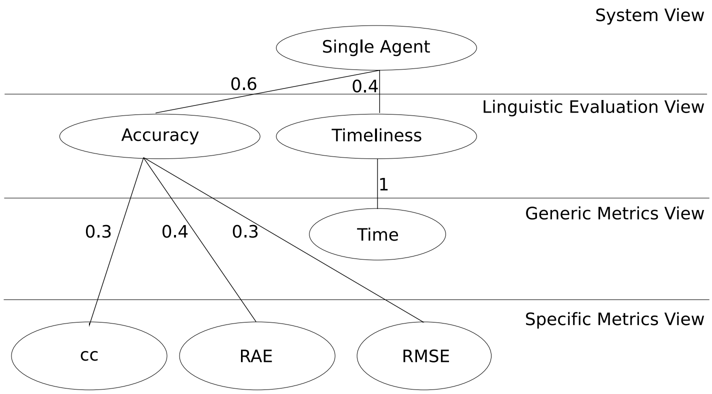


Fig. 2. Resulted metrics graph for Mertacor evaluation

the corresponding fuzzy scales in *Step 7* of the methodology. For the generic and specific metrics, we provide the following scales:

1. CC: The correlation coefficient is the degree at which the forecasted bid and the resulted price are correlated. The *cc* lies in the [-1,1] interval.
2. RAE: The Relative Absolute Error is a percentage indicator for the deviation of the above mentioned variables.
3. RMSE: The Root Mean Square Error is another well-known DM metric for the above mentioned variables.
4. Time: In TAC SCM auctions, bids are normally submitted just before the end of each predefined auction interval. One could argue that, since this time constraint exists, all agents have a time barrier to bid and therefore all bidding calculation procedures should be characterized either as successful or failed. In that context, timeliness is only a binary metric that provides no further performance indication. However, due to the modular architecture of Mertacor, the earliest possible decision on the bid, allows the agent to perform other game-related tasks in this interval. We therefore define *Time* as the time interval between the first call of the related bidding API function and the determination of the bidding value, in milliseconds.

Step 4: Specify measurement method and parameters Estimation of the winning price of the bids can be modeled as a regression problem, where the desired output is the agent's bidding price for clients' RFQs and the inputs are the parameters related to the bid that are known to the agent. The initial set of attributes considered are the demand (Total PCs requested each day), the demand in the product's market range, the due date of the order, the reserve price of components, and the maximum and minimum prices of same type PCs sold in the last days (2 previous days for maximum 4 for minimum), as shown in Table 2.

Table 2. Set of SCM auction attributes for DM

Attribute description	Attribute name
Demand (Total PCs requested the day the RFQ was issued)	demandAll
Demand in the product's market range	demandRange
Duedate	dueDate
Reserve price	reservePrice
Maximum price of PCs of same type sold in the last 1 day	max1
Maximum price of PCs of same type sold in the last 2 days	max2
Minimum price of PCs of same type sold in the last 1 day	min1
Minimum price of PCs of same type sold in the last 2 days	min2
Minimum price of PCs of same type sold in the last 3 days	min3
Minimum price of PCs of same type sold in the last 4 days	min4
Winning price of the bid	price

Finally, for training purposes, four different classification (regression) and two meta-classification schemes were applied, in order to decide on the one that optimally meets the problem of predicting the winning bid of an order:

1. Linear Regression
2. Neural Networks
3. SMOreg (Support Vector Machines)
4. the M5' algorithm
5. Additive Regression
6. Bagging

Step 5: Execute experiments In order to experiment on the data with a variety of training techniques and algorithms, the WEKA [17] was selected, providing with a wide range of filters for pre-processing, model evaluation, visualization and post-processing. The results of the experimental procedures are presented in Table 3.

Table 3. Results of experiments

Algorithm	CC	RAE (%)	RMSE	Time(ms)
Linear Regression	0.93	28.99	90.17	108
Neural Networks	0.93	32.91	94.69	111
Support Vector Machines	0.93	26.47	89.08	157
M5'	0.95	22.77	61.09	140
Additive Regr.	1.00	3.21	22.12	192
Bagging	0.98	14.89	52.02	201

Step 6: Define weights in the graph This step requires a subjective, expert-initiated attribution of weights to the corresponding edges of the metrics graph. Driven by our experience in the field, we assign a higher weight to *Accuracy* (0.7) and a lesser weight to *Timeliness* (0.3). The corresponding weights are illustrated in Figure 2

Step 7: Define fuzzy scales and convert measurements accordingly We provide the following fuzzy sets for the selected metrics:

– Fuzzy variables *very low*,*low*,*medium*,*high* and *very high* for the *RAE* and *RMSE* metrics
– Fuzzy variables *low* and *high* for the *CC* metric
– Fuzzy variables *low*, *medium* and *high* for the *Time* metric

We provide fuzzy variables *very low*,*low*,*medium*,*high* and *very high* for the *CC*, *RAE*, *RMSE* and *Time* metrics

Step 8: Select and apply aggregation operators on the collected measurements The final step of the methodology consists of the application of the selected aggregation method. As described in [14], the application of weighted operators result into a single characterization for every linguistic metric. After summarizing the results it can be seen that *Additive Regression* exhibit the best performance for all data subsets, as it balances between large accuracy and small time responses.

5 Conclusions and Future Work

As the number of application that integrate DM and AT increase, the need for assessing the overall system performance is imperative. In this paper, we have presented a generalized methodology for evaluating agents and MAS that employ DM techniques for knowledge extraction and knowledge model generation. The proposed methodology comprises a set of concise steps that guide an evaluator through the evaluation process. A novel theoretical representation tool introduces a metrics graph and appropriate selection guidelines for measurement and aggregation methods. A real world DM-enriched agent in the field of Supply Chain Management has used to demonstrate the applicability of the proposed methodology. Future work in this direction include the specification of a unique metrics ontology for the proposed metrics representation graph and the expansion of the graph with a complete set of real world metrics, borrowed either from the software engineering discipline or existing, ad-hoc efforts in intelligent systems evaluation. Finally, the proposed methodology must be thoroughly tested in a number of diverse and representative case studies.

References

1. Boris Galitsky and Rajesh Pampapathi: Deductive and inductive reasoning for processing the claims of unsatisfied customers. In Paul W. H. Chung, Chris J. Hinde, and Moonis Ali, editors, *IEA/AIE*, volume 2718 of *Lecture Notes in Computer Science* Springer (2003) 21–30

2. Alvaro A. A. Fernandes: Combining inductive and deductive inference in knowledge management tasks. In *DEXA '00: Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, page 1109, Washington, DC, USA (2000) IEEE Computer Society
3. Bob Kero, Lucian Russell, Shalom Tsur, and Wei-Min Shen: An overview of database mining techniques. In *KDOOD/TDOOD* (1995) 1–8
4. Andreas L. Symeonidis and Pericles A. Mitkas: *Agent Intelligence Through Data Mining*. Springer Science and Business Media (2005)
5. Lotfi A. Zadeh: In quest of performance metrics for intelligent systemsa challenge that cannot be met with existing methods. In *Proc. of the Third International Workshop on Performance Metrics for Intelligent Systems (PERMIS)*, 13-15 August 2002
6. Andrew L. Nelson, Edward Grant, and Thomas C. Henderson: Competitive relative performance evaluation of neural controllers for competitive game playing with teams of real mobile robots. In *Proc. of the Third International Workshop on Performance Metrics for Intelligent Systems (PERMIS)*, 13-15 August 2002
7. Noah Zimmerman, Craig Schlenoff, Stephen Balakirsky, and Robert Wray: Performance evaluation of tools and techniques for representing cost-based decision criteria for on-road autonomous navigation. In *Proc. of the Third International Workshop on Performance Metrics for Intelligent Systems (PERMIS)*, 3-15 August 2002
8. Jean Scholtz, Brian Antonishek, and Jeff Young: Evaluation of human-robot interaction in the nist reference search and rescue test arenas. In *Proc. of the Fourth International Workshop on Performance Metrics for Intelligent Systems (PERMIS)*, 13-15 August 2002
9. Jennifer L. Burke, Robin R. Murphy, Dawn R. Riddle, and Thomas Fincannon: Task performance metrics in human-robot interaction: Taking a systems approach. In *Proc. of the Fourth International Workshop on Performance Metrics for Intelligent Systems (PERMIS)*, 13-15 August 2002
10. Markus C. Huebscher and Julie A. McCann: Evaluation issues in autonomic computing. In *Proceedings of Grid and Cooperative Computing Workshops (GCC)* (2004) 597–608
11. Timothy K. Shih. Evolution of mobile agents. In *Proc. of the First International Workshop on Performance Metrics for Intelligent Systems (PERMIS)* 12-14 August 2000
12. Klaus Krippendorff: *A Dictionary of Cybernetics.* The American Society of Cybernetics, Norfolk, VA, USA (1986)
13. Barbara Ann Kitchenham: Evaluating software engineering methods and tool, part 2: selecting an appropriate evaluation method technical criteria. *SIGSOFT Softw. Eng. Notes*, 21(2) (1996) 11–15
14. Michel Grabisch, Sergei A. Orlovski, and Ronald R. Yager: Fuzzy aggregation of numerical preferences (1998) 31–68
15. Ioannis Kontogounis, Kyriakos Chatzidimitriou, Andreas L. Symeonidis, and Pericles A. Mitkas: A robust agent design for dynamic scm environments. In *LNAI, Vol. 3955* Springer-Verlag (2006) 127–136
16. Arunachalam R. Sadeh N. Ericsson J. Finne N. Collins, J. and S. Janson: The supply chain management game for the 2005 trading agent competition. Technical report, CMU (2004)
17. Ian H. Witten and Eibe Frank: *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco (2005)

Towards an Ant System for Autonomous Agents

Zoheir Ezziane

College of Information Technology, University of Dubai
P.O. Box 14143 Dubai, United Arab Emirates
zezziane@ud.ac.ae

Abstract. Over the past few years, Electronic Commerce has become an increasingly central part of the economy. More and more transactions, both from business to consumer and between businesses, are taking place online. Simple fixed cost business transactions are often automated at one or both ends and auctions are mainly conducted by automated auctioneer software. Bid evaluation is a costly optimization problem. It involves the examination of all possible combinations of bids and subsequently the selection of the best combination; based on factors such as price, supplier reliability, schedule risk, etc. If an exhaustive search approach is used to derive the optimum solution, the generation of possible bid combinations will explode combinatorial as the number of bids grow. This paper shows how to build an ant system for autonomous agents. In addition, it assigns customer agents' services to supplier agents' bids in the best possible way while considering several factors.

Keywords: Distributed agents, agent economies, e-commerce, ant colony, bid evaluation, e-auctions.

1 Introduction

Over the past few years, electronic commerce (e-commerce) has become an increasingly central part of the economy. An Internet presence is considered central part of doing business, rather than exotic add-on to a company. More and more transactions, both from business to consumer and between businesses, are taking place online. Simple fixed cost business transactions are often automated at one or both ends and auctions are mainly conducted by automated auctioneer software [7].

Internet auctions appeared on the scene in the mid 1990s, and quickly became one the most successful applications of e-commerce. Ebay, the premier consumer-to-consumer (C2C) Internet auction site, is generally held up as an exemplar for the industry. However, it is widely predicted that the potential transaction volume in business-to-business (B2B) auctions will be much greater than in C2C channel [8].

Auctions are exemplary negotiation mechanisms, which are particularly well suited to automation. A variety of auctions, models and techniques exist due to the differences in the trading products, in the rules for users as well as in the pricing policies. Furthermore, a broad categorization of auctions can be based on auctions pricing policies and will identify the increasing price (or *English*), decreasing price (*Dutch*),

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 88–93, 2007.
© Springer-Verlag Berlin Heidelberg 2007

the first (second)-price sealed bid, the call markets and continuous double (or *CDA*) auctions. Web-based implementations of different auction types involve the specification of an auction model as well as appropriate design of methodologies and techniques which will support the overall auction process.

Generally, an auction may sell a single item or multiple items. Auction such as English and the traditional Dutch sell individual item whereas combinatorial auction sell multiple items that are complimentary. In addition, evaluating bids in combinatorial auction (optimal allocation) is known to be NP-hard. Bid evaluation is a costly optimization problem. It involves the examination of all possible combinations of bids and subsequently the selection of the best combination based on factors such as price, supplier reliability, schedule risk, etc. If an exhaustive search approach is used to derive the optimum solution, the generation of possible bid combinations will explode combinatorial as the number of bids grow [1].

Although ants are blind, they navigate complex environments and can find food some distance from their nest and return to their nest successfully. They do this by laying pheromones while they navigate their environment. This process modifies their environment to permit communication between the ants and the colony as well as memory for the return trip to the nest. Ants tend to take the best route between their nest and some external landmark. As more ants use a particular trail to an external landmark, the trail becomes higher in pheromone concentration. The higher the concentration of pheromone, the more ants will choose this route over others that might be available. This iterative process achieves sub-optimal to optimal trails between the endpoints.

The use of ant colonies is found to be effective in finding near-optimum solutions to combinatorial optimization problems such as TSP [2], cooperating agents [3] and discrete optimization [4]. This paper adopts the ant colonies solution to the bid evaluation problem in an autonomous multiagent environment.

2 Ant Algorithms

An ant is a single agent within the ant algorithm and is commonly part of a larger population (colony) used to solve a given problem. The ant is endowed with a set of simple rules that define how it chooses its path through the graph (or network). The ant maintains a tabu list, which is simply a list of nodes that it has visited. This list is kept so that the ant moves through each node only once. A path between two nodes on a graph where each node is visited only once is known as a Hamiltonian path.

The ant also maintains the list of nodes in its current tour in the order in which the ant has traveled. This is used later to identify the tour length through the nodes. In nature an ant will lay pheromone on the edges of the graph once the tour is complete.

Ant colony optimization (ACO) algorithms are such adaptive construction heuristics, in the sense that a colony of ants modifies the solution representation assigning higher pheromone trail strength to connections of services to bids, which are contained in better solutions. During the solution construction, ants select connections which have a high pheromone strength and by combining such connections they generate promising starting solutions for the local search algorithm. The set of good initial solutions generated by the ant algorithm is used as a startup by the local search algorithm in order to search for a local optimum. Therefore, for a given number of iterations many more local searches can be run rather than by starting from randomly generated solutions.

2.1 Agents in Electronic Markets and Online Auctions

Markets play an essential role in the economy by facilitating the exchange of information, goods, and services. Furthermore, there is growing evidence that software agents will play an increasing role as mediators in electronic markets [5], [6]. Mediators typically provide services such as searching for a product or supplier, negotiating the terms of a deal, providing payment services, and ensuring delivery of goods. Automated negotiation, particularly bid evaluation [1] is one market function where software agents' capability can be explored. Furthermore, in the long term, software agents will be more effective than human bidders in online auctions.

The bidding algorithm developed here is an ant system combined with a local search to undertake the reasoning behind finding the solution. The reason for this choice is that, in some online auctions, there is no optimal bidding strategy. This is because an agent's decision making about bidding involves multiple factors that are affected by the attitudes toward risk of its opponents, the nature of the markets supply (demand), and the preferences of the other bidders. Since it is a competitive environment, no agent can have all this information in advance, and the best that can be achieved is a satisfying strategy.

3 Ant Algorithm Adopted for Autonomous Agents

Autonomous trading involves negotiation among agents, who are assumed to be self-interested and possess limited rationality. Here it is assumed that there are two types of agents: the customer and the supplier. The customer agent requests for resources or services, and the supplier agent(s) may fulfill the request by offering resources or services for specified prices over specified periods. During negotiation, the supplier agent attempts to gain the highest possible price while the customer agent tries to bargain for the lowest price. For example a customer agent issues a list of requested services (S_0 …S_4). The supplier agent(s), on the other hand, formulates and submits sealed bids (B_0, B_1 and B_2) for the services. Upon receiving the bids, the customer agent then evaluates the bids based on factors such as prices, supplier, reliability, etc. Subsequently, it awards the tasks to the lowest bidder. The customer agent has the choice of either awarding the whole set of services to a specific supplier agent or granting specific resources to different suppliers.

3.1 Start

The initial step is the generation of the following:

- a number *Nc* of customer agents and number *Ns* of supplier agents
- a complete list of services where each one is associated with its own price.
- a list of requested services by each customer agent
- sealed bids for each supplier agent along with its relative reliability factor.

3.2 Bid Evaluation Function

An evaluation function calculates the bid-task combination (*bTc*) as follows:

$$Min\ f(x) = bTc = bTp + sR\ . \quad (1)$$

where *bTp* and *sR* represent bid-task price and supplier reliability respectively.

$$bTp = total\_bid\_prices \times weight\_\ factor\ . \quad (2)$$

$$sR = average\_supplier\_rel \times weight\_\ factor\_supplier\ . \quad (3)$$

A bid usually involves some risks, such as recovery cost, loss of value due to delays and cost of plan failure. Here in this research, supplier reliability is considered, which refers to the reputation of the supplier agent. In estimating the total cost of *bTc*, *bTp* and *sR* are multiplied by its corresponding weight factor (customer agent defined).

3.3 Ant's Characteristics

An ant algorithm, used here, represents an environment for the supplier agents to compete and win for the bids requested by the client agents. Each ant has the following characteristics:

1. When it chooses to assign service *j* to bid *i*, it leaves the pheromone τ_{ij}.
2. It chooses the bid to which a given service is to be assigned with a probability which is a function of the connection between *i* and *j*, and of the pheromone.
3. Construct a schedule excluding services and bids already assigned, and stop when all services have been assigned.

The ant algorithm uses a population of *n* ants which construct solutions step by step, assigning a service to each bid. When all ants have constructed their respective schedules, the best assignments are rewarded so as to encourage the identification of ever better solutions in the next cycles. This process stops when there are no changes in the assignments of bids to their corresponding services.

To satisfy the requirement that the ants assign service to a different bid, a tabu list is associated to each ant. Its purpose is to memorize the bids already used and stops the ant assigning them a new service before a cycle is complete (i.e., schedule). Once a schedule is completed, the tabu list is cleared and the ant is free to choose its connections again. A $tabu_x$ is needed to contain the tabu list for the x^{th} ant and $tabu_x(y)$ as the y^{th} element of the tabu list for the x^{th} ant (the bid occupied by the y^{th} service in the assignment made by the x^{th} ant).

3.4 Probability of Coupling

The calculation of the fitting between *i* and *j*, initially when no trace is observed will be modified by the experience acquired by the ants. Max-Min, an improvement of local search for ACO used to solve the TSP [9], is adopted here for the multiagent system. At each step, ant *k* first randomly chooses a service *i* among those not yet assigned, and then places it on a free bid *j* with a probability

$$P_{ij}^{k}(t)=\frac{\tau_{ij}(t)}{\sum_{l\in N_i^k}\tau_{il}(t)} \text{ if } j\in N_i^k$$

τ_{ij} is interpreted as the desirability of assigning service *i* to bid *j*, and N_i^k represents those bids that are still not assigned.

3.5 Pheromone Updates for Potential Assignments

Once a schedule is complete, the process of updating the trails is performed. This includes not only updating the trails for the pheromone deposited by ants, but also existing pheromone that has evaporated in time. This process helps in slowly removing any assignments that are part of unsuccessful bids through within the solution (final schedule). Equation 4 is used for pheromone evaporation.

$$\tau_{ij}(t)=\tau_{ij}(t)\times(1-\rho) \ . \tag{4}$$

After all ants have constructed a schedule (temporary solution), the pheromone trails are updated according to the following equations (Eq. 5.1 and Eq. 5.2).

$$\tau_{ij}(t)=\tau_{ij}(t)+\Delta\tau_{ij}^{best} \ . \tag{5}$$

Where $\Delta\tau_{ij}^{best}=\frac{1}{f_{\varphi}^{best}}$ if service *i* is assigned bid *j* in solution φ^{best} otherwise it is set to zero.

$$\tau_{ij}(t)=\rho\times\tau_{ij}(t) \qquad \text{Constant } \rho \text{ is a value between 0 and 1.} \tag{6}$$

f_{φ}^{best} represents the objective function value of φ^{best} which is here the iteration-best solution φ^{ib}. Thus, if in the best solution, services are often put on specific bids, these couplings (connections) will have a high amount of pheromone. Equation 5 indicates the process of adding new pheromone to the assignment (i.e., connection), and equation 6 reduces the intensity of pheromone deposited by the ant. Furthermore, to avoid search stagnation, the allowed range of the pheromone trail strengths is limited to the interval $[\tau_{min}, \tau_{max}]$, so that $\forall\tau_{ij}\ \ \tau_{min}\le\tau_{ij}\le\tau_{max}$. Initially, the pheromone trails are set to the upper trail limit, which triggers a wider search exploration at the beginning of the algorithm [9]. If after the pheromone update, $\tau_{ij}>\tau_{max}$ then $\tau_{ij}=\tau_{max}$; and if $\tau_{ij}<\tau_{max}$ then $\tau_{ij}=\tau_{min}$.

3.6 Restart

When the ant schedule is complete, the assignments updates based upon the schedule evaluations function values, and evaporation on all assignments has been performed, the algorithm is restarted. All tabu lists are cleared and the schedules evaluation values are reset to zero. The ants are permitted to connect services and bids using equations (5) and (6) as a guide to the next assignment to be searched. This process can be performed for a constant number of schedules, or until no changes have been seen for some number of schedules. The best schedule represents the solution.

4 Conclusion

An ant system has been proposed to solve the problem of finding the best possible schedule, in which customer agents get their services assigned to the lowest biddings possible submitted by the supplier agents. A solution is constructed by artificial ants and a local search algorithm guided by a refined probability which does not use any heuristics in its function. This simplified probability function lets ants make decisions and move relatively faster in assigning services to bids. This algorithm can also be extended and adopted to be used in real multiagent environment such as an online auction where customer agents are searching for the lowest bids from a pool of competing supplier agents. This research can be extended by conducting a comparison study between existing different approaches and demonstrates the efficiency of the ant system when used to find the best bidders in a multiagent environment.

References

1. Alvarez, Ma. Fe R., Bulos, R.de Dios: Genetic Local Search in an Automated Contracting Environment. LNCS Vol. 2738, Springer Verlag, (2003) 125–134
2. Dorigo, M.: Ant Colonies for the Traveling Salesman Problem. Biosystems, vol. 43 (1996) 73–81
3. Dorigo, M., Maniezzo, V., Colorni, A.: The Ant System: Optimization by a Colony of Cooperating Agents. Available at http://www.agent.ai/doc/upload/200302/dori96.pdf
4. Dorigo, M., Di Caro, G: Ant Algorithms for Discrete Optimization. Available at http://www.idsia.ch/~luca/ij_23-alife99.pdf
5. Easwaran, A.M., Pitt, J.: Simultaneous Optimization of Multiple Criteria for Efficient Agent Service Brokering. Proc. 5th Int'l Conf. Autonomous agents, ACM Press, (2001) 69–71
6. Ezziane, Z.: Designing Peer-To-Peer Agent Auctions Using an Object-Process Methodology. Frontiers in Artificial Intelligence and Applications Vol. 138 IOS Press (2006) 92–98
7. Priest, C., Bartolini, C., Byde, A.: Agent-Based Service Composition through Simultaneous Negotiation in Forward and Reverse Auctions. Proc. of the 4th ACM conference on Electronic commerce, ACM Press, San Diego, CA, USA (2003) 55–63
8. Rosenthal, R.: 2002 Competitive Assessment of eRFx Solutions Providers. Available at http://www.idc.com
9. Stutzle, T., Hoos, H.H.: Improvements on the Ant System: Introducing the Max-Min Ant System. Artificial Neural Networks and Genetic Algorithms, Springer Verlag, New York (1998) 245–249

Semantic Modelling in Agent-Based Software Development

Peter Graubmann[1] and Mikhail Roshchin[1,2]

[1] Siemens AG, CT SE 2, Otto-Hahn-Ring, 6, 81739 Munich, Germany
[2] Volgograd State Technical University, 400131 Volgograd, Russia
peter.graubmann@siemens.com, roshchin@gmail.com

Abstract. To facilitate automated agent-based software development and to support high effective and low cost selection, adaptation and integration of required functionality extensions into existing multi-agent software systems, we present our semantic modelling approach. It defines a structured annotation process, proposing concepts, techniques and a methodical support for the formal description of static and dynamic semantic information of software agents and their services emerging from heterogeneous environments. This information is organized by description patterns which are – according to our Logic-on-Demand concept – based on a variety of inference mechanisms which offer variability in expressiveness, reasoning power and the required analysis depth for the identification of agent properties and qualities.

Keywords: Agent-based Software Development, Semantic Web Techniques, Semantic Modelling, Heterogeneous environments.

1 Introduction

In multi-agent systems, agents are characterized by their need to communicate and interact in a cooperative and coordinated way. Thus, the benefits that can be gained from using the software agent paradigm for developing a service solution or an application program relies in the way in which multiple agents can flexibly come together, discover each other and organize their interoperation in order to achieve complex tasks and to provide valuable combined services for their customers. In addition, agents with their concepts of “autonomy”, “reactivity” and “social ability” and their usually very loosely coupling blend very well into service-oriented architectures (like integrated web services) where the very same problems occur: how to detect services currently available for a certain problem solution, how to determine the best-fitting service among this lot and how to establish the interaction properly.

Agent collaboration is impossible without precise matching of their interfaces. Adaptation of syntactically different but semantically identical descriptions includes reasoning on the semantic level of agent properties. Today, this is done manually. Thus, agent-based software development based on selection, adaptation, and integration, requires sufficient detailed semantics descriptions and techniques for an appropriate expressive representation. Properly organized semantic methods will support the composition of

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 94–99, 2007.
© Springer-Verlag Berlin Heidelberg 2007

different agents from heterogeneous domains, their dynamic configuration and real-time integration in existing environments.

There are several semantic approaches – primarily dedicated to services – which propose different description principles and means, based on their different objectives. They rely on different knowledge representation formalisms. Their usability in our context for service and agent description depends to a large extent on their expressiveness and the associated inference mechanisms. But, one of the approaches, OWL-S [5] is not sufficiently expressive for representing semantics of services since it does, for instance, neither provide enough logic constructs for describing dynamic quality properties nor supports the description of process modelling requirements. Aiming in overcoming the limitations of OWL-S, another approach WSMO [10] uses richer modelling languages like F-Logic [7], which, however, turn out to be undecidable and too complex for customers to express their requirements.

In our approach, we assume that all the information needed by an automated composition approach is made available through the *semantic modelling* of the software agent and its related services. Actually, this paper extends our approach to agent-based systems which was originally formulated for software components (as, for instance, in [9]). In this approach, semantic modelling means to provide structured semantic descriptions for agents called *annotations* [1]. Annotations are intended to extend existing syntactic standards (as, for instance, CORBA IDL or WSDL) by semantics, supported with knowledge-based techniques for description of and reasoning about their properties. This includes a particular focus for dynamic annotations, that is, for information that depends on time or environment parameters and thus may change probably even during execution time.

2 Semantic Modelling Approach

Agent-based software development comprises basically (a) the specification of the requirements for the functionality that has to be achieved (definition of the composition goal); (b) the collection and interpretation of the information available about existing agents (agent discovery); (c) the adaptation and execution of selected services from particular agents (the proper system integration). In order to enable the automation of the composition task, we aim at the development of methods and techniques for an intelligent machine-based agent discovery with support for integration and interoperation adaptation.

2.1 Software Agent Annotation

Our modelling paradigm is concerned with both the meta-modelling and the modelling level. Meta-modelling, which is the essential semantic modelling process, includes the definition of the “universe” for the semantic models. This requires presenting the necessary description means for software agents and their services in the form of templates, thereby identifying and collecting what kind of information is needed for the description of agents and their services. It also comprises the definition of the ontologies tailored to the semantic models.

The modelling level is concerned with annotating concrete software agents, and based upon templates defined by knowledge engineers to ensure a correct and adequate semantic description. The composition task – in order to achieve a more complex behaviour of agents – considers these templates and the corresponding ontologies and inference mechanisms. For the incorporation of agents from different application domains, a semantic mapping among the templates has to be achieved. Thereby, commonly available ontologies as well as publicly accessible domain-specific ontologies are to be used. The mapping mechanism of course also recurs to the semantics of the involved agents and the logics, used for their description.

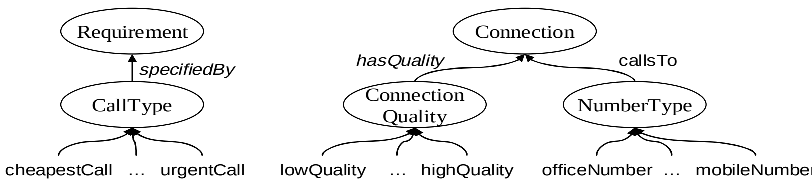


Fig. 1. Example: Requirement and Connection Ontologies for a Telecommunication Service

The semantic modelling process strives for static and dynamic annotations. Static annotations present concrete information about a given agent and its service. This information is readily usable, in particular, for comparison of requirements with the agent's properties. Dynamic annotations define a set of rules to derive dynamically changing information whenever this information is needed (assume, for instance, that the usage cost of a service is based on a currency other than the requestor's currency, then, it might be necessary to calculate the actual costs, using the exchange rate valid at the time of the service request). The use of dynamic annotations is discussed in section 2.3 where the integration of static and dynamic annotations together with their associated ontologies into the semantic modelling is described.

Fig. 1 shows an example that provides ontologies for main term definitions and their possible values from a telecommunication domain. Accordingly to that description, requirements for a call are specified by the calling type, whereas a connection is defined by quality and the type of the called number (that is, refers to a mobile or office number, etc.).

2.2 The Concept of Logic-on-Demand

As argued in the previous chapter, syntax and semantics of software agents have to be defined precisely by using the appropriate logic formalisms for knowledge representation. A variety of logic families has emerged; each of these languages is designed to solve particular aspects of ontology modelling. Very often, logic-based approaches focus primarily on the expressivity of the model, thereby neglecting the impact that expressivity has on the performance of reasoning algorithms and the facility to integrate logic formalisms into industrial applications. Due to these problems, there has up to now not yet been a large number of successful applications of ontology-based techniques in industry.

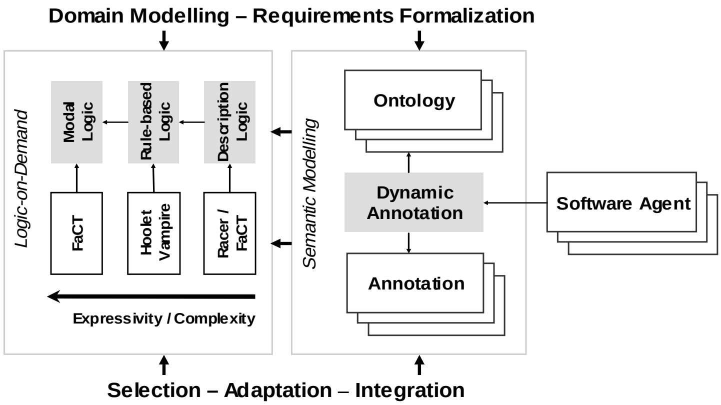


Fig. 2. Logic-on-Demand and Dynamic Annotation among Semantic Modelling

The concept of Logic-on-Demand (LoD) is supposed to overcome the problems by accommodating the expressivity of the proposed ontology languages to the varying needs and requirements of the description, in particular with respect to decidability. The main purpose of the LoD concept is to provide the agent annotation process with an adequate and adaptive way, based on uniform principles, for describing all the notions, relations and rules, the behaviour and whatever proves necessary. To achieve this, LoD means to define a basic logical formalism that is adequate and tailored to the application domain and to incorporate additional logic formalisms and description techniques with further expressivity as optional features that can be used whenever needed. These additional formalisms share notions and terms with the basic formalism which will be grounded syntactically in OWL [3] and semantically in description logic [4].

DL provides us with the basic features for the modelling of software agents. It is sufficient to define a terminology, hierarchical structures of terms, and definitions of concepts and their properties through terms. It plays an essential role in our modelling framework. As an example for the basic formalism, see Formula 1, which provides a formal definition of the ontologies given in Fig. 1 (examples for extensions are given in Formulas 2 and 3):

$$\begin{gathered} \mathit{Requirement} \subseteq \exists\, \text{specifiedBy}(\mathit{CallType})\,. \\ \mathit{Connection} \subseteq \exists\, \text{hasQuality}(\mathit{ConnectionQuality})\ \textbf{AND}\ \exists\, \text{callsTo}(\mathit{NumberType})\,. \end{gathered} \tag{1}$$

However, the expressivity of DL is, for instance, not sufficient to describe implication rules, modalities and probabilities which are needed for a proper reasoning in software agent composition and therefore, we propose here to extend DL with these description means. We thereby follow an approach of using the Semantic Web Rule Language (SWRL) [8], which extends the set of OWL axioms by the possibility to express rules. Formula 2 shows an example which defines the procedure of comparing (matching) requirements with an offered service. It expresses that “urgent calls” should be done on a mobile phone with high connection quality:

Requirement(?requirement) **AND** specifiedBy(?requirement, *urgentCall*)
AND *Connection*(?call) **AND** callsTo(?call, *mobileNumber*)
AND hasQuality(?call, *highQuality*) → matches(?call, ?requirement) . (2)

Traditional techniques and methods for software component description do not provide sufficient expressivity to describe non-functional properties and requirements, which usually are based on real-time and probability constraints. To express this, and to integrate it into the reasoning process, we have to resort to modal logic. Formula 3 formulates the constraint that if a service is provided by a not trusted service provider (identified by the variable *?webService*) then this service will be not trusted as well.

ServiceProvider(?webService) **AND** isTrusted(?webService, **NO**)
AND isProvidedBy(?service, ?webService) → $\Diamond$ hasTrust(?service, **NO**) . (3)

A semantic modelling approach does not aim at introducing completely new semantic models but propose to support current techniques. With support we thereby envisage such interface and component property description techniques that help to automatically generate new services or software applications out of a given set of available components, agents and services through their interacting and interoperating. Fig. 2 shows how Logic-on-Demand with its hierarchy of logics and related inference engines is based on the structured annotation process for semantic modelling.

2.3 Coping with Dynamics in Semantic Modelling

At the time an ontology is modelled, no one can anticipate future service functionality or behaviour. Similarly, it is impossible to predict and formulate in an annotation all concrete values that may occur in future scenarios. Therefore, we propose a new modelling element that is designed for flexibility and extensibility, and call it *Dynamic Annotation*. With the help of dynamic annotations, a software agent can manipulate and reason about a particular future state of the environment in order to determine what to do next. Thus, considering the general modelling paradigm, we assume that a complete semantic model comprises three different levels of abstraction about software agents: a set of ontologies, a dynamic annotation, and a set of (static) annotations (see Fig. 2).

Ontologies as a means for meta modelling are independent from actual circumstances based upon changing state of the environment or the actual time. However, such dependencies from actual, dynamically changing circumstances do have an important influence in the compositional approach. Hence, rules determining how to cope with this dynamicity have to be provided if one has to include it into the reasoning, they are specified in the dynamic annotation. Thus, dynamic annotations play the role of mediators between the ontology and the static semantic annotations that describes the agent properties and its qualities, and in particular its requirements with respect to composition. As an example, consider delay time characteristics of a service which depend on the platform it will be executed on: delay time relevant platform dependency for such a particular service has to be specified on the level of dynamic annotations. Concrete values can then be derived when the circumstances of a concrete execution become known. As another example, Formula 2 and 3 show how a rule-based formalism and modal logic can be used to provide the matching procedure and trust verification in circumstances where a reaction on an evolving situation becomes necessary (that is, how to react on an urgent call or whether to trust a particular service).

Dynamic annotations, when being confronted with a concrete situation or being addressed by some set of requirements, will be resolved to annotations – static description of the properties and qualities of the particular service.

3 Conclusion

The development of multi-agent systems becomes more and more complex while the number of software agents increases exponentially. The detection of potentially collaborative agents is less feasible. Thus, the central objective to support the reuse of software agents becomes overwhelmingly important. As it was shown, the use of semantic modelling techniques can facilitate most scenarios of agent composition.

Our approach defines an annotation process and its semantic extensions through knowledge-based techniques extended by necessary representation constructs. The annotation process introduces flexibility with respect to the description mechanisms what allows for a trade-off between expressivity and complexity and the selection of the appropriate reasoning tools. The concept of dynamic annotation allows for dealing with a changeable environment and composition conditions by introducing dynamic behaviour. Dynamic notions produce relevant values due to time-dependant requirements or particular scenario. We assume that dynamic annotations are an additional abstraction modelling layer to the basic modelling paradigm.

References

1. Graubmann P., Pfeuffer E., Roshchin M.: Web Services Annotation and Reasoning. Position paper, W3C Workshop on Frameworks for Semantics in Web Services, Innsbruck (2005)
2. Pahl C.: Ontologies for Semantic Web Components. ERCIM News N 51 (2002)
3. OWL, http://www.w3.org/TR/owl-features/
4. Description Logic, http://dl.kr.org/
5. OWL-S, http://www.daml.org/services/owl-s/1.0/
6. Semantic Web Services Initiative (SWSI), http://www.swsi.org/
7. F-Logic, http://www.ontoprise.de/content/e5/e190/e191/tutorial_flogic_ger.pdf
8. SWRL, http://www.daml.org/2003/11/swrl/
9. Graubmann P., Roshchin M.: Semantic Annotation of Software Components. IEEE Computer Society. Proceedings of SEAA06, Dubrovnik, Croatia (2006)
10. WSMO, http://www.w3.org/Submission/WSMO/

Combination Methodologies of Multi-agent Hyper Surface Classifiers: Design and Implementation Issues

Qing He [1,2] , Xiu-Rong Zhao [1,2] , Ping Luo [1,2] , and Zhong-Zhi Shi [1,2]

[1] The Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
[2] Graduate University of Chinese Academy of Sciences,
Beijing 100039, China
{heq,zhaoxr,luop,shizz}@ics.ict.ac.cn

Abstract. This paper describes a new framework using intelligent agents for pattern recognition. Based on Jordan Curve Theorem, a universal classification method called Hyper Surface Classifier (HSC) has been studied since 2002. We propose multi-agents based technology to realize the combination of Hyper Surface Classifiers. Agents can imitate human beings' group decision to solve problems. We use two types of agents: the classifier training agent and the classifier combining agent. Each classifier training agent is responsible to read a vertical slice of the samples and train the local classifier, while the classifier combining agent is designed to combine the classification results of all the classifier training agents. The key of our method is that the sub-datasets for the classifier training agents are obtained by dividing the features rather than by dividing the sample set in distribution environment. Experimental results show that this method has a preferable performance on high dimensional datasets.

1 Introduction

The combination of multiple classifiers can be considered as a generic pattern recognition problem in which the input consists of the results of the individual classifiers, and the output is the combined decision. For this purpose, many developed classification techniques can be applied; in fact, classification techniques such as neural networks and polynomial classifiers have served to combine the results of multiple classifiers. In this area, the early work can be found in [24, 25]. This approach almost immediately produced promising results. In this domain technological developments has increased and grown tremendously [7]-[19], partly as a result of the coincident advances in the technology itself. These include the production of very fast and low cost computers that have made many complex algorithms practicable, among which are many pattern recognition algorithms. In paper [20], L. Xu, A. Krzyzak, and C. Y. Suen propose to conduct a more systematical investigation into the problem of multi-classifier combination. Their idea consists of two parts. The first part, being closely dependent on the specific applications, includes the problems of "How many classifiers are chosen for a specific application problem? What kind of classifiers should be used? And for each classifier what types of features should be chosen?" as well as other problems that

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 100–113, 2007.
© Springer-Verlag Berlin Heidelberg 2007

relate to the construction of those individual and complementary classifiers. In paper [15], J. Kittler presents a theoretical framework for combining soft decision outputs of multiple experts employing mixed (some shared and some distinct) representations of patterns to be classified. In paper [22], Ching Y. Suen and Louisa Lam examine the main combination methods that have been developed for different levels of classifier outputs - abstract level, ranked list of classes, and measurements. In paper [23], Shaodan Lin etc. suggests a technology of information fusion using multiple agents, each of which uses a quite different classification algorithm such as decision tree algorithm, simple Naïve Bayes algorithm and the newly emerging classification algorithm based on atomic association rules. In paper [21] Ahmed Al-Ani and Mohamed Deriche present a classifier combination technique based on the Dempster-Shafer theory of evidence, which is a powerful method for combining measures of evidence from different classifiers.

In this paper, we propose multi-agents based technology to realize the combination of Hyper Surface Classifiers (HSC). Agents can imitate human beings' group decision to solve problems in the same way as group consultations of doctors. In [1] [2] a new classification method based on hyper surface (HSC) is put forward. In this method, a model of hyper surface is obtained during the training process and then it is directly used to classify large database according to whether the wind number is odd or even based on Jordan Curve Theorem. Experiments show that HSC can efficiently and accurately classify large-sized data sets in two-dimensional space and three-dimensional space. Though HSC can classify higher dimensional data according to Jordan Curve Theorem in theory, it is not as easy to realize HSC in higher dimensional space as in three-dimensional space. However, what we really need is an algorithm that can deal with data not only of massive size but also of high dimensionality. Thus in [3] a simple and effective kind of dimension reduction method without losing any essential information is proposed, which is a dimension changing method in nature. In paper [29], based on the idea of ensemble, another solution to the problem of HSC on high dimensional data sets is proposed and proven to have a preferable performance by experiments. However multi agent technique has not been utilized in paper [29]. Recently, Distributed Data Mining (DDM) [30][31] has attracted lots of attention among the data mining community. DDM refers to the mining of inherently distributed datasets, aiming to generate global patterns from the union set of locally distributed data. However, the security issue among different local datasets and the huge communication cost in data migration prevent moving all the datasets to a public site. Thus, the algorithms of DDM often adopt a computing paradigm of local processing and global synthesizing, which means that the mining process takes place at a local level and then at a global level where local data mining results are combined to gain global findings. Considering the problem of distributed computing environment and that agents can imitate human beings' group decision to solve problems just as group consultations of doctors, in this paper we extend on our work in [29]. The multi-agent techniques are introduced to realize the combination of Hyper Surface Classifiers. A combination of classifiers is a set of classifiers, whose individual classification decisions are combined in some way, typically by a weighted or equal voting, to classify new examples. Generally speaking, there're two ways of combination, horizontal combination and vertical combination. Here, vertical combination is adopted. The combination of HSC is constructed by dimension dividing rather than dimension reduction for high dimensional data. We use two types of agents: the classifier training agent and the classifier combining agent. Each classifier training agent is responsible to read a vertical slice of the samples and train the local classifier, while the classifier combining agent is designed to combine the classification results of all the classifier training agents.

The rest of this paper is organized as follows: In section 2, we give an outline of hyper surface classifier. Then in Section 3, after discussing the problems with HSC on high dimensional data sets, we give the combination technique used in HSC. Our experimental results are presented in Section 4, followed by our conclusions in Section 5.

2 Overview of the Classification Method Based on Hyper Surface

HSC is a universal classification method based on Jordan Curve Theorem in topology. The main differences between the well-known SVM algorithm and HSC are that HSC can directly solve the nonlinear classification problem in the original space without having to map the data to a higher dimensional space, and thus without the use of kernel function.

Jordan Curve Theorem. Let X be a closed set in n-dimensional space R^n. If X is homeomorphic to a sphere in $n-1$ dimensional space, then its complement $R^n \setminus X$ has two components, both connected, and one of them is called inside, the other called outside.

Classification Theorem. For any given point $x \in R^n \setminus X$, x is in the inside of X $\Leftrightarrow$ the wind number i.e. intersecting number between any radial from x and X is odd, and x is in the outside of X $\Leftrightarrow$ the intersecting number between any radial from x and X is even.

How to construct the separating hyper surface is an important problem. An approach has been given in [1]. Based on Jordan Curve Theorem, we have put forward the following classification method HSC [1].

Step1. Let the given samples distribute in the inside of a rectangle region.

Step2. Transform the region into a unit region.

Step3. Divide equally the region into some smaller units. If some units contain samples from two or more classes then divide them into a series of smaller units repeatedly until each unit covers at most samples from the same class.

Step4. Label each region according to the inside sample's class. Then the frontier vectors and the class vector form a string for each region.

Step5. Combine the adjacent regions of the same class and obtain a separating hyper surface then save it as a string.

Step6. Input a new sample and calculate the intersecting number of the sample about separating hyper surface. Drawing a radial from the sample can do this. Then the class of the sample is decided according to whether the intersecting number between the radial and the separating hyper surface is even or odd.

HSC has the following properties.

1) *High Efficient and Accuracy*

The classification algorithm based on hyper surface is a polynomial algorithm if the same class samples are distributed in finite connected components. Experiments show that HSC can efficiently and accurately classify density large dataset in two-dimensional or three-dimensional space for multi-classification. For large three-dimensional data up to 10^7, the speed of HSC is still very fast [2]. The reason is that the time of saving and extracting hyper surface is very short and the need for storage is very little. Another reason is that the decision process is very easy by using Jordan Curve Theorem.

2) *Ability of Generalization*

The experiment of training on small scale samples and testing on density large scale shows that HSC has strong ability of generalization[2]. According to statistic learning theory[4][5], the higher VC dimension is, the larger confidence domain is. So the difference between real risk and experimental risk possibly increase. This is the reason of excessive learning problem. So machine learning process is not only minimizing the experimental risk, but also reducing the dimension of VC. But the strategy is not useful in HSC. Because the hyper surface made by linear segmentation function. The function set has infinite VC dimension because the set can separate any more h samples that distribute in anyway. This shows that the conclusion about the bound of generalization given by Vapnik is loose when the VC dimension is too big.

3) *Robustness*

Though the data noise can not be completely clear, it can be controlled in a local region. If a noised sample locates in the inside of the hyper surface, the hyper surface will change into complex hyper surface. In this case the classification theorem is still efficient, the noise may make mistake in classification, but the influence has been controlled in a local small unit.

4) *Independ on The Distribusion Feature of Samples*

In fact, HSC can solve the nonlinear classification problem which the samples distribute in any shape in a finite region. The method has nothing to do with the shape of the data, even though the shape is interlock or crisscross. The common condition as other classification methods is that the samples must reflect the feature of data distribution.

3 Hyper Surface Classifiers Combination

3.1 Problems with HSC on High Dimensional Data Sets

According to Jordan Curve Theorem, HSC can deal with any data sets regardless of their dimensionality on the theoretical plane. But in practice, there exist some problems in both time and space in doing this directly. It is not as easy to realize HSC in higher dimensional space as in three-dimensional space. However, what we really need is an algorithm that can deal with data not only of massive size but also of high dimensionality. Fortunately, there exist many methods of dimensionality reduction for us to use. And another important and effective kind of dimensionality reduction method without losing any essential information is proposed in [3]. This method rearranges all of the numerals in the higher dimensional data to lower dimensional data without changing the value of all numerals, but only change their position according to some orders, and thus very suitable for HSC. In paper [29], based on the idea of ensemble, another solution to the problem of HSC on high dimensional data sets is proposed and proven to have a preferable performance by experiments. However multi agent technique has not been utilized in paper [29].

Another problem of high dimensional data is that the different dimension data for the same object usually spread in a distribution environment. DDM refers to the mining of inherently distributed datasets, aiming to generate global patterns from the union set of locally distributed data. However, the security issue among different local datasets and the huge communication cost in data migration prevent moving all the datasets to a public site. Thus, the algorithms of DDM often adopt a computing paradigm of local processing and global synthesizing, which means that the mining process takes place at a local level and then at a global level where local data mining results are combined to gain global findings.

In our work, we solve the problem based on the idea of combination and perfect results are gained. This method will be described in further detail in the next part.

3.2 The Combination of HSC

We propose multi-agents based technology to realize the combination of Hyper Surface Classifiers. Agents can imitate human beings' group decision to solve problems in the same way as group consultations of doctors. According to this, every classifier is designed to be an agent with given condition attributes. We use two types of agents: classifier training agent and classifier combining agent. Classifier training agent learns a classification model using a specified algorithm on a dataset and predicts an unlabelled instance using the learned model. Classifier combining agent realizes the information fusion of multiple classifier training agents' prediction results and gives the final class label of the unlabelled instance. When agents collectively do a classifying job, first of all, every classifier training agent must complete its learning on its own and produce an independent classifying model with given different condition attributes. Then when a classifying prediction task is performed, the classifier combining agent sends the instances to be forecast to every classifier training agent. Classifier training agents use classifying model to forecast the label of the instance, meanwhile, it evaluates the instance's properties and sends the overall evaluation results with the class label to the classifier combining agent. The classifier combining agent decides the final class label based on the results of all classifier training agents in terms of control logic such as voting or weighted voting. Thus, classifier combining agent probably surpasses the capability limits of single classifiers

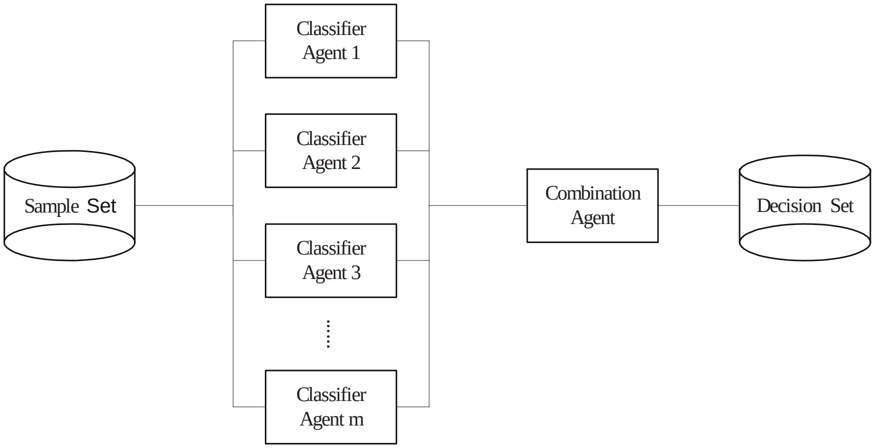


Fig. 1. The Combination of multiple classifier agents

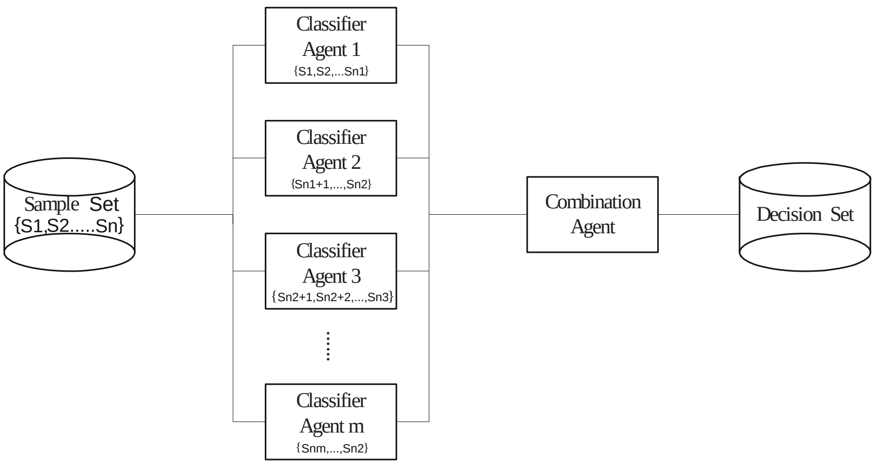


Fig. 2. The horizontal combination of multiple classifier agents

(see Fig1). Hence, the agent-based system of heterogeneous HSC classifiers can achieve a high classifying accuracy without additional fusion training of classifier group. Generally speaking, there're two ways of combination, horizontal combination (see Fig2) and vertical combination (see Fig3). The most important difference sub-datasets are obtained by dividing the features rather than by dividing the sample set, so in the case of no inconsistency, the size of each sub-dataset is equal to the original sample set, with totally occupying a little more storage space than the original sample set. Here, vertical combination is adopted.

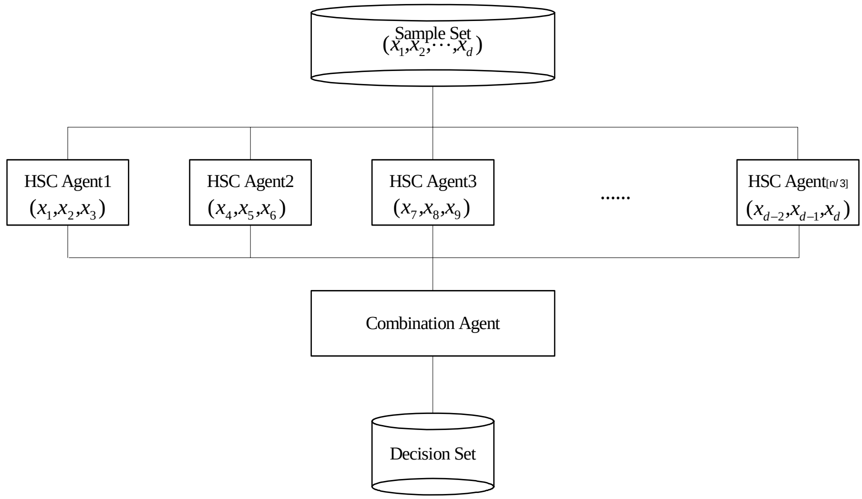


Fig. 3. The vertical combination of HSC agents

Aiming at the memory shortage problem of HSC on high dimensional data sets, we provide a solution based on the idea of combination. By attaching the same importance to each feature, firstly we group the multiple features of the data according to some rules to form some sub-datasets, then start a training process and generate a classifier for each sub-dataset, and the final determination is reached by integrating the series of classification results in some way. The following is its detailed steps.

The Agent Training Process

Step1. Get the dimension of conditional attributes d from the data set of training samples.

Step2. Divide the features into $\lceil d/3 \rceil$ subsets, where $\lceil d/3 \rceil$ is the smallest integer number that is greater than $\lceil d/3 \rceil$. The i-th subset covers the features $3i-2$, $3i-1$, $3i$, $i = 1,2,3,\cdots,\lceil d/3 \rceil$, and the decision attribute. If d cannot be divided exactly by 3, then one or two features in the $d-1$-th subset are added to the last subset. After this, the data set of training samples has been divided into $\lceil d/3 \rceil$ sub-datasets, each of which has the same size, with three conditional attributes and one decision attribute.

Step3. For each sub-dataset, eliminate the inconsistency that may have been caused in Step2. For each sample in this sub-dataset, if there exist some other samples in which the values of conditional attributes are all the same with those of this sample but only the value of decision attribute is different, then we say inconsistency occurs. In that case, we just have to delete these samples from this sub-dataset.

Step4. For each sub-dataset after Step3, start a HSC training process independently and save the training result as the model. And till now, we have got $\lceil d/3 \rceil$ HSC classifiers, which we define as classifiers combination.

The Agent Classification Process

Step1. Get the dimension of conditional attributes d from the data set of testing samples.

Step2. Divide the features into $\lceil d/3 \rceil$ subsets, and the i-th subset covers the features $3i-2$, $3i-1$, $3i$, $i=1,2,3,\cdots,\lceil d/3 \rceil$. If d cannot be divided exactly by 3, then one or two features in the $d-1$-th subset are added to the last subset. After this, the data set of testing samples has been divided into $\lceil d/3 \rceil$ sub-datasets, each of which has three conditional attributes.

Step3. For each sub-dataset after Step3, start the corresponding HSC classifier and save the result of classification. So now we have got $\lceil d/3 \rceil$ classification results for each testing sample in the original data set.

Step4. The final decision for each testing sample in the original data set is reached by voting. Here we adopt the *plurality* voting scheme in which the collective decision is the classification result reached by more classifiers than any other. Notice that we attach the same importance to all the classifiers, so in the case of the same number of votes for two or more classification results, we can randomly choose one of them.

HSC Combination classifies high dimensional data sets by trying to analyze multiple slices of the training and testing samples. Furthermore, the combination can be far less fallible than any single HSC classifier.

3.3 Voting Methods

Among the combination methods for HSC Agents, majority vote is the simplest to implement, since it requires no prior training, and it has been used as early as 1974 [26]. The use of this method is especially appropriate in situations where other quantifiable forms of output cannot be easily obtained from individual classifiers agent, or where the use of other accurate combination methods may be too complex. Obvious examples of the former are some structural classifiers. This combination method has also been found to be highly effective. From this process of simple majority vote in which the decision of each classifier carries equal weight, various refinements can be made. This can be done by assigning different weights to each classifier agent to optimize the performance of the combined classifiers agent on the training set. For the first refinement, weights can be generated by a genetic algorithm and assigned to the vote of each classifier to determine the optimal values for an objective function. This function can incorporate conditions on the recognition and error rates. We adopted plurality voting method. The combined classifier agent decides for the testing sample belonging to class C_j if the number of classifiers that support it is considerably bigger than the number of classifiers that support any other class.

3.4 Development Toolkit and System Implementation

MAS environments for Combination HyperSurface Classifiers System (CHCS) is very important. The reason for adopting MAS technology is that it is in with the following characteristics and requirements. MAS is a natural distributed computing environment, allowing agents running on different computing hosts simultaneously to achieve high availability. Moreover, MAS middleware helps to form a unified computing interface and makes the distributed applications more easily built. MAS is an open system In an open system, the structure of the system itself is capable of dynamically changing. Its components may not be known in advance, could change over time, and may be highly heterogeneous in that they are implemented by different people, at different times, using different software tools and techniques. In training and testing complex process, the output of an early job is the input of the following jobs. Thus, distributed execution of this process involves complex process control and management. To address this problem, a number of agents, specializing in solving constituent sub-problems in autonomous ways, need to be developed. They are then coordinated through standardized Agent Communication Language, which is well supported in a MAS environment by some communication protocols.

We adopted MAGE [27] as toolkit and MAS environment. MAGE is a middleware that facilitates the development of MA systems. Each instance of the MAGE runtime environment for each computing host is called a container, as it can contain several agents. Each container on that host presents the corresponding computing resource. Several containers can connect with each other through networks to form a natural distributed computing environment. From the view of programmers, it seems that all the containers are running on the same computer. We encapsulate each algorithm into the behavior of different agents. Such an agent can be regarded as an algorithm entity that can run on any host. To initiate an agent on a host means to perform the behavior of that agent, and thus utilize the corresponding computing resource. In paper [28], we implement this system in an established multi-agent system environment, in which the reuse of existing data mining algorithms is achieved by encapsulating them into agents.

3.5 Architecture of the CHSC System

We use two types of agents: the classifier training agent and the classifier combining agent. Each classifier training agent is responsible to train the local classifier, while the classifier combining agent is designed to combine the classification results of all the classifier training agents. The whole system is organized into two layers as shown in Figure 3: the HSC Agent function layer and the Combination Agent management and control layer. HSC Agent function layer provides software resources of the HSC algorithm in the form of classifier training agents. When a host does not have the software copy of a classifier training agent, it sends a request to the agent provider of software resources, and then the provider answers it with the mobile agent equipped with the required software copy. Thus it increases the computing capability of the host. Classifier training agents perform various operations in all phases of the data mining process, including sampling, preprocessing, training, classifier combining agent perform model combination and evaluation.

4 Experiments

As we can predict, the main problem with HSC Combination is inconsistency in sub-datasets. In that case, we have to delete some samples to keep consistent. As a result, the actual number used for training or testing decreases, it will affect our final results. Take the data set of Pima-diabetes from the UCI machine learning repository for example, the Table1 below illustrates this point.

Table 1. Classification results of HSC Combination on the data set of Pima

Rate / Agent	Actual No. of Samples for Training	Recall	Accuracy
HSC Agent 1	566	100%	90.00%
HSC Agent 2	542	100%	85.05%
HSC Agent 3	542	100%	94.85%
HSC Agents Combination	568	88.73%	87.50%

Note: Pima has 8 dimensions and 768 samples, 568 of which are used for training and the rest for testing.

Notice that even though the recall rate and accuracy of each sub-classifier are just high, the overall rates are much less than those of other data sets. This is mainly because that in this data set there exists more inconsistency after dividing the data set into sub-datasets, and as a result the actual number of samples participated in training or testing process decreases. But when computing the recall rate and accuracy, we need to divide the results by the number of overall samples, which are bigger than the number of actual samples in sub-dataset.

The accuracy is also related with another factor. During the voting process, it is quite posssible that two or more classification results have the same number of votes. In that case, we can randomly choose one of them as the final result, but which one we choose actually affects the accuracy very much in practice.

However, HSC Combination performs perfectly well on data sets that have little inconsistency after dividing them into sub-datasets. Now we show some results on several high dimensional data sets taken from the UCI machine learning repository in the following tables.

Table 2. Classification results of HSC Combination on high dimensional data sets

Data Sets	Dimensions	Number of Training Samples	Number of Testing Samples	Number of Sub-Classifiers	Recall	Accuracy
Iris	4	100	50	2	100%	98.00%
Wine	13	128	50	5	100%	100%
Wdbc	30	369	200	10	100%	100%
Sonar	60	158	50	20	100%	100%

From this table, we can see that HSC Combination works fairly well on these high dimensional data sets, and the recall rate and accuracy are very good. Specially, for the data set of iris, there are only two classifiers, the accuracy is lower than other high dimensional data sets. And it is obvious that the higher the dimension is, the higher accuracy is. Now,we give the classification results on each data set in details.

Table 3. Classification results of HSC Combination on the data set of Iris

	Recall	Accuracy
HSC Agent 1	100%	98.00%
HSC Agent 2	100%	92.00%
HSC Agents Combination	100%	98.00%

Table 4. Classification results of HSC Combination on the data set of Wine

	Recall	Accuracy
HSC Agent 1	100%	88.00%
HSC Agent 2	100%	84.00%
HSC Agent 3	100%	96.00%
HSC Agent 4	100%	94.00%
HSC Agent 5	100%	94.00%
HSC Agents Combination	100%	100.00%

Table 5. Classification results of HSC Combination on the data set of Wdbc

	Recall	Accuracy
HSC Agent 1	100%	99.00%
HSC Agent 2	100%	98.00%
HSC Agent 3	100%	97.00%
HSC Agent 4	100%	95.00%
HSC Agent 5	100%	94.00%
HSC Agent 6	100%	86.50%
HSC Agent 7	100%	97.50%
HSC Agent 8	100%	99.50%
HSC Agent 9	100%	89.00%
HSC Agent 10	100%	96.50%
HSC Agents Combination	100%	100%

For the data set of Wdbc, ten HSC Agents have been obtained, two of which are presented in the following Fig.4 and Fig. 5.

From experimental results and analysis above, we can see that HSC Combination performs very well on high dimensional data sets and is perfectly suitable for the data sets in which samples are different in each slice.

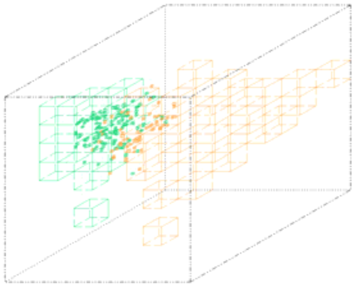


Fig. 4. HSC Agent 1 Model of the data set of Wdbc

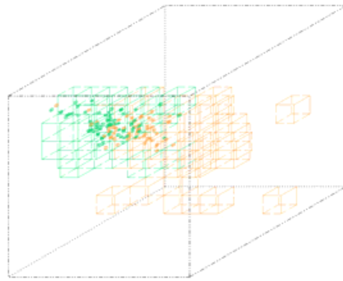


Fig. 5. HSC Agent 10 Model of the data set of Wdbc

5 Conclusions

For high dimensional data that the different dimension data for the same object usually spread in a distribution environment, a new framework using intelligent agents for pattern recognition is proposed. Considering that agents can imitate human beings' group decision to solve problems in the same way as group consultations of doctors, we propose multi-agents based technology to realize the combination of HSC Agents for high dimensional data sets. Two types of agents are introduced: classifier training agent and classifier combining HSC Agent. Each classifier training agent is designed to be an agent with given conditional attributes instead of all conditional attributes, so it is responsible to read a vertical slice of the samples and train the local classifier. The classifier combining agent is designed to combine the classification results of all the classifier training agents by way of voting. The most important difference between HSC combination and the traditional combination is that the sub-datasets are obtained by dividing the features rather than by dividing the sample set, so in the case of no inconsistency, the size of each sub-dataset is equal to the original sample set, with totally occupying a little more storage space than the original sample set. Experiments show that this method has a preferable performance on high dimensional data sets, especially on those in which samples are different in each slice. The time complexity of HSC Combination is $o((nd+n^2d)/3)$, where n denotes the size of the sample set; d denotes dimensional number.

Acknowledgements

This work is supported by the National Science Foundation of China (No. 60435010, 90604017, 60675010), the 863 Project (No.2006AA01Z128), National Basic Research Priorities Programme (No. 2003CB317004) and the Nature Science Foundation of Beijing (No. 4052025).

References

1. He, Q., Shi, Z. Z., Ren, L. A.: The Classification Method Based on Hyper Surface. In: Proc. 2002 IEEE International Joint Conference on Neural Networks, Las Vegas (2002) 1499–1503
2. He, Q., Shi, Z. Z., Ren, L. A., Lee, E. S.: A Novel Classification Method Based on Hyper Surface. Int. J. of Mathematical and Computer Modeling 38 (2003) 395–407
3. He, Q., Zhao, X. R. and Shi, Z. Z.: Classification based on dimension transposition for high dimension data. Soft Computing, 11, 4 (2006) 329–334
4. Vapnik, V. N., Golowich, S., Smola, A.: Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing. In: M. Mozer, M. Jordan, and T. Petsche (eds.): Neural Information Processing Systems, Vol. 9. MIT Press, Cambridge, MA (1997)
5. Vapnik V. N.: The Nature of Statistical Learning Theory. Springer-Verlag, New York, (1995)
6. Fulton, W.: Algebraic topology. Springer-Verlag, New York (1995)
7. Alkoot, F. M., Kittler, J.: Experimental Evaluation of Expert Fusion Strategies. Pattern Recognition Letters 20 (1999) 1352–1369
8. Cordella, L. P., Foggia, P., Sansone, C., Tortorella, F. and Veneto M.: Reliability Parameters to Improve Combination Strategies in Multi-Expert Systems. Pattern Analysis and Applications 2 (1999) 205–214
9. Ho, T. K., Hull, J. J. and Srihari, S. N.: Decision Combination in Multiple Classifier Systems. IEEE Trans. PAMI 16 (1994) 66–75
10. Huang, Y. S. and Suen, C. Y.: A Method of Combination of Multiple Experts for the Recognition of Unconstrained Handwritten Numerals. IEEE Trans. PAMI 17 (1995) 90-94
11. Ji, C. and Ma, S.: Combinations of Weak Classifiers. IEEE Trans. Neural Networks 8 (1997) 32–42
12. Kang, H. J., Kim, K. and Kim, J. H.: Optimal Approximation of Discrete Probability Distribution with kth-order Dependency and Its Application to Combining Multiple Classifiers. Pattern Recognition Letters 18 (1997) 515–523
13. Kang, H.J. and Lee, S.W.: Combining Classifiers Based on Minimization of a Bayes Error Rate. In: Proceedings of the 5th International Conference on Document Analysis and Recognition, Fort Collins, Colorado, USA (1999) 124–129
14. Kittler, J.: Combining Classifiers: A Theoretical Framework. Pattern Analysis and Applications 1 (1998) 18–27
15. Kittler, J.: On Combining Classifiers. IEEE Trans. PAMI 20 (1998) 226–239
16. Lam, L. and Suen, C. Y.: Application of Majority Voting to Pattern Recognition: An Analysis of Its Behavior and Performance. IEEE Trans. Systems, Man, and Cybernetics 27 (1997) 553–568
17. Rahman, A.F.R. and Fairhurst, M.C.: An Evaluation of Multi-expert Configurations for the Recognition of Handwritten Numerals. Pattern Recognition 31 (1998) 1255–1273
18. Rahman, F.R. and Fairhurst, M. C.: Serial Combination of Multiple Experts: A Unified Evaluation. Pattern Analysis and Applications 2 (1999) 292–311
19. Wang, D., Keller, J. M., Carson, C.A., McAdoo-Edwards, K.K., and Bailey C. W.: Use of Fuzzy-Logic-Inspired Features to Improve Bacterial Recognition through Classifiers Fusion. IEEE Trans. SMC 28B (1998) 583–591
20. Xu, L., Krzyzak, A., and Suen, C. Y.: Methods of Combining Multiple Classifiers and Their Application to Handwritten Numeral Recognition. IEEE Trans. Systems, Man, and Cybernetics 22 (1992) 418–435

21. Al-Ani, A, Deriche, M: A New Technique for Combining Multiple Classifiers Using the Dempster-Shafer Theory of Evidence. Journal of Artificial Intelligence Research 17 (2002) 333–361
22. Suen, C. Y. and Lam, L: Multiple Classifier Combination Methodologies for Different Output Levels. In J. Kittler and F. Roli (Eds.): LNCS Vol.1857 (2000) 52–66
23. Lin, S. D., Han G. Q., Yuan, X.: Information Fusion of Agent Based Heterogeneous Multi-Classifiers. In Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou (2005) 18–21
24. Gader, P. D., Hepp, D., Forrester, B., and Peurach, T.: Pipelined Systems for Recognition of Written Digits in USPS ZIP Codes. In Proc. U. S. Postal Service Advanced Technology Conf. (1990) 539–548
25. Mandler, E. and Schuermann, J.: Combining the Classification Results of Independent Classifiers Based on the Dempster/Shafer Theory of Evidence. Pattern Recognition and Artificial Intelligence (1988) 381–393
26. Spanjersberg, A. A.: Combination of Different Systems for the Recognition of Handwritten Digits. In Proc. 2nd Int. Joint Conf. on Pattern Recognition, Copenhagen (1974) 208–209
27. Shi, Z. Z., H. Zhang, Y. Cheng, Y. Jiang, Q. Sheng and Z. Zhao: MAGE: An Agent-Oriented Programming Environment. In Proceedings of the IEEE International Conference on Cognitive Informatics, New York (2004) 250–257
28. Luo, P., Lu¨, K., Huang, R., He, Q., and Shi, Z. Z.: A Heterogeneous Computing System for Data Mining workflows in Multi-Agent Environments. Expert Systems 23 (2006) 258–272
29. Zhao, X. R., He, Q. & Shi, Z. Z: HyperSurface Classifiers Ensemble for High Dimensional Data Sets. In Proc. International Symposium on Neural Networks. Lecture Notes in Computer Science, Vol. 3971. Springer-Verlag, Berlin Heidelberg New York (2006) 1299–1304
30. Fu, Y.: Distributed data mining: An overview. IEEE TCDP newsletter (2001)
31. Cannataro, M., Congiusta, A., Pugliese, A., Talia, D., Trunfio, P.: Distributed data mining on grids: Services, tools, and applications, IEEE Transactions on Systems, Man and Cybernetics 34 (6) (2004) 2451–2465

Security in a Mobile Agent Based DDM Infrastructure

Xining Li

State Key Laboratory of Novel Software Technology (Nanjing University)
Department of CIS, University of Guelph, Canada
xli@cis.uoguelph.ca

Abstract. Mobile agent technology offers a new approach to mining information from data sources distributed over the Internet. However, the potential benefits of mobile agents must be weighted against the real security threats. An agent based Distributed Data Mining (DDM) system must cope with exposed data services and insecure communication channels in the Internet to protect the privacy, integrity and availability of agents and distributed resources. In this paper, we present the design of a mobile agent infrastructure for DDM applications and discuss the implementation of security mechanism that has been effectively integrated with the mobile agent virtual machine.

Keywords: mobile agent, distributed data mining, virtual machine, security.

1 Introduction

Mobile agent systems are generalized distributed systems in the sense that they are mainly intended for network computing, applications such as data mining, knowledge discovery or E-commerce, distributed over large-scale computer networks. An agent is an autonomous process acting on behalf of a user. A mobile agent roams the Internet to access data and services, and carries out its assigned task remotely. Distributed Data Mining (DDM) is one of the promising application areas of deploying intelligent mobile agent paradigm [1][2][3]. Most existing DDM projects focus on approaches to apply various machine leaning algorithms to compute descriptive models of the physically distributed data sources. Although these approaches provide numerous algorithms, ranging from statistical model to symbolic/logic models, they typically consider homogeneous data sites and require the support of distributed databases.

As the number and size of databases and data warehouses grow at phenomenal rates, one of the main challenges in DDM is the design and implementation of mobile agent based infrastructure that scales up to large, dynamic and remote data sources. However, the potential benefits of mobile agents cannot be fully explored without proper measures against the real security threats. In general, security threads can be classified into four categories: (1) comprised hosts and mobile agents attempt representing different parties that may exhibit malicious behavior toward one another, (2) exposure of mobile agents to third party intruders through the network, (3) agents interfere with each other or gain unauthorized access to internal state, and (4) agents carrying malicious mobile code try to corrupt remote hosts or invoke unauthorized

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 114–123, 2007.
© Springer-Verlag Berlin Heidelberg 2007

access to resources. Clearly, an agent based DDM system must cope with exposed data services and insecure communication channels in the Internet to protect the privacy, integrity and availability of agents and distributed data sources [4]. For example, the security architecture in Ajanta system is built on Java security model [5]. It provides a set of mechanisms to protect servers, agents and name service respectively. In order to protect security-sensitive interface methods presented by agent servers, a challenge-response based authentication is introduced. In addition, Ajanta agents don't have direct references to resources. A proxy acts as the mediator between the agents and a corresponding resource. Based on the proxy's security policy and the client agent's credentials, access control mechanism of the proxy will decide whether accept or reject the agent requests. Another example is D'Agent [6]. Its security model is responsible for identifying agent owners, protecting agents, and protecting server resources against unauthorized access. To protect agent hosts, D'Agent system implements two security policies. One is authentication, which verifies the identity of an agent's owner. The other is authorization and enforcement, which assigns resource limits to the agent and enforces resource limits. The objective of our research is to provide basic mechanisms that are generally incorporated in mobile agent systems to support security and to equip mobile agents with system tools such that those agents can search for data sites, move from hosts to hosts, gather information and access databases, carry out complex data mining algorithms, and generate global data model or pattern through the aggregation of the local results.

The rest of the paper is structured as follows. Section 2 gives an overview of the design and implementation of the IMAGO (Intelligent Mobile Agent Gliding On-line) system, an infrastructure that couples a logic-based programming interface for DDM applications. In Section 3, we discuss the security properties related to agent migration, namely, issues concerning the identification and authentication of the moving agent and the secure migration channel between the sending and receiving servers. In Section 4, we present the access control mechanism in the IMAGO system, that is, a runtime security manager that directly control mobile agents from their dynamic behaviors during different stages of their lifecycle to protect the host server and its resources. We conclude the paper in Section 5 with an outline of future work.

2 Overview of the IMAGO DDM Infrastructure

IMAGO system is an experimental mobile agent infrastructure for DDM applications. An evaluation release of the IMAGO project is available at the IMAGO Web site (http://draco.cis.uoguelph.ca/main.html). Unlike the traditional client/server model, the IMAGO system exhibits a server/server architecture where each server is installed with the MLVM - an efficient Multi-threading Logic Virtual Machine [7]. An IMAGO server hosts mobile agents and provides a protected agent execution environment and data services. From application point of view, the IMAGO system consists of three kinds of agent servers: stationary server, database server and service discovery server. The stationary server of a DDM application is the home server where the application is invoked. In order to access information and knowledge from distributed data sources, data mining agents of an application are able to migrate to remote database servers. Like a web server, a database server must have either a well-known name or a name searchable through the service discovery server.

One important aspect of DDM is not only how to retrieve, investigate and integrate relevant information but to discover previously unknown, implicit and valuable data sites in the Internet. This in turn leads us to the service discovery problem, that is, how to find data sites available to a DDM application. A variety of Service Discovery Protocols (SDP) are currently under development by some companies and research groups [8][9][10][11]. Some of them are extended and applied by several mobile agent systems to solve the service discovery problem. In the IMAGO system, we have implemented a new data service discovery model DSSEM (Discovery Service via Search Engine Model) for mobile agents [12]. DSSEM is based on the search engine technology, a global Web search tool with centralized index and fuzzy retrieval. Database service providers manually register their services in a service discovery server. A mobile agent locates a specific service by migrating to the service discovery server and subsequently submitting requests with the required description. Before a service can be discovered, it should make itself public. This process is called service advertisement. A service advertisement should consist of the service identifier, plus a simple string describing what the service is, or a set of strings for specifications and attributes. The most significant feature of DSSEM is that we enrich the service description by using web page's URL to replace the traditional string-set service description in mobile agent systems. On the other hand, a mobile agent can move to a service discovery server, communicate with the discovery module to obtain an itinerary that includes a list of ranked host addresses of the service providers. Based on the given itinerary, the mobile agent may travel from host to host to carry out a DDM task. Fig. 1 shows an example of service discovery and data mining process.

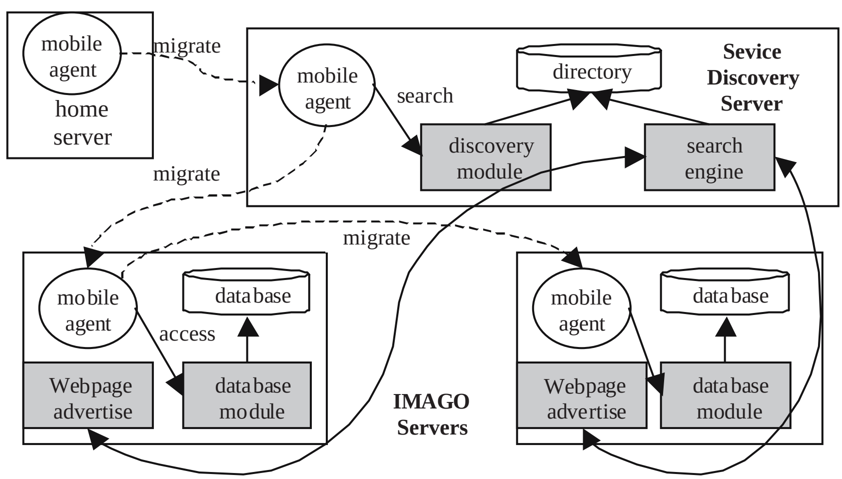


Fig. 1. An Example of IMAGO Service discovery and Data Mining Process

In order to bridge data mining agents with database systems, the IMAGO system provides a set of database predicates, which enables agents to establish connection with data sources and make requests for desired information. The IMAGO system offers two different ways of database access operations, *i.e.*, the set retrieval and the tuple retrieval. The former returns the entire matching data set to the requesting agent, whereas the latter allows the requesting agent to consume matching data on the tuple by tuple basis. The database management module not only provides flexible interface for accessing data, but also manipulates database connections efficiently. At the current stage, the database model in the IMAGO system is MySQL, the most popular open source DBMS system in the world [13].

In a DDM application, agents are not working alone and they need to communicate with each other to cooperate and generate a global data aggregation for further analysis. Most existing mobile agent systems adopt some kind of communication models/protocols from traditional distributed systems. However, we use a different strategy to cope with this issue. The IMAGO system inter-agent communication model has focused on simplicity and ease of use, and at the same time offers more expressive power. The idea is to deploy intelligent mobile messengers for inter-agent communication [14]. Messengers are thin agents dedicated to deliver messages. Like normal agents, a messenger can move, clone, and make decisions. Unlike normal agents, a messenger is anonymous and its special task is to automatically track down the receiving agent and reliably deliver its carried message in a dynamic, changing environment. The IMAGO system provides a set of built-in messengers as a part of its programming interface. A mobile agent at any remote sites and at any time may dispatch messengers to deliver data to designated receivers. For example, suppose that a mobile agent has completed its data mining task at a remote database server, it can either move back to its home server, or dispatch a messenger to deliver result to the home agent. The main advantages of system built-in messengers lie in their reliability and efficiency. Built-in messengers are carefully designed and fully tested. The varieties of built-in messengers are sufficient for most mobile agent applications. In addition, code of built-in messengers has been cached in every IMAGO server. As a consequence, migration of a built-in messenger only involves moving its data and run time stack. We call this technique as zero-code migration that greatly reduces the overhead of agent migration. The concept of messengers offers flexibility to simulate different patterns of agent communication. Users not only have choices to dispatch different messengers for various purposes, but also have the freedom to accept messengers either selectively or in some priority order. Communication among agents takes place by means of Agent Communication Language (ACL). The essence of an ACL is to make agents understanding the purpose and meaning of their messages. In order to facilitate open standards of ACL's, the IMAGO agent-based communication model is in compliance with the FIPA ACL message structure specification [15]. The advantage of the declarative ACL is that inter-agent communication is closely bound with the underlying logic programming language and fulfills the semantics defined by the content expression.

In brief, an IMAGO server is a Multi-threading Logic Virtual Machine. Each MLVM-thread is a kernel level thread (sometimes called a light weight process) responsible for certain system function, such as execution engine, memory manipulation, migration, security, and so on. An important design consideration is the specification and implementation of agent migration protocol. It is the only protocol governing both agent migration and inter-agent communication. To make it easier to deal with issues involved in different levels of the secure migration channel, a multi-layer architecture has been chosen and security policies have been attached and implemented at the connection and security layer. In addition, the MLVM has two more kernel level threads, the agent verifier and the security manager, to deal with privacy, physical access restrictions, application availability, content integrity, and access policy. The purpose of these security modules is to protect mobile agents against a malicious host as well as protect hosts against malicious agents.

3 Secure Agent Migration Channel

In a large-scale open network, we have to assume that a mobile agent system is subject to the fundamental security threats of disclosure, modification, denial of service, misuse, abuse, repudiation, and so on. The security challenge here is much more severe than the one in traditional client/server model. In addition, security mechanism should be able fit the purpose of various applications with different security requirements. Therefore, we are encouraged to develop the security mechanism based on the most well known techniques. In the current design of the IMAGO system, the integrated security mechanism consists of three kernel components: secure migration protocol, agent verifier and security monitor. Fig. 2 shows the logic relationship of security components with other system modules.

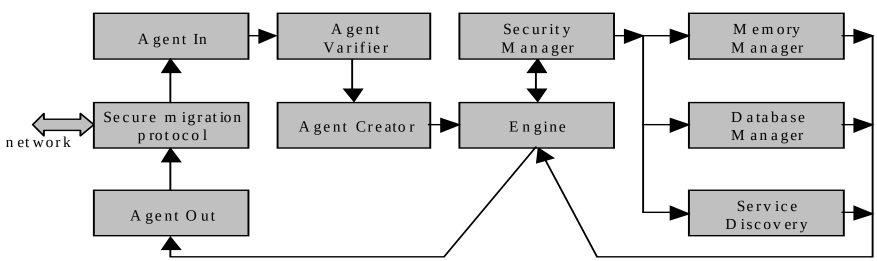


Fig. 2. Security Components in IMAGO System

As the primary identifying characteristic of IMAGO system is the sole protocol to support both agent mobility and inter-agent communication, we provide a secure and authenticated connection between two end points of a migration channel to protect the privacy and integrity of a moving agent. The technique incorporated with the IMAGO migration protocol is the excellent OpenSSL [16]. It is a full-featured toolkit implementing socket and transport layer security protocols as well as a set of general-purpose cryptography algorithms. Based on OpenSSL, we construct a secure server-to-server agent migration protocol with ability of authentication by using the RSA and DSA algorithms, which are based on a pair of complementary numerical keys.

When an agent intends to move to another host, it will be transferred to the Agent Out module. However, before the agent migration takes place, the sending server requires some initial configuration if the receiving server is first met. For each new destined server, the initial configuration is performed only once and then cached in a special internal table. The main purpose of the initialization is to exchange public keys of both parties and negotiate the data encryption option. OpenSSL offers multiple symmetric cryptography algorithms. For the sake of flexibility and availability, several cryptography options, such as, from the most secure to the least secure, 3DES, BlowFish, IDEA, DES and RC4, will be negotiated with respect to their preference order, until an option is agreed by both sending and receiving servers.

On the other hand, if the configuration information of receiving server is ready to use, the agent will be passed to the migration protocol stack immediately. The current implementation of migration protocol consists of five layers. They are, from top to bottom, marshaling, rendezvous, security, routing and connection. The whole stack sits on the top of TCP/IP. When a Protocol Data Unit (PDU) from the upper layer of migration protocol stack being passed to the security layer, a 256-bit randomly generated session key *Ks* is used for encryption/decryption operations. Let *d* be the PDU and *E* the agreed encryption algorithm, then $E_{Ks}(d)$ is the cipher to be transmitted. However, encryption can only insure that any third party sniffing the migration channel cannot intercept the agent PDU, we need to authenticate and protect the integrity of data as well. Thus, a DSA-based signature is generated for verifying and filtering of PDUs based on their origin. In order to spare computation cost, the DSA signature is not applied on the entire PDU, but only on the session key *Ks*. We use the sender’s DSA private key to form the signature $DSA_{sig}(Ks)$. Since the session key is generated on the sender side and unique for every PDU, it must be transmitted to the receiving server for the symmetric decryption operation. Needless to say, the session key must be encrypted prior to transit. By applying the RSA asymmetric encryption algorithm with the receiver’s public key *R+*, we encrypt the session key and digital signature together, that is, $RSA_{R+}(Ks, DSA_{sig}(Ks))$. Now, we are ready to form the new PDU generated by the security layer, that is,

$$E_{Ks}(d) + RSA_{R+}(Ks, DSA_{sig}(Ks)).$$

When an encrypted PDU arrives at the destination host, its security layer will use its own RSA private key to decrypt the session key and DSA signature. Then it invokes the DSA algorithm to verify the decrypted signature. If the verification process failed, this PDU is regarded useless and discarded thereafter. Otherwise, the corresponding symmetric cryptography algorithm is invoked to decrypt the cipher and restore the original PDU.

4 Security Manager

The secure migration channel guarantees the confidentiality, integrity, and passwordless authentication of an agent moving from host to host. However, the use of mobile agents raises another two important security concerns, *i.e.*, agents need protection from malicious hosts and hosts need to be protected against malicious agents.

Unfortunately, the problem posed by malicious hosts to agents seems impossible to be fully solved, because there is no easy way to enforce a host, especially a malicious host, to obey security requirements. Once an agent has arrived at a host, it submits itself completely to the host and cannot stop the host from malicious attacking. Even though some solutions have been proposed, many of them are used for attack detection, rather than prevent agents against misbehaving servers. In IMAGO system, we assume that IMAGO servers behave truthfully and honestly. However, a measure can be taken to let an agent owner at least detect if the mobile agent has been tampered in its life cycle. The IMAGO system adopted a verification technique similar to the read-only state proposed by [5]. In fact, the read-only state of an agent consists of its code and initial data and is signed by a MD5 digest calculated from the state. The home server then encrypts the digest with its private key before the agent moves out to other servers. When the agent finishes its data-mining task and eventually returns to its home server, the home server can compare the ready-only state against the signed agent digest to determine whether some malicious hosts have tampered with the agent code and initial data.

On the other hand, the IMAGO system takes a great measure to protect hosts against malicious agents. First of all, the MLVM is constructed as a sandbox, a commonly used technique to control agent execution at the byte-code instruction level. In MLVM, agents are user-level threads and executed within their own virtual memory space, thereby preventing them from interfering with each other. Secondly, there are two types of security policies for detecting malicious agents. One is agent verification and the other is runtime monitoring. Agent verification is mainly used to check an incoming mobile agent that has just arrived from a foreign host. An incoming agent will be transferred to the Agent Verifier before it can be scheduled for execution. This module checks whether the agent obeys the security rules of MLVM and detects potential security holes, such as illegal instructions, illegal memory requirements, bearing an expired life time limit, or whose stationery server being found in a blacklist. Clearly, we do not need to verify every incoming agent, for example, system built-in messengers are cached at each IMAGO server and they are always safe to execute.

Unfortunately, the verification process cannot find all potential dangers to the host server. It is possible that some agents are deliberately coded to damage remote servers and some are poorly coded to cause unexpected, harmful side effects. Even though the verifier can protect hosts against illegal instructions, it is not power enough to prevent denial of service attacks. For example, an agent might generate a huge number of clones to exhaust memory of the host, or it might infinitely migrate from host to host and do nothing. Such ill-behaved agents use up the host’s resources or services, such as by constantly consuming network bandwidth, CPU time and memory resources, and eventually cause the server malfunction. In order to prevent such potential risks, a run-time check is required to monitor the agent’s execution. We adopt limitation techniques [17] to control the persistent survivability of mobile agents, that is, to directly control mobile agents from their dynamic behaviors during different stages of their lifecycle.

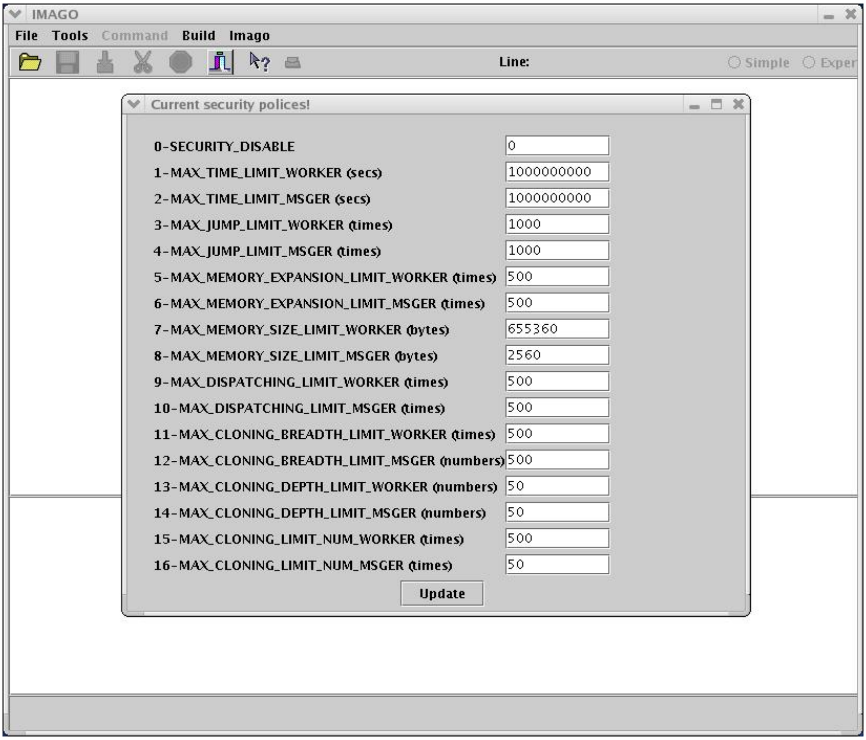


Fig. 3. The Interface of IMAGO Customized Security Policies

To prevent agents from running wild, the MLVM security manager enforces the followings limitations:

- ***Lifetime Limit:*** A lifetime limit is defined by setting a maximum amount of time that IMAGAO system allows a mobile agent to exist in an application. For example, a mobile agent that has been roaming the network past its timed life span shall be terminated.
- ***Range (Jump) Limit:*** A range or a jump limit is the maximum number of destinations or network hops that a mobile agent is allowed to travel.
- ***Dispatching Limit:*** A dispatching limit defines the maximum number of messengers an agent can dispatch.
- ***Cloning (Duplication) Limit:*** A cloning or a duplication limit is the maximum number that IMAGO system allows a mobile agent to be cloned. If an agent is terminated because it violates the cloning limitation, security manager will destroy all its clones as well.
- ***Memory Limits:*** Memory limits consist of the maximum size of memory that a mobile agent is allowed to apply and the number of times that a mobile agent can request for expanding memory during execution.

By using limitation techniques, the IMAGO security manager directly monitors activities of hosted agents and effectively protects the server against intentional or accidental attacks from agents. Of course, security requirements range much in different agent-based applications. For example, if a DDM application only involves servers in an internal network, the security challenges are much less than in the general open systems. For mission-critical, real world applications, the system administrator can configure various limitations. For this purpose, the IMAGO IDE (Integrated Development Environment) provides a user-friendly interface for customizing the security policies, as shown in Fig. 3. System administrator can incorporate security manager comprised of their choice of security policies, refined from a set of system default limitations.

5 Conclusions and Future Work

In this paper, we discussed the scheme of deploying mobile agents in DDM applications and security measures in the design and implementation of IMAGO system. The advantage of adopting mobile agents for DDM is to scale up to large, dynamic and remote data sources, such as various databases distributed over the Internet. In order to verify the feasibility and availability of the mobile agent based DDM proposal, we comprehensively identify the potential security threats from underlying networks, agents and hosts. Possible solutions, such as secure migration channel, sandbox technique, agent verification, and runtime security monitoring, are introduced. The security mechanism of IMAGO system has been integrated with MLVM, the virtual machine running on each IMAGO server. We also provide a user-friendly interface to facilitate customizing security policies according to the requirements of DDM applications. To evaluate the performance of the IMAGO security mechanism, we measured the overhead of security services through a set of benchmarks. The results show that the overhead caused by security modules are in the range of 5 to 10 percent of total execution time for most benchmarks.

Research on the agent based DDM involves further extensions to the IMAGO security policy and mechanism. First, the current technique of protecting agents against malicious hosts is quite weak. We are considering adding execution trace for detecting unauthorized changing of data and code of an agent through the faithful recording of the agent's execution on each visited server. Secondly, since databases may require their own authentication and authorization, retrieving such kind of information from service discovery is a pending problem. In addition, we are making investigations on adding more programming languages to the IMAGO system, as well as introducing more flexible and efficient communication tools, such as mobile socket, to facilitate DDM applications.

Acknowledgments. I would like to thank members in the IMAGO Lab for their contributions to the IMAGO project. I would also like to express my appreciation to the Natural Science and Engineering Council of Canada and State Key Laboratory of Novel Software Technology (Nanjing University) for supporting this research.

References

[1] Klusch, M., Lodi, S., Moro, G.: The Role of Agents in Distributed Data Mining: Issues and Benefits, IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03) (2003)
[2] Park, B., Kargupta, H.: Distributed Data Mining: Algorithms, Systems, and Applications, Data Mining Handbook, N. Ye, Ed. (2002)
[3] Krishnaswamy, S., Zaslasvky, A., and Loke, S, W.: Internet Delivery of Distributed Data Mining Services: Architectures, Issues and Prospects, Chapter 7 in the book Architectural Issues of Web-enabled Electronic Business, Murthy, V.K. and Shi, N. (eds.), Idea Group Publishing, (2003) 113–127
[4] Borselius, N.: Mobile Agent Security, Electronic & Communication Engineering Journal, Vol. 10 (2002) 211–218
[5] Karnik, N., Tripathi, A.: Security in the Ajanta mobile agent system, Software Practice and Experience, Vol. 31, No. 4 (2001) 301–329
[6] Gray, R., Kotz, D., Cybenko, G., Rus, D.: D'Agents: Security in a Multiple Language Mobile-agent System, Mobile Agents and Security, Lecture Notes in Computer Science, Springer-Verlag, No. 1419 (1998) 154–187
[7] Li, X.: On the Implementation of IMAGO System, International Journal of Computer Science and Network Security, Vol. 6, No. 2a (2006) 107–118
[8] Bettstetter, C., Renner, C.: A Comparison of Service Discovery Protocols and Implementation of the Service Location Protocol, Proc. of EUNICE 2000, sixth EUNICE Open European Summer School (2000) 13–15
[9] Salutation Consortium, Salutation Architecture Overview, White Paper, (1998) http://www.salutation.org/whitepaper
[10] Guttman, E., Perkins, C., Veizades, J., Day. M.: Service Location Protocol, Version 2, White Paper, IETF, RFC 2608 (1999)
[11] OASIS UDDI, UDDI White Paper, (2005) http://www.uddi.org
[12] Song, L., Li, X, Ni, J.: A Database Service Discovery Model for Mobile Agents, International Journal of Intelligent Information Technologies, Vol. 2, No. 2 (2006) 16–29
[13] MySQL AB *MySQL Reference Manual*, user’s manual, (2005) http://dev.mysql.com/doc/mysql/en/
[14] Li, X., Autran, G.: Inter-agent Communication in IMAGO Prolog, Lecture Notes in Artificial Intelligence, Vol. 3346 (2005) 163–180
[15] FIPA ACL, Agent Communication Language Specifications, *FIPA*, (2005) http://www.fipa.org
[16] OpenSSL, http://www.openssl.org/
[17] Greenberg, M., Byington, J., Harper, D.: Mobile Agents and Security, IEEE Communication Magazine, Vol. 36, No. 7 (1998) 76–85

Automatic Extraction of Business Rules to Improve Quality in Planning and Consolidation in Transport Logistics Based on Multi-agent Clustering

Igor Minakov[1], George Rzevski[2], Petr Skobelev[1], and Simon Volman[1]

[1] The Institute for the Control of Complex Systems, RAS,
Sadovaya Str., 61, 443020, Samara, Russia
{minakov,skobelev,volman}@magenta-technology.ru
[2] MAGENTA Technology, Gainsborough House, 59 - 60 Thames Street
Windsor, Berkshire, SL4 1TX, UK
{george.rzevski}@magenta-technology.com

Abstract. The article describes multi-agent engine for data clustering and IF-THEN rules generation and their application to transportation logistics. The developed engine can be used for investigating customer source data, pattern discovery in batch or in real time mode and ongoing forecasting and consolidation of orders and in other cases. Engine basic architecture fits well for both batch and real time clustering. The example of data clustering and generation of IF-THEN rules for one of UK logistics operators is considered. It is shown how the extracted rules were applied to automatic schedule generation and how as a result the quality of schedules was improved. The article also describes an approach, which allows getting orders consolidation from extracted rules. Algorithm of rule search and the obtained results analysis are other points mentioned.

Keywords: multi-agent planning, multi-agent learning, data mining, task and resource allocation, if-then rules, logistics, schedule generation, multi-agent real-time clustering, pattern extraction, orders consolidation.

1 Introduction

Nowadays, transportation scheduling is an urgent and significant task for great number of modern companies that have to operate on unstable and highly dynamic market where demand and supply change rapidly.

The complexity of problem solving results from large quantity and diversity of domain knowledge that should be taken into consideration during decision making process added to absence of problem-solving procedures that represent all peculiarities of contemporary business environment. During schedule generation process, it is necessary to consider individual preferences (including strategies, restrictions, preferences) of all vehicles, drivers, orders and clients. Clear example of such peculiarities is a set of rules used by one of companies that were under consideration:

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 124–137, 2007.
© Springer-Verlag Berlin Heidelberg 2007

- Some roads become unavailable for double-deckers after rain, since wet tree branches lower and hinder road traffic.
- Avoid allocating a driver to itinerary that goes near driver's home, since drivers tend to call at their places for a cup of tea.

All these factors make manual re-working an obligatory procedure for schedules created with systems for automatic schedule generation while leaving the re-working process quite a complicated task.

To make full use of problem domain features it is required to formalize all knowledge and criteria that serve as a base for human decision-making in schedule development. However, the implementation difficulties are in the following:

- Sometimes a man fails to formalize rules he or she uses for decision making
- Sometimes a man reports incorrect rules
- If several persons engaged into decision making process, there are rules that do not belong to knowledge of any particular person – these rules belong to a system on the whole

Therefore, to produce high-quality schedules in short terms, the system (or agents of this system – if it is an agent-based system) should automatically detect and take into consideration peculiarities of problem domain. For automatic search for such data dependencies and peculiarities, it is possible to use methods and approaches of Data Mining. Of all classical Data Mining methods, systems for revealing if-then rules give the most comprehensible, the best understood results that are easy to interpret, and if necessary, the results can be manually modified by an expert. That's why in this paper we'll discuss methods of automatic extraction of IF-THEN rules and show how such rules can be applied in real-life tasks.

1.1 Brief Overview of Algorithms, Applicable for If-Then Rules Extraction

Today several software products solve a problem of finding if-then rules on datasets. Mathematical algorithms set an underlying logic for these products, for example, algorithms of bounded combinatorial search (i.e. WizWhy), algorithms of developing decision trees (i.e. See5/C5.0) or algorithms based on association rules extraction (i.e. Apriori, Generalized Rule Induction). The description of these tools is available in [1]. However, classical systems have the following drawbacks:

- These algorithms provide tools to search for if-then rules with THEN part consisting of a single condition. It's a very weak type of rule, as a schedule represents a sophisticated entity made of multiple interdependent fields.
- Algorithms require significant data pre-processing before any analysis can be made, sometimes manually selecting target field, sometimes representing data as a set of zero's and one's etc, which is hard for problem domain experts, as well as tiresome, time consuming and error-prone.
- These algorithms have severe constraints on the number of data fields or unique values for each data field. But usually a schedule is a very complex network with multiple data fields and different values for each field.
- Many of these algorithms have constraints on a very high minimum support level for rule generation, thus missing many important-to-user rules, which have high confidence, but represented in the system by small number of records.

- These algorithms offer tools to search for if-then rules only in a batch mode while present-day requirements for systems of automatic schedule generation demand real time operations to react rapidly on market dynamics .

Hence, it is necessary to develop methods that will allow automatic detection of if-then rules with multiple IF part and will ensure rules' applicability for schedule generation. In addition, these methods should include adaptive mechanisms for extracting and adjusting rules in runtime.

To solve the problem we propose to reject searching and looking through all possible rules, but to find rules that characterize the densest groups of records. The value of discovered rules can be evaluated through calculating probabilistic and representative characteristics for these rules. The proposed approach uses multi-agent clustering to reveal dense groups of records.

To estimate the effectiveness of discovered rules, it is necessary to compare system-generated schedule and schedule produced manually by an operator. Nevertheless, acceptable criteria for metric proximity (similarity) for two schedules resists formalization, therefore such estimations should be executed through expertise.

1.2 Brief Overview of Clustering Algorithms

Clustering is a task aimed at revealing groups of high density in data space. These groups are to be further applied in methods of data analysis, knowledge retrieval or methods for effective information extraction. The following groups of algorithms are available up to the present moment:

- Partitioning algorithm – these algorithms decompose dataset of N objects into K clusters. Examples of such algorithms are: FOREL 2, KRAB, KRAB Heuristic, K-means, K-medoid, CLARANS (Clustering Large Application based on RANdomaized Search)
- Hierarchical decomposition and ordering - these algorithms use iterations to decompose dataset into smaller sets until termination conditions are satisfied. Examples of such algorithms are: FOREL, FOREL OPT, KRAB Heuristic 2, BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies). Decision Trees algorithms can be also placed into this group - C4.5, See5/C5.0, CART (Classification and Regression Trees). Similarly, the following algorithms can be classified into this group: DBSCAN, CURE (Clustering Using Representatives)
- Algorithms which quantize the space - these algorithms quantize the space into finite number of cells and then run all operations on produced cells. Examples of such algorithms are: STING (STatistical INformation Grid-based method), CLIQUE [1, 2, 3].

It is more preferable and more effective to find clusters within sub-spaces of original space for the initial task since multi-dimensional data can include noise or have evenly distributed values for some dimensions (and often there is no way to exclude non-meaningful fields from initial analysis). Among all algorithms mentioned above only CLIQUE can detect clusters on sub-spaces, still restrictions of this algorithm make it unsuitable for the current problem:

- The algorithm operates in a batch mode.
- The algorithm subdivides the space into finite number of cells in advance. This approach is non-applicable in our case, since quality of the obtained results will depend upon initial subdivision of space into cells.
- Found clusters are not organized into hierarchical groups.
- To overcome these restrictions, we decided to develop a new clustering algorithm on the base of multi-agent approach [4, 5].

2 Problem Domain Specification (Transport Logistics Domain)

The task concerns developing an optimal schedule to allocate transportation orders (in this case, optimality of a schedule is considered in terms of a specific criterion). An order contains information on cargo properties (such as weight, volume, type, name of client and specific requirement for cargo delivery conditions) and requirements for source location and its time window, destination location and its time window. Orders can be allocated to a limited number of trucks of different types. In some cases, a transportation company can execute an order with 3rd party carriers, however, this leads to profitability decrease. Some orders include additional restrictions on appropriate trucks (for example, chilled cargoes are to be transported on trucks equipped with refrigerating units). Truck drivers also have schedules of their own that regulate working hours, besides driver workday is a subject of safety requirements and labor code. Orders come into system in time moments that are unknown in advance; scheduling is carried out in real time, this means that planning process and execution process go in parallel. Nevertheless, event planning is to be accomplished before event execution commences or before the time when event execution can be considered as delayed. If necessary, event processing breaks off when a new event appears, for example, in case of a truck’s breakdown, no more allocations is made to the truck, already allocated orders are re-planned while allocations on other trucks continue to go in parallel.

The key criterion in estimating a schedule is maximization schedule value to an enterprise. The following factors exert influence on enterprise-relevant value of a schedule:

- Cost of schedule (cost of transportation each orders according to schedule)
- Total mileage of all trucks
- Customer satisfaction level of individual clients
- Percentage of successfully allocated orders
- Planning time

More details on the subject are available in [5, 6].

3 Suggested Solution

On the basis of problem domain data a table is created, where each row (record) represents a unique entity of problem domain, for example, a transportation order that includes information on source and destination location of transportation and cargo-related information. Thus, the problem domain is presented as a multi-dimensional space, where each record is an element of this space. This space is heterogeneous, since axes of the given space can be of different types (integer, real, strings, currency, time etc).

3.1 Multi-agent Clustering Algorithm

Within the approach offered, agents of records and clusters start to play the main role. An agent is a software object capable of analyzing a situation, making decisions, performing actions over objects of a world and of communicating with other agents. Entering clusters, agents form virtual communities similar to temporary hierarchies, which can be organized in different ways.

In the most simple case, records try to find the most “profitable” clusters with the maximum density, which is a metric for its decisions (in more complex cases, metric can include number of records, number of sub-dimensions for cluster, time of life in cluster, type of attributes etc). The process of search is always started with the nearest points and extends gradually. When a record finds a proper cluster, it makes an offer and waits for a reply. The cluster reconsiders its locality, calculates its variant and either accepts or rejects the offer. Thus instead of a centralized optimal “front-office” decision of a classic algorithm, the offered approach provides solutions taken at the lowest level. These solutions are only based on some current local balance of interests of a particular record and some cluster. If each of the sides agrees, the record enters the cluster, if not – the record searches for other variants. The stages of this process in a simple 2D case are shown in Fig. 1, where: a – the first record’s entering, b – the second record’s entering and creating of the cluster, c – the third record’s entering, forming the second cluster of the first cluster and the third record, d – the second cluster “lures” the records from the first cluster to enter it (as it’s two-dimensional and based on selected value formula is more “profitable” for records), the first cluster is dropped out, the fourth record comes, e – a new cluster is formed, f – thea new cluster “lures” the records from the inner cluster, a new record comes, g – a new record comes and the process starts from the very beginning, h – the final cluster.

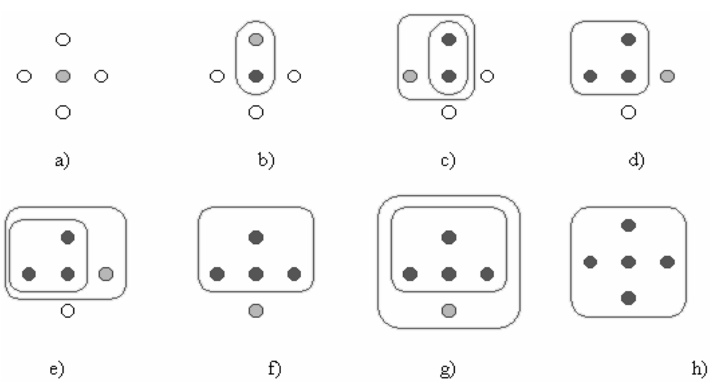


Fig. 1. The stages of forming a cluster within the approach developed

In more complicated cases, agents can interact in the virtual market and use money, which regulate the records' abilities or restrict them in making decisions in a natural way. For example, in transportation logistics, the sum of money available for a record can be set as a price for order, which customer is due to pay. Then the record of VIP order is "richer" than a record of a single order from random customer. This makes it possible for the first record to form clusters with distant records, taking into account bigger context, and generating more clusters as a result. In this situation different models of the microeconomics of cluster or records can be applied. For example, clusters can take charges from records for the right to enter them in accordance with the "club system" model (the amount of payment does not depend on the situation, richness of a record or the number of the club members) or according to the "shareholder" model (the amount depends on the situation). In the latter case, a record can even increase its capital entering or quitting the cluster in the right time. It is obvious that if a record sees a more beneficial cluster and it does not have enough money to enter it, it can leave the current cluster, which may cause a wave of the process of dropping out cluster. In the process of interaction of clusters and records real negotiation is conducted, with parties meeting each other half-way, for example, by mutual concessions. All this provides different variants of taking approximate decisions and the control of the correlation between the accuracy and cost and time effectiveness.

A case is even more complicated in the situation with time dependencies, when the processes of self-organisation can be supplemented by the processes of evolution. In this case, clusters and records pay time taxes for acting in the system and for staying in clusters. This will have an effect on their financial resources, making some clusters and records disappear from the system, decreasing thus the load on the system, while some of them will be growing and becoming stronger. In this case, clusters and records are supposed to think not only about "money", but about their "lifespan", balancing between the criteria. For example, if a cluster is under threat to be dropped out of the system, when it does not receive any offers from records for a long time and thus is not paid the entrance fee, it can considerably decrease this fee to attract new members. This will help the cluster to stay in the system for some time.

These examples make it clear that clusters and records can make decisions according to different rules, combining, for example, a simple local density and "public value" (the number of clusters a record participates in or the number of records in a cluster), "richness" or 'lifespan" of a cluster or a record. These combinations of rules depend on what a user wants to see: bigger and more steady groups of clusters or, on the contrary, the most dynamic and miniature; densest or the most disperse; "the richest", the most "die-hard" or any others.

The clustering processes for a stable data structure resembles crystallization processes – records create various structures at the micro level, the produced structures on their part start to participate in clustering process. The process stops when the whole structure is crystallized. The outcome of the process generates high-level structures, which are more or less stable, but are adjusted in real time as new records (events) come into system. Further information on clustering algorithm is available in [4, 7].

3.2 If-Then Rules Extraction Algorithm

Further, it is necessary to transform clusters into a form that can be easily under-stood by a problem domain expert or by an automatic system, in other words – rules. To produce a rule on the base of cluster, all axes of the space are subdivided into two groups regarding particular features of problem domain – first group represent fields, which can be part of IF part of rule, and second – of THEN part (if such groups are not specified, it is considered as each field can be either in IF or THEN part of different rules). Next, retrieval of data for various dependencies starts – how fields of one category depend upon fields of another category. The guideline for space subdivision is quite simple – one group includes fields over which we have a control and which we can manipulate (information is being defined at schedule generation stage) while the other group includes fields unavailable for control (order information). In analysis we neglect clusters made of fields only from one category.

Let us consider cluster as a logical rule of the following sort IF (condition A_1) and (condition A_2) and … (condition A_N), THEN (condition B_1) and (condition B_2) and … (condition B_M), where A_i – conditions relevant to axes from the second category (fields over which we have no control), while B_i – conditions relevant to axes of the first category (fields over which we have a control). For example, if an order requires transportation of a cargo of 5 kilograms, then this order should be allocated to the truck of “Gazel” type that belongs to Trans-GAZ carrier-company.

We can distinguish three main characteristics of each rule that help to evaluate the rules produced:

- Representativeness – shows number of elements that correspond to the given rule (for IF part).
- Confidence level – shows how many elements of domain space that correspond to the left part of the rule (IF part) meet conditions of the right part of the rule (THEN part). This parameter depends upon applied pattern (for example, among all ones that are parous, all are women; while not all women are parous).
- Completeness – shows how many elements of space meet conditions of the right part of the rule (THEN part) and do not meet left part of the rule (IF part). For example, “if you are a human being, then you are mortal” rule has high level of representativeness but a low level of completeness (since not all mortal beings are human beings).

The higher value of representativeness and confidence level has a cluster, the more valuable the revealed interdependency is for an expert (with the exception of obvious rule or tautology).

When the rule is detected, we can try to move some conditions from the reason part of the rule (IF) into the consequence part of the rule (THEN). If this operation does not decrease a confidence level of the rule, then the modified rule is more preferable as it contains fewer conditions in IF part. For example, the following rule “If an order requires transportation from source location Krasnoyarsk then destination location of an order is Moscow and this order should be planned on truck of ZIL type” is more preferable than the rule “If an order requires transportation from Krasnoyarsk to Moscow, then this order should be planned to truck of ZIL type” (if confidence level after transformation does not decrease).

Note that any rule formalized in terms of multi-dimensional space will represent a certain group of elements of high density, in other words, a cluster. Hence, any rule with high level of confidence represents a cluster (inverse statement is not true in general case); therefore, if clustering algorithm finds all dense groups, it finds all rules with high level of confidence, so it's reasonable to use clustering algorithm to extract IF-THEN rules.

4 Integration with Logistics i-Scheduler

A developed product for extraction rules in logistics domain (it was named Pattern Seeker) is also able to integrate with logistic schedulers, thus monitoring a planning application in real-time mode and providing rules, which can be applied to improve the quality of planning in a constantly changing dynamic environment. A possible scheme of integration with operational planner Magenta Operational i-Scheduler is given below (Figure 2). Magenta Operational i-Scheduler is an agent-based Intelligent Scheduler for Road Transportation real-time Applications [6].

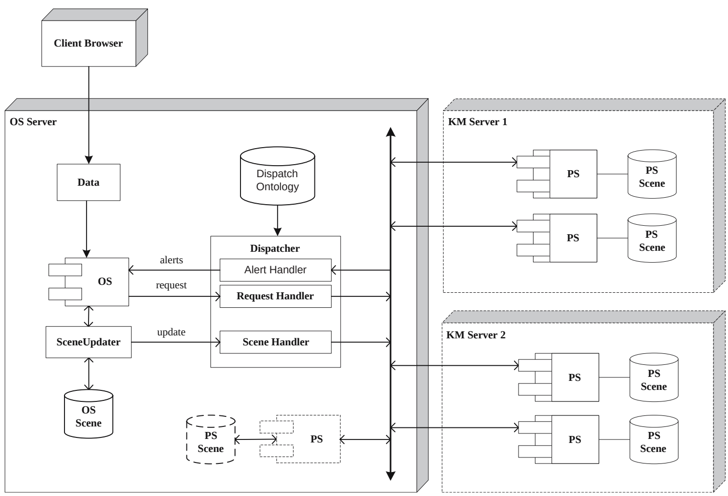


Fig. 2. Integration of Pattern Seeker with Magenta Logistics i-Scheduler. (OS – Magenta Operational i-Scheduler; PS – Pattern Seeker; Dispatcher – main module, which is responsible for parallel work of several applications; OS Scene – ontological scene of OS, which represent a transportation network; PS Scene – ontological scene, which is given as an input to PS)

Pattern Seeker is located on a separate computer or several computers (as its algorithm is parallel). In some cases, it can exist on one computer together with OS, but this variant can lead to OS performance decrease, which will require considerable calculation resources.

The two systems interact through dispatcher module. Both modules interact with dispatcher module by means of asynchronous messages.

Dispatcher availability gives the following advantages:

- An ability to add new Pattern Seeker instance.
- An ability to change any Pattern Seeker instance location.
- Modules interaction transparency and unification gives an ability to integrate the system with modules of other producers.

OS and Pattern Seeker work independently

When the two systems work together, the exchange of the following messages takes place:

OS sends Pattern Seeker the following messages:

- Notification about new order that has come into the system
- Notification about schedule changes
- Query about some certain order planning limitations
- Query about preferred consolidation for non-allocated order
- Pattern Seeker can send OS the following messages for agents:
- Proposals on joining of non-allocated orders into one consolidation
- Proposals on adding new orders to a trip for backhaul
- Advice to re-plan some certain orders

This process provides a learning basis for OS agents, as each instance of each agent type applies its strategies for selecting best route or best order or consolidation option, and depending on recommendations selects best appropriate in the current market conditions, changing set and/or priorities of strategies.

5 Data of Logistics Company Focused on Transportation Scheduling

During the cooperation with the logistics company a problem was set to automatically generate a schedule that would be similar to a schedule developed by an operator manually. However, the customer failed to define metric criteria to estimate proximity and similarity of schedules, thus, schedule estimation was run through expert evaluation.

The following points specify project purpose in more details:

- Plan customer orders and provide schedule using own and 3rd party fleets
- Ensure that the generated schedule is sufficiently workable from cost perspective.
- Consider all feasibility constraints and preferences, which are now possessed by operators (patterns of delivery, preferred carriers, trip shape)
- Consider bringing future orders to optimize today's trips
- Validate the use of human-like heuristics in a real environment (allocation of constrained orders/resources first, working from most distant/close points, etc)

The customer provided a dataset made of 920 orders planned manually by operators. To enable the system of automatic schedule generation produce a schedule similar to the one created by a human operator, we considered a task of extracting hidden rules from the test dataset that served as a base for a schedule development. Further information on the system for automatic schedule generation is available in [6].

A table with information on orders and the way how orders were planned was created where each row represents an order. The following fields were included into the table (each field represents space axis):

- From – the source location of cargo transportation;
- To – the destination location of cargo transportation;
- Pallets – number of units to be transported;
- CarrierType – name of carrier company that was selected for order execution;
- Depot – location of vehicle that should execute transportation;
- VehicleType – type of vehicle that should execute transportation;
- Orders – total amount of orders transported in parallel on a vehicle;

Based upon characteristics of the problem domain, we selected the following pattern: Form, To, and Pallets axes are set as axes of the second category (we do not have control over these fields), while CarrierType, Depot, VehicleType, and Orders axes are set as axes from the first category (we do have control over these fields).

The developed algorithm found 218 rules. Figure 3 shows how found rules are distributed by confidence level.

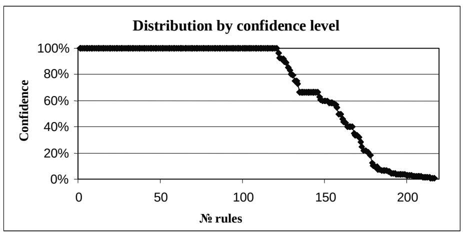


Fig. 3. Distribution of System-detected Rules by Confidence Level

As the graph shows, more than half of rules have confidence level of 100%, these figures ensure high value of the detected rules.

Table 1 shows an example of system-found rules, companies and geographical locations which were renamed by customer demand.

Table 1. Example of the Rules Detected by the System

№	Pattern	%	cou
182	<To> = "Bryansk" -> <From> = "Moskow" ,<Carriertype> = "Bryansk TRANS" ,<Depot> = "Bryansk" ,<Pallets> = "26" ,<Vehicletype> = "ZIL"	100	37
110	<To> = "Samara" -> <Carriertype> = "Moskow" ,<Depot> = "Moskow" ,<Vehicletype> = "ZIL" ,<Orders> = "11"	100	33
52	<To> = "Archangelsk 1" -> <From> = "Volgograd" ,<Carriertype> = "TRANS GAZ" ,<Depot> = "Archangelsk" ,<Pallets> = "26" ,<Vehicletype> = "Gazel" ,<Orders> = "3"	100	30
111	<From> = "Archangelsk 3" ,<To> = "Kiev" -> <Carriertype> = "TRANS GAZ" ,<Depot> = "Kiev" ,<Pallets> = "26" ,<Vehicletype> = "Kamaz" ,<Orders> = "3"	100	30
8	<To> = "Bryansk" ,<Pallets> = "26" -> <From> = "Kiev" ,<Carriertype> = "TRANS GAZ" ,<Depot> = "Kiev"	100	28
101	<From> = "Samara" ,<To> = "Volgograd" -> <Carriertype> = "TRANS GAZ" ,<Depot> = "Volgograd" ,<Vehicletype> = "Gazel" ,<Orders> = "11"	100	28
132	<To> = "Bryansk" ,<Pallets> in [1.0 .. 3.0] -> <Carriertype> = "Bryansk TRANS" ,<Depot> = "Bryansk" ,<Vehicletype> = "Kamaz" ,<Orders> = "11"	100	26
143	<From> = "Krasnoyarsk" ,<To> = "Moskow" ,<Pallets> in [1.0 .. 3.0] -> <Carriertype> = "Bryansk TRANS" ,<Depot> = "Bryansk" ,<Vehicletype> = "ZIL" ,<Orders> = "11"	100	24

*Names of geographical locations are obfuscated to keep confidentiality of client's data

System-detected rules were demonstrated to an expert who confirmed most of the rules and proved the revealed dependencies to be intrinsic characteristics of the problem domain. Even more, experts did not report in advance approx. 8-12% of the rules, which were found by the system, although these rules have great value of confidence level.

All obtained rules were loaded into the knowledge base of the system for automatic schedule generation.

Schedule calculation time:

- Without Pattern Seeker patterns, for 1 week period - ~ 5 hours
- With usage of Pattern Seeker patterns, for 1 week period - ~3 hours

Moreover, expert evaluations for the following were performed:

- Analysis of schedule created by experienced human-operator;
- Analysis of system-generated schedule (without use of extracted rules);
- Analysis of system-generated schedule with extracted rules applied;

Schedules were developed on another dataset of the customer company that was different from one served as base for rule detection. The expert declared the system-generated schedule where revealed rules were applied, as the one that was the most similar to the schedule created by experienced human-operator.

Implementation of the algorithm brought gains in schedule quality. The following figures show quality grows after application of Pattern Seeker-found rules:

- Manual rework needed – decreased by 32%
- Trip quality – increased by 17%
- Gaps presence in the trips – decreased by 11%
- Fleet mileage – decreased by 16%
- Fleet usage – decreased by 8%

In overall, it brought approx. 20% of increase in schedule quality (To obtain more information of how schedule quality is calculated, and by which parameters it can be affected, please refer to [6]).

The application of this system will considerably decrease the time required to customize logistic i-scheduler to customer problem domain. According to estimations, the customization period will drastically decline from 1-2 months to 10-15 days.

6 Consolidation Example

Finding possible options for consolidation within transport logistics is one of the specific cases for applying clustering algorithm and running clustering analysis. The clustering algorithm is applied by setting specific filters on IF and THEN parts of the rules. This is useful when the rules consolidating fields for geographical coordinates, time windows and Journey-Time Matrix information are of particular interest. In that case, each cluster can be considered as a group of orders that are potentially suitable for consolidation.

The following parameters should be considered:

- Nearness of source locations by geography and by distance/time (JTM)
- Nearness of destination locations by geography and by distance/time (JTM)
- Fully adequate intersection of time intervals so that if one truck takes all orders of consolidation, it manages to deliver them in time
- Different sizes of trucks that can take the group

At the next step, the heuristics are applied when the consolidation groups are found. It is necessary to define for each cluster the way in which consolidation should be treated within this group – whether all orders in the group are to be shipped by one truck or orders in the group are to be shipped by several trucks that are similar in characteristics. The decision depends upon distance between locations, since time is required to load each cargo and to deliver each cargo, whilst driver's working hours are bounded. In addition, the decision depends upon cargo properties, in some cases, a special-purpose truck is required to deliver a cargo, for example, for chilled cargoes.

Additional clustering run on the consolidation clusters helps to bring to light more dependencies and details. When tested on real data the system demonstrated the following characteristics:

- 90 % of orders found consolidations with at least 1 order
- 423 good consolidations (20-26 pallets) were found

Examples of consolidations found can be seen in Fig. 4.

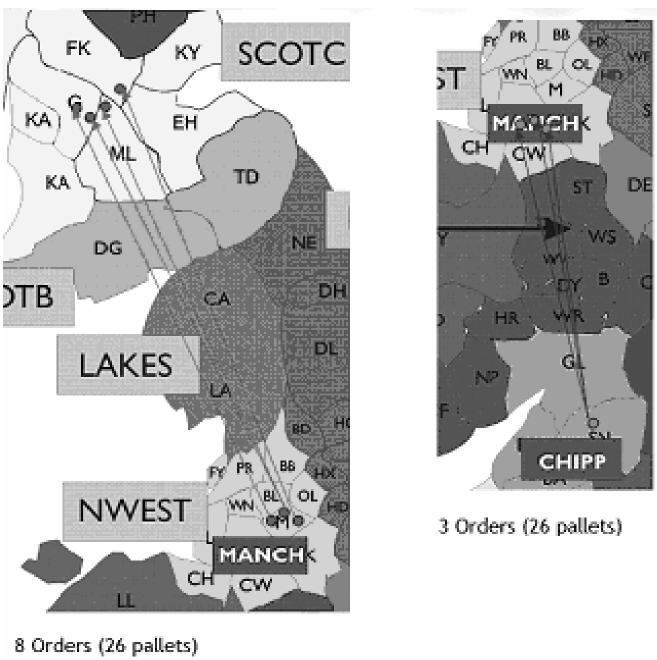


Fig. 4. Example of Consolidations Found

Table 2. Results of Consolidations Found

Pallets	Orders			
	Natural Consolidation	Inbound	Outbound	Pattern Seeker
26	715	736	1202	1250
25	61	52	96	79
24	35	63	71	77
23	83	77	48	44
22	66	66	71	67
21	74	57	60	50
20	60	51	24	30
Total >= 20	1094	1102	1572	1597

More detailed results are given in Table 2

As a result with a help of Pattern Seeker it was possible to find more than 45% of consolidations, resulting in improvement of generated schedule quality by more than 15 %.

7 Conclusion

The following advantages for client use were achieved through applying offered approach to various real-world logistics applications from different problem domains:

- Significant improvement in quality of automatically generated schedules due to the usage of extracted business patterns;
- Considerable assistance for operators in developing schedules in manual mode or through semi-automatic mode that increases efficiency and accuracy of decision making;
- Ability to validate existing human heuristics and find unknown ones;
- Enhanced knowledge on problem domain that leads to great improvements of scheduling model;
- Decreasing time required to customize scheduler for client problem domain
- Revealing interdependencies hidden in problem domain and capability to spot in time changes in existing dependencies to affect quality of integrated scheduler.

The current approach can be further used for dynamic validation or adjustment of patterns and to improve agents planning strategies according to new incoming events. The approach also applies to a preliminary investigation of problem domain to find more dependencies in data through data analysis and rule retrieval. The produced results can be further used for schedule development either manually or with systems for automatic schedule generation.

Furthermore, the presented investigation showed the following scientific results:

- The new algorithm of searching for if-then rules with multiple THEN part was developed and implemented.

- Revealed rules were applied to the system for automatic schedule generation. Problem domain experts assured that the schedules generated through automatic systems where these rules were applied proved to be similar to the manually created schedules.
- At the same time, system-generated schedules produced on the base of rules exceeded in quality these system-generated schedules where no rules were applied.
- A framework was developed as a base for a new generation of software products that adapt, learn and evolve over their life cycle.

References

1. Larose D.: Discovering Knowledge in data. Wiley Interscience (2005)
2. Agrawal R., Gehrke J., Gunopulos D., Raghavan P.: Automatic subspace clustering of high dimensional data for data mining applications. Proceeding of the ACM SIGMOD Conference on Management of Data, June 2-4, Seattle, Washington, USA, Publisher of proceedings (1998) 94-105
3. Ester M., Kriegel H., Sander J., Xu X.: A Density-Based Algorithm for Discovering Clustering in Large Spatial Database with Noise. Proceeding of 2nd International Conference on KDD, August 2-4, Portland, Oregon, USA, Publisher of proceedings (1996) 226–231
4. Minakov I., Rzevski G.: Skobelev P.: Data Mining. Patent Reference No: GB 2 411 015 A. Published 17.08.2005
5. Rzevski G., Skobelev P.: Agent Method and Computer System For Negotiating in a Virtual Environment, Patent Reference No: WO 03/067432 A1. Published 14.08.2003
6. Himoff J., Rzevski G., Skobelev P.: Multi-Agent Logistics i-Scheduler for Road Transportation: Proceedings of the AAMAS’06, May 8-12, Hakodate, Hokkaido, Japan, Publisher of proceedings (2006) 1514 – 1521
7. Andreev V., Batishchev S., Ivkushkin K., Minakov I., Rzevski G., Safronov A., Skobelev P.: MagentA Multi-Agent Engines for Decision Making Support // International Conference on Advanced Infrastructure for Electronic Business, Science, Education and Medicine on the Internet (ISBN 88-85280-63-3), L’Aquila, Italy (2002) 64–76

Intelligent Agents for Real Time Data Mining in Telecommunications Networks

Luis E. Rocha-Mier[1], Leonid Sheremetov[2], and Ildar Batyrshin[3]

[1] Department of Analytics, SAS Institute
Paseo de la Reforma 505 Piso 38, Torre Mayor Col Cuauhtemoc
C.P. 06500 Mexico DF, Mexico
luisenrique.rocha@sas.com
[2] St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences,
39, 14th Line, St. Petersburg, 199178, Russia
lsheremetov@aim.com
[3] Institute of Problems of Informatics, Academy of Sciences of Tatarstan, 20, Mushtari,
Kazan, 420012, Russia
batyr1@qmail.com

Abstract. Over the last years, the data generated in Telecommunications Networks has reached unmanageable limits of information. Data Mining (DM) techniques have showed their advantages on helping to manage this information and transforming it in useful knowledge. However, due to the dynamics of the environment of Telecommunications Networks, the simple application or adaptation of DM techniques is not enough to obtain timely a deeper knowledge. In this paper, this problem is addressed by applying DM techniques in real time. First, we propose a methodology taking into account all the processes involved in transforming telecommunications data into information, and finally to knowledge. Second, we propose a framework for the utilization of Intelligent Agents to help the process of DM in real time. To illustrate our approach, we describe a real-life case study based on the integration of Intelligent Agents and DM technologies for obtaining in real time knowledge that is critical for managing telecommunication networks.

Keywords: Real-time Data Mining, Intelligent Agents, Telecommunications Networks.

1 Introduction

Over the last years, the telecommunications industry has experienced an unthinkable growth. Nowadays, it generates and stores a tremendous amount of data [1]. These data are classified in three types: call detail data, network data and customer data. The massive generation of network-management data typically hides knowledge that is critical to some of the key tasks involved in effectively managing a communication network [2]. Data Mining (DM) is the exploration and analysis of large quantities of data in order to discover meaningful patterns and rules [3]. Different DM models have been successfully applied in the telecommunication industry to solve problems like: fraud detection, marketing/customer’ profiling, network fault isolation and forecasting telecommunication equipment failures [4], [5], [6], [7]. However, these works do not

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 138–152, 2007.
© Springer-Verlag Berlin Heidelberg 2007

take into account the eventual changes of Telecommunications Networks' environment and the knowledge obtained from their DM models can become obsolete if they are not updated in a timely manner.

In order to analyze the Telecommunications Networks data, these data must be transformed in time series. Recently, a variety of methods of Time Series Data Mining (TSDM) have been developed [8], [9], [10], [11]. The specific of problems in Telecommunications Networks is that it is impossible to analyze them by using only statistics. Moreover, if traditional DM techniques are applied at a considered time, a high risk exists that theses techniques could became obsolete in a future time as we above-mentioned. The Telecommunications Networks environment is dynamic and can change dramatically in a very short time. Therefore, traditional DM techniques based on static off-line training usually do not give satisfactory results predicting network behavior. Another disadvantage of traditional methods is the need of a experienced user to train the models and interpret their results.

This work is an attempt to overcome these limitations of traditional approach using real-time DM [6] and Intelligent Agent (IA) technology [12]. The agents have proved their efficiency to solve problems in very complex and dynamic environments. An IA is able to sense the environment, process the data/information obtained from the environment, and act to execute the most appropriate actions that contribute to the improvement of his behavior: they are especially useful for highly dynamic environments as telecommunications networks [13]. That is why IA technology has been widely used as a framework for distributed DM and data fusion [14], [15].

The approach of this work is to automate the DM process of Telecommunications Networks data using cognitive and cooperative capabilities of IA and Web services (WS) encapsulating specific DM models throughout the DM process cycle. The contribution of this research work is twofold: i) we propose a theoretical framework decomposing the DM process of Telecommunications Networks data into several steps, and ii) we develop a hybrid multi-agent system supporting the DM process. Moreover, we describe a case study based on the integration of Intelligent Agents and DM technologies for mining Telecommunications Networks data.

The rest of the paper is organized as follows. Section 2 describes the proposed theoretical framework. The perceptual and combined forecasting as well as the neural network model for prediction are described in more detail. In Section 3 the multi-agent infrastructure supporting the proposed approach is described. In Section 4 we present our experimental results and their analysis. Finally, conclusions and future work are summarized.

2 Theoretical Framework

The problem discussed in this paper consists in predicting a network variable (network utilization level in our case) based on the historical observations of the network data. The predictions are made on a daily basis with one-hour shifts. The proposed framework for real-time DM contains seven steps (see Fig. 1):

1. Collection of Telecommunication Network variables from distributed data sources (*e.g.* network data logs from network segment routers) and their integration in a Network Data Mart [16].

2. Preparation of the collected data and construction of time series. Sometimes the collected data must suffer some transformations (*e.g.* delete of repeated or duplicated data, normalization of time scales, etc.) to be used in the DM process.
3. Correlation analysis of the network variables to eliminate the redundant network variables. At this step, the network variables should be selected for further optimization of the Data Mining forecasting and predictive models.
4. Importance analysis of the most relevant network variables with respect to the target variable (the variable to be predicted).
5. Forecasting of the network variables to be used in the predictive/descriptive model. From the time series of each network variable, it is important to select the most appropriate forecasting method or combine forecasting methods.
6. Prediction/Description Modeling. At this step, the most appropriate model (*e.g.* Neural Networks, Decision Trees, Lineal Regression, etc.) is determined and applied to predict a characteristic network variable.
7. Evaluation of the DM process and interpretation of the results.

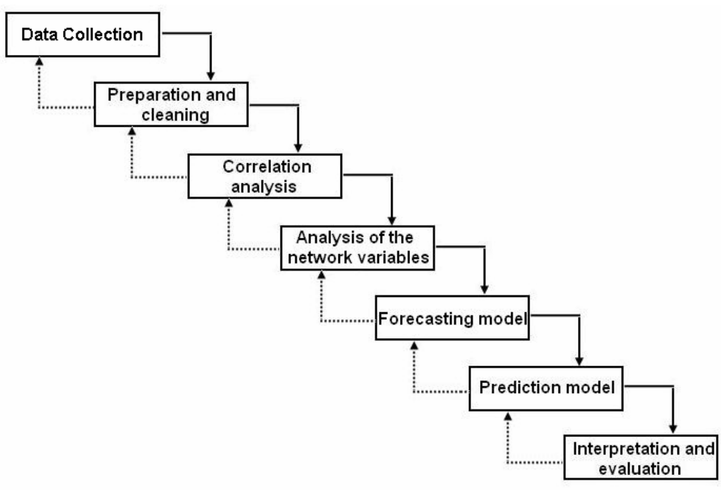


Fig. 1. Steps of the theoretical framework

The first four steps of the theoretical framework are the collection of data and the construction of time series data that contain several irredundant/relevant network variables that describe the environment state in a telecommunications network.

A time series is a sequence of vectors, $x(t)$, $t = 0,1,...$, where t represents an elapsed time. Theoretically, x may be a value, which varies continuously with t, such as the utilization of network links. In practice, for any given physical system, x will be sampled to give a series of discrete data points, equally spaced in time. Time series are generally sequences of measurements of one or more visible variables of an underlying dynamic system [17]. Examples of the variables used in the case study presented below include: Ethernet Traffic, Data Base Load, CPU utilization, and TCP transmissions.

The time series from Telecommunications Networks usually have redundant and irrelevant variables that contain the same information and contribute in the same way to the target problem (prediction or description of an event). This represents a problem of selection of the important number of variables to take into account. In order to address this problem, some correlation technique to detect the redundant variables should be used. The Spearman correlation could be an alternative. This technique indicates that there is high correlation between two variables if the result is close to one and a low correlation if the result is close to zero. A threshold of 0.8 was chosen to detect the variables that were highly correlated.

Once the redundant variables are eliminated at the Correlation analysis step, there are still irrelevant statistics that do not contribute to the target prediction. Besides, the large number of input variables can affect forecasting and prediction models. Once artificial neural networks (ANN) are used in the prediction model, they are affected in two ways. First, the more variables are used as inputs into the network, the larger the network needs to be, increasing the risk of overfitting and increasing the size of the training set. Second, the more variables are used, the longer it takes the network to converge to a set of weights. And, with too many variables, the weights are less likely to be optimal. This variable selection problem is a common difficulty for statisticians and decision trees provide a good method for choosing the best variables [3].

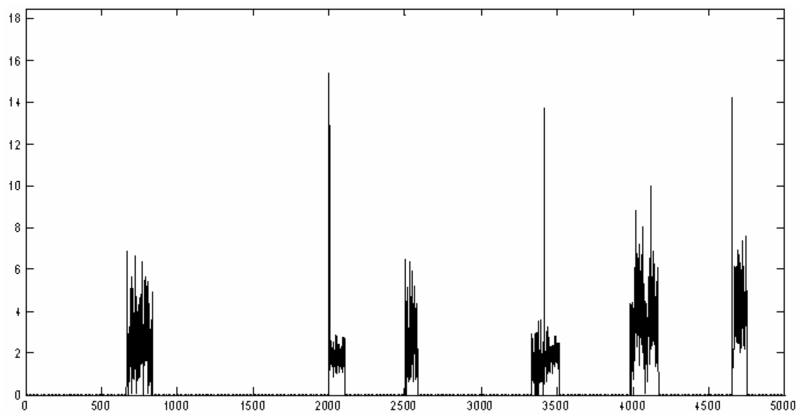
The predictive model has to be based on the irredundant/relevant network variables obtained at the previous step of the proposed framework. However, the values of each variable are not known at the moment of the prediction; it is where forecasting takes place [18], [19]. Although the next time points following the known ones are the output information we only want for forecasting, practically many of the existing forecasting algorithms usually supply or are able to supply some other related information supporting the forecasting. For example, a Bayes algorithm may also supply M values of post-probabilities for each possible time point. We can take advantage of this information in order to compare and/or combine the results of multiple classifiers.

As an example, parallel perceptual forecasting is implemented within the framework. Forecasting model using Bayesian approach is constructed to combine the results. Finally, a predictive model based on a multilayer perceptron (MLP) neural network is used to predict the network utilization level. After the prediction of network events, the results must be interpreted and used to improve the performance of Telecommunication Networks (actionable knowledge). We describe forecasting and predictive models in more details below.

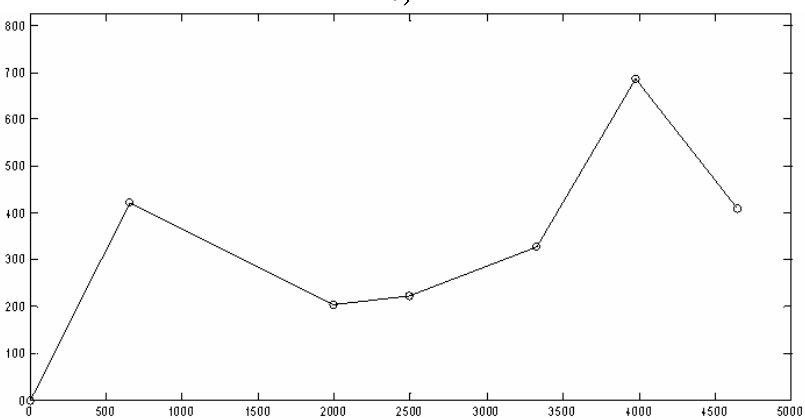
2.1 Perceptual Forecasting

Both the stochastic nature of the TS and the need in real time forecasting make it feasible to approximate the original TS with piece-wise linear patterns. Recently, several algorithms for piece-wise linear representation of time series have been developed in TSDM. For example, the method of symbolic representation of time series called Symbolic Aggregate approximation SAX was proposed in [8], [10]. In this paper, a perceptual forecasting method is developed on the basis of the Moving Approximation (MAP) transform reported in [11]. The idea of the method consists in the following. We replace the time series values by slope values. The time pattern before forecasting time moments is called a goal pattern. As a result, the goal pattern of time series values is replaced by a perceptual pattern of slope values describing a class of time series patterns with the shape of the goal pattern. In the history of time series, the method looks for the subsequence of slope values most similar to the perceptual image of the goal pattern.

To apply this idea to the considered problem, first, all TS of measured variables should be pattern TS. Due to their stochastic nature of the TS (Fig. 2a), we both smooth and divide patterns using the beginning of the impulse and its average intensity as pattern descriptors, as shown in example from Fig. 2b. Next, we analyze the TS of patterns for each day to construct classes of signals. For each class we identify the forthcoming classes with the corresponding probability. Also, a corresponding average signal TS is constructed for each class.



a)



b)

Fig. 2. Example of initial and pattern TS for the variable “Web Browsing.CPU Utilization”: a) original TS, b) pattern TS.

As a distance between the TS of patterns shown in Fig. 2b the following measure was used:

$$d(x,y) = \sqrt{\sum_{i=1}^{m} \alpha(t_{xi} - t_{yi})^2 + \beta(q_{xi} - q_{yi})^2}\,, \tag{1}$$

where m is a number of patterns in a considered class of TS of patterns x and y, i is an index of the current pattern, t and q are time points and intensity of patterns respectively, α and β are normalizing parameters.

The forecasted values are extracted from the most similar pattern following that of the goal. The location of the value following the found pattern is extended on the location of a forecasted value following the goal pattern. Then the slope value directly following this found subsequence is used for calculation of forecasted value following the goal pattern. We can forecast both the whole patterns following the goal one, or applying this method sequentially, forecast values in several sequential time points.

2.2 Combined Forecasting

Since each forecasting method exhibits strengths and shortcomings that are time series dependant, collaboration among heterogeneous methods is needed to bring forth their benefits, so that DM process can be brought to higher ground. In this paper this collaboration refers to the collaboration among several agents assigned to the classification of the patterns in the perceptual forecasting methods described above. Towards this end, each agent A_r is responsible for pattern identification and the results from multiple agents are combined.

When multiple agents work in parallel, a conflict resolution strategy is needed to arbitrate among the contradictory decisions generated by them so that one consolidated solution can be generated. The problem of decision fusion for multiple agents has been considered in the literature, but most of the authors concentrated on the combined classification [14]. In this paper, we extend the model to combined forecasting based on a Bayesian approach [15], [20]. This approach differs considerably from that of collaborative forecasting [21].

Suppose we have C_j , $j = [1,...,J]$ class patterns identified during TS analysis. In general, the predictions from each forecasting agent A_r can be recorded in a confusion matrix (CM) as follows:

$$CM^r = \begin{pmatrix} n_{11}^r & n_{12}^r & \cdots & n_{1(J+1)}^r \\ n_{21}^r & n_{22}^r & \cdots & n_{2(J+1)}^r \\ \vdots & \vdots & & \vdots \\ n_{J1}^r & n_{J2}^r & \cdots & n_{J(J+1)}^r \end{pmatrix}, \tag{2}$$

where n_{ij}^r, $i = [1,...,J]$ and $j = [1,...,J+1]$ indicates the number of samples belonging to class C_i that have been assigned to class C_j by Agent A_r. The diagonal elements of CM^r are then the correct predictions from A_r. The confusion matrix is computed for each method during the offline training stage. The information in CM^r is used in real-time as follows: the conditional probability that implies $\boldsymbol{x} \in C_i$ given the evidence $e_r(x) = j$ from agent A_r can be computed as [20]:

$$P(x \in C_i | e_r(x) = j) = \frac{n_{ij}^r}{n_{\bullet j}^r} = \frac{n_{ij}^r}{\sum_{i=1}^{J} n_{ij}^r}, \quad i \in [1, ..., J]. \tag{3}$$

where $P(x \in C_i | e_r(x) = j)$ can be considered the confidence of a classifier regarding the assignment of sample $\boldsymbol{x}$ to class C_i. Each agent $r \in [1, R]$ has its own confusion matrix, CM^r, and R evidences are produced in real-time. Each agent expresses its predictions supporting the proposition that $\boldsymbol{x} \in C_j$ in the form of conditional probability. The combined probability PE that supports $\boldsymbol{x} \in C_i$ from the collaboration among the agents can then be written as:

$$P(x \in C_i | A_r(x) = j_r, r = 1, ..., R) = \frac{\prod_{r=1}^{R} P(x \in C_i | A_r(x) = j_r)}{\sum_{i=1}^{J} \prod_{r=1}^{R} P(x \in C_i | A_r(x) = j_r)}. \tag{4}$$

Based on (4), a sample $\boldsymbol{x}$ is classified into class j depending on the combined conditional probability. The class C_i with the highest PE can be selected as the optimal combined prediction.

2.3 Predictive Model Using MLP Neural Network

Once the values of the forecasting variables are obtained, they are used as inputs to the predictive model of the network utilization level variable (or other network event). The predictive model used in this step was chose by an Intelligent Agent thanks to its higher prediction ability with respect to other predictive models such as Decision Trees.

The neural network architecture is a multilayer perceptron architecture (Fig. 3). In the case study presented in section 4, we used a multilayer perceptron architecture

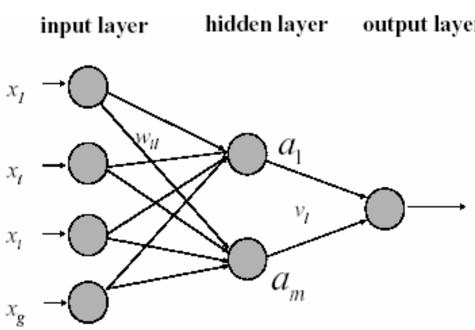


Fig. 3. MLP architecture

with seven inputs (independent variables), seven elements of the hidden layer and a network link utilization as an output variable (dependent variable). This architecture uses a linear combination function in the hidden and output layers and uses a sigmoid activation function in the hidden layer.

After being normalized to [0,1], input signals at the j–th time instant from the input nodes are multiplied by proper weights w_{il}, corresponding to connections between input neurons, from which the signal has been dispatched, and neurons in the hidden layer. In each of the m hidden nodes, the weighted sum of all the incoming signals and threshold values (w_{0l}) is computed and then transformed by – in case of this study – the logistic function, giving the value of a_l dispatched by l-th neuron:

$$a_l = \left[1 + \exp(-w_{0l} - \sum_{i=1}^{g} w_{il} x_i \right]^{-1} \quad (5)$$

Afterwards, the signals a_l, multiplied by proper weights v_l, are transferred to the neuron of the third layer. In this final stage, the new weighted sum is computed

$$y_{j+T} = v_0 + \sum_{l=1}^{m} a_l v_l \quad (6)$$

and after de-normalization of the output, the forecasted value y_{j+T} is determined.

Another prediction models such as Decision Tree or Linear Regression models can also be used at this stage as we above-mentioned. Once other techniques are implemented, the approach to combined prediction similar to that described in Section 2.2 can be applied.

3 Multi-agent Framework

The theoretical framework from Fig. 1 can be represented as a business process scheme using components described in the previous section as shown in Fig. 4.

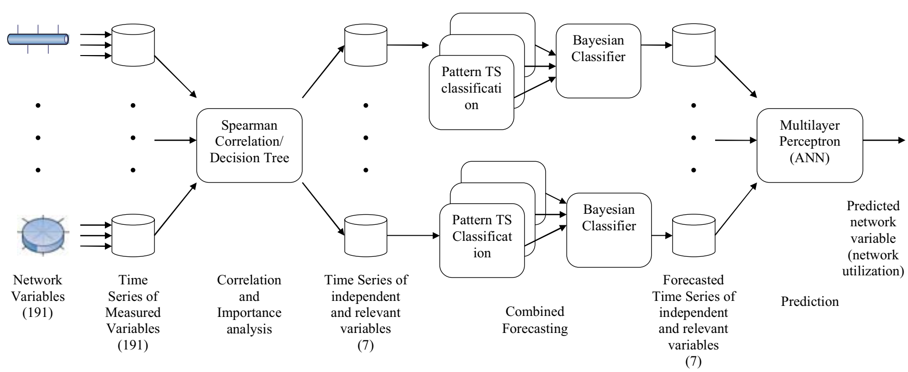


Fig. 4. Business process scheme of the telecommunication network data mining

In order to implement the scheme from Fig. 4, we developed a multi-agent framework. This framework uses the problem solving components both self-made and from the shelf implemented as Web and agent services along with the agents' ability to encapsulate different decision making techniques. Agents were developed using CAPNET agent platform [22]. In order to enable the integration with Web services, the model of agent and Web service integration implemented within the CAPNET is used [23]. The business process model developed for the orchestration of the Web and agent services follows the DM process described above according to the implementation scheme shown in Fig. 5, where each of agents addresses a different aspect of the DMP. The collaboration among the agents makes the framework capable of providing multi-faceted automation for DMP of telecommunication data.

Now we discuss the components of the framework in more details. The first step in the process is the network data collection from the subnets composing the particular network topology. Since the log files always contain huge amount of data (e.g. typical net log with the main parameters of a medium size subnet for one hour can measure several Gb), data collection agents are located in the monitoring nodes, extracting the set of network variables using business logic defined as extraction rules. These variables are summarized in Table 1 described in Section 4. CAPNET Expert System Shell component is used as inference engine.

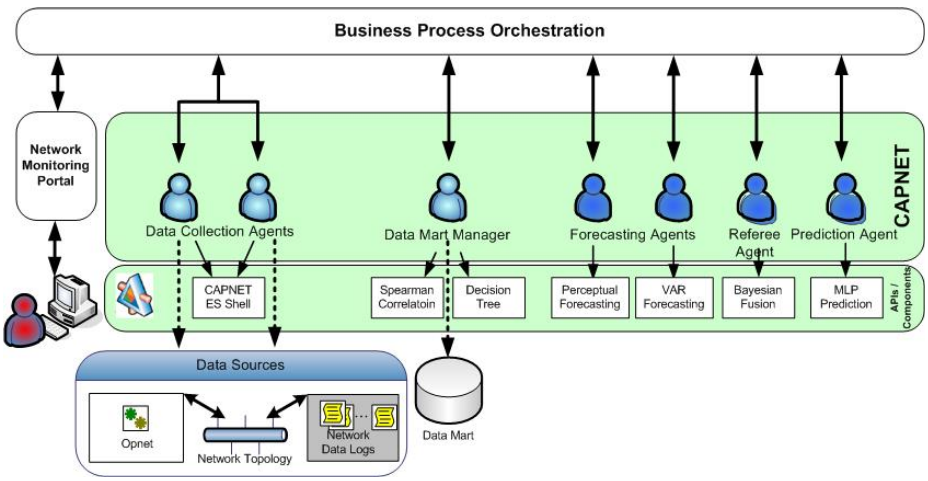


Fig. 5. DM process implementation scheme

Once data filtering and cleaning is finished, data collection agents send the information to the Data Mart manager agent for data integration and storing in the Network Data Mart. Correlation and importance analysis methods are implemented as Web Services, their function is to eliminate redundant and irrelevant variables from the Network Data Mart. By now, only Spearman Correlation and Decision Tree methods are implemented for the discrimination of redundant and irrelevant variables respectively. Once more methods are implemented as components, the agent can decide the best method (service) to use.

The next step is the forecasting of network variables. As described above, perceptual and ANN forecasting methods are implemented. They are invoked by the forecasting agents. In the case of perceptual forecasting, two agents are assigned to each TS collected on the daily basis, both using the method described in Section 2.1. One of them as a proximity measure uses the distance to the closest pattern of the cluster, while another one uses the distance to the average pattern of the cluster. They provide their forecasted patterns to the referee agent generating combined forecasting variables to use in each particular analysis as described in Section 2.2. As a result of the forecasting, the predicted time series for network parameters for the next time period are generated and stored in the Data Mart by the manager agent. At the final step of the process, these data are used for the prediction of the link utilization variable for the traffic prediction in network links by the MLP agent.

This process is dynamic and iterative. It means that for each time period $t+d$, both training and evaluation are applied to the TS data for a time period preceding the predicted one. This way, each time the prediction is made using the most recent data taking into account the typical fluctuations of the network parameters during the day. Moreover, each time the best model is selected as a product of the negotiation among agents.

This process is illustrated in the following section by the case study example. The only difference with the described process is that we use Opnet Modeler to generate log files instead of real network monitoring parameters.

4 Data Network Scenario Description and Experimental Results

We decided to create a network scenario using Opnet Modeler. This tool was used to generate the statistics and values of the network data variables. However, the same network data variables can be obtained from network data logs by using Intelligent Agents. The sequences of measurements of these variables represent the times series based on Data Network statistics. Opnet let us generate the network traffic and several kinds of services and applications in order to create several network traffic load levels.

4.1 Case Study

We propose a partial network topology from a Mexican retail enterprise. This enterprise has several branches distributed around the country. In this scenario the information and services that gives the enterprise, are requested from a particularly branch, in this case Mexico City.

Our network topology has four subnets. The network model is shown in Fig. 6, which has followed a hierarchical design where each of the branches is represented by a subnet. Opnet organizes the statistics in three groups: global, node and link statistics. The following statistics were chosen according to the objectives of this work (Table 1).

Table 1. Network statistics

Global statistics	Data Base, Email, Ftp, Ethernet and http
Node statistics	Client DB, Client Email, Client Ftp, Client http, CPU, Ethernet Channel, Ethernet, LAN, Server DB, Server Email, Server Ftp, Server http, Server performance and Switch.
Link statistics	Queuing delay, throughput and utilization.

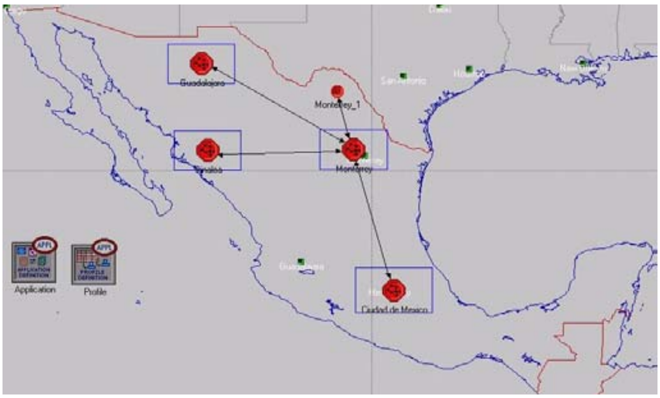


Fig. 6. Network scenario

There are some variables that can define the network performance such as link utilization, workload of the servers, network delay, response time of the applications, etc. However, the utilization variable was defined as the target because we want to know timely when a network link is overloaded.

There are two important properties to define in the simulation: duration and values per statistic. The duration of the simulation is limited to the number of events that can be used, and as the number of events is increased according to the traffic load in the network. We established duration of 500 minutes and 5000 values per statistic.

4.2 Simulation Results

Opnet lets to export the statistic data of the simulation. These statistics represent the measurements of the variables that are the components of the times series based on Data Network statistics. According to the statistics selected and the number of devices used in the network we got 191 statistics, each of them representing a network variable. These time series are preprocessed like explained in Section 2. At the first stage, we got 22 irredundant network variables (see Table 1) as a result of the correlation analysis of the 191 variables that integrated the TS. Further on, seven variables were selected for predicting the link utilization variable (Fig. 4).

The 80 % of the data was used to train the model and the 20 % for its validation. Such random partitioning is not appropriate for TS data since the order of the samples cannot be changed. In order to preserve the chronological order of the samples in the training and validation, we do the partitioning in the SAS BASE code node.

The input variables of the VAR component are the logarithms of the original variables. The relation among the variables is not always linear, so the inputs should be standardized for correct operation of the model. To train the VAR model the inputs are the lagged values of the targets, so if we want to forecast the input variable values at the time *t*, the input values have to be its logarithm in the time *t*-1.

The results of the forecasting comparing different models are shown in Table 2 for two simulation runs with different generated statistics. We show the average forecasting results for all seven variables. As it can be seen, the Bayesian combined forecasting obtained slightly better results then the VAR model.

Table 2. Performance evaluation of each forecasting agent

Forecasting agent	Recognition rate		
	Simulation run 1	Simulation run 2	Average
VAR forecasting model	81.34	83.56	82.45
Distance to the closest pattern of the cluster	79.15	75.16	77.155
Distance to the average pattern of the cluster	82.26	80.01	81.135
Bayesian Combination	84.13	82.78	83.455

4.3 Predictive Model Results

In Fig. 7, we show a comparison of the link network utilization level obtained in the first simulation run and the prediction link network utilization of the ANN model. Besides, we can notice that the neural network model is well suited to anticipate high utilization in order to avoid important network failures.

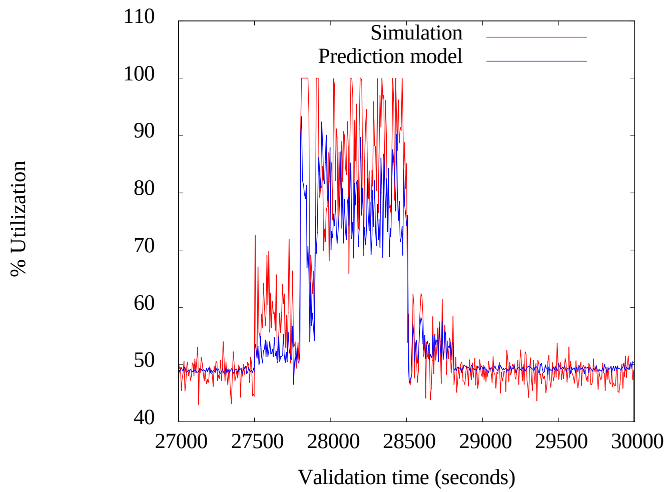


Fig. 7. Prediction of the Utilization level of the network link

For the implementation of the prototype the following tools were used:

- MS .Net Framework 1.1.
- MS SQL Server 2000.
- MS IIS Web server used as WS container for CAPNET.
- CAPNET Agent Platform with CAPNET Expert System Shell as the inference component for Data Mart composition agent.
- PerceptMiner Perceptual Forecasting Component.
- SAS export components for VAR forecasting and MLP prediction models.
- OPNET Modeler.

5 Conclusions

In this research work, we presented a theoretical framework showing the importance of the use of Intelligent Agents to analyze the big amount of data available in Telecommunications Networks in real time. This with the aim of obtaining knowledge and intelligence needed to attend network management tasks such as the traffic prediction in network links. Bayesian approach used to combine different forecasting methods takes advantage of collaboration among agents. Through this, the strengths of the different methods can be integrated while the drawbacks of the individual methods diminished through collaboration. The decomposition of the DMP of Telecommunications Networks data along with the multi-agent implementation framework permit applying the DM techniques in real time taking advantage of on-line data extraction, automatic learning and higher degree of parallelism. Another key benefit of the multi-agent framework is that it enables efficient computational integration of complex components.

The work is in progress; the first prototype version with a limited number of competing models has been developed so far and is reported in this paper. Though we use in the paper Opnet Modeler for simulation of the network parameters, multi-agent infrastructure is transparent for using data collection agents directly in the subnets in real-life scenarios. At this moment, we are working on more complex network scenarios and the implementation of rules extraction algorithms to facilitate the explanation of rules generated from the neural networks models. Another direction of current efforts is the implementation of another forecasting/prediction methods. Finally, the DM process workflow using Agent services and Web services integration capabilities of the CAPNET agent platform is under development.

Acknowledgments. Partial support for this research work has been provided by the SAS Institute.

References

1. Weiss, G. M.: Predicting Telecommunication Equipment Failures from Sequences of Network Alarms. In W. Kloesgen and J.Zytkow (eds.): Handbook of Knowledge Discovery and Data Mining, Oxford University Press (2001) 891–896

2. Garofalakis, M. and Rastogi, R.: Data Mining Meets Network Management: The NEMESIS Project. In ACM SIGMOD Int'l Workshop on Research Issues in Data Mining and Knowledge Discovery, May (2001)
3. Berry, M. and Linoff, G.S.: Data Mining Techniques, Second edition, Wiley (2004).
4. Weiss, G. M.: Data Mining in Telecommunications. In O. Maimon and L. Rokach (eds.): Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, Kluwer Academic Publishers, (2005) 1189–1201
5. Lee, W., Stolfo, S., Chan, P., Eskin, E., Fan, W., Miller, M., Hershkop, Sh. and Zhang, J.: Real Time Data Mining-based Intrusion Detection, The 2001 DARPA Information Survivability Conference and Exposition (DISCEX II), Anaheim, CA, June (2001)
6. Hatonen, K., Klemettinen, M., Mannila, H., Ronkainen, P. and Toivonen, H.: Knowledge discovery from telecommunication network alarm databases. In Proc. of the 12th Int. Conf. on Data Engineering, New Orleans, Louisiana, IEEE Computer Society (1996) 115–122
7. Klemettinen, M.: A Knowledge Discovery Methodology for Telecommunication Network Alarm Databases, PhD thesis, University of Helsinki, Finland (1999)
8. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In Proc. of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. San Diego, CA. June (2003)
9. Last, M., Klein, Y. and Kandel, A.: Knowledge discovery in time series databases, IEEE Trans. SMC, Part B, 31(1) 160–169 (2001)
10. Keogh, E. Lin J. and Fu A.: HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. In Proc. of the 5th IEEE International Conference on Data Mining (ICDM 2005), Houston, Texas, Nov 27-30 (2005) 226–233
11. Batyrshin, I. and Sheremetov, L.: Perception based time series data mining with MAP transform. In Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence, Vol. 3789. Springer Verlag (2005) 514–523
12. Weiss, G.: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press (2000)
13. Rocha-Mier L.E., Amy B. and Pellegrini, C.: Neural Networks for learning in a Collective Intelligence: Internet packet routing application, The 3rd Int. Conf. on Cognitive Science, Beijing, China, August (2001)
14. Gorodetsky, V., Karsaev, O., Kotenko, I. and Samoilov V.: Multi-Agent Information Fusion: Methodology, Architecture and Software Tool for Learning of Object and Situation Assessment. Proc. of the 7-th International Conference Fusion-04, Stockholm, Sweden, June 28-July 2 (2004) 346–353
15. Ng Y. S. and Srinivasan R.: Multi-agent Framework for Fault Detection & Diagnosis in Transient Operations. In V. Plesu and P.S. Agachi (eds.): Proc. of the 17th European Symposium on Computer Aided Process Engineering, Bucharest, Romania, 27-30 May, Elsevier B.V. (2007)
16. Samtani, S., Mohania, M., Kumar V. and Kambayashi Y.: Recent Advances and Research Problems in Data Warehousing. Lecture Notes in Computer Science, Vol. 1552. Springer Verlag (2004) 81–92
17. Frank, R.J., Davey N. and Hunt, S.P.: Time Series Prediction and Neural Networks. J. of Intelligent and Robotic Systems, Springer Verlag, 31(1-3), May (2001) 91–103
18. Han J. and Kamber, M.: Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers (2000)
19. Sims, C.A.: Macroeconomics and Reality, Econometrica, Vol. 48 (1980)

20. Xu, L., Krzyzak, A., Suen, C.Y.: Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Transactions on Systems, Man, and Cybernatics 22(3): 418–435 (1992)
21. Hassan, U. and Soh, B.: Using adaptive web service multi agents for collaborative forecasting and countering bullwhip effect in SCM. In Proc. of the Int. Conf. Integration of Knowledge Intensive Multi-Agent Systems, April 18-21 (2005) 567–572
22. Contreras, M., Germán, E., Chi M. and Sheremetov, L.: Design and implementation of a FIPA compliant Agent Platform in .NET. J. of Object Technology, ETH Zurich, 3(3), March-April (2004)
23. Sheremetov, L. and Contreras, M.: Industrial Application Integration Using Agent-Enabled SSOA: CAPNET Case Study. In: W. Shen (ed.): Information Technologies for Balanced Manufacturing Systems, IFIP Series, Springer Verlag (2006) 109–118

Architecture of Typical Sensor Agent for Learning and Classification Network

Vladimir Samoylov

SPIIRAS, 39, 14-th Liniya, St. Petersburg, 199178, Russia
samovl@mail.iias.spb.su

Abstract. Distributed decision making and learning procedures operate with distributed data sources. Intelligent access to distributed data sources is one of the important requirements to any distributed learning and decision making systems. In many such applications, input data arrive from spatially distributed sensors structured in sensor networks. An increasing interest to what is called *intelligent sensors* for which agent-based technology seems to be rather attractive implementation paradigm forms a marked trend in the sensor network area. Indeed, the latter provides a natural mapping from the set of nodes of intelligent sensor network to the set of collaborating agents and, at the same time, enriches the network nodes by the methods of collaboration developed under multi-agent approach and provides the network nodes with the capability to communicate in terms of high level language. The paper proposes a typical (reusable) architecture of the intelligent sensor agents as well as typical protocols supporting interaction of sensors with other software components of agent-based network intended for distributed learning and classification. The main solutions proposed in the paper are demonstrated via prototyping of two case studies.

1 Introduction: Data Sources, Data Sensors and Scalability

Distributed learning and decision making procedures operate with distributed data sources. In many above applications, input data result from processing of data perceived by many spatially distributed sensors structured in a sensor network. Continuously growing interest to what is called intelligent sensors is a marked trend, in the sensor network area. A sensor is meant as intelligent if some important functionalities are performed by the sensors itself, i.e. by the software installed on the sensor. Such sensors produce not only filtered but also semantically enriched information, thus, reducing communication overhead and upper level processor overload.

Other current trend is to provide sensors with a capability to interact with their neighbors, thus, taking on some functions of distributed data processing while interacting without use of a central processor. Such architecture assumes Peer-to-Peer (P2P) interaction of sensor-based software. Such an architecture seems prospective as providing for scalability, fault tolerance, supporting transient population of sensors and other data sources, preserving privacy of data when necessary, etc.

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 153–164, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Multi-agent technology looks like an attractive framework for software implementation of intelligent sensor networks. Indeed, it provides a natural mapping from the intelligent sensor network set of nodes to the network of agents and, at the same time, enriches the network nodes with the methods of collaboration developed by multi-agent approach and by the capability to communicate in terms of high level language.

On the other hand, intelligent sensor network put forward some new problems. If this network is intended for solving a distributed decision making task, e.g., distributed classification, the transient set of sensor network nodes should lead to the transient set of solvers and, therefore, to the necessity to re-train both local and centralized solvers when new data source/sensor enters or exits the network. Although the last issue as a whole is beyond the paper scope, its indirect impact on intelligent sensor network design is in the paper focus. More exactly, transient population of the sensor network nodes leads to the necessity of embedding new sensors when "embedding" is understood not only in purely physical sense, but mainly means embedding the sensor-based software, its adaptation to the existing intelligent sensor network structure in many respects, e.g., to networking, neighbors detection, service advertising and search, task-dependent coalition formation, etc.

Solution of the aforementioned problems simplifies if architecture of the network node representing intelligent sensor is reusable. This architecture is composed of the sensor(s) itself and agent-based software supporting interaction between sensor and data source, source data processing and interaction of the network node with other entities of the network intended for distributed learning and decision making. Data source processing software, as a rule, is strongly application dependent whereas the sensor(s) itself and agent-based software supporting interaction of intelligent sensor(s) with outside world are less dependent and thus may be reused in various applications. Development of typical architecture for intelligent sensor as well as typical protocols of its interaction with outside world is the subjects of the paper.

Thus, the first objective of the presented research is to develop the typical architecture of the software component which takes the raw sensor data as input and transforms them into feature space. This component is called below a sensor agent. A local, or, what is the same, base classifier utilizes the data transformed by the sensor agent.

The second objective is a development of typical protocols responsible for collaboration of the sensor agent with other nodes of the network. The first subtask in this collaboration is a hierarchical combining of the decisions produced by all or only some base classifiers during normal operation. The second subtask is collaboration of the network nodes in training of meta-procedures responsible for decision combining. In both cases the collaboration is performed via weakly application dependent protocols.

The here described research is aimed at the development of typical sensor agent architecture and protocols supporting interaction of intelligent sensor agent with other components of classification network in distributed classification and learning modes. Both aforementioned components of the intelligent sensor agent are specified using state machine formal framework implementing basic functionalities (behavior) of agent–sensor and its interactions with outside world in various scenarios. The developed reusable solutions are demonstrated through two case studies. The first one is agent-based classification network intended for detection of intrusions in computer network using multiple heterogeneous data sources resulting from preprocessing of raw traffic, and the second one is detection of ground objects based on infrared images perceived by airborne surveillance system.

The rest of the paper is organized as follows. Section 2 describes the main peculiarities of the target application domains of typical sensor agent, i.e. distributed decision making and distributed learning. Section 3 presents basic use cases for typical intelligent sensor agent. Section 4 considers formal specification of the typical intelligent agent–sensor *architecture* and the protocols. Section 5 demonstrates the implementations of typical sensor agent within two case studies. Conclusion summarizes the paper results and outlines perspectives.

2 Typical Architectures of Distributed Classification Systems

Distributed decision making based of multiple data sources as well as distributed learning of combining the decisions produced by source-based classifiers conventionally uses a hierarchical architecture. The decision combing process is here built as two–level system. At the lower level, each particular classifier produces decision using data available from a local data source, at that, several classifiers operating with the same data may be used. As a rule, data of particular source are unavailable to the classifiers working with other data source, e.g., due to privacy issue. At the second upper level the decisions of multiple classifiers are combined thus producing decisions implicitly using the all data available to the system. Additional advantage of this architecture is that it reduces the complexity.

Other approach considers a use of one–level architecture with the horizontal interaction of the classifiers. The latter are considered as peer classifiers responsible for both local data source–based classification and meta-level classification assuming involvement of the data and/or decisions produced by some other local classifiers. Such an architecture is built on P2P (serverless) basis. Situations when a classifier requests for information produced by other classifiers depend on applications[1]. While having this information received the agent–requestor somehow combines it with its own information and makes decision itself.

The P2P architecture of decision combing and learning of decision combining is so far a weakly investigated problem, although it is considered as an approach of the future [4]. Indeed this assessment is argued by the rapidly increasing area of sensor network applications, in particular, critical applications dealing with spatially distributed monitoring of large regions intended for prediction of force majeure situations like tsunami, flood, earthquake, etc. In many applications such cooperating sensor network assumes operation in "alarm detection" mode and the essence of cooperation is in combining the alarms produced by various sensor–based agents to recover the whole picture of the situation within the monitored region.

---

[1] To optimize search for relevant information, learning procedure has to be used in order to re-configure the agent communication structure, but this aspect is out of the paper scope.

3 Generic Use Cases of Sensor-Based Distributed Decision Making

This section outlines a typical use cases of sensor operation within both hierarchical and P2P architectures. Fig. 1 presents typical variants of sensor usage for distributed learning and classification systems in UML 2.0 notation. Let us note that the diagram represents only use cases involving sensors and associated data sources.

In the use case corresponding the distributed learning, the *Learning procedure* is the central one referencing to *Data Source* intended for *Data Preparation*. This

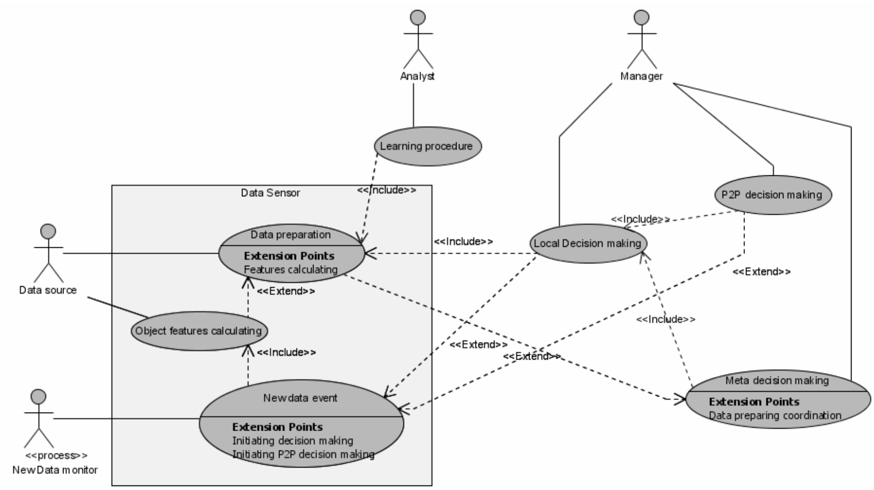


Fig. 1. Typical use cases of data sensor in learning and decision making

process supposes data sampling using certain criteria. In some cases it is necessary to compute additional attributes that are not explicitly stored in local data source. In Fig. 1 this procedure is called *Object Features Calculating*.

Local Decision Making is the basic procedure of any distributed decision making system. It produces classification using data formed by *Data Preparation procedure*. In P2P and hierarchical distributed classification systems, both, *P2P Decision Making* and *Meta Decision Making* processes use *Local Decision Making* functionality of local nodes involved in decision combining.

For hierarchical architecture, an extended variant of the *Meta Decision Making* process can take place. In it, management of preparation of all the necessary data is performed in centralized manner and carried out by *Data Preparing Coordination* functionality actually producing a command for local data sensors to prepare data needed for the decision generation and to send the result to the upper level.

In the aforementioned use cases, the sensor is a *passive entity* providing the data upon request. If the sensor is provided with the capability to get new data it performs *New Data Monitor* functionality. In Fig. 1 this use case is called *New Data Even*t. The latter extends the set of variants of hierarchical and P2P decision making systems. A distinguishing feature of these variants is that upon an arrival of new batch of data the sensor has to initiate calculation of the secondary features (using functionality *Object Feature Calculating*) and then to initiate the functionality *Local Decision Making* based on local data source.

4 Architecture and Protocols of Typical Intelligent Sensor Agent of Distributed Learning and Decision Making Systems

According to the Gaia methodology [5] used in this project for the agent-based classification networks design, the meta–level specification of a multi-agent system (MAS) architecture is realized in terms of roles executed by system agents and protocols (distributed algorithms) in whose performance the roles participate. If the roles are allocated to particular classes of agents one can tell about the protocols in which this or that agent class participates.

Typical sensor agent architecture assumes that there exist typical roles the sensor agents has to perform and typical protocols in whose performance the roles participate and the above mentioned architecture and protocols are weakly application dependent. Of course, "typical" does not mean that role or protocol is used in many different applications without any alterations. In fact, they are specialized to particular applications but some basic solutions concerning agent architecture and protocols remain reusable. Within the systems in question, "specialization" of typical protocol corresponds to specification of content of the messages which the participating sensor agents exchanged as well as specification of the ontology forming the terminology of the messages' contents. As for "role specialization", it assumes specification and implementation of the mechanisms used for data access and specification of particular data processing algorithms used by this or that protocol.

The proposed architecture of the typical sensor agent participating in typical use cases outlined in previous section (see also Fig. 1) is described below. These specifications are done in notation used in the MADK 3.0 [2], a software tool that was used in design and implementation of the software prototypes of MAS applications[2].

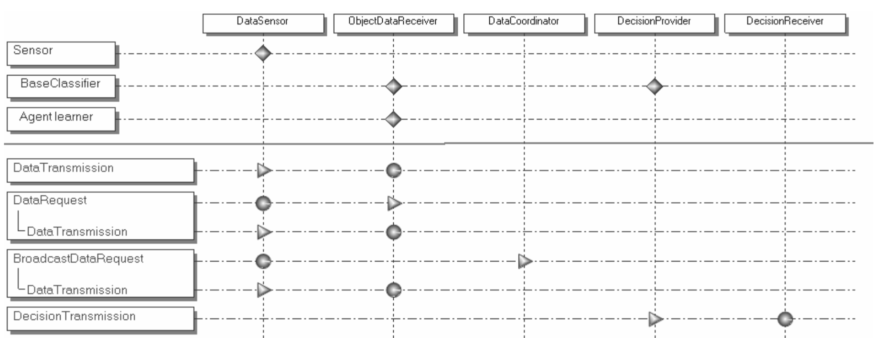


Fig. 2. Meta–model of typical sensor agent for distributed learning and classification system

[2] The author of this paper participated in development of the MASDK 3.0 software tool.

Interactions of typical agent classes, roles and the interaction protocols are shown in Fig. 2. According to the Gaia methodology [5] this figure presents the meta-model of an agent-based classification network in graphical notation used in MADK 3.0. In this figure typical roles are described in abstract terms in order not to associate them with a particular application of the class of interest.

In this meta–model the following typical roles of the distributed learning and classification MAS associated with operation on data sources are introduced:

DataSensor – the provider of information to be further used for classification. Information provided by it is a preprocessed raw physical sensor data represented mainly in terms of classification feature space. It may also provide other information about external world.

ObjectDataReceiver – the consumer of information provided by sensor agent, i.e. consumer of object features and other information produced by the sensor agent.

DecisionProvider – the source of decisions. While having input information received, this role produces classification, i.e. the label of the class to which the input object belongs.

DecisionReceiver – the consumer of decisions on the input object classes. As a rule, this role is assigned to the agent responsible for combining of several decisions produced by the agents performing the *DecisionProvider* role.

DataCoordinator – the coordinator of data circulating in classification network. This role is necessary in some special cases of hierarchical decision making systems.

Let us outline the roles interaction protocols used in distributed decision making systems while focusing on the protocols only involving sensor agents.

DataTransmission – the protocol intended for transmission of object features information from sensor agent to a data consumer. This protocol is used in case of a so-called "*active sensor agent*" which is driven by events arriving from external world. As a rule, active sensor agent is used in real-time systems.

DataRequest – the protocol intended for transmission of object features from sensor agent to data requester in case when the system but not sensor agent is "*active entity*".

BroadcastDataRequest – the protocol used for transmission of object features information from sensor agent in case of "active" hierarchical decision making system, containing special software component performing the coordinator role. In this case, after receiving the coordinator command, sensor agent computes object features and sends this information to other agents consuming it.

DecisionTransmission – the protocol intended for transmission of the decisions to the consumers of information of a kind.

In the developed meta–model of distributed decision making MAS sensor agent is denoted by identifier “*Sensor*". It performs the single role called *DataSensor* in Fig. 2. Specification of the typical architecture of the sensor agents is given in Fig. 3 in terms of "*liveness expressions*" [5] and state machines. Such notation is used in MASDK 3.0 software tool that has been utilized for formal specification of sensor agent architecture.

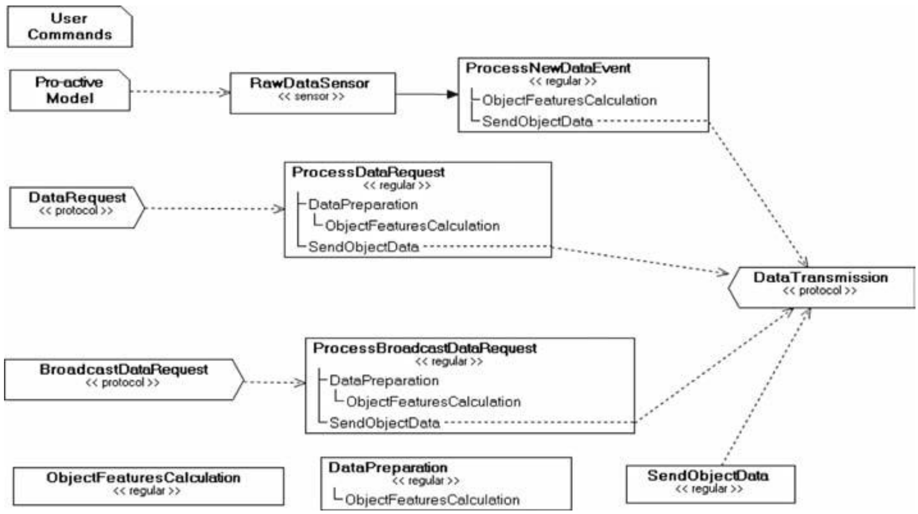
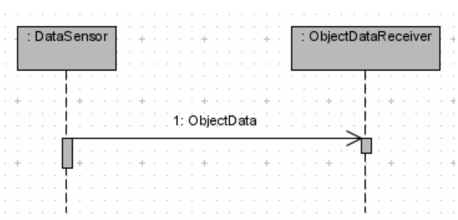


Fig. 3. Typical sensor agent architecture

For each particular MAS application the typical state machined depicted in Fig. 3 has to be specialized. This specialization concerns implementation of specific mechanisms of the access to data (*RowDataSensor*, *ProcessNewDataEvent*), preprocessing algorithm (*ObjectFeaturesCalculation*), implementation of "on request" data search mechanism (*DataPreparation*) as well as processing of the protocol–associated messages represented in terms of MAS application ontology (*ProcesDataRequest*, *ProcessBroadcastDataRequest*, *SendObjectData*). The aforementioned specialization has to be done for every protocol involving the sensor agent.

It is important to note that the sensor agents do not directly interact with each other. They interact through mediation of agents performing the *ObjectDataReceiver* and *DataCoordinator* roles. The latter roles use information from the sensor agents for local classifiers learning as well as for local classifier–based decision making. It is



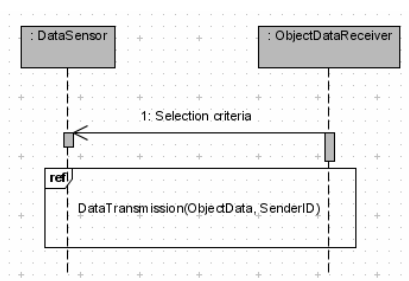


Fig. 4a. Data transmission protocol

Fig. 4b. Data request protocol

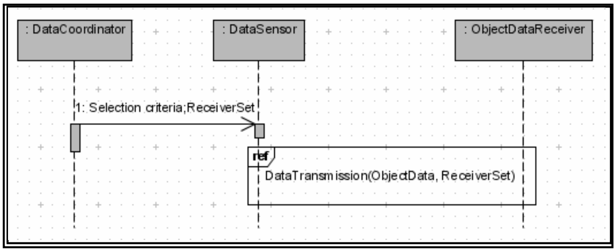


Fig. 5. Broadcast data request protocol

important to note that sensor's data may be consumed by several agents. For example, several agents–classifiers may produce their decisions using data of the same sensor agent as input though using different algorithms. In these cases, the typical protocols presented in Fig. 2 are used for interaction with sensor agents.

Figures 4.a, 4.b and 5 present more detailed specifications of typical protocols in which *DataSensor* role participates. In these figures the interaction diagrams are presented only in the part concerning the *DataSensor* role performed by sensor agents and also the roles the sensor agents directly interact with. Let us note that the protocols given in Fig. 4.a, 4.b and 5 are represented in terms of one of the standard languages used for protocol specification.

5 Case Studies Demonstrating Use of Typical Sensor Agent

Let us consider two case studies demonstrating usage of the developed typical sensor agent architecture. The first case study is multi-agent system intended for detection of intrusions in computer network based on traffic analysis. The second one solves the task of distributed detection of ground objects presented in infra red images. These case studies are used for the developed architecture validation. They also show the

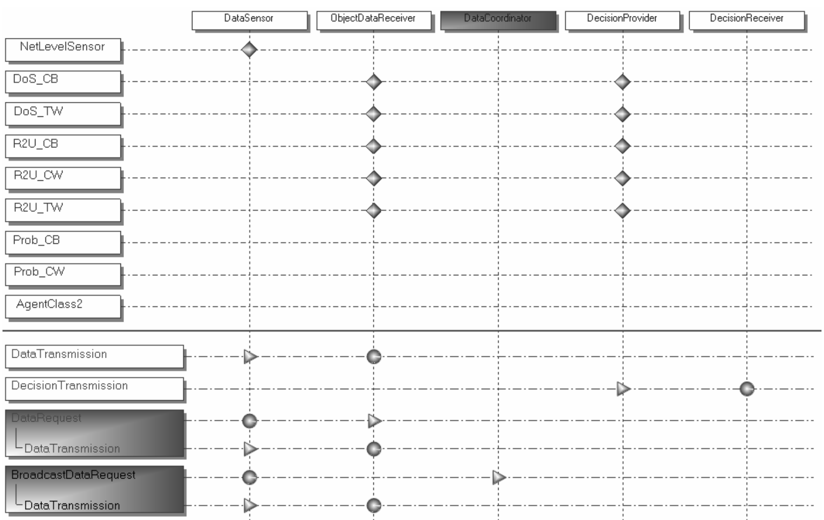


Fig. 6. Meta–model of typical data sensor agent in Intrusion detection system

common part of as well as difference between specialized sensor agent architectures used in different applications. The "common" part is exactly that one which can be reused in different applications as a ready solution applicable to distributed learning and classification MAS.

Let us consider the first case study that is intrusion detection system (IDS). Meta–model of this MAS is presented in Fig. 6. It contains single instance of the '*DataSensor*" role (like Fig. 2) and single agent class of the "*Sensor*" type.

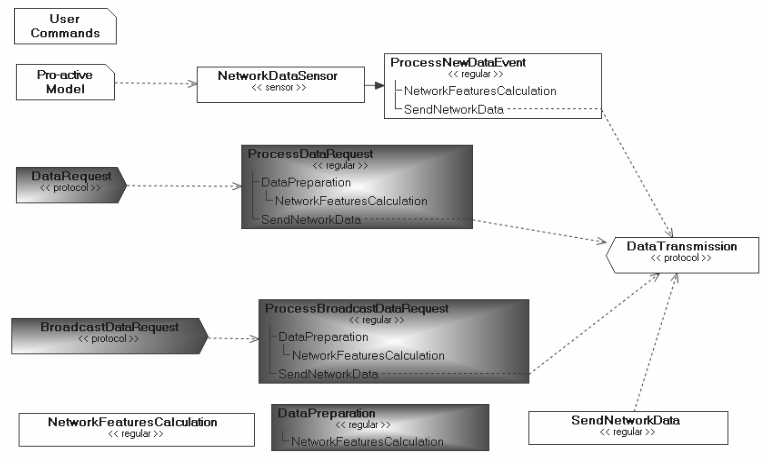


Fig. 7. Sensor agent architecture used in Intrusion detection system

Not going into details of the architecture represented by the meta–model given in Fig. 6, let us only note that the roles and protocols depicted in this figure by dark color are not used in IDS case study implementation. Architecture of the sensor agent of IDS specified in terms of state machines is presented in Fig. 7. Like in Fig. 6 the components not used in agent network implementing IDS are shown in this figure by dark color. Let us also note that IDS is built using "*active*" sensor agent option, because it is a real time system driven by the events of "external world" corresponding to the facts of arrival of traffic data new portion to be perceived by sensor agents.

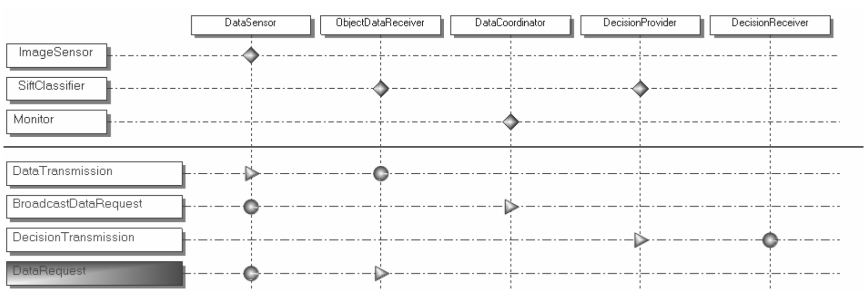


Fig. 8. Meta–model of typical data sensor agent in ground objects detection system

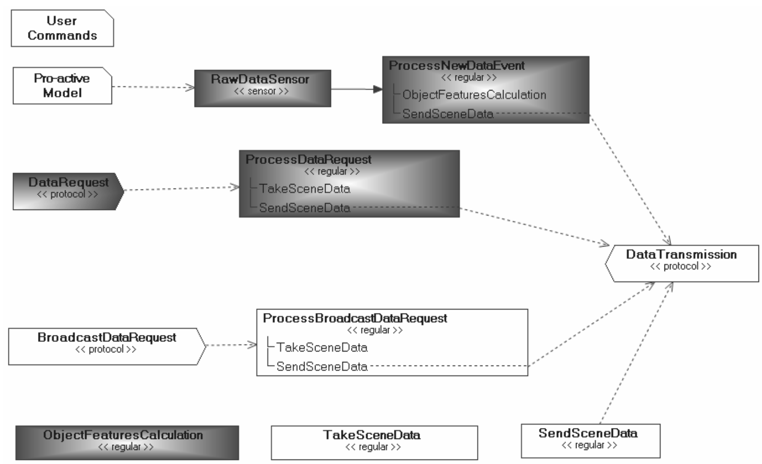


Fig. 9. Sensor agent architecture in ground objects detection system

The second case study, distributed ground object detection, is also formulated as distributed decision making task producing classification of the detected objects using data arriving from different sources. In this case study each sensor agent perceives the ground scene as infra red image extracts from it different features and therefore produces classification based on particular data source.

Additional peculiarity of the distributed ground object recognition system is that each classifier is trained to detect the object of a single class and only if the latter is observed from specific direction, i.e. either from the front, or from the back, or from the left or from the right. Decisions produced by particular classifiers are then combined at the upper level of hierarchical architecture. Fig. 8 and 9 present the meta–model of the ground object detection MAS and architecture of the sensor agent specified in terms of state machines respectively.

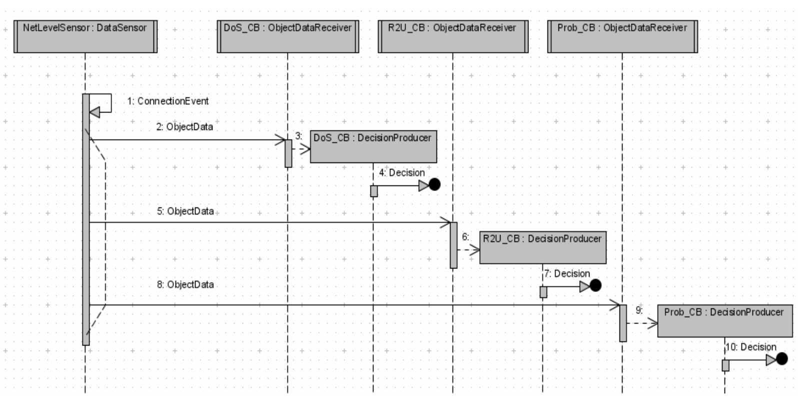


Fig. 10. Message sequence diagram of sensor agent in Intrusion detection system

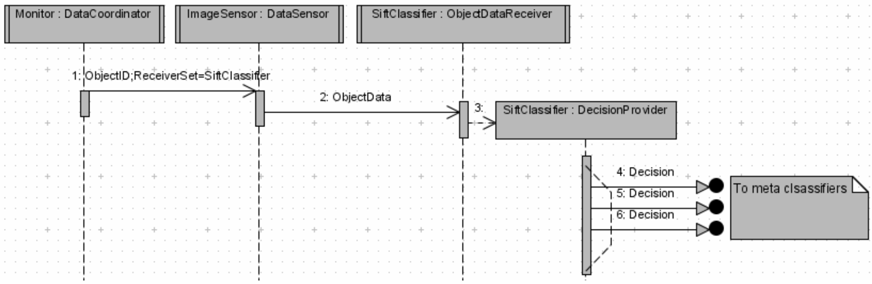


Fig. 11. Sequence diagram of the messages arriving to the sensor agent of the ground object detection system

Like above given figures, in Fig 8 and 9, the components of the MAS meta–model and sensor agent architecture which are not used in the case study in question are shown in dark color. Let us note that, the developed software prototype of the ground object detection system uses the options where the active entity is the system itself and not the sensor agent as in IDS MAS. Indeed, the data flow processes are controlled here by queries from the upper level.

Fig. 10 demonstrates the sequence of messages arrioving to the sensor agent of the network level initiated by *ConnectionEvent* event corresponding to the completion of a TCP connection (session of a user). This sequence is generated by the *DataTransmission* and *DecisionTransmission* protocols initiated one after another (see also meta–model presented in Fig. 2 indicating particular roles participating in performance of last protocols). Fig. 10 also presents specialization of typical interactions between the sensor agents in IDS system.

The similar diagram of interaction of the sensor agent of the ground object detection multi-agent system is shown in Fig 11. This version of a message sequence corresponds to systems with active coordinator. In this figure the role of *DataSensor* is performed by agent denoted by *ImageSensor*.

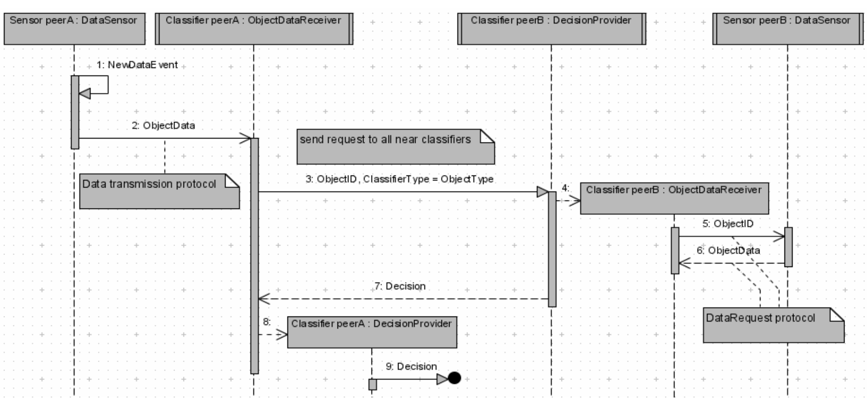


Fig. 12. Messages sequence diagram of sensor agent in agent-based P2P classification network intended for intrusion detection

6 Conclusion

The paper summarizes twofold research contribution. The first one proposes typical components of the sensor agent architecture demonstrated by two different versions of multi-agent applications from the distributed learning and distributed decision making scope. The second contribution concerns development of typical protocols' set supporting standard interactions of the sensor agent for various scenarios and options of distributed learning and decision making systems.

The future efforts will concern further development of the networks of agents intended for distributed learning and classification assuming that the network nodes interact on P2P basis, i.e. on the serverless basis. The limited experience accumulated in this respect proved that the proposed typical architecture of the sensor agent is also applicable to P2P classification networks. In particular, both case studies considered in the paper were also implemented as P2P agent-based classification network in which information exchange between agents was organized within service-oriented architecture. Fig 12 demonstrates diagram of interaction between sensor agents of the P2P agent-based classification network intended for detection of intrusions into computer network.

Acknowledgements. This research was supported by the Russian Foundation for Basic Research (project # 04–01-00494a, 2004-2006) and Program # 14 of the Presidium of the Russian Academy of Sciences (Project #236).

References

1. Gorodetsky, V., Serebryakov, S.: Methods and Algorithms of Collective Recognition: A Survey, Transactions of SPIIRAS, No. 3, vol. 1, St. Petersburg, Nauka Publishers (2006) 139–171 (in Russian)
2. Gorodetsky, V., Karsaev, O., Samoylov, V., Konushy, V., Mankov, E., Malyshev, A.: Multi-Agent System Development Kit, In R. Unland, M. Klusch, M. Calisti (Eds.), Software Agent-Based Applications, Platforms and Development Kits. Whitestein Publishers (2005)
3. Ortega, J., Coppel, M., Argamon, S.: Arbitraining Among Competing Classifiers Using Learned Referees", International Journal "Knowledge and Information Systems", No. 3 (2001) 470–490
4. Tate, A.: The "Helpful Environment": Geographically Dispersed Intelligent Agents That Collaborate", Special Issue of IEEE Intelligent Systems "Future of Artificial Intelligence", No. 3 (2006) 57–61
5. Wooldridge M., Jennings N.R., Kinny D.: The Gaia Methodology for Agent–Oriented Analysis and Design, Journal of Autonomous Agents and Multi-Agent Systems, No. 3, vol. 3 (2000) 285–312

Self-organizing Multi-agent Systems for Data Mining

Ichiro Satoh

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
ichiro@nii.ac.jp

Abstract. A framework for deploying and coordinating agents for data mining over a distributed system by using the notion of dynamics between components is presented. The key idea behind the framework is to provide agents with deployment policies for their deployment and coordinations as relations between them and other agents. As a result, a federation of distributed agents can migrate and coordinate over a distributed system in a self-organizing manner. This paper also presents a prototype implementation of the approach and its applications.

1 Introduction

Multi-agent system technology provides a powerful method for building and managing data mining in a distributed system. Agents autonomously exchange data with other agents and discovering their interesting data. To do this, appropriate agents need to be deployed at appropriate computers and interact with appropriate agents, which are running at the same or different computers. However, computers cannot initially be equipped with all agents, which may be wanted in future due to limitations in their individual computational capabilities. The scale and complexity of modern distributed systems impair our ability to deploy agents to appropriate computers using traditional approaches, such as those that are centralized and top-down. Furthermore, the structure of a distributed system may also be frequently changed by adding or removing computers and changing the network topology. The requirements of the applications tend to vary and changed dynamically. Multi-agent systems for distributed data mining must adapt to such changes without any centralized management mechanisms.

To solve these problems, we propose a framework for dynamically deploying and coordinating agents over a distributed system. It provides agents with their own relocation policies without the need of any global policies. As a result, it enables individual agents or a group of agents to migrate at appropriate computers over a network in a self-organizing manner without losing their previous coordination. This paper addresses the deployment and coordination of multiple agents for data mining over a distributed system instead of multiagent-based data mining techniques. We have presented earlier versions of the framework in this paper in our previous papers [12,13]. These previous versions aimed at providing a dynamic configuration mechanism for ubiquitous computing. We also presented a bio-inspired framework for deploying agents in our other previous papers [14,15]. The previous framework provides agents with their own deployment policies but it aimed at applying bio-inspired approaches, e.g., lamellipodia

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 165–177, 2007.
© Springer-Verlag Berlin Heidelberg 2007

in cells the movement of myxomycete, into mobile agents. These previous versions had no mechanism for coordinating agents, which was essential in multi-agents for data mining over a distributed system.

This paper describes our design goals (Section 2), the design of the framework, and a prototype implementation (Section 3). We also describe our experience with it (Section 4). We then briefly review related work (Section 5), provide a summary, and discuss some future issues (Section 6).

2 Basic Approach

A multiagent system for data mining is constructed as a federation of agents, which may be running at the same or at different computers. Each agent is autonomous and exchange data with other agents. To reduce communication traffic in a network, agents for data mining should be deployed at computers close to the producers of the data, i.e., sensors and database systems. When a data mining system is made up of multiple agents, the location of one may affect the others. For example, two agents are required to be at the same or nearby computers, when the agents frequently interact with each other.

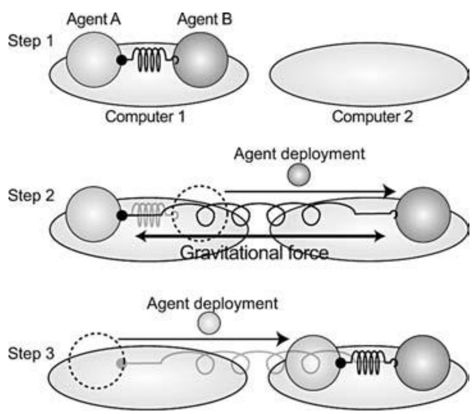


Fig. 1. Deployment policy

2.1 Inter-agent Communications

This framework introduces a mechanism into agents to possibly track moving targets and to interact with these. Most interactions between computing entities in object-oriented systems within a computer can be covered by three primitives: event passing, method invocation, and stream communication. Achieving syntactic and (partial) semantic transparency for remote interactions requires the use of proxy objects that have the same interfaces as the remote agents. Our framework also provide each agent with policies for inter-agent communications. These policies are activated when agents migrate at different computers. For example, if an agent has a policy for another agent, when the latter agent migrate to another location, the policy creates a clone of the latter at the latter's source location. The framework does not assume any existing agent communication language (ACL), but it can support various ACL representations, including XML-based formats, because it exchange arbitrary data representations between agents.

2.2 Self-deployment of Agents

Agents should be deployed at appropriate computers only while they are wanted. Computing devices have limited memory, secondary storage, and processing capacity. The deployment of agents for data mining depends on the locations of data that they need to discover and analyze as well as their applications. The framework therefore enables each agent to explicitly specify a policy for deploying agents. The current implementation provides four types of hooks. As you can see in Figure 1, the first enables an agent to follow another agent. An aggregation of agents, each with its own deployment policies, can change its structure and move over a distributed system in response to changes in the underlying system and the requirements of the application (Figure 2).

Remark

This framework was inspired by our earlier versions presented in previous papers [12,13]. The previous papers aimed at presenting the middleware for building and operating a large-scale system as a federation of one or more mobile agents like the framework presented here, but they addressed ubiquitous computing environments whose computers are heterogenous rather than large-scale distributed systems. The previous versions offered some of the gravitational relocation policies supported by this framework, but lacked any of the repulsive policies, which are essential in supporting load-balancing and fault-tolerant mechanisms. In fact, when many components are organized and deployed over a distributed system by only using the gravitational relocation policies, they tend to gather at several computers.

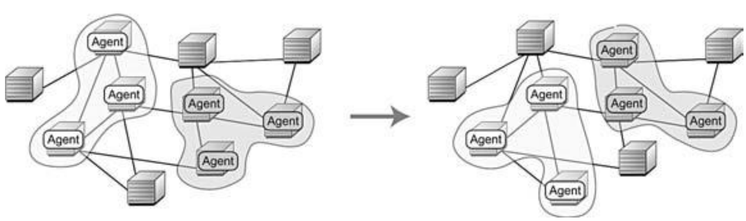


Fig. 2. Group migration

3 Design and Implementation

This framework consists of two parts: runtime systems and agents. Each agent in the current implementation is a collection of Java objects. The current implementation of the framework is general-purpose middleware for distributed systems and is constructed based on mobile agent technology.

3.1 Runtime System

Each runtime system is running on a computer and is responsible for executing and migrating agents to other computers. It establishes at most one TCP connection with each of its neighboring computers and exchanges control messages, agents, and inter-agent communications with these through the connection.[1] Figure 3 outlines the basic structure of a runtime system. Each agent in the current implementation is a collection of Java objects in the standard JAR file format and can migrate from computer to computer and duplicate itself by using mobile agent technology [11].[2] When an agent is transferred over the network, the agent runtime system on the sending side marshals the code of the agent and its state, e.g., instance variables in Java objects, into a bit-stream and then transfers them to the destination. The agent runtime system on the receiving side receives and unmarshals the bit-stream so that the agent can continue its execution at the destination.

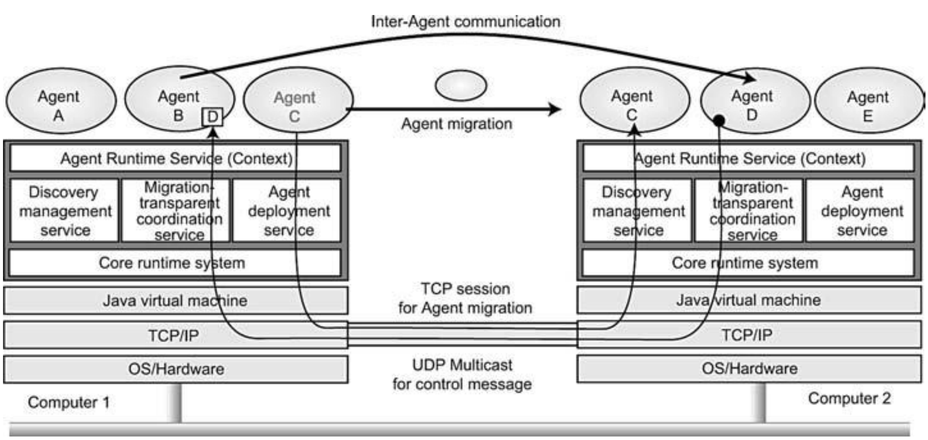


Fig. 3. Agent runtime system

3.2 Agent Model

Each agent runtime system governs all the agents inside it and maintains the life-cycle state of each agent. When the life-cycle state of an agent changes, e.g., when it is created, terminates, or migrates to another computer, the runtime system issues specific

[1] TCP connections are light-weight in comparison with other approaches, e.g., RMI.
[2] JavaBeans can easily be translated into agents in the framework.

events to the agent. This is because the agent may have to acquire various resources, e.g., files, windows, or sockets, or release ones it had previously acquired. The current implementation uses Java's object serialization package for marshaling agents. This package can save the content of instance variables in an agent program but does not support the capturing of the stack frames of threads. Consequently, runtime systems cannot serialize the execution states of any thread objects. Instead, when an agent is marshaled or unmarshaled, the runtime system propagates certain events to its agents instructing them to stop their active threads and then automatically stops and marshals them after a given period of time. Each agent must be an instance of a subclass of the `MAgent` class. Here, we will explain the programming interface characterizing the framework.

```
class MAgent extends MobileAgent implements Serializable {
  void go(URL url) throws NoSuchHostException { ... }
  void duplicate() throws IllegalAccessException { .. }
  setPolicy(ComponnetProfile cref,
    MigrationPolicy mpolicy) { ... }
  setTTL(int lifespan) { ... }
  void setAgentProfile(AgentProfile cpf) { ... }
  boolean isConformableHost(HostProfile hfs) { ... }
  void send(URL url, AgentID id, Message msg) throws
    NoSuchHostException, NoSuchAgentException, .. { .. }
  Object call(URL url, AgentID id, Message msg) throws
    NoSuchHostException, NoSuchAgentException, .. { .. }
  ....
}
```

An agent executes `go(URL url)` to move to the destination host specified as `url` by its runtime system, and `duplicate()` creates a copy of the agent, including its code and instance variables. The `setTTL()` specifies the life span, called time-to-live (TTL), of the agent. The span decrements TTL over time. When the TTL of an agent reaches zero, the agent automatically removes itself. Each agent can have more than one listener object that implements a specific listener interface to hook certain events issued before or after changes in its life-cycle state. That is, each agent host invokes the specified callback methods of its agents when the agents are created, destroyed, or migrate to another host.

3.3 Interagent Communication

The current implementation offers several communication policies for interagent interactions as follows:

- If an agent declares a *transparent* policy, agent, even when the agent migrates to another location, the messages that are sent to the agent are forwarded to the agent.
- If an agent declares a *non-transparent* policy, agent, even when the agent terminate or migrates to another location, the messages that are sent to the agent are disposed.
- If an agent declares a *relief* policy, when the agent terminates or migrates to another location, a clone of the agent is created at the source side computer and the messages that are sent to the agent are received by to the clone.

Each agent can one of the above policies and can have the transparent policy when it does not explicitly declare the others. The following policies support coordination between agents.

- If an agent declares a *forward* policy for another agent, when specified messages are sent to other agents, the messages are forwarded to the latter as well as the former.
- If an agent declares a *delegate* policy for another agent, when specified messages are send to the former, the messages are forwarded to the latter but not to the former.

The former policy is useful when two agents share the same information and the latter policy provides a master-slave relation between agents. The framework provides three interactions: publish/subscribe for asynchronous event passing, remote method invocation, and stream-based communication as well as message *forward* and *delegate* policies.

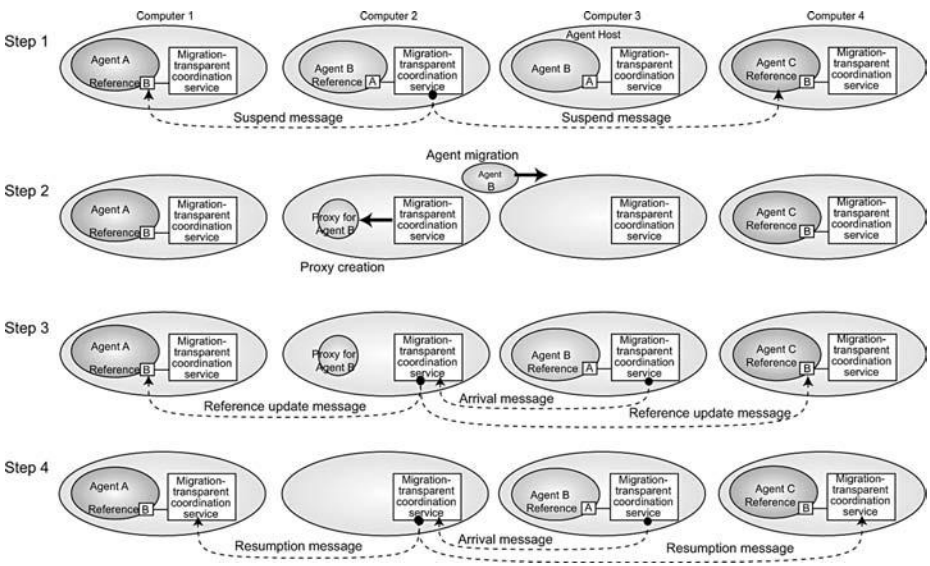


Fig. 4. Forwarding messages to migrated component

Each runtime system offers a remote method invocation (RMI) mechanism through a TCP connection. It is implemented independent of Java's RMI because Java's RMI lacks any reference updating mechanism for migrating agents. Each runtime system can maintain a database that stores pairs of identifiers of its connected agents and the network addresses of their current runtime systems. It also provides agents with references to the other agents of the application federation to which it belongs. As you can see in Figure 4, each reference enables the agent to interact with the agent that it specifies, even if the agents are on different computers or move to other computers.

1) When an agent request the current runtime system to migrate to another computer, the system searches its database for the network addresses of the runtime systems with the agents. It sends *suspend* messages to these systems to block any new uplinks from them to the migrating agent with the address of the destination.

2) If the moving agent contains references, the current runtime system sends the address of the destination to the runtime systems that run agents specified in the references so that they can update their databases.
3) After the agent arrives at its destination, it sends an *arrival* message with the network address of the destination to the departure runtime system.
4) When the departure system receives the arrival message, it sends `resumption` messages with the address of the destination to the runtime systems that may hold references to the moved agent.

When an agent begins to interact with a moving agent, the former can send messages to the source of the moving agent before the above basic algorithm is completed. To solve this problem, a migrating agent create and leave a proxy agent at the departure runtime system for the duration the algorithm takes to finish. The proxy agent receives uplinks from other runtime systems and forwards them to the moved agent. Since all agents do not have to be tracked for other agents to communicate with them, agents can leave proxy agents a long their trail under their control.

3.4 Deployment Policy

While an agent is running, it can declare its own deployment policy by invoking the `setPolicy` method of the `MAgent` class. Let us now explain four policies, called *deployment* policies (Figure 5).

– If one agent declares a *follow* policy for another agent, when the latter exists or migrates to a host, the former migrates to the latter's current or destination host.
– If an agent declares a *dispatch* policy for another agent, when the latter migrates to another host, a copy of the former is created and deployed at the latter's destination host.
– If an agent declares a *shift* policy for another agent, when the latter migrates to another host, the former migrates to the latter's source host.
– If an agent declares a *fill* policy for another agent, when the latter migrates to another host, a copy of the former is created and deployed at the latter's source host.

Figure 5 outlines four deployment policies. These policies are related to phenomena in biological processes. For example, a `follow` policy enables an agent to come near another agent. For example, when multiple agents declare a policy for a leader agent, they can swarm around it. A *shift* policy enables an agent to follow the movement of another agent. The former agent can track the latter as it moves. A `dispatch` policy enables an agent to stay in the current location and then deploy its clone at the destination of another moving agent. The framework is open to define policies as long as they are subclasses of the `MigrationPolicy` so that we can easily define new policies.

Agents duplicated by the dispatch or fill policy have this for their original agents. Each agent can specify a requirement that its destination host must satisfy by invoking `setAgentProfile()`, with the requirement specified as `cpf`, where it is defined in CC/PP (composite capability/preference profiles) form [17], which describes the capabilities of the agent host and the requirements of the agents. The class has a service method called `isConformableHost()`, which the agent uses to determine whether

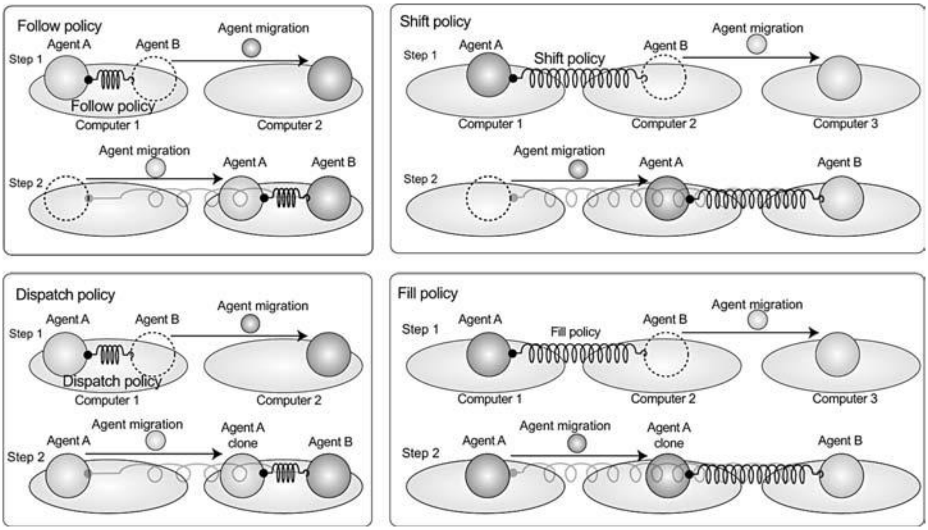


Fig. 5. Deployment policies

the capabilities of the agent host specified as an instance of the `HostProfile` class satisfy the requirements of the agent. Runtime systems transform the profiles into their corresponding LISP-like expressions and then evaluate them by using a LISP-based interpreter. When an agent migrates to the destination according to its policy, if the destination cannot satisfy the requirements of the agent, the runtime system recommends candidates that are hosts in the same network domain to the agent.

3.5 Agent Deployment Management

Agent deployment policies are managed by runtime systems without any centralized management servers. The framework permits each agent to dynamically declare at most one policy for at most another agent. Each runtime system periodically advertises its address to the other systems through UDP multicasting, and then these systems return their addresses and capabilities to the computer through a TCP channel.[3] Each agent registers its deployment policy to the current runtime system.

1) When an agent requests it current runtime system to migrate itself to another computer, the runtime system multicasts query messages to computers in the current sub-network (and neighboring sub-networks via gateways).
2) When another runtime system has agents specified in the policy of the visiting agent, it asks the destination-side computer about the computational resource of the destination-side computer if the policy is *follow* or *dispatch* or the source-side computer about the of the computer if the policy is *shift* or *fill*.

[3] We assume that the agents comprising an application are initially deployed to computers within a localized space smaller than the domain of a sub-network.

3) The runtime system receives information about the resource of the computer and then instructs the agent if the information satisfy the requirement of the agent and it instructs the specified agents to migrate to the destination-side (if the policy is *follow* or *dispatch*) or the source-side computer (if the policy is *shift* or *fill*).

Moreover, when the capabilities of a candidate destination do not satisfy all the requirements of the agent, the agent itself decides, on the basis of its own configuration policy, whether it will migrate itself to the destination and adapt itself to the destination's capabilities.

3.6 Security

The current implementation is a prototype system to dynamically deploy the agents presented in this paper. Nevertheless, it has several security mechanisms. For example, it can encrypt agents before migrating them over the network and it can then decrypt them after they arrive at their destinations. Moreover, since each agent is simply a programmable entity, it can explicitly encrypt its individual fields and migrate itself with these and its own cryptographic procedure. The Java virtual machine could explicitly restrict agents so that they could only access specified resources to protect computers from malicious agents. Although the current implementation cannot protect agents from malicious computers, the runtime system supports authentication mechanisms to migrate agents so that all runtime systems can only send agents to, and only receive from, trusted runtime systems.

3.7 Current Status

A prototype implementation of this framework was constructed with Sun's Java Developer Kit, version 1.4 or later version.[4] Although the current implementation was not constructed for performance, we evaluated the migration of two agents based on deployment policies. When an agent declares a follow, dispatch, shift, or fill policy for another, the cost of migrating the former or its clone to the destination or the source of the latter after the latter begins to migrate is 92 ms, 116 ms, 89 ms, 118 ms, or 136 ms if the policy is follow, dispatch, shift, fill, or exclusive, where the cost of agent migration between two computers over a TCP connection is 35 ms and the cost of duplicating an agent in a computer was less than 7 ms.[5] This experiment was done with three computers (Pentium M-1.8 GHz with Windows XP and JDK ver.5) connected through a Fast Ethernet network. Migrating agents included the cost of opening a TCP-transmission, marshaling the agents, migrating them from their source computers to their destination computers, unmarshaling them, and verifying security.

4 Experience

This section presents several example applications that illustrate how the framework works.

[4] The functionalities of the framework, except for subscribe/publish-based remote event passing, can be implemented on Java Developer Kit version 1.1 or later, including Personal Java.

[5] The size of each of the three agents was about 8 KB in size.

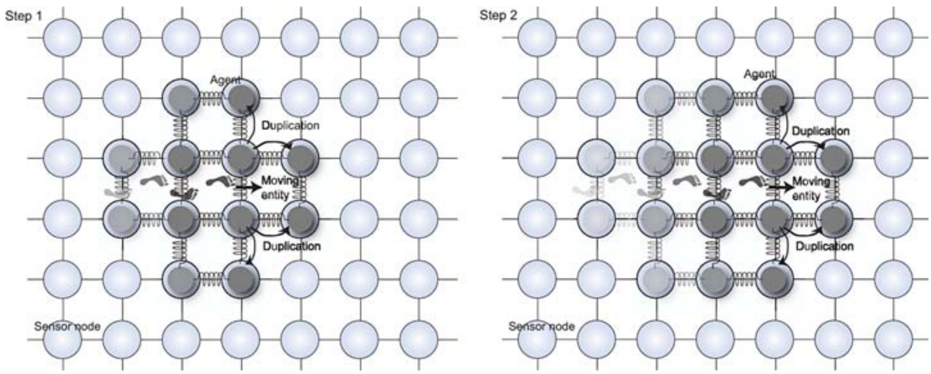


Fig. 6. Agent diffusion for moving entity over a sensor network

4.1 Agent Diffusion in Sensor Networks

The first example is the speculative deployment of agents like cell-lamellipodia. This provides a mechanism that dynamically and speculatively deploys agents at sensor nodes when there are environmental changes. This mechanism was inspired by lamellipodia in cells. It assumes that the sensor field is a two-dimensional surface composed of sensor nodes and it monitors environmental changes, such as motion in objects and variations in temperature. It is a well known fact that after a sensor node detects environmental changes in its area of coverage, some of its neighboring nodes tend to detect similar changes after a period of time. It deploys agents for monitoring sensors at sensor nodes, where each agent can control and monitor its current sensor node and has its own TTL. These agents can support data preparation in data mining involves data selection, data cleansing, data preprocessing, and data representation. Since our agents are general program entities, they can support a variety of data preparation. As you can see Fig. 6, diffusion occurs as follows:

i) When an agent detects the presence of its target, it creates a specified number of its clones, e.g., two clones, where this number depends on the number of neighboring sensor nodes
ii) When the target moves to another location, the monitoring agents located at the nodes near the target detect the presence of the target or changes in temperature and create their clones in the same way.
iii) Each clone is associated with a resource limit that functions as a generalized TTL field. Although a node can monitor changes in interesting environments, it sets the TTLs of its agents to their own initial values. It otherwise decrements TTLs as the passage of time. When the TTL of an agent becomes zero, the agent automatically removes itself.

The system provides agents for data mining. These agents declare follow polices to monitoring agents. They receive data from the monitoring agents and then analyze and exchange with them.

4.2 Ant-Based Routing Mechanisms

Ants are able to locate a path to a food source using the trails of chemical substances called pheromones that are deposited by other ants. Several researchers have attempted to use the notion of ant pheromones for network-routing mechanisms [1,2]. Our framework allows moving agents to leave themselves on their trails and to become automatically volatile after their life-spans are over. an agent corresponding to an ant, A, corresponding to a pheromone is attached to another agent corresponding to an ant according to the *fill* policy. When the latter agent randomly selects its destination and migrates to the selected destination, the former creates a clone and migrates to the source host of the latter. Since each of the cloned agents defines its life-span by invoking `setTTL()`, they are active for a specified duration after being created. If there are other agents corresponding to pheromones in the host, the visiting agent adds their time spans to its own time span. When another agent corresponding to another ant migrates over the network, it can select a host that has agents corresponding to pheromones with the longest time-spans from neighboring hosts. We experimented with ant-based routing for agents using this prototype implementation with more than eight hosts. However, we knew that it would be difficult to quickly converge a short-path to the destination in real distributed systems, because routing mechanisms tend to diverge.

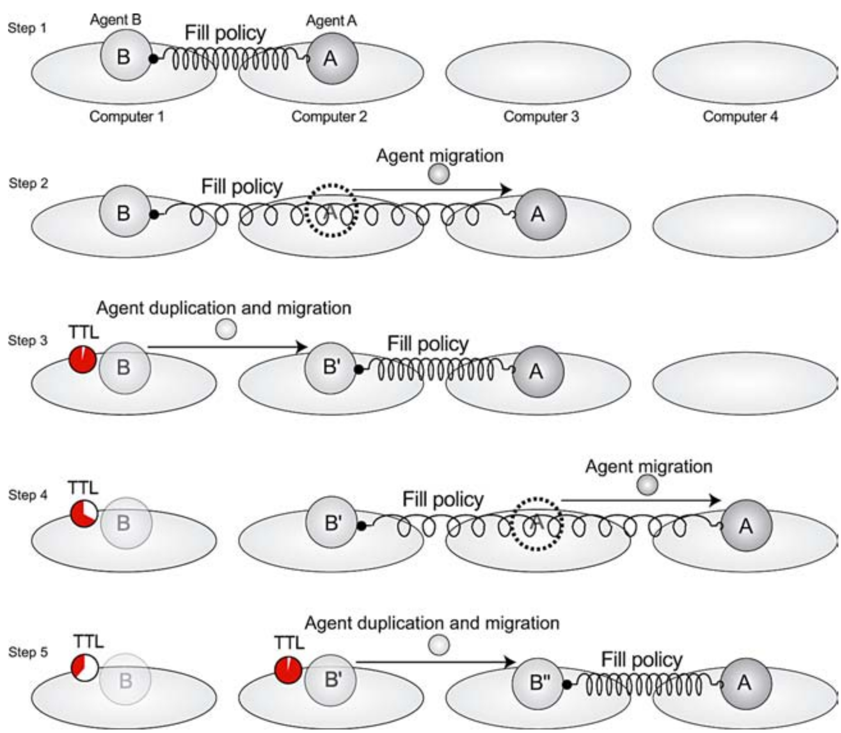


Fig. 7. Ant-based routing mechanism

5 Related Work

There have been several attempts to develop infrastructures for dynamically deploying agents or components between computers in large-scale computing environments, e.g., workstation-clusters and grid computing. Most of them have aimed at dynamically deploying partitioned applications or systems to different computers in distributed systems to balance computational loads or network traffic. However, as they have explicitly or implicitly assumed centralized management approaches to deploying partitioned applications or systems to different computers, they have not allowed all partitioned applications or systems to have its own deployment approaches.

Of these, the FarGo system introduces a mechanism for distributed applications dynamically laid out in a decentralized manner [5]. This is similar to our relocation policy in the sense that it allows all components to have their own policies, but it is aimed at allowing one or more components to control a single component, whereas ours aims at allowing one component to describe its own migration. This is because our framework treats components as autonomous entities that travel from computer to computer under their own control. This difference is important, because FarGo's policies may conflict if two components can declare different relocation policies for one single component. Our framework is free of any conflict because each component can only declare a policy for its own relocation, and not for other components. Several researchers have introduced the dynamic deployment of partitioned applications as a technology that enables distributed computers to support various services, which they may not have initially been designed for, rather than to balance computational loads and traffic in a distributed system.

The Co-Field project [7] by the University di Modena e Reggio Emilia proposed the notion of a computational force-field model for coordinating the movements of a group of agents, including mobile devices, mobile robots, and sensors. However, the model only seems to be available within the limits of simulation and not within a real distributed system. Hive [8] is a distributed agent middleware for building decentralized applications and it can deploy agents at devices in ubiquitous computing environments and organize these devices as groups of agents. Although it introduced metaphors drawn from ecology, it cannot change the structure of agents dynamically whereas ours can.

6 Conclusion

We described a framework for dynamically aggregating distributed applications in a distributed system based on physical dynamics. It was used to build a data mining system from agents, which can explicitly have policies for their own deployment and coordination. It enables a federation of agents to be dynamically structured in a self-organized manner and to be deployed at computers as agents that have gravitational and repulsive forces between them. We designed and implemented a prototype system for the framework and demonstrated its effectiveness in several practical applications. We believe that the framework provides a general and practical infrastructure for building distributed and mobile applications.

Finally, we would like to identify further issues that need to be resolved. The current implementation relies on Java's security manager. Nevertheless, we are interested in security mechanisms for agents that have their own deployment policies and plan on introducing various such policies to support adaptive applications over a distributed system.

References

1. Babaoglu, O., Meling, H. and Montresor, A.: Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems. Proceeding of 22th IEEE International Conference on Distributed Computing Systems, July 2002
2. Di Caro, G. and Dorigo, M.: AntNet: A Mobile Agents Approach to Adaptive Routing. Proceedings of Hawaii International Conference on Systems, Computer Society Press, January (1998) 74–83
3. Gamma, E., Helm, R., Johnson, R. and Vlissides, J.: Design Patterns, Addison-Wesley (1995)
4. Garlan, D., Siewiorek, D., Smailagic, A. and Steenkiste, P.: Project Aura: Towards Distraction-Free Pervasive Computing. IEEE Pervasive Computing, vol. 1 (2002) 22–31
5. Holder, O., Ben-Shaul, I., and Gazit, H.: System Support for Dynamic Layout of Distributed Applications. Proceedings of International Conference on Distributed Computing Systems (ICDCS'99) IEEE Computer Soceity (1999) 403–411
6. Krasner, G.E. and Pope, S.T.: A Cookbook for Using the Model-View-Controller User Interface Paradigma in Smalltalk-80. Journal of Object Oriented Programming, Vol. 1 No. 3 (1988) 26–49
7. Mamei, M., Leonardi, L., Zambonelli, F.: Co-Fields: A Unifying Approach to Swarm Intelligence. International Workshop on Engineering Societies in the Agents World (ESAW 2002), Lecture Notes in Computer Science, Vol. 2577, Springer Verlag (2003)
8. Minar, N., Gray, M., Roup, O., Krikorian, R., Maes, P.: Hive: Distributed Agents for Networking Things. International Symposium on Agent Systems and Applications / International Symposium on Mobile Agents (ASA/MA'99) (1999)
9. Román, M., Hess, C.K., Cerqueira, R., Ranganat, A., Campbell, R.H., Nahrstedt K.: Gaia: A Middleware Infrastructure to Enable Active Spaces. IEEE Pervasive Computing, Vol. 1 (2002) 74–82
10. Román, M., Ho, H., Campbell, R.H.: Application Mobility in Active Spaces. Proceedings of International Conference on Mobile and Ubiquitous Multimedia (2002)
11. Satoh, I.: MobileSpaces: A Framework for Building Adaptive Distributed Applications Using a Hierarchical Mobile Agent System. Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS'2000) (April 2000) 161–168
12. Satoh, I.: Dynamic Federation of Partitioned Applications in Ubiquitous Computing Environments. Proceedings of 2nd International Conference on Pervasive Computing and Communications (PerCom'2004), IEEE Computer Society (March 2004) 356–360
13. Satoh, I.: Dynamic Deployment of Pervasive Services. Proceedings of IEEE International Conference on Pervasive Services (ICPS'2005), IEEE Computer Society (July 2005) 302–311
14. Satoh, I.: Bio-inspired Organization for Multi-agents on Distributed Systems. Proceedings of 2nd Workshop on Biologically Inspired Approaches to Advanced Information Technology (BioADIT'2006), Lecture Notes in Computer Science (LNCS), Vol. 3853, Springer (January 2006) 355–362
15. Satoh, I.: Self-organizing Software Components in Distributed Systems. to appear in 20th International Conference on Architecture of Computing Systems (ARCS'2007), Lecture Notes in Computer Science, Springer, March 2007
16. Szyperski, C.: Component Software, Addison-Wesley (1998)
17. World Wide Web Consortium (W3C), Composite Capability/Preference Profiles (CC/PP), http://www.w3.org/TR/NOTE-CCPP (1999)

Role-Based Decision Mining for Multiagent Emergency Response Management

Alexander Smirnov, Mikhail Pashkin, Tatiana Levashova,
Nikolay Shilov, and Alexey Kashevnik

St.Petersburg Institute for Informatics and Automation of the
Russian Academy of Sciences (SPIIRAS), 14 line, 39,
199178 St.Petersburg, Russia
{smir,michael,oleg,nick,alexey}@iias.spb.su

Abstract. Emergency response operations require availability of tools that would allow fast and clear description of situation, generation of effective solutions for situation management, selection of a right decision maker and supplying him/her necessary data. During decision making support a large amount of raw data describing current situation, users preferences, found solutions and final decisions done by decision maker are accumulated in the repository. To make these data useful, methods of data mining can be applied. The goal of decision mining is to find "rules" explaining under which circumstances one activity is to be selected rather than the other one. The paper presents results of a research concerning mining of decisions stored in user profiles to find common preferences for different roles of decision makers participating in emergency response operations. These preferences could be used as a basis for building of decision trees allowing (semi)automatically selection of best decisions in a typical situation. Modeling of an emergency response management system implementing the research results has been done using software agents playing roles of different types of the system users. Case studies related to fire and accident response operations have been used.

Keywords: Decision mining, profiling, context management, intelligent agents.

1 Introduction

Decisions in emergency response operations are characterized as well-specified, ad hoc, quickly made, based on past experience and taking into account alternatives. They are made in rapidly changing, sometimes unexpected, situations. Such operations require availability of systems / tools that would allow fast and clear description of situation, generation of new and reuse of previously made effective solutions for situation reformation, selection of a right decision maker and supplying him/her necessary data. Such tools include components for actual situation description, user modeling, finding appropriate methods for problem solving, integration of data from heterogeneous sources, finding / generation of insufficient data, removing of uncertainties, estimation of solutions, etc. During decision making process a large amount of auxiliary raw data are accumulated in repositories. Methods

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 178–191, 2007.
© Springer-Verlag Berlin Heidelberg 2007

of data mining are used in such systems for different purposes: finding associative rules between decisions and factors influencing them, user clustering using decision trees and neural networks, recognition of common users' features / interests and others [1], [2], [3]. Validation of the obtained results can be done using simulation software modules.

The paper presents results of the research from the area of data mining in the context-sensitive emergency response management system (ERMS) implementation. The developed approach assumes usage of (i) ontologies for problem domain description (notions, relations between notions, data sources and methods), (ii) user profiles to accumulate raw data and build the system's vision of the user, (iii) context for actual situation description and (iv) intelligent agents for modeling of the users interactions. The developed methodology is oriented to producing an ontology-driven context model so that the decision makers would be provided with the information and knowledge required in the situation they are interested in and according to the roles they fill.

Within the presented approach the context is a weak-structured information containing three constituents: (i) ontology elements describing actual situation; (ii) user data describing his/her role, preferences and additional constraints for actual situation description; and (iii) data / information / knowledge extracted from available sources. Context is built when an emergency situation occurs and used for response operation planning. This is a basis for generation and estimation of alternative solutions and presentation of them to the decision maker for selecting the best one from his/her point of view. Each situation is described by the following components: context, solutions built upon the context and final decision – selection of solution(s). These components are stored in the user profile.

Finding influence of the context to the final decision made by the decision maker with a certain role is an actual task because it would help to (i) find and model typical scenarios of interaction between users; (ii) reveal standard situations within large amount of raw data; and (iii) cluster existing decision makers into groups that would allow to reduce the number of supported user models and increase the quality of data presentation. Finally these results lead to increasing of decision quality during emergency response operations when decisions have to be made under pressure of time. To find the above inference the described here approach applies decision mining techniques as a part of data mining.

Design of multi-agent systems is a hot research topic in the area of development of distributed applications. It provides such benefits as interoperability (since agent interactions are based on clear specifications) and scalability (since multiagent systems can be extended with agents of new types and/or functions of existing agents can be modified). Pro-active agents' functions allow modeling of customer behavior corresponding to actual situations and their intentions.

This paper gives a brief introduction of tasks solved in the area of decision mining and motivation of using agents for modeling system scenarios (section 2); introduces the developed context-sensitive methodology for emergency response management (section 3); describes a developed profile model; its role in decision making, and methods based on the analysis of alternatives intended for identification of user preferences (section 4); and presents modeling of ERMS that implements the research results via usage of software agents presenting decision makers with different roles (section 5).

2 Technological Background

Analysis of different kinds of decisions is one of direction of data mining for business processes. The goal of decision mining is to find “rules” explaining under which circumstances one activity is to be selected rather than the other one. For instance, decision mining aims at the detection of data dependencies that affect the routing of a case in the event log of the business process executions [4]. There is a set of tools implementing different tasks of decision mining: “Decision Miner” [4], two projects of Decision Mining software (“Risky Business” and “GoldPan”) [5] and others.

Decision mining covers a wide range of tasks. Estimation of data quality and interpretation of their semantics is one of the major tasks of decision mining. It requires the following interpretation of data attributes: whether it is relevant, what it actually means, in what quantities it is measured, etc. Classification of decisions is also one of important task for decision mining. Solving of these tasks requires development of decision models and (semi)automatic analysis techniques. For instance, concept of decision trees has been adapted to carry out a decision point analysis [4], spatial analysis has been adapted for criminal event prediction [6].

The developed project proposes considered application of decision mining to the area of emergency response management. Classification of decisions allows to check correspondence between decision makers and their roles. Finding of preferences of decision makers helps to build decision trees allowing to make a right decision in a critical situation (semi)automatically.

Multiagent architecture is widely used for distributed application design [7], [8], [9]. Multiagent systems are used both for research purposes and for development of industrial application. This usage is motivated by the following reasons: (i) agents are able to operate alone and to be a member of a team of agents, (ii) there are standard protocols and messages for agents’ interaction, (iii) agents can appear and disappear without destroying of agent community, (iv) there are a lot of tools for multiagent systems development and many other reasons.

A short overview of famous software for Multiagent system development for business applications is presented in Table 1.

The major tasks of users participating in emergency response operation are: monitoring of actual situation, detection of emergency situation, estimation of dangerous, finding of solutions, selection of solution and execution of assigned tasks. Participation in operations assumes interactions between different roles, negotiation of conditions of tasks execution and forming of teams for tasks execution. Modeling of different roles and their interaction using software agent could be a suitable solution to test to be developed information models, hypothesis and scenarios of ERMS. JADE (Java Agent DEvelopment Framework) was selected for implementation of agent-based modeling because it is fully FIPA [10] compliant, widely spread in science community and is available as open source software.

Table 1. Features of famous multiagent development tools (adapted from [11])

Project	Developer	Features	Application
JACK [12]	The Agent Oriented Software Group	• Completed • Commercial • Java-based • Custom messages and add-on for FIPA ACL • Support of ontology	Intelligent Agent-controlled flight by a Codarra 'Avatar' unmanned aerial vehicle
MAGENTA [13]	MAGENTA Technology	• Completed • Java-based • Custom messages • Ontology editor	i-Scheduler system for Tankers UK Ltd.
JADE [14]	Telecom Italia Lab	• On-going project • Open source • Java-based • Fully FIPA compliant • Support of ontology • Huge community of developers that are using JADE	It is widely spread and used in the academic environment but several international R&D projects, from different financing European frameworks (IST, ACTS, etc.), have been based on JADE

3 Context-Sensitive Decision Support in Emergency Response Management System

The idea of using context model for operational decision support arose from the context definition. Context is defined as any information that can be used to characterize the situation of an entity where an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [15].

In modern information systems ontologies are widely used as a basis for domain knowledge description. In the presented research formalism of Object-Oriented Constraint Networks (OOCN) is used for a formal ontology representation. According to this formalism ontology is represented by sets of classes, class attributes, attribute domains, and constraints. The set of constrains consists of constraints describing "class, attribute, domain" relation; constraints representing structural relations as hierarchical relationships "part-of" and "is-a", classes compatibility, associative relationships, class cardinality restrictions; and constraints describing functional dependencies. Each of the entities in a class is considered as an instance of the class.

The developed methodology proposes integration of environmental information and knowledge in context [16]. The context is purposed to represent only relevant information and knowledge from the large amount of those. Relevance of information and knowledge is evaluated on a basis how they are related to a modelling of an ad hoc problem. Two types of context are used: 1) *abstract context* that is an ontology-based model integrating information and knowledge relevant to the problem, and 2) *operational context* that is an instantiation of the abstract context with data provided by the information sources or calculated based on functions specified in the abstract context.

Before a problem can be solved an ontology describing areas covered by ERMS has to be built. The ontology combines the domain knowledge and problem solving knowledge.

Decision making in dynamic domains is characterized by a necessity to dynamically process and integrate information from heterogeneous sources and to provide the user with context-driven help for the analysis of the current situation. Systems of context-driven decision making support are based on usage of information / data, documents, knowledge and models for problem identification and solving.

Another necessary prerequisite for the system operation is its connection to information sources. It consists of three operations: finding information sources, connection of the found sources to the system using Web-services, and assigning information source references to attributes of the ontology, in other words, defining which sources the ontology attributes take values from.

Once the above operations are completed the ERMS is considered to be ready for problem processing. Personalization is one of the important features of operational decision making support. For this reason users are described in the system via user profiles and associated with different roles that set certain limitations to provide only information that is useful for a particular user (see sec. 4.1).

After registration in the system the user profile is updated or created if it did not exist before. After that depending on the user role and available information the user either describes a problem or selects a situation. In the latter case the abstract context is assumed to exist and it is activated. In order to ensure usability of the previously created abstract context the context versioning technique described in the next section is used. In the former case few more steps are required. The problem description stored in the profile for further analysis is recognized by the system and recognized meaningful terms are mapped to the vocabulary of the ontology; the vocabulary includes class names, attribute names, string domains, etc. Based on the identified ontology elements and developed algorithm a set of slices relevant to the problem is built. Based on these slices a set of abstract contexts is generated. Since some of the problem solving methods can be alternative, integration of such slices leads to a set of alternative abstract contexts. The generated abstract contexts are checked for consistency, their attributes are assigned information sources based on the information from the ontology, and it is saved in the context archive. Role constraints are applied to the generated abstract context to remove information that is not interesting or not supposed to be seen by the user.

The abstract contexts are checked if it is enough for solving the current user problem. Then the required information is acquired from the appropriate sources and calculations are performed. Due to the chosen notation of OOCN a compatible constraint solver can be used for this purpose. The constraint solver analyses information acquired from sources and produces a set of feasible solutions eliminating contradictory information.

The above operations result in forming operational contexts for each abstract context. Among the operational contexts one with values for all attributes is selected. This operational context is be used at the later stages of the problem solving. All the operational contexts are stored in the context archive and references to them are stored in the user profile.

Based on the identified operational context the set of solutions is generated. Among the generated solutions the user selects the most appropriate one – makes a decision. The solutions and the final decision are stored in the user profile for further analysis. Stored in the profile information can be used for various purposes including audit of user activities, estimation of user skills in certain areas, etc. In the presented research this information is used for decision mining.

In proposed ERMS a problem is described via user request entered in the system. In the rest of the paper a term “request” is used instead of “problem description”.

4 Revealing of User Preferences

Auxiliary data related to each task processed by ERMS are stored in user profile. Analysis of the stored data allows to define factors influencing decisions made by decision makers with similar roles.

4.1 Profile Model

Based on the analysis of user properties used in profile models, means used for profile representations, views on user preferences depending on the application domain, and the role of profiles in collaborative work a user profile reference model and a formal model of user profile have been developed.

The user profile reference model (Fig. 1) consists of the following categories:

- *User context* actualizing information about the user. It includes four subcategories: (i) personal and contact information, (ii) system information about the user (current system information), (iii) feedback (information about quality of the user work), and (iv) user preferences (user preferences identified by ERMS).
- *Decision making history* containing history about the user requests, related operational contexts, solutions generated by the system, decisions made by the user, system information and user preferences at the point of time of the decision making process initiation (moment of user problem definition).

Subcategory *Personal and contact information* contains the following properties:

- *First name, last name, gender,* and *birth date* represent personal information about the user.
- *List of languages* represents languages for contacting with the user.
- *Phone number, e-mail, instant messaging ID, and homepage* properties are used for contacting with the user (user can participate in chats, audio and video conferences using his/her instant messaging ID).

Subcategory *System information* contains the following properties:

- *Role* defines the current role of the user during decision making process. It defines a list of tasks related to this role. The list of task corresponds to the tasks represented in tasks & methods ontology this role deal with.

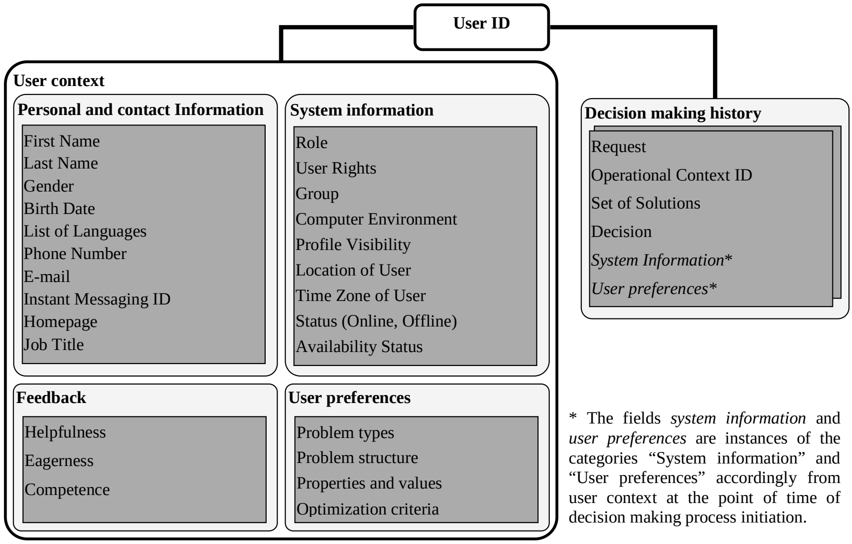


Fig. 1. Developed reference model of the user profile

- *User rights* define current user rights for decision making. Problems to be solved may contain confidential data and not all users can be allowed to view it.
- *Group* determines the current user group. Groups join users with the same role and rights to simplify selection of users for decision making. Selection of user for decision making depends on the user context. Grouping of users helps team leaders to select users for make decision for this problem.
- *Computer environment* defines interfaces for the current session in the system. It allows team leader to involve the user into the decision making process if he/she has suitable local computer environment.
- *Profile visibility* defines users or groups of users, who can view the profile of this user. It makes it possible to hide the decision making history for some users or groups.
- *Location of user* defines current location of the user. It can be taken into account for estimating rapidity and quality of decision making process in a particular situation.
- *Time zone of user* defines current user time. It determines, for example, time left until the user's work day is over and if it is enough to solve the current problem.
- *Status (online, offline)* defines current status of user in the system.
- *Availability status (available, not available)* defines current occupancy of the user and his/her ability to participate in the decision making process.

When the user registers in ERMS he/she inputs his/her personal and contact information. This information is saved in the user profile. User login and password are saved in the subcategory "System information". When the user is authenticated in ERMS, the fields "Computer environment", "Location of user", "Time zone of user", and "Status" are filled in and the appropriate values are saved in the profile.

Subcategory *Feedback* contains the following properties estimated by the team leader:

- *Helpfulness* - degree to which the team leader feels that the user is capable to solve problems. It defines helpfulness of the user's contribution into previous decision making processes. Helpfulness can be estimated by the team leader based on previous decisions of the user (decision maker). This estimation can be based on the percentage of executed tasks from all the tasks.
- *Eagerness* - a positive feeling of the user to push ahead with something. Team leader can determine eagerness of the user based on his/her participation in previous decision making processes.
- *Competence* – skills and knowledge acquired by the user. Team leader can estimate the level of competence based on the analysis of the number of problems solved by the user and time of user's work in a given problem area.

Subcategory *User preferences* represents preferences of the user identified by the system based on abstract context in the being developed methodology. It is a model of the problem being solved for the role. Subcategory user preferences contains following properties:

- *Problem types* represent user preferences regarding problem areas. It contains a list of classes of abstract context "preferred" by the user earlier (estimated by the system).
- *Problem structures* represent user preferences for relationship types. It contains a list of types of relations of the abstract context "preferred" by the user earlier (estimated by the system).
- *Properties and values* represent user preferences for values of certain properties. Contains list of properties and/or their values "preferred" by the user earlier (estimated by the system). Properties are related to attributes of the abstract context.
- *Optimization criteria* represent user preferences for optimization criteria. Contains list of criteria "preferred" by the user earlier (estimated by the system).

Category *Decision making history* (Fig. 2) contains sets of tuples with the following properties (each tuple stored in the user profile is related to one decision making process):

- *Request* can be used for further reuse of the abstract context for the same or similar requests to the system.
- *Operational context ID* refers to instantiated problem model the user worked with in a particular situation.
- *Set of solutions and decision* are used to analyze users' work (other users can see solutions generated in particular situations) and to identify user preferences (via analyzing differences of selected decisions from other offered alternative solutions).
- *System information* stores information about the user at the moment of decision making initiation. It contains a snapshot of all the properties of the category "System information".
- *User preferences* store user preferences at the moment of decision making initiation. It contains a snapshot of all the properties of the category "User preferences".

id	task_text	date_of_assignment	task_status	user_id	disaster_type	N_Of_Victims	point_in_map	task_preferences	oper._context	solutionXML	SolutionID
73	Fire at Fragmenton	2006-10-26 15:33:57	4	3	1	9	232	0	<MEMO>	<MEMO>	1
75	New task	2006-10-26 16:36:43	2	3	1	9	68	0	<MEMO>	<MEMO>	<NULL>
78	Accident around Novi	2006-10-27 09:57:00	5	3	0	5	38	0	<MEMO>	<MEMO>	1
79	Fire in Village Oaks Lake	2006-10-27 10:53:32	4	3	1	9	142	1	<MEMO>	<MEMO>	1
81	Test example	2006-10-27 11:31:13	4	3	1	9	142	0	<MEMO>	<MEMO>	1
82	Disaster in Plymouth	2006-10-27 14:30:26	4	3	0	15	77	0	<MEMO>	<MEMO>	1
83	Accdent around the Franklin	2006-10-30 10:39:28	4	3	0	15	314	1	<MEMO>	<MEMO>	1
84	Description	2006-10-30 16:12:01	4	3	1	15	187	0	<MEMO>	<MEMO>	1
85	Fire in RedFord	2006-10-30 16:25:27	2	3	1	9	302	0	<MEMO>	<MEMO>	<NULL>
86	Description	2006-10-30 16:43:17	2	3	1	9	302	0	<MEMO>	<MEMO>	<NULL>
87	Fire at the neares place from Fragment(	2006-10-30 16:45:31	2	3	1	9	255	0	<MEMO>	<MEMO>	<NULL>
88	Small fire	2006-11-01 17:11:53	3	3	1	9	233	0	<MEMO>	<MEMO>	1


```
  <?xml version="1.0" encoding="utf-8" ?>
- <SOLUTIONS>
  - <SOLUTION>
      <ID>1</ID>
      <CONTEXTID>34</CONTEXTID>
      <TIME>11.58349457</TIME>
      <MEDICALTIME>32.323893482708</MEDICALTIME>
      <COST>9546.6267697435</COST>
    - <DISASTER>
        <POINTID>232</POINTID>
        <X>488.500000000000</X>
        <Y>237.125045776367</Y>
      </DISASTER>
      <RAIN>1</RAIN>
      <WIND>0</WIND>
      <TEMPERATURE>16</TEMPERATURE>
    - <INUNDATED_ROADS>
        <INUNDATED_ROAD point1id="62" point2id="51" />
        <INUNDATED_ROAD point1id="51" point2id="52" />
        <INUNDATED_ROAD point1id="52" point2id="82" />
```


Fig. 2. Part of user profile presented at time the user logged in ERMS

From the above set of properties the following context specific properties has been identified: role, time zone of user, location of user, operational context ID, set of solutions, decision. They are used as “raw” data by data mining techniques for clustering of decision makers into roles and for user preferences mining.

4.2 Profile-Based Decision Mining

The proposed model of profile has been already adapted for user clustering using text mining techniques [17]. The result of user clustering is identification of roles for decision maker grouping. Grouping shows similarities between different users and allows to find alternative decision makers with similar competence, knowledge, etc. Inputs for clustering algorithm are user requests and the ontology.

To take into account other data stored in user profiles another method based on the analysis of alternatives has been developed. Unlike the mentioned above method searching for similarities of the decisions selected by the user, this method assumes an analysis of differences between each decision made (the solution selected by the user from the offered set of solutions) from the remaining solutions. These differences are stored and appropriate statistics is accumulated. Based on the statistics the user preferences can be identified as most common differences of his/her decisions from the remaining solutions taking into account information from the operational context.

Let $DM = \{Sol_{1,\ldots,n}, Dec\}$, where Sol_i is a solution and Dec is a decision selected by the user. The decision Dec is considered to be excluded from the set of solutions $Sol_{1,\ldots,n}$.

Each decision as well as the solution consists of objects (class instances) I with instantiated attributes P (attribute values V): $Sol_i = \{<I, P, V>\}$.

Comparison of the decision with each solution results in sets:

$Diff_i^+ = \{<I, P, V>\}$, $i = 1, \ldots, n$, containing objects from the decision and object attributes that differ from those of the solution ($\forall <I, P, V> \subset Diff_i^+ : <I, P, V> \subset Dec$, $<I, P, V> \not\subset Sol_i$).

$Diff_i^- = \{<I, P, V>\}$, $i = 1, \ldots, n$, containing objects from the solution and object attributes that differ from those of the decision ($\forall <I, P, V> \subset Diff_i^- : <I, P, V> \not\subset Dec$, $<I, P, V> \subset Sol_i$).

Then, these sets are united and number of occurrences (k) of each tuple $<I, P, V>$ is calculated:

$Diff^+ = \{<I, P, V>, k_l\}$ – objects and their values preferred by the user.

$Diff^- = \{<I, P, V>, k_m\}$ – objects and their values avoided by the user.

When the volume of the accumulated statistics is enough (it depends on the problem complexity and is defined by the administrator) a conclusion about usually preferred and/or avoided objects and their parameters from the problem domain by the user can be made. This conclusion is considered as user preference.

The resulting preferences can be described via rules, e.g.:

```
if Weather.WindSpeed > 10 then EXCLUDE HELICOPTERS
if Disaster.Type = “Fire” then INCLUDE FIREFIGHTERS
if Disaster.Number Of Victims > 0
  then Solution.Time -> min
```

When users are clustered into roles and their preferences are revealed it is possible to compare preferred objects in each cluster and build hypothesis how decision trees have to look like. The extracted preferences are used for building decision trees allowing to make a right decision in a critical situation (semi)automatically. An example of usage of decision trees by agents presenting members of emergency response operation is given in sec. 5. Also such decision trees could be useful for solutions estimation and ranking to support of decision maker. The question of building of decision trees is out of scope of the paper. Today there are a lot of algorithms and tools implementing them to solve this task (e.g., CART – Classification and Regression Tree algorithm [18], C4.5 [19] and others).

Fig. 3 presents an example of results of decision mining.

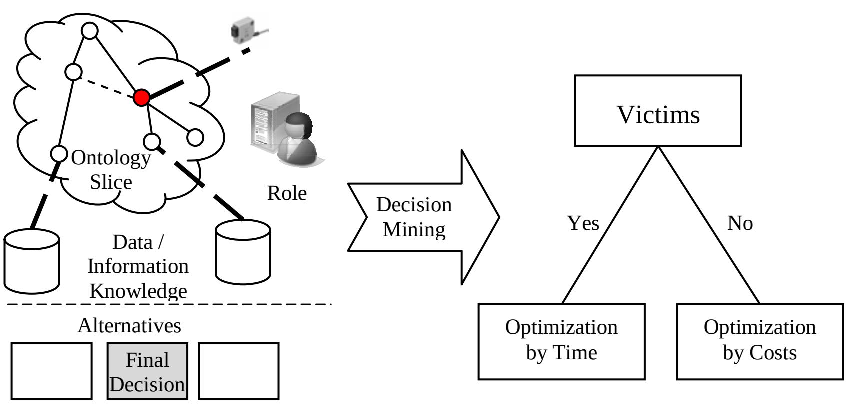


Fig. 3. Example of the mined rule influencing user decision

Depending on information about victims decision makers having the same role used the same optimization criterion to make a final decision. The rule looks like “if where some victims in accident are then an emergency operation has to be executed in the shortest time”.

5 Agent-Based Modeling in Emergency Response Management System

For the implementation of the case study a set of software services have been developed (Table 2). They are used for performing context-dependent operations.

Table 2. Prototype services

Services	Description
Core service	Mediator for solution generation. Role agent inputs a request into the system. The service is responsible for interaction between other services, collection of data, and preparation of solution for role agent.
Wrapper Web service	These services are used to extract (i) information from available sources (databases, Web sites, etc.), (ii) data from sensors, and (iii) documents from corpus.
Ontology Web service	These services are used (i) to build abstract and operative contexts, ontology and role of user, and (ii) to perform operations with stored slices: versioning, change propagation.
Solver Web service	These services are used to solve optimization tasks: e.g., resource allocation, transportation tasks.

Referring to the case study, the following types of agents have been used in the prototype (Table 3): dispatcher agent (DA), a set of role agents (RA) presenting decision makers and participant agent (PA). In real life a participant could be Medical Technician, Firefighter and other. But for agent-based modeling a real profession of the participant does not matter.

Table 3. Developed agents of ERMS

Type of agent	Functions of agent
Dispatcher agent	• Detection of new event appearance • Auxiliary data preparation • Selection of RA
Role agent	• Call of core service • Negotiate schedule and capacity with PA • Assign tasks for PA
Participant agent	• Accept or decline requirements from RA • Send data to user

Depending on the agent type the following functions are performed by agents: (i) presentation of user models (read and update user profile), (ii) detection of new event appearance, (iii) negotiation for agent team creation and (iv) cooperative task execution.

Agents of the ERMS have the following behavior characteristics:

- benevolence – willingness to help each other;
- veracity – agents do not process knowingly false information;
- rationality – willingness to achieve the goal and not to avoid it.

The following general structure has been proposed for agent-based modeling: (i) identifier, (ii) name, (iii) list of scenarios the agent can perform, (iv) list of methods the agent can execute, and (iv) list of messages the agents can exchange. Structure of a RA has been extended by (i) role of decision maker it presents and (ii) decision tree built using user preferences – result of decision mining – for decision making.

Fig. 4 shows a general scenario demonstrating interactions of agents specified within the case study. The DA receives information about disaster, selects an appropriate RA, prepares auxiliary information and sends a message to the RA. Information about disaster includes its location, potential number of victims and other additional descriptions.

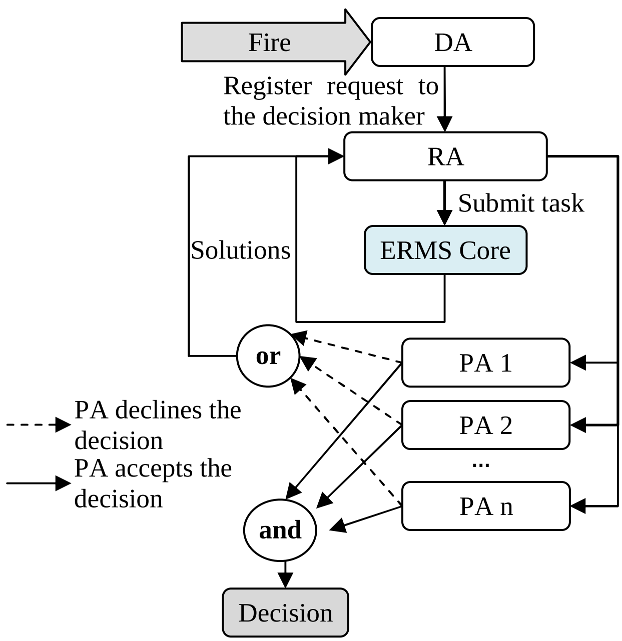


Fig. 4. Interactions of agents performing different roles

The RA analyzes received information. It adds constraints related to its role and prepares and sends a request to core service of ERMS. The core service builds an abstract context and fills with data from the information sources and with the values calculated based on the functional constraints. Thus an operational context (or the emergency situation) is produced. This situation is presented to the RA. It can add additional constraints (e.g., criteria for optimization by time) and launch the solution generation to core service. Using the decision tree built as a result of the decision mining it estimates solutions and selects the best one. This solution is considered to be the decision.

Based on this decision, the RA forms action plans for PAs of the operation and negotiates it with them. The PAs can either accept or decline the plans. In the latter case the RA is informed that it has to make another decision.

6 Conclusion

The paper presents results of a research related to the area of data mining in the context-sensitive emergency response management system implementation. In the proposed system after each decision making process auxiliary data are stored in user profile: ontology elements describing actual situation, preferences of user (decision maker), set of alternative solutions and final decision. A method for role-based decision mining for revealing of user preferences influencing to the final user decision has been proposed. Modeling of the system using software agent has been done to validate obtained results.

In the future other methods belonging to area of data mining can be developed or adapted to extract new results from accumulated data and tested using developed agent community.

Acknowledgments. The paper is due to the research carried out as a part of CRDF partner project # RUM2-1554-ST-05 with US ONR and US AFRL, projects funded by grants # 05-01-00151 and # 06-07-89242 of the Russian Foundation for Basic Research, as well as projects # 16.2.35 of the research program "Mathematical Modelling and Intelligent Systems", # 1.9 of the research program “Fundamental Basics of Information Technologies and Computer Systems” of the Russian Academy of Sciences, # 78 of the scientific program of St.Petersburg Scientific Center, and # 26/06 of the Science and High School Committee of St.Petersburg.

References

1. Thomassey, S., Fiordaliso, A. A Hybrid Sales Forecasting System Based on Clustering and Decision Trees. Decision Support Systems. Vol. 42, N 1 (2006) 408–421
2. Li, X.: A Scalable Decision Tree System and Its Application in Pattern Recognition and Intrusion Detection. Decision Support Systems. Vol. 41, N 1 (2005) 112–130
3. Chiang, W., Zhang, D., Zhou, L.: Predicting and Explaining Patronage Behavior Toward Web and Traditional Stores Using Neural Networks: a Comparative Analysis with Logistic Regression. Decision Support Systems. Vol. 41, N 2 (2006) 514–531
4. Rozinat, A., van der Aalst, W.M.P.: Decision Mining in ProM. In Dustdar, S., Fiadeiro, J., Sheth, A. (eds.): International Conference on Business Process Management (BPM 2006). Lecture Notes in Computer Science, Vol. 4102. Springer-Verlag, Berlin (2006) 420–425

5. Decision Mining software corporate Web site, URL: http://www.decisionmining.com, (2007)
6. Brown, D., Liu, H., Xue, Y.: Spatial Analysis with Preference Specification of Latent Decision Makers for Criminal Event Prediction. Special Issue: Intelligence and Security Informatics (2006) 560–573
7. Reyes, R., Sison R.: A Case-Based Reasoning Approach to an Internet Agent-based Tutoring System. In Proceedings of the World Conference on Artificial Intelligence in Education (2001) 122–129
8. Wooldridge, M. Responding to Real-World Complexity: Introduction to Multi-agent Technology. Magenta White Paper (2004). URL: http://www.magenta-technology.com/resources/downloads/ (2007)
9. Smirnov, A., Pashkin, M., Chilov, N., Levashova, T., Krizhanovsky, A.: Agent-Based Intelligent Support to Coalition Operations: A Case Study of Health Service Logistics Support. In: Shalamanov, V., Johnson, G., Fay, J., (eds.) Information & Security. An International Journal. IT in Coalition and Emergency Operations. ProCon Ltd., Sofia, ISSN: 1311-1493, Vol. 16 (2005) 41–61
10. FIPA: The Foundation for Intelligent Physical Agents. URL: http://www.fipa.org/, (2007)
11. The AgentLink III Committee, URL: http://eprints.agentlink.org/ view/type/software.html (2007)
12. The Agent Oriented Software Group (AOS) Web site, URL: http://www. agent-software.com.au/shared/home/, (2007)
13. Magenta Technology Web site, URL: http://www.magenta-technology.com/, (2007)
14. JADE: Java Agent DEvelopment Framework.URL: http://jade.tilab.com/, 2007
15. Dey, A.K., Salber, D., Abowd, G.D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, Context-Aware Computing. In: Moran , T.P., Dourish P. (eds.) A Special Triple Issue of Human-Computer Interaction. Lawrence-Erlbaum. Vol. 16, (2001) 97–166
16. Smirnov, A., Pashkin, M., Chilov, N., Levashova, T.: Constraint-driven methodology for context-based decision support. Design, Building and Evaluation of Intelligent DMSS (Journal of Decision Systems), Lavoisier, Vol. 14, N 3 (2005) 279–301
17. Smirnov, A., Pashkin, M., Chilov, N., Levashova, T., Krizhanovsky, A., Kashevnik, A.: Ontology-Based Users and Requests Clustering in Customer Service Management System. In: Gorodetsky, V., Liu, J., Skormin, V. (eds.): Autonomous Intelligent Systems: Agents and Data Mining: International Workshop, AIS-ADM 2005. Lecture Notes in Computer Science, Vol. 3505. Springer-Verlag GmbH (2005) 231–246
18. Breiman, L., Friedman J.H., Olshen R.A., Stone C.T.: Classification and Regression Trees. Wadsworth, Belmont, California (1984)
19. Ross Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc. (1993)

Virtual Markets: Q-Learning Sellers with Simple State Representation

Natalia Akchurina[1] and Hans Kleine Büning[2]

[1] International Graduate School of Dynamic Intelligent Systems,
University of Paderborn, Warburger Str. 100, 33098 Paderborn, Germany
anatalia@mail.uni-paderborn.de
[2] Faculty of Computer Science,
University of Paderborn, Warburger Str. 100, 33098 Paderborn, Germany
kbcsl@uni-paderborn.de

Abstract. As shopbots spread through Internet collecting information about lowest prices / highest qualities human sellers will turn out to be too slow to tune the prices and thus unprofitable in comparison with smart agents — pricebots. One of the most promising approaches to building pricebots is Q-learning. Its advantages: flexibility to act under changing conditions of virtual markets, Q-learning sellers can take into account not only immediate rewards but also profits far ahead, and don't need information neither on buyer demand nor on competitors' behaviour. But up to now Q-learning sellers used state representation exponential in the number of sellers acting in the market and could function successfully only with one competitor which no doubt is unrealistic. We are proposing a new state representation independent of the number of sellers that allowed to 10 agents to find the prices that maximize cumulative profits under conditions of high competition in three moderately realistic economic models. It was also shown that due to their flexibility Q-learning sellers managed to collect more profit than pricebots based on two other generally used approaches even though one of them possessed much more information about buyer demand and competitors' behaviour. The proposed representation doesn't depend on the number of sellers and in principle Q-learning sellers using it can function in the markets with arbitrary number of competitors.

Keywords: reinforcement learning, Q-learning, adaptive multi-agent systems, agent economies, virtual markets, pricebots.

1 Introduction

In 1995 came into the world Bargain Finder the first Internet shopping agent that searched the Internet for the lowest price for its owner's favorite music compact disk. Now shopbots have spread into a wide range of markets. As a consumer expresses interest in a particular good or service, a typical shopbot

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 192–205, 2007.
© Springer-Verlag Berlin Heidelberg 2007

queries several hundred web sites [1], [2], [3], [4], [5], and then returns to the user the list of the most attractive offers — all within seconds. It would take hours for the most persistent, resolute consumer to acquire far less scope of information on the good of interest.

Due to the digital nature of information and tireless activity of shopbots virtual commerce will obtain some new characteristics not typical in tradition markets [6], [7]:

First, for the menu costs (the costs to firms of evaluating, updating, and advertising prices) are very low, prices can no longer be fixed but completely dynamic reflecting the latest change in actual costs and buyer demand.

Second, because of shopbots' lack of normal human affections and inertness and their ability to scan hundreds of sites looking for the lowest price and the highest quality [3] the market will be much more open to competition.

Third, companies will be motivated to invest in research on how to lower the costs and improve the quality, rather than into advertising (no advertising campaign can move these strong-headed price-quality hunters or retain them).

In order to be profitable in the environment where the majority of customers hunt for the lowest price or / and the highest quality [3] (and their quality criteria are known to them alone) through hundreds of sites businesses will need to continually adapt to changing market conditions. It was shown [8] that in this environment the faster the seller providing a commodity with some characteristics sets the price the more profit he earns, thus humans would undoubtedly be too slow and expensive and would be substituted by smart programs — pricebots.

Pricebots [8] are economically motivated agents that adjust prices to maximize the profits of firms trading in the Internet, just as shopbots search for the best offers to minimize costs of consumers.

But what can be the essence of such a program?

Traditional game-theoretic approach is out of the question. It can't be applied because of the uncertainties of the real business world: sellers don't posses perfect knowledge on buyer demands let alone competitors' strategies.

We need a pricebot flexible enough to act under changing conditions of virtual market without a model of it.

Sellers based on Q-learning — one of the most significant and actively investigated reinforcement learning algorithms [9] have the potential to adapt to the changing buyer characteristics that determine the buyer demand. The advantage of Q-learning [10] is that one does not need a model of the environment. One can simply observe actual rewards and state transitions and base learning on that. Setting price Q-learning sellers can also look ahead: in principle, the Q-function represents the expected cumulative reward taking into account also profits far ahead.

In this paper we are proposing a simple state representation for Q-learning sellers that allowed to 10 sellers to find the prices that maximize their cumulative profits under condition of high competition in three moderately realistic economic models. Previous approaches to building pricebots based on Q-learning had tremendous drawback — the resulting seller could function only in a market with one competing agent [11], [12] that was undoubtedly too unrealistic. Our representation doesn't depend on the quantity of sellers and in principle pricebots based on our approach can function in markets with any number of competitors.

In Sect. 2 we present Q-learning algorithm. Section 3 is devoted to the introduction of three models of virtual market in which we compared our pricebot with selling agents based on two other widely used approaches (presented in Sect. 4). Section 5 is devoted to the results of the experiments and their analysis and Sect. 6 — to the conclusion.

2 Q-Learning Algorithm

One of the most important breakthroughs in reinforcement learning [13] was the development of Q-learning algorithm. It allows to calculate a tabular function $Q(s, a)$ that returns the greatest value for the action a that should be taken in each particular state s so as to maximize expected cumulative reward.

Q-learning Algorithm.

- Initialize $Q(s, a)$ arbitrarily
- Observe the current state s
- Repeat forever:
  - Choose a from s using policy derived from Q (e.g., ϵ-greedy)
  - Take action a, observe reward r and succeeding state s' provided by the environment
  - $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$
  - $s \leftarrow s'$

Here $\gamma \in [0, 1)$ is the *discount factor* that determines the relative value of delayed versus immediate rewards.

Parameter α in Q-learning algorithm must satisfy $0 < \alpha \leq 1$.

Choosing a from s using ϵ-greedy policy derived from Q means that most of the time we choose an action a that has maximal estimated action value $Q(s, a)$, but with probability ϵ we instead select an action at random.

3 Three Models of Virtual Markets

In all three economies that we'll consider S sellers offer for sale a commodity (a single product or service) that is of interest to B buyers, $B \gg S$.

At any given moment t:

- Each *buyer* purchases a unit of good from the seller that has the best offer (maximizing *utility* — different for each model) or doesn't buy anything at all if no offer is sufficiently attractive (*utility* is negative— e.g., if the sellers' prices are greater than the buyer's valuation of the good). If several sellers offer the good of the same maximal level of attractiveness then the

customer chooses one at random. Every buyer possess up-to-date information about the sellers' prices and characteristics of their goods. (The first can be collected by shopbots and the second can be derived from consumer surveys, is reported by the seller (assumed to be honest), or could be measured by an independent third party).
- Each *seller* s adjusts its price p_s and / or product characteristics (depends on the model). Each seller receives p_s for each purchase, and pays a cost c_s (for production and delivery). All sellers are made aware of the buyers' choices along with the prices offered by all the sellers. (Also can be obtained by means of shopbots).

3.1 Model 1

In economy [8] the value of the good to buyer b is v_b.

A buyer b's utility for a good is a function of price:

$$u_b(p) = v_b - p \tag{1}$$

The *buyer population* consists of:

- w_{T_1} (%) buyers of type T_1 who select seller at random
- w_{T_2} (%) buyers of type T_2 who are "*bargain hunters*" and select seller with the lowest price — correspond to those who take advantage of shopbots

We'll assume that $v_b = v$ for all buyers b.

The probability $h_s(\boldsymbol{p})$ [8] that buyer selects seller s is a function of the price vector $\boldsymbol{p}$:

$$h_s(\boldsymbol{p}) = w_{T_1} f_{s,T_1}(\boldsymbol{p}) + w_{T_2} f_{s,T_2}(\boldsymbol{p}) \tag{2}$$

where $f_{s,T_1}(\boldsymbol{p})$ and $f_{s,T_2}(\boldsymbol{p})$ are the probabilities that the seller s is selected by buyers of type T_1 and T_2, respectively. The probability that a buyer of type T_1 select a seller s is $f_{s,T_1}(\boldsymbol{p}) = 1/S$. Buyers of type T_2, however, select a seller s if and only if s is one of the lowest price sellers. Let's define the following functions:

- $\lambda_s(\boldsymbol{p})$ is the number of sellers charging a lower price than s
- $\tau_s(\boldsymbol{p})$ is the number of sellers charging the same price as s, excluding s itself

Now buyers of type T_2 select seller s iff s is s.t. $\lambda_s(\boldsymbol{p}) = 0$, in which case a buyer selects such a particular seller s with probability $1/(\tau_s(\boldsymbol{p})+1)$. Therefore,

$$f_{s,T_2}(\boldsymbol{p}) = \frac{1}{\tau_s(\boldsymbol{p}) + 1} \delta_{\lambda_s(\boldsymbol{p}),0} \tag{3}$$

where $\delta_{i,j}$ is the Kronecker delta function, equal to 1, whenever $i = j$, and 0, otherwise.

If all buyers agree upon valuation v, and for all sellers the cost is the same — c, then [8] the profit function π_s for seller s:

$$\pi_s(\boldsymbol{p}) = \begin{cases} (p_s - c) h_s(\boldsymbol{p}) B & \text{if } p_s \leq v \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where

$$h_s(\boldsymbol{p}) = w_{T_1}\frac{1}{S} + w_{T_2}\frac{1}{\tau_s(\boldsymbol{p})+1}\delta_{\lambda_s(\boldsymbol{p}),0} \quad (5)$$

3.2 Model 2

The first model takes into account both types of buyers those who use shopbots and those who don't. The second and the third consider only customers who collect information with the help of shopbots. In the model proposed in [14] the commodity may be characterized by a number of different attributes. Nevertheless, it is assumed that for every combination of the attributes' values we can obtain a scalar quality that all the buyers agree upon (corresponds to the case when a numerical rating of the product is available — these ratings could be based on the consumer surveys or measured by an independent third party).

Each buyer b has a utility function $u_b(p, q)$, which returns a number that represents the value that b assigns to the good at that particular price p and quality q:

$$u_b(p,q) = \begin{cases} \gamma_b(q-\bar{q}_b) + (1-\gamma_b)(\bar{p}_b - p) & \text{if } (q-\bar{q}_b)(\bar{p}_b - p) \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (6)$$

where $\bar{p}_b$ is the buyer's price ceiling (the maximum price it is willing to pay), $\bar{q}_b$ is its quality floor (the minimum quality it can put up with), γ_b is a parameter that ranges between 0 and 1. A buyer with $\gamma_b = 0$ is extremely price-sensitive: it will search for the seller with the lowest price, subject to the seller provides the minimal quality level $\bar{q}_b$. A buyer with $\gamma_b = 1$ is extremely quality-sensitive: it will choose the seller with the highest quality, but the buyer can afford good only at price $\bar{p}_b$ at the most.

A buyer b can be characterized completely by its set of three parameters $(\bar{p}_b, \bar{q}_b, \gamma_b)$.

3.3 Model 3

The only difference from the model presented in Sect. 3.2 is that the authors of this virtual market model [15] emphasized practical impossibility to enforce a quality measure that all buyers can agree upon. And it is thus up to the buyer b to assess the quality q_b of the good basing on its own comparative preferences of possible characteristics.

4 Three Types of Sellers

Here we present two commonly used approaches to building pricebots along with our idea of using simple state representation for sellers based on Q-learning. The selling agents are arranged in the order as the amount of information they need on the model is decreasing.

4.1 Myopically Optimal Sellers

The myopically optimal, or *myoptimal*, pricing strategy uses information about all the buyers' characteristics and competitors' prices but assumes that the latter will remain fixed. For example, for the model presented in Sect. 3.1 we use 4 and 5 to calculate the price $p^\star$ that maximizes the expected profit π.

4.2 Q-Learning Sellers

Previous approaches to building pricebots on the base of Q-learning used presentation [11], [12] where a state consisted of a vector of the latest prices set by all competitors functioning in the market (state space of each Q-learning seller correspondingly included $|price\ range|^{number\ of\ competitors}$ states) that didn't allow the agent to further function as a new competitor wanted to present its good to the market or a existing one — to leave it. It also restricted applicability of Q-learning sellers to the case when only two pricebots compete for buyers [11], [12]. As the number of competitors increased their state spaces grew exponentially and they couldn't find prices that would maximize their cumulative profits in such a big search space. In contrast to this we propose state presentation that reflects only price of the most successful seller in the last round (and state space of each Q-learning seller consists accordingly only of $|price\ range|$ states). The competitors' prices can be again collected by shopbots. For models of virtual markets presented in Sect. 3.1 and 3.2 action space of each Q-learning seller consists of all the prices that the selling agent can set in each particular model. In model from Sect. 3.3 Q-learning agent can take actions that simultaneously adjust the price and the characteristics of the product. Rewards for all the models are the profits the agent gets as the whole buyer population accomplishes the purchase of the commodity at their favourite sellers.

4.3 Derivative-Following Sellers

The *derivative-following* strategy requires neither knowledge about the competitors' prices nor about the buyer characteristics. A pricebot based on this strategy simply increases (or decreases) its price until the observed profitability rate falls, at which point the sign of the increment is changed.

5 Experiments

As it is assumed in all three models that the number of buyers is very large. For all experiments the prices are discrete (one unit can correspond to the difference in prices discernible by the customers). For the models presented in Sect. 3.1 and 3.2 the sellers can assign discrete prices from interval 0 to 100. For the model from Sect. 3.3 — from 0 to 9. As it is not quite clear how to change price and quality characteristics simultaneously using derivative-following strategy and myoptimal sellers possessing all necessary information about customer demand can find optimal prices / parameters for the commodity very easily the comparison between the three types of sellers is made only on the basis of the first two models.

The parameters of Q-learning algorithm were the same for all three models: $\gamma = 0.9$, $\epsilon = 0.1$ $\alpha = 0.1$

5.1 Model 1

Let's consider a market where $S = 10$ sellers offer a good to B buyers, $B \gg S$. The cost of the production to all the sellers is the same. Let it be some base one, e.g., $c = 0$. The maximal price the buyers will grudgingly pay is 100 and the buyer population is divided into those who absent-mindedly select the first seller they encounter (at random) ($w_{T_1} = 25\%$) and those who will get to know all the offers (by means of shopbots) and select the seller with the lowest price ($w_{T_2} = 75\%$). Under this conditions sellers can assign prices in interval from 0 to 100.

Figure 1 shows how Q-learning sellers' average profit rates per round (corresponds to the time interval when each customer buys exactly one unit of the commodity) even out as they tend to assign the prices that maximize their cumulative profits in this highly competitive environment — 9 sellers are inclined to set the maximal price 100 (profit of each seller is $100 \cdot \frac{25\%}{10} \cdot B = 2.5 \cdot B$) while 1 seller — price 3 (profit $3 \cdot (\frac{75\%}{1} + \frac{25\%}{10}) \cdot B = (2.25 + 0.075) \cdot B = 2.325 \cdot B$). But these prices don't form a Nash equilibrium (the seller with the price 3 will gain more profit if it increases its price but then it can be undercut by the buyer with initial price 100 and its cumulative reward will be undoubtedly smaller).

Having increased prices to the maximum 100 population of derivative-following sellers as well as population of myoptimal sellers have formed monopolies. But they turned out to be not flexible enough to compete with Q-learning sellers. Figures 3 and 5 show that Q-learning sellers have gained much more profit in mixed society of sellers and more over shared profit between themselves. (This result holds for every proportion of Q-learning sellers to pricebots based on myoptimal or derivative-following strategies)

5.2 Model 2

Quality-Sensitive Buyers. In this experiment the B buyers are extremely quality-sensitive ($\gamma_b = 1$) and seek the *highest-quality* seller which price does not exceed the *price ceiling* $\bar{p}_b$ but can put up with the lowest quality $\bar{q}_b = 0$ in case of deficiency of other alternatives. We also assume that the highest prices they are willing to pay are distributed uniformly between 0 and 100 among the buyer population.

In the market only five sellers offer the good that is of interest to the mentioned buyers and these sellers are only free to set the *prices*, their *qualities* and *costs* are fixed:

	Seller 1	Seller 2	Seller 3	Seller 4	Seller 5
Quality	90	60	50	30	25
Cost	30	20	16	10	5

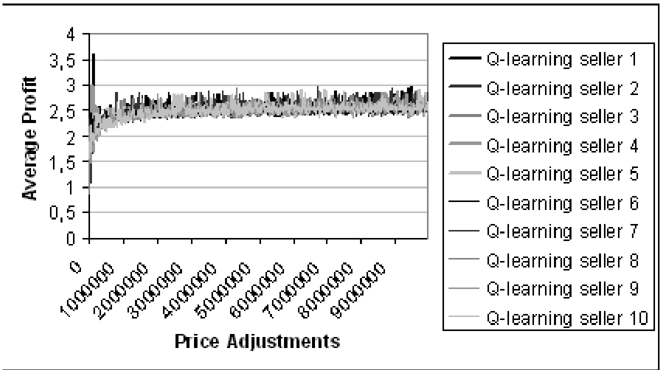


Fig. 1. Model 1: average profit rate of Q-learning sellers

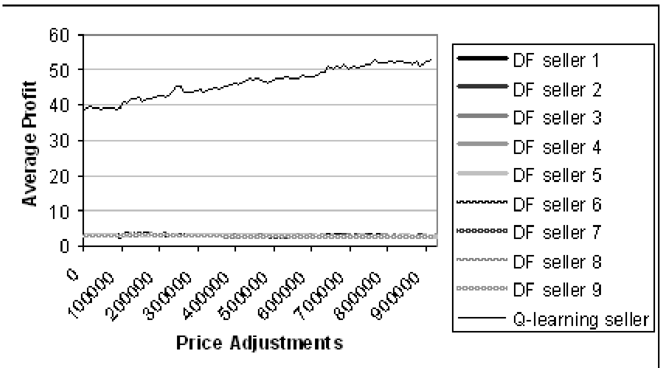


Fig. 2. Model 1: Q-learning seller versus derivative-following sellers

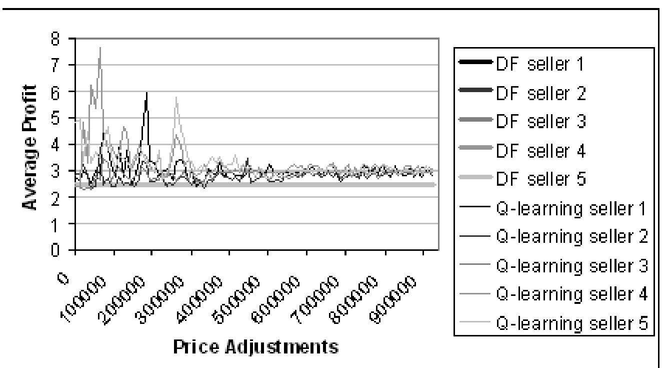


Fig. 3. Model 1: Q-learning sellers versus derivative-following sellers

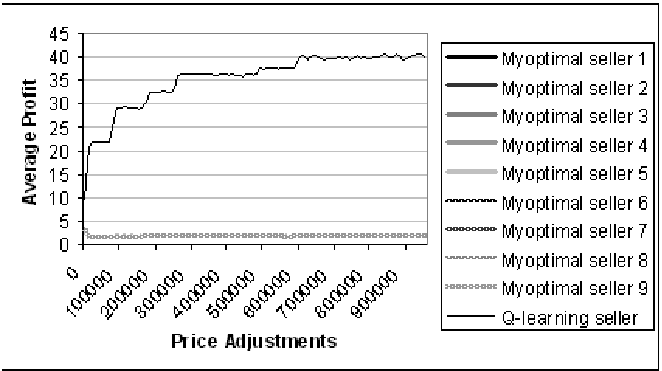


Fig. 4. Model 1: Q-learning seller versus myoptimal sellers

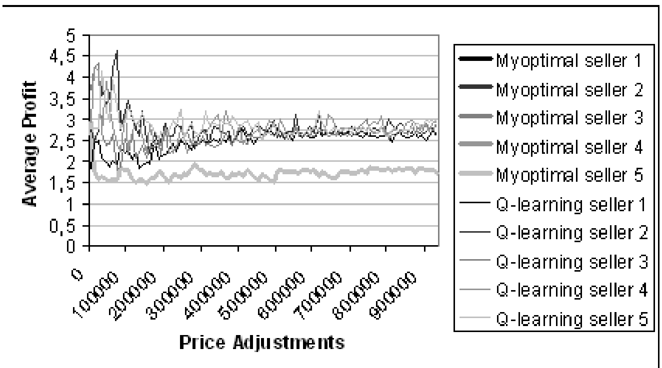


Fig. 5. Model 1: Q-learning sellers versus myoptimal sellers

Let's calculate the prices that will bring maximum profits to them.

The first seller has no competitors. All buyers who can afford the good at its price p_1 (their $\bar{p}_b$ must be not less than p_1) will rush for its offer (they are looking for the good of the highest quality). So the only thing the first seller must care for is to find the optimum between the rate of the buyers who can afford the good at its price $(1.00 - \frac{p_1}{100})$ and the profit it will get per one buyer $(p_1 - 30)$.

Its profit can be expressed as a function of a price p_1:

$Profit_1(p_1) = (1.00 - \frac{p_1}{100}) \cdot (p_1 - 30) \cdot B$

where B — the number of the buyers.

It will get maximum profit at price $p_1 = 65$.

After the first buyers bought the good of the highest quality there remain buyers who couldn't afford it. These are customers for the second seller. Repeating the above argumentation with the only difference that it can attract only $(0.65 - \frac{p_2}{100})$ percent of buyer population and the profit it will get per one buyer is $(p_2 - 20)$ we get that the best price for the second seller is $p_2 = 42.5$. Continuing we can calculate preferable prices for all the sellers: $p_1 = 65, p_2 = 42.5, p_3 = 29.25, p_4 = 19.63, p_5 = 12.31$.

These optimal prices constitute also Nash equilibrium (no seller will gain if it is the only one that changes its price).

Their maximal *profits* per round (corresponds to the time interval when each customer buys exactly one unit of the commodity) will be accordingly: $(12.25 \cdot B, 5.06 \cdot B, 1.76 \cdot B, 0.93 \cdot B, 0.54 \cdot B)$

Q-learning sellers have converged to the optimal prices. Their profit rates per round can be seen at Fig. 6. Derivative-following sellers were not flexible enough to find the optimal prices. MyOptimal sellers possessing all the information about buyers' demand and competitors' prices have converged to the optimal prices.

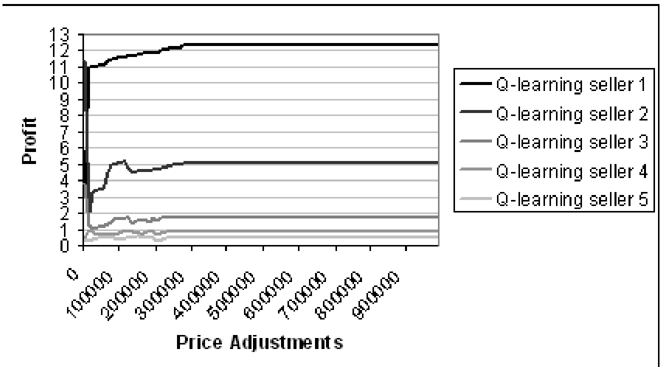


Fig. 6. Quality-sensitive buyers: profit rate of Q-learning sellers

Price-Sensitive Buyers. Suppose that the sellers introduced in Sect. 5.2 want to start selling in the market where all the buyers care only for prices $(\gamma_b = 0)$ and are satisfied with the lowest quality $\bar{q}_b$ (distributed uniformly between 0 and 100 among the buyer population). They nevertheless are rather reasonable and are willing to pay for the quality they demand (we expressed it as $\bar{q}_b = \bar{p}_b$).

In this case we can also calculate the prices maximizing the profits.

10% of buyer population (those whose lowest bearable quality $\bar{q}_b$: $91 < \bar{q}_b \leq 100$) won't be able to buy anything from our sellers (their requirements are too exacting to be satisfied even by the first seller (it can provide the good only of 90 quality rate)).

Though the first seller can count on 30% of customers (those with $\bar{q}_b$: $60 < \bar{q}_b \leq 90$) and for them it doesn't have to compete. No other offer will satisfy their rather high requirements.

Its profit function will be $Profit_1(p_1) = (0.90 - \frac{p_1}{100}) \cdot (p_1 - 30) \cdot B$ and it gets maximum at price $p_1 = 60$.

The second seller can count on those whose quality floor $\bar{q}_b \leq 60$ and its profit function will be $Profit_2(p_2) = (0.60 - \frac{p_2}{100}) \cdot (p_2 - 20) \cdot B$ that has maximum at price $p_2 = 40$. But quality requirements of 10% customers (those with $\bar{q}_b$: $40 < \bar{q}_b \leq 50$) can be satisfied both by the second and the third selling agents. And to get them the second seller must set lower price than the third one. The optimal price for the third agent will be 33 (profit function $Profit_3 = (0.50 - \frac{p_3}{100}) \cdot (p_3 - 16) \cdot B$). The second seller can of course try to undercut till the price 31 (it is the last point where its profit will be greater than the one it will get sticking to price $p_2 = 50$ — the following inequality $Profit_2(p_2) > Profit_2(50)$ holds only while $p_2 > 30$). In case the price offer of the second seller surpasses the third's one, the latter will lose profit entirely and also will sell the good at a lower price. As the third seller undercuts, the second will find itself selling the good at an extremely lower price $30 < p_2 < 33$ to 10% of buyers whose quality requirements could not be satisfied by the third offer $50 < \bar{q}_b \leq 60$ getting less than half the profit it would have gained selling the good at price 50 to them. The ensuing price-wars will inevitably lead to a considerable loss in the cumulative profit. So it is wiser for the second agent to stick to price 50.

Being guided by the same considerations we will get the prices that maximize the cumulative reward: $p_1 = 60$, $p_2 = 50$, $p_3 = 33$, $p_4 = 25$, $p_5 = 15$.

But they don't form a Nash equilibrium (the second seller will gain if it changes its price to 32 but the ensuing price-wars will inevitably lead to a considerable loss in the cumulative profit).

Maximal *profits* per round will be accordingly: $(9.00 \cdot B, 3.00 \cdot B, 2.89 \cdot B, 0.75 \cdot B, 1.00 \cdot B)$

Q-learning sellers have converged to the above prices (corresponding profit rates per round can be seen at Fig. 7). Derivative-following sellers were not flexible enough to find the prices maximizing the profits. MyOptimal sellers also haven't come to them. The average profits that myoptimal sellers get are $(9.00 \cdot B, 2.42 \cdot B, 1.81 \cdot B, 0.75 \cdot B, 1.00 \cdot B)$ (see Fig. 8) what reflects the loss of profit by the second and the third sellers as a result of their ceaseless price-wars.

5.3 Model 3

Let the good be characterized by two *attributes*:

- attribute a_1 changes its values v_{a_1} in the interval 0 to 9 and its each increase on one unit adds $\frac{1}{3}$ to the basic cost of the production ($c = 0$)
- attribute a_2 changes its values v_{a_2} in the interval 0 to 9 and its each increase on one unit adds $\frac{2}{3}$ to the basic cost of the production ($c = 0$)

Customers of the following two types constitute the *buyer population* in equal proportions:

- T_1 buyers: $\gamma_{T_1} = 1.0$ (T_1 buyers look for quality), $\bar{q}_{T_1} = 0$, $\bar{p}_{T_1} = 9$ and quality function $q_{T_1}(v_{a_1}, v_{a_2}) = 0.75 \cdot v_{a_1} + 0.25 \cdot v_{a_2}$ (characteristics a_1 is three times more important for them than a_2)

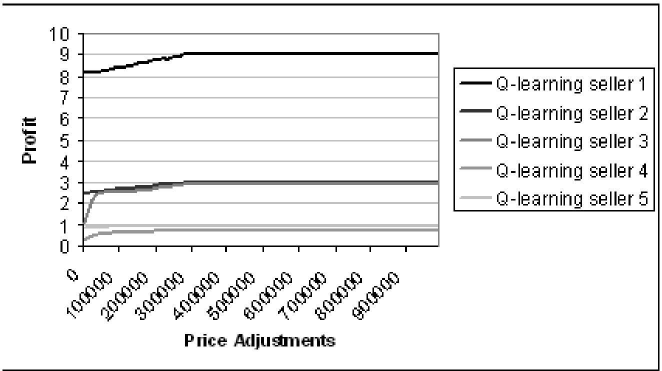


Fig. 7. Price-sensitive buyers: profit rate of Q-learning sellers

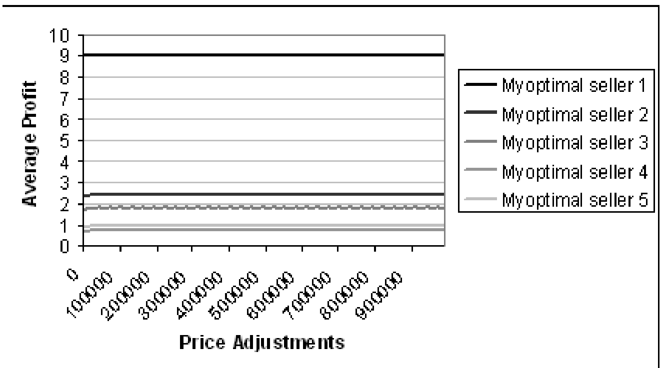


Fig. 8. Price-sensitive buyers: average profit rate of myoptimal sellers

– T_2 buyers: $\gamma_{T_2} = 0.5$ (T_2 buyers select the golden mean between price and quality), $\bar{q}_{T_2} = 0$, $\bar{p}_{T_2} = 9$ and quality function $q_{T_2}(v_{a_1}, v_{a_2}) = 1.00 \cdot v_{a_2}$ (the attribute a_1 is absolutely of no interest to them)

After 200000 price adjustments Q-learning sellers came to the following distribution:

– 5 sellers offer the good with characteristics $v_{a_1} = 9$ and $v_{a_2} = 8$ at price 9 (trying to attract buyers of type T_1)
– 5 sellers offer the good with characteristics $v_{a_1} = 0$ and $v_{a_2} = 9$ at price 7 (counting on buyers of type T_2)

Buyers of the first type are quality hunters and maximal price they can pay is 9. Q-learning sellers are extremely competitive. So through constant quality increasing they have come to selling good of the extremely high characteristics that still brings profit (good with parameters $v_{a_1} = 9$ and $v_{a_2} = 9$ wouldn't bring any profit while good with characteristics $v_{a_1} = 8$ and $v_{a_2} = 9$ is not so attractive to buyers of type T_1 because the first characteristics is three times more desirable to them).

Sellers of the second type are trying to attract buyers of type T_2. They have set $v_{a_1} = 0$ for the first parameter only increases cost and of no interest to the buyers T_2. But why they decided to offer a good at price 7 but with the highest quality 9? There can be different golden means between price and quality that buyers T_2 appreciate so much. For the seller one unit increase in quality adds only $\frac{2}{3}$ to its cost (so decreases its profit on $\frac{2}{3}$) and one unit reduction in price will decrease the profit exactly on one unit. And as buyers have no preferences between decreasing the price or increasing the quality on the same amount it is the most profitable to sell the good of the highest possible quality. And 7 is the lowest price that allows to stop competitors from undercutting but at the same time to make some profit.

The total profit per round for sellers of the second type is $(7-9\cdot\frac{2}{3})\cdot\frac{B}{2} = 1\cdot\frac{B}{2}$, where B — the number of the buyers.

While selling the good of the same level of attractiveness ($v_{a_2} = 8$ and price 6) would bring only $(6 - 8\cdot\frac{2}{3})\cdot\frac{B}{2} = 0.67\cdot\frac{B}{2}$

But why have they distributed in this way (5 sellers of the first type and 5 of the second)?

Every round each of the sellers of the first type gets profit: $\frac{9-(9\cdot\frac{1}{3}+8\cdot\frac{2}{3})}{5}\cdot\frac{B}{2} = \frac{9-(3+5.33)}{5}\cdot\frac{B}{2} = 0.134\cdot\frac{B}{2}$

Each of the sellers of the second type gets: $\frac{7-(9\cdot\frac{2}{3})}{5}\cdot\frac{B}{2} = 0.20\cdot\frac{B}{2}$

It's very close to the true Nash equilibrium where the sellers of the first type must number 4 and second — 6 (profits per seller $0.167\cdot\frac{B}{2}$ and $0.167\cdot\frac{B}{2}$ accordingly).

6 Conclusion

This paper is devoted to creating a pricebot — an economically motivated agent that adjusts prices so as to maximize the profits of a firm trading in Internet. One of the most promising approaches to building pricebots is Q-learning. New state representation for Q-learning sellers in virtual markets was proposed that allows to increase the number of sellers arbitrarily (before their applicability was restricted to virtual markets with only two sellers, what was too unrealistic). It was shown that Q-learning sellers with the proposed state representation are capable of finding the prices and the commodity characteristics maximizing their cumulative profits under conditions of high competition even when the buyers' tastes are of the most sophisticated kind. The comparison of Q-learning sellers with pricebots based on two widely used strategies — derivative-following and myopically optimal showed that the latter both are not flexible enough to compete with Q-learning sellers let alone the fact that myopically optimal sellers require precise information about buyer demand and competitors' prices that is unobtainable in the changing conditions of virtual markets.

References

1. http://dealtime.com
2. http://whenu.com
3. http://www.bizrate.com
4. http://www.mysimon.com
5. http://www.pricescan.com
6. DeLong, J.B., Froomkin, A.M.: Speculative Microeconomics for Tomorrow's Economy. First Monday, **5** (2) (2000)
7. Rosenschein, J.S., Zlotkin, G.: Designing conventions for automated negotiation. AI Magazine, **15** (3) (1994) 29–46
8. Greenwald, A.R., Kephart, J.O.: Shopbots and pricebots. In Dean, T., ed.: IJCAI, Morgan Kaufmann (1999) 506–511
9. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. The MIT Press (1998)
10. Watkins, C.J.C.H., Dayan, P.: Technical note q-learning. Machine Learning, **8** (1992) 279–292
11. Tesauro, G., Kephart, J.O.: Pricing in agent economies using multi-agent q-learning. Autonomous Agents and Multi-Agent Systems **5** (3) (2002) 289–304
12. Tesauro, G.: Pricing in agent economies using neural networks and multi-agent q-learning. In Sun, R., Giles, C.L., eds.: Sequence Learning. Volume 1828 of Lecture Notes in Computer Science. Springer (2001) 288–307
13. Mitchell, T.: Machine Learning McGraw-Hill (1997)
14. Sairamesh, J., Kephart, J.O.: Price dynamics of vertically differentiated information markets. In: ICE '98: Proceedings of the first international conference on Information and computation economies, New York, NY, USA, ACM Press (1998) 28–36
15. Vidal, J.M., Durfee, E.H.: Learning nested agent models in an information economy. Journal of Experimental and Theoretical Artificial Intelligence, **10** (3) (1998) 291–308

Fusion of Dependence Networks in Multi-agent Systems – Application to Support Net-Enabled Littoral Surveillance

Mohamad K. Allouche and Éloi Bossé

Defence Research & Development Canada - Valcartier
2459 Pie XI North, Québec, QC, G3J 1X5 Canada
Mohamad.Allouche@drdc-rddc.gc.ca, Eloi.Bosse@drdc-rddc.gc.ca

Abstract. In a multi-agent system, agents need to identify their relationships in order to coordinate their actions. Actually, coordination is the most challenging issue in multi-agent design. Very often, the definition of agents' relationships depends on the properties of the tasks they are able to perform. Knowing its task, an agent is able to build a dependence network that helps identify other agents with whom to interact and cooperate. An agent can also fuse branches in its dependence network in order to derive dependence relations between other agents. The fusion of dependence relations is a powerful tool that could be used by agents to coordinate their tasks and enhance the whole system performance.

Keywords: Fusion Dependence Networks, Multi-Agent Systems, Social Reasoning.

1 Introduction

Making a group of agents work together is one of the most challenging issues in multi-agent design. Success depends essentially on their capacity to coordinate their activities during the system functioning. Dependence networks theory [1-2] was a major finding in the field. Simply put, interactions between agents should be based and motivated by the identification of relationships that relate them. In this work, we are interested in dependence relations between agents that are derived from data mining activities and from properties of the tasks that these agents are likely to perform.

The fusion of dependence relations is a powerful tool for social reasoning among agents. It could be used for better resource management, planning and coordination [3-4]. The fusion of dependence relations is illustrated with an example related to support large-area surveillance missions, which require efficient deployment of multiple search resources and rapid extraction and distribution of key information.

2 Dependence Relations

In this study, we consider a set of agents $A = \{\ldots, a_u, a_v, a_w, \ldots\}$ and a set of tasks $T = \{\ldots, p_i, p_j, p_k, \ldots\}$. Each agent a_u is able to execute a set tasks defined by the function

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 206–211, 2007.
© Springer-Verlag Berlin Heidelberg 2007

resp: A → T where *resp*(a_u) is the set of tasks that a_u is able to perform. The assignment of tasks to agents (*resp* values) is beyond the scope of this document. The current work is directly based on [3-4]. Three different types of dependence relations were identified: Needs, Competes, and Helps.

In this work, we suppose that agents perform some data mining activities in order to identify their dependence relations, and hence to build their dependence networks, but also to know how to use these networks in their social reasoning and problem solving. The dependence relations are based on relations between tasks. We are interested in two kinds of relations: the subtask relation and the non-compatibility relation. The function *subtask*(p_i) returns all the subtasks of p_i. The *opp*(p_i, p_j) has a true value if p_i and p_j are non-compatible, that is, only one execution of the two tasks will succeed if both tasks are executed simultaneously. We suppose that we have a set of tasks with predefined relation between them. In the rest of the paper, we use the operator *do* to indicate that an agent is executing a task over an interval of time: $do_I(a_u, p_i)$ means that agent a_u is executing task p_i over the interval I.

The Needs dependence is defined between two agents a_u and a_v, and regarding two tasks p_i and p_j where p_j is a subtask of p_i and is not part of a_u tasks (*resp*(a_u)). The Needs relation states that in order to execute task p_i agent a_u needs the services of a_v to execute task p_j. A Needs dependence is not exclusive: two agents may depend on the same agents for the same task and regarding two different tasks. The atemporal and temporal definitions of a Needs dependence are given by Eqs. 1-2 in Table 1 (see [5] for more details).

The Competes dependence states that two agents a_u and a_v are both capable of executing the same task p_i (Eq. 3 in Table 1). A Competes dependence is activated over any interval I during which a_u or a_v is executing task p_i. We used an exclusive or ($\succ\prec$) because only one the two agents is supposed to execute p_i and not both of them (Eq. 4 in Table 1). The *d*, *s*, *f*, *e* correspond to Allen's relations [5] between intervals: "*during*", "*start*", "*finish*", "*equal*".

Finally, the Helps dependence is defined for two agents a_u and a_v, regarding two non-compatible tasks p_i and p_j. By non-compatible we mean that those tasks cannot be achieved successfully at the same time. For example, any tasks that must use a single resource are considered as non-compatible tasks. A Helps dependence is used to prevent an agent from undertaking a task while another agent is executing a non-compatible task (Eq. 5 in Table 1).

The non-compatibility of the tasks p_i and p_j is defined by the *opp* relation. In the above definition a_u is not able to execute p_j and a_v is not able to execute p_i. A Helps dependence is activated over any interval of time I during which either a_u is executing p_i or a_v is executing p_j (Eq. 6 in Table 1).

3 Fusion of Dependence Relations

We define the fusion operator ⊕ to derive a new dependence relation from two existing ones. The fusion is defined for Competes-Competes, Competes-Needs, Competes-Helps, Needs-Needs and Helps-Helps dependence relations. All other combinations will not be considered since they do not present any interesting property and agents cannot use them in their social reasoning.

3.1 Competes-Competes Fusion

An agent having two Competes relations with other agents regarding the same task can fuse these relations to derive a Competes relation between these agents regarding the same task. The a-temporal and temporal definitions that rule the activation-deactivation of the derived dependence relation are given by Eqs. 7-8 in Table 1.

The activation of the derived dependence relations is ruled by the activation of the original dependence relations that relate a_u to a_v and a_w. If one of these relations is activated over an interval I, then both derived relations are activated over this interval.

The context of use of a Competes-Competes fusion is the following:
If a_u is executing task p_i then both Competes(a_u, a_v, p_i) and Competes(a_u, a_w, p_i) are activated. Agent a_u will also activate Competes(a_w, a_v, p_i) and Competes(a_v, a_w, p_i), that is, after the execution of p_i, both agents a_v and a_w will be informed about termination of task p_i ; if agent a_v (resp. a_w) is executing task p_i then Competes(a_u, a_v, p_i) (resp. Competes(a_u, a_w, p_i)) is activated. Agent a_u will also activate Competes(a_w, a_v, p_i) and Competes(a_v, a_w, p_i), that is, agent a_u knows that once p_i is executed agent a_w (resp. a_v) will be informed of the termination of task execution.

3.2 Competes-Needs Dependence Fusion

As for the Competes-Competes fusion, it is possible to fuse a Competes with a Needs relation in order to derive a Needs relation (Eq. 9 in Table 1). This definition means that if a_u has a Competes relation with a_v regarding task p_i and a Needs dependence with a_w, the same dependence relation between a_v and a_w could be derived. The activation rule for the derived dependence relation is given by Eq. 10 in Table 1.

If the Competes dependence between a_u and a_v is activated and the agent a_v is executing the task p_i then a_u can activate the derived relation between a_v and a_w. Actually, this activation means that a_v will need the services of a_w to execute task p_j. The *o*, *m*, *di*, *e* correspond to Allen's interval relations: "*overlap*", "*meet*", "*during inverse*" and "*equal*". The context of use of a Competes-Needs fusion is the following: the activation of Competes(a_u, a_v, p_i) means that a_u should not activate Needs(a_u, a_w, p_i, p_j) as long as Competes(a_u, a_v, p_i) is activated: a_u knows that task p_j will be executed by a_w upon the request of agent a_v (Needs(a_v, a_w, p_i, p_j)); the activation of Needs(a_u, a_w, p_i, p_j) is not relevant and could not be of any use for agent a_u. Actually, this means that p_j is being executed by agent a_w.

3.3 Competes-Helps Dependence Fusion

An agent a_u having a Competes relation with a_v regarding task p_i and a Helps dependence with a_w, may derive the same dependence relation between a_v and a_w as in Eq. 11 in Table 1. This definition means that if a_u has a Competes relation with a_v regarding task p_i and a Helps dependence with a_w because p_i and p_j are non-compatible. From these relations, a_u can derive that a_v and a_w have the same dependence relation. The activation rule for the new derived Helps dependence is the following: if $Competes_I$(a_u, a_v, p_i) is activated, then a_v is executing task p_i. Since Helps(a_v, a_w, p_i, p_j) is activated then a_v prevents a_w from executing task p_j for non-compatibility reason. In this case, a_u doesn't need to activate Helps(a_u, a_w, p_i, p_j). The activation of Helps(a_u, a_w, p_i, p_j) has no effect on the activation of the derived dependence relation (Eq. 12 in Table 1).

3.4 Needs-Needs Dependence Fusion

An agent a_u having the following relations with agent a_v and a_w, Needs(a_u, a_v, p_i, p_j) and Needs(a_u, a_w, p_k, p_j), can derive two new dependence relations between a_v and a_w, Competes(a_v, a_w, p_j) and Competes(a_w, a_v, p_j) (Eq. 13 in Table 1). The activation rule for the derived relations is given by Eq. 14 in Table 1. The reasoning is the same if Needs(a_u, a_w, p_k, p_j) is activated instead of Needs(a_u, a_v, p_i, p_j). The context of use of these derived relations is the following: suppose that Needs(a_u, a_v, p_i, p_j) is activated. The activation of Competes(a_v, a_w, p_j) and Competes(a_w, a_v, p_j) means that either a_v or a_w is executing p_j. If a_u needs to execute p_k, it doesn't need to ask a_w to execute p_j again (subtask of p_k). It will wait for p_j execution is terminated and then use the result for both p_i and p_k. The reasoning is the same if we suppose that Needs(a_u, a_w, p_k, p_j) is activated instead of Needs(a_u, a_v, p_i, p_j).

3.5 Helps-Helps Dependence Fusion

An agent a_u having the following relations with agent a_v and a_w, Helps(a_u, a_v, p_i, p_j) and Helps(a_u, a_w, p_k, p_j), can derive two new dependence relations between a_v and a_w, Competes(a_v, a_w, p_j) and Competes(a_w, a_v, p_j) (Eq. 15 in Table 1). It is important to mention that since agent a_u has a Helps relation with agent a_w, Helps(a_u, a_w, p_k, p_j), it has also another Help relation regarding p_i, Helps(a_u, a_w, p_i, p_j). This is due to the fact

Table 1. Needs, Competes and Helps equations

$$Needs(a_u,a_v,p_i,p_j) \triangleq p_i \in resp(a_u) \wedge p_j \in resp(a_v) \cap subtask(p_i) \wedge p_j \notin resp(a_u) \quad (1)$$

$$Needs_I(a_u,a_v,p_i,p_j) \Leftrightarrow \exists J,K \mid Needs(a_u,a_v,p_i,p_j) \wedge do_J(a_u,p_i) \wedge do_K(a_v,p_j) \wedge I = J \cap K \quad (2)$$

$$Competes(a_u,a_v,p_i) \triangleq p_i \in resp(a_u) \cap resp(a_v) \quad (3)$$

$$Competes_I(a_u,a_v,p_i) \Leftrightarrow \exists J \mid competes(a_u,a_v,p_i) \wedge (do_J(a_u,p_i) \succ\!\prec do_J(a_v,p_i)) \wedge I\{d,s,f,e\}J \quad (4)$$

$$Helps(a_u,a_v,p_i,p_j) \triangleq opp(p_i,p_j) \wedge p_i \in resp(a_u) - resp(a_v) \wedge p_j \in resp(a_v) - resp(a_u) \quad (5)$$

$$Helps_I(a_u,a_v,p_i,p_j) \Leftrightarrow \exists J \mid Helps(a_u,a_v,p_i,p_j) \wedge (do_J(a_u,p_i) \vee do_J(a_v,p_j)) \wedge I\{s,d,f,e\}J \quad (6)$$

$$Competes(a_u,a_v,p_i) \oplus Competes(a_u,a_w,p_i) \triangleq Competes(a_w,a_v,p_i) \wedge Competes(a_v,a_w,p_i) \quad (7)$$

$$Competes_I(a_u,a_v,p_i) \succ\!\prec Competes_I(a_u,a_w,p_i) \Rightarrow Competes_I(a_w,a_v,p_i) \wedge Competes_I(a_v,a_w,p_i) \quad (8)$$

$$Competes(a_u,a_v,p_i) \oplus Needs(a_u,a_w,p_i,p_j) \triangleq Needs(a_v,a_w,p_i,p_j) \quad (9)$$

$$Competes_I(a_u,a_v,p_i) \wedge Needs(a_u,a_w,p_i,p_j) \wedge do_J(a_v,p_i) \wedge J\{o,m,di,e\}I \Rightarrow Needs_I(a_v,a_w,p_i,p_j) \quad (10)$$

$$Competes(a_u,a_v,p_i) \oplus Helps(a_u,a_w,p_i,p_j) \triangleq Helps(a_v,a_w,p_i,p_j) \quad (11)$$

$$Competes_I(a_u,a_v,p_i) \wedge Helps(a_u,a_w,p_i,p_j) \wedge do_J(a_v,p_i) \wedge J\{o,m,di,e\}I \Rightarrow Helps_I(a_v,a_w,p_i,p_j) \quad (12)$$

$$Needs(a_u,a_v,p_i,p_j) \oplus Needs(a_u,a_w,p_k,p_j) \triangleq Competes(a_v,a_w,p_j) \wedge Competes(a_w,a_v,p_j) \quad (13)$$

$$Needs_I(a_u,a_v,p_i,p_j) \wedge Needs(a_u,a_w,p_k,p_j) \Rightarrow Competes_I(a_v,a_w,p_j) \wedge Competes_I(a_w,a_v,p_j) \quad (14)$$

$$Helps(a_u,a_v,p_i,p_j) \oplus Helps(a_u,a_w,p_k,p_j) \triangleq Competes(a_v,a_w,p_j) \wedge Competes(a_w,a_v,p_j) \quad (15)$$

$$Helps_I(a_u,a_v,p_i,p_j) \wedge Helps(a_u,a_w,p_k,p_j) \Rightarrow Competes_I(a_v,a_w,p_j) \wedge Competes_I(a_w,a_v,p_j) \quad (16)$$

$$Competes(GS, HMCS, Surveillance) \oplus Needs(GS, AURORA, Surveillance, UAV) \triangleq Needs(HMCS, AURORA, Surveillance, UAV) \quad (17)$$

$$Competes(HMCS, GS, Surveillance) \oplus Needs(HMCS, AURORA, Surveillance, UAV) \triangleq Needs(GS, AURORA, Surveillance, UAV) \quad (18)$$

that p_i and p_j are non-compatible according to the relation Helps(a_u, a_v, p_i, p_j) between a_u and a_v. The activation rule for the derived relations is given by (Eq. 16 in Table 1).

This rule means that the derived relations will be activated as long as one of the Helps relations is activated. The reasoning is the same if Helps(a_u, a_w, p_k, p_j) is activated instead of Helps(a_u, a_v, p_i, p_j). The context of use of the derived Competes relations is the following: suppose that Helps(a_u, a_v, p_i, p_j) is activated. The activation of Competes(a_v, a_w, p_j) and Competes(a_w, a_v, p_j) means that either a_v or a_w is executing p_j. This means that a_u has to wait until these relations are de-activated before executing p_k because it is non-compatible with p_j.

4 Application to Net-Enabled Littoral Surveillance

In this example we consider three agents, the first agent, called GS, is assigned to a Ground Station. The second agent, called HMCS, is assigned to Her Majesty's Canadian Ship. The last agent, called Aurora, is assigned to the Aurora CP-140 aircraft. We consider also two tasks: the first one is called Surveillance and consists in performing the surveillance by deploying necessary assets in a specific area. The second task, which is a subtask of task Surveillance, is called UAV. It consists in dispatching a UAV in a specific area in order to collect information of primary information to both operators and commanders in the GS and onboard the HMCS.

A = {GS, HMCS, Aurora}, T = {Surveillance, UAV}

The only relation between tasks, we use in the example is that UAV task is a subtask of the global Surveillance task. --- *subtask*(Surveillance) = UAV. The function *resp* is defined as follows: *resp*(GS) = Surveillance, *resp*(HMCS) = Surveillance, *resp*(Aurora) = UAV. Based on the previous definitions of *subtask* and *resp* functions, each agent is able to build a dependence network. Agent GS has one Competes relation with HMCS regarding Surveillance task and a Needs dependence with Aurora regarding tasks Surveillance and UAV.

Competes(GS, HMCS, Surveillance), Needs(GS, Aurora, Surveillance, UAV)

Agent HMCS has also two dependence relations: One competes relation with GS regarding task Surveillance and also a Needs dependence with agent Aurora regarding tasks Surveillance and UAV. Finally, agent Aurora has an empty dependence network.

Competes(HMCS, GS, Surveillance), Needs(HMCS, Aurora, Surveillance, UAV)

Agent GS derives a Needs relation between agents HMCS and AURORA regarding tasks Surveillance and UAV (Eq. 17 in Table 1). Agent HMCS derives a Needs relation between agents GS and AURORA regarding tasks Surveillance and UAV (Eq. 18 in Table 1). Agent AURORA cannot derive any dependence relation with an empty dependence network.

In order to illustrate the efficiency of dependence relation fusion in order multi-agent coordination, we consider the following situation: The Marine Operations Center has received information that reports a suspicious activity 20 knots NE from the drug vessels. The GS has been charged to send the necessary assets in order to assess the situation. The HMCS is already executing a Surveillance task in the area. Agent GS has the relation Competes(GS, HMCS, Surveillance) activated. The derived relation Needs(HMCS, AURORA, Surveillance, UAV) is also activated according to the activation rule described in Section 3.2. Due to this activation, GS will not activate the relation Needs(GS, AURORA, Surveillance, UAV) by tasking the AURORA to send another UAV to the area. It will instead ask reports from the same UAV tasked the HMCS in the area.

5 Conclusion

This paper presented the fusion of dependence relations as a powerful tool that could be used by agents to coordinate their tasks and enhance the whole system performance. The fusion of dependence relations was illustrated with an example related to support large-area surveillance missions, which require efficient deployment of multiple search resources and rapid extraction and distribution of key information.

References

1. Castelfranchi, C., Miceli, M., and Cesta, A.: *Dependence relations among autonomous agents*. In E. Werner and Y. Demazeau, editors, Decentralized AI 3, Elsevier Science Publishers (1992) 215–227
2. Jennings, N.R.: Coordination Techniques for Distributed Artificial Intelligence. In G.M.P. O'Hare and N.R. Jennings, editors, Foundations of Distributed Artificial. John Wiley and Sons (1996)
3. Allouche, Mohamad K. : “Une société d’agents temporels pour la supervision de systèmes industriels”, Ph.D. Thesis, Ecole Nationale Supérieure desMines de St-Etienne, St-Etienne, France (1998)
4. Allouche, M., Boissier, O., Sayettat, C.: Temporal Social Reasoning in Dynamic Multi-Agent Systems. ICMAS 2000: 23–30, Boston, MA.
5. Allen, J.F.: Maintaining Knowledge about Temporal Intervals.' Communications of the ACM 26, 11, November (1983) 832–843

Multi-agent Framework for Simulation of Adaptive Cooperative Defense Against Internet Attacks

Igor Kotenko and Alexander Ulanov

Computer Security Research Group, St. Petersburg Institute for Informatics and Automation
39, 14 Liniya, St.-Petersburg, 199178, Russia
{ivkote,ulanov}@comsec.spb.ru

Abstract. The paper proposes the framework for investigation of prospective adaptive and cooperative defense mechanisms against the Internet attacks. The approach suggested is based on the multi-agent modeling and simulation. According to the approach the defense and attack systems are represented as interacting teams of intelligent agents that act under some adaptation criterion. They adjust their configuration and behavior in compliance with the network conditions and attack (defense) severity. The paper represents the architecture and software implementation of simulation environment that combines discrete-event simulation, multi-agent approach and packet-level simulation of various Internet protocols. The environment allows to simulate complex attack and defense scenarios. The paper describes the experiments aimed on the investigation of adaptive "Distributed Denial of Service" attacks and defense mechanisms.

Keywords: adaptation and learning in multi-agent systems, applications, security simulation, network attacks and defense.

1 Introduction

From the conceptual point of view the main requirements to a computer network defense are defined by the following main conditions: system objects have to be defended from unauthorized access and attacks, there have to be implemented the mechanisms of integrity support of defense system and defended objects, availability of information resources has to be provided.

Traditional tools and mechanisms for computer network security are more oriented on the defense from particular threats and attack types and, as usual, are implemented as a set of software and hardware components functioning independently. The approaches used in present reside in a whole series of drawbacks. As investigations show they fail to solve effectively the problem of security management in real time. These drawbacks are caused mainly by particular specialization of security tools, uncultivated mechanisms of defense verification on the stages of creation and support, the static mechanisms of defense and configuration of computer networks, the unequal mechanisms of risk analysis, security assessment, monitoring of networks state and adaptation to the change of work conditions.

According to the contemporary view the prospective network defense system has to be interconnected and multi-echeloned system. It has to detect and react operative

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 212–228, 2007.
© Springer-Verlag Berlin Heidelberg 2007

on remote and local computer attacks and unauthorized operations. It is needed to carry continuous control of network functioning and analysis of possible risks, collect knowledge about counteraction, detection and reaction methods and use them for defense reinforcement. This system has to be adaptive and evolve dynamically with the change of work conditions. The change of conditions is interpreted in the wide sense: on the one hand, they are connected with the change of business processes that are to be defended; on the other hand, they are defined by the change of environment, including the evolution of software and hardware, threats modification, current oscillations of network state, etc.

To implement these possibilities (and, first of all, adaptive security possibilities) in prospective defense system one must implement the dynamic behavior, autonomy and adaptation of particular components, the use of methods based on negotiations and cooperation that lie in the basis of multi-agent systems and (or) autonomic computing. Furthermore, the prospective defense system has to provide at least three levels of security. First level contains "traditional" static defense tools (components) implementing functions of identification and authentication, cryptographic protection, access control, integrity, auditing, firewalls, etc. Second level includes tools (components) of proactive defense that provide the collection of needful information, security assessment, network state monitoring, attack detection, counteraction to their realization, malefactor deception, etc. Third level corresponds to such tools (components) of defense management that do integral evaluation of network state, defense management and adaptation of single defense mechanisms and security policies. So this third adaptive defense level is built on top of various non-adaptive security mechanisms, which makes it applicable for a wide range of network security defenses. We have to mention that these three levels can be applied both to particular defense subsystems and to the integrated defense system as a whole.

The development of defense mechanisms that correspond to the second and especially to the third level and implement in essence the intelligent superstructure above traditional defense mechanisms in present is the foreground task in the domain of theoretical and application research in information security. Therefore it is very important to *develop flexible, cooperative, adaptive, distributed defense mechanisms as well as to determine the optimal defense strategies by their research simulation.*

A major threatening type of network attacks is distributed denial-of-service (DDoS) flooding ones [28]. They aim to overload host or network resources by flooding the system with huge quantity of network packets. These attacks are implemented by many software agents (bots or daemons) distributed on hosts that the attacker has compromised before. Effective DDoS defense which includes attack prevention, attack detection, tracing the attack source and attack counteraction is a very complicated problem. The main task of defense is to accurately detect DDoS attacks, quickly respond to them and recognize the legitimate traffic that shares the attack signature and deliver it reliably to the victim [28]. Adequate protection can only be achieved by cooperation of different distributed components.

The paper proposes the multi-agent framework and software environment for simulation of malefactors and defense system counteraction. They are intended to investigate the cooperative adaptive defense systems against the Internet DDoS attacks.

The paper is structured in the following way. *Second section* describes the relevant papers, research goal and the features of proposed approach. *Third section* presents the framework for multi-agent simulation of defense systems in the Internet, main aspects of its application for the investigation of defense against “Distributed Denial of Service” attacks and the architecture and current implementation of multi-agent simulation environment. *Forth section* outlines the adaptation and agent learning mechanisms suggested. *Fifth section* describes the parameters for testing of defense mechanisms and presents the results of experiments conducted. *Conclusion* outlines the main results and future work directions.

2 Related Work and Research Goal

The approach proposed in the paper is based on research in several fields. First of all they are multi-agent systems and agent-oriented simulation, defense mechanisms against distributed attacks, autonomic computing, adaptive behavior, etc.

The main basis for the research is *multiagent systems and the agent teamwork theory*. There are three well-known approaches to the formalization of the agent teamwork – joint intentions theory [6], shared plans theory [12] and the hybrid approaches [35, 39] which use the combination of joint intentions and shared plans theories. A lot of teamwork approaches are implemented in various multi-agent software, e.g. GRATE*, OAA, CAST, RETSINA-MAS, COGNET/BATON, Team-Soar, etc.

To implement the multi-agent modeling and simulation, we intended to develop a multi-level simulation environment which differs from the well-known *agent-oriented simulation tools* (e.g. CORMAS, Repast, Swarm, MadKit, MASON, NetLogo, etc.) [25, 27], first of all, by the use of simulation tools that allow to adequately simulate the network protocols and security processes. From the other side there are a lot of tools which can be used for *network simulation*: NS2 [31], OMNeT++ INET Framework [32], SSF Net [38], J-Sim [17], etc. To choose the necessary tool we fulfilled the detailed analysis of these simulation environments [22].

Traditional defense from *DDoS* includes detection and reaction mechanisms. To detect abnormal network characteristics, many methods can be applied (for instance, statistical, cumulative sum, pattern matching, etc). The examples of detection methods are Hop counts Filtering (HCF) [14], Source IP address monitoring (SIPM) [36], Bit per Second (BPS), etc. As a rule, the reaction mechanisms include filtering, congestion control and traceback. As detection of DDoS is most accurate when it is close to the victim hosts and separation of legitimate is most successful close to the sources, adequate victim protection to constrain attack traffic can only be achieved by *cooperation of different distributed components* [28]. There are a lot of architectures for distributed cooperative defense mechanisms [4, 19, 20, 29, 34, 42, etc.].

The problem of adaptation is considered in many papers. A generic model for *dynamic adaptation* at runtime was presented in [15]. The basic principles of *autonomic computing* are represented in [13, 18, 40]. They are self-healing, self-configuration, self-optimization and self-protection. Very important direction of research in cyber defense – *survivability and intrusion tolerance* – focuses on augmenting existing computer systems with adaptive defenses. In many intrusion-tolerant systems, auto-adaptive capabilities are a standard feature [3]. The papers [1, 2, 3, 41] describe a middleware-based survivability approach and toolkit which lets an application and its underlying infrastructure respond to attacks based on a defense strategy determined by its survivability requirements. The Willow architecture [21] achieves intrusion tolerance using a combination of fault avoidance (by disabling vulnerable network elements), fault elimination (by replacing system software elements), and fault tolerance (by reconfiguring the system). S.Cheng et al. consider in [5] a solution that introduces a self-management coordination architecture to support a composition of different self-management modules. These modules can monitor system behavior and adapt the system at run time to configure the system, improve its performance, recover from faults, etc. The incremental adaptation approach which is based on using externalized mechanisms is suggested in [7, 8, 11, 33]. Another significant adaptation approach is in using of *artificial immune systems (AIS)* due to their ability to adapt to continuously changing environments [16]. Many applications in computer network security including distributed intrusion detection have been developed by combining AIS with different computational intelligence methods (fuzzy systems, neural networks, evolutionary computation, DNA computing, etc.) [30].

The series of papers are devoted to the implementation of adaptive approach to DDoS defense. The technical report [37] examines the ability to change behavior dynamically to sustain service in response to DDoS. The Saber system [19] uses intrusion detection, automatic code patching, process migration, and attack filtering for coherent defense. In [9] an approach and a system for granularity-adaptive attack detection are considered. Zou et al. introduced in [43] an "adaptive defense" principle based on cost minimization and presented adaptive defense designs for defending against SYN flood DDoS and worm infection.

This paper proposes the multi-agent approach and software environment for simulation of malefactors and defense systems counteraction in the Internet. The results of this investigation are based undoubtedly on the mentioned relevant papers. But the authors have not found the close approaches to simulation of adaptive cooperative defense from distributed attacks. *We have to emphasize that the goal of this paper is not in developing the new adaptive DDoS defense methods, but in investigation of possibility of application the agent-oriented approach and simulation environment being developed to the defense mechanisms simulation and, first of all, to cooperative adaptive DDoS defense mechanisms.* The proposed approach is based on the representation of network systems as the complex of interacting intelligent agent teams that can be in the states of antagonistic counteraction, indifference and cooperation. Aggregated system behavior lies in local interaction of single agents. The behavior of antagonistic teams is based on the use of some adaptation criterion. According to this criterion, the antagonistic teams (attack and defense systems) adjust their configuration and behavior according to the network conditions and behavior of adversarial team, for example, attack or defense severity. The main paper results are shown on the basis of investigating the adaptive cooperative defense against DDoS. In the previous papers of authors [22-24] the approach to simulation of malefactors and defense systems counteraction was not based on the use of adaptation criterions. There were examined only simplified non-cooperative defense scenarios. The further improvement of the formal framework and simulation environment allowed to implement the complex adaptive attack and defense scenarios described in this paper.

3 Multi-agent Framework and Environment

The *general conceptual model of cybernetic agents counteraction and cooperation* includes [10]: (1) The ontology of application containing application notions and relations between them (we differentiate the problem ontology, the shared application ontology, the application ontology of particular team and particular agent); (2) The protocols of teamwork for the agents of different teams; (3) The models of scenario behavior of agents for team, group and individual levels; (4) The libraries of agent basic functions; (5) The communication platform and components for agent message exchange; (6) The models of environment – the computer network, including topological and functional components. Coordination of agent actions in team (group, personal) is based on a general plan and the procedures of actions coordination, monitoring, recovery of agent functionality and communication selectivity.

The paper uses the following assumption about the structure and functionality of attack and defense agent teams [22, 23]. *DDoS attacks agents* are divided at least into two classes: "daemon" and "master". Daemons are attack executors. Master coordinates them. The attack mode can be specified, for example, by the intensity of packet sending (packets per second) and the method of IP address spoofing. According to the *general DDoS defense approach* the defense agents are classified into the following classes [24]: information processing ("sampler"); attack detection ("detector"); filtering ("filter"); investigation and deactivation of attack agents ("investigator").

Defense agent teams can interact due to various schemas. For example, one of the schemas is as follows. At detecting the attack start, the detector of victim network acts. It sends to sampler agents of other teams the request to get the information that could be relevant to the attack. Samplers send the needed data. If a likely attack source is detected, the detector of victim network sends the attack agent address to the detector of the team where the attack agent could appear. This team tries to defeat it.

The simulation environment architecture is represented in figure 1.

Simulation framework is the discrete event simulator. Other components are expansions or models for Simulation Framework.

Internet Simulation Framework is the modular simulation suite with a realistic simulation of Internet nodes and protocols. The highest IP simulation abstraction level is the network itself, consisting of IP nodes. IP node corresponds to the computer representation of Internet Protocol. IP node can represent router or host. The modules of IP node are organized as operating system processes IP datagram.

Multi-agent Simulation Framework allows to realize agent-based simulation. It consists of modules representing the intelligent agents implemented as applications. There were used the elements of abstract FIPA architecture during agent modules design and implementation. Agent communication language is implemented for the agent interactions. The message transmission occurs above the TCP protocol (transport layer) implemented in Internet Simulation Framework. Agent directory is mandatory only for agent that coordinates other agents in its team.

Subject Domain Library is used for imitation of subject domain processes. It contains modules that extend functionality of host: filtering table and packet analyzer.

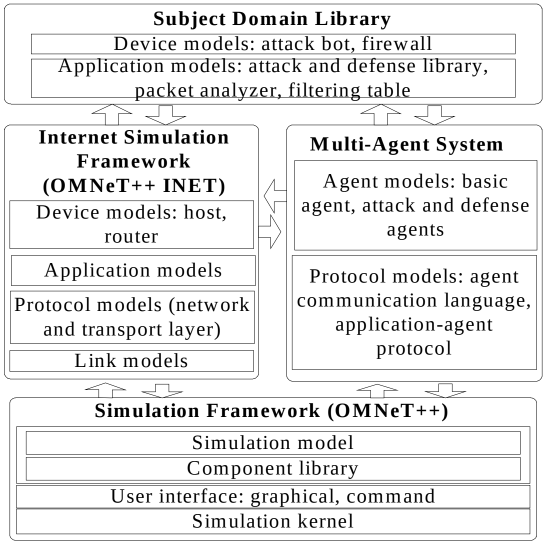


Fig. 1. Simulation environment architecture

This architecture was implemented for multi-agent simulation DDoS attack and defense mechanisms with the use of OMNeT++ INET Framework and software models developed in C++. Agent models implemented in Multi-agent Simulation Framework are represented with generic agent, attack and defense agents. Subject Domain Library contains various models of hosts, e.g. attacking host, firewall etc., and also the application models (attack and defense mechanisms, packet analyzer, filtering table).

The example of *multi-window user interface of the simulation environment* is depicted on the figure 2. The *window for simulation management* (figure 2, upper left) allows looking through and changing simulation parameters. It has the time base displaying the main system events: start or end of TCP connection, attack commands, etc. At the bottom left one can see the fragment of *simulation network window*. Network consists of hosts and data channels. The applications including agents are deployed on the hosts. During simulation one can observe various model parameters, e.g. network traffic (figure 2, upper right), agent parameters (figure 2, middle right), agent teamwork parameters (bottom right), etc.

Figure 2 (bottom left) shows the fragment of simulated network. The configuration of this network includes: 10 routers; 10 clients hosts (the caption below them is “i_cli[]”), generating typical network traffic; 4 defense teams; 3 attack teams (hosts with attack agents are distributed among all network; one can distinguish them by “Master” or “Daemon” marking above them). The current state of agents is displayed above their hosts. The network is created due to algorithms that allow to create the configurations that are close to the real Internet networks [26].

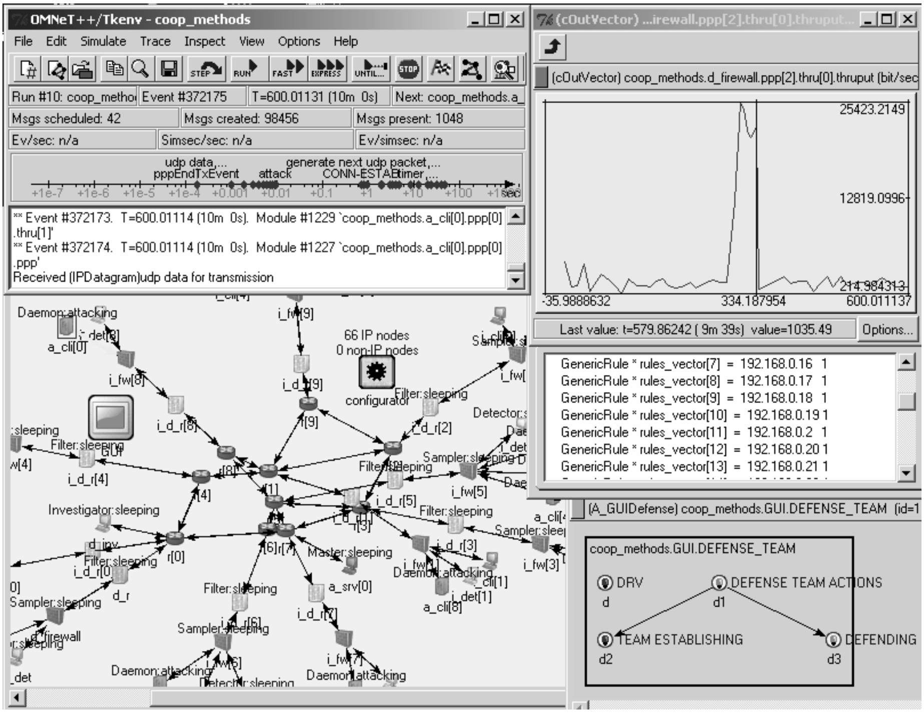


Fig. 2. Multi-window user's interface of environment

Figure 3 shows the hierarchy of simulated objects (from left to right: nested objects "network", "host", "agent"). During simulation one can switch from one level of hierarchy of objects representation to another and analyze the functioning parameters of various objects.

There are used the following *specification elements to define the investigated network models, attack and defense mechanisms*:

- *Network topology:* quantity and types of hosts, channels between them and their types. The possibility to deploy certain type of application (or agent) depends on host type.
- *Defense team parameters:* quantity of daemons; master's address and port used for interactions; daemon's port used to send attack packets; victim's address and port; time of attack; attack intensity; address spoofing technique.
- *Attack realization parameters:* victim type (application, host or network; one must define the IP-address and port of victim); type of attack (brute force (UDP/ICMP flood, smurf/fraggle, etc.) or semantic (TCP SYN, incorrect packets, hard requests, etc.)); attack rate dynamics (can be constant or variable); adaptation scheme depending on attack severity, etc.
- *Defense team parameters:* address of defended host; detector's address and port for interactions; server's reply size and delay time; adaptation scheme (changing of defense mechanisms) depending on attack severity, etc.

- *Defense mechanisms parameters*: deployment location (source, intermediate or defended subnets); the stages the defense method can implement (attack prevention, attack detection, tracing the attack source, attack counteraction); attack detection technique (misuse and anomaly detection; one chooses one particular detection method or the set of methods), etc.
- *User team parameters:* quantity of users; server's address and port; time to start; quantity of requests to server, interval between them and their size; interval between connections.
- *Defense team cooperation parameters:* scheme of cooperation.
- *Simulation parameters:* simulation duration; quantity of experiments; initialization of random number generator.

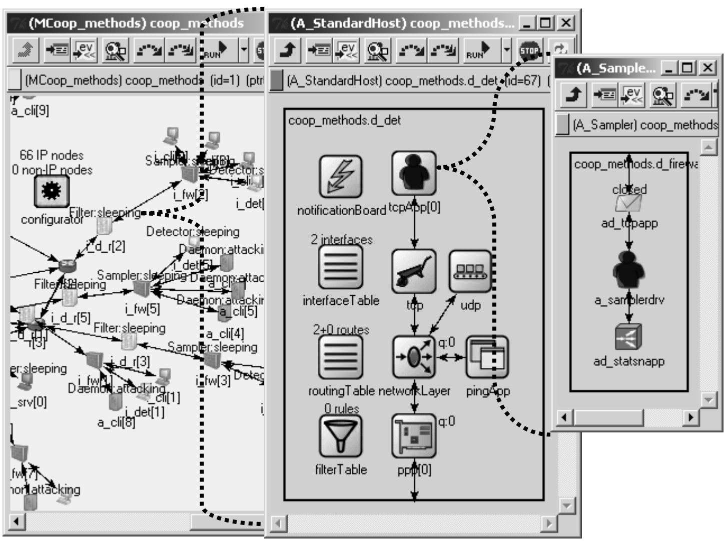


Fig. 3. Hierarchy of simulated objects: "network" → "host" → "agent"

4 Mechanisms of Agent Adaptation and Learning

The agents are supposed to implement the mechanisms of self-adaptation and to evolve during functioning. The team of agents-malefactors evolves due to generation of new instances and types of attacks and to scenarios of their realization to overcome the defense subsystem. The team of defense agents adapts to malefactors actions due to changing the executed security policy, forming of new defense mechanisms and profiles instances.

This paper uses simplified adaptation models of agent team. The main goal is to investigate the possibilities of proposed approach to multi-agent simulation. The general approach to adaptation is the generalization of approach proposed in [43].

There are the following *adaptation parameters for defense teams*:

- $S(t)$ – attack severity at time t. We define attack severity as the amount of attack traffic;
- $K_D(t) = \{M_i, TK_j\}$ – defense system configuration at time t, M_i – defense method (SIPM [36], HCF [14] and BPS) and its parameters (acquired during learning), TK_j – cooperation schema (no cooperation, filter-level, sampler-level, full cooperation);
- $C_i(S(t),K_D(t))$ – the i-th component of defense cost (i=1,…,n).

We define the following *components (factors) of defense cost*:

- $C_{FP}(S(t), K_D(t))$ – defense system false positives rate;
- $C_{FN}(S(t), K_D(t))$ – defense systems false negatives rate;
- $C_T(S(t), K_D(t))$ – duration of attack.

The main idea of adaptation is in the following. When $S(t)$ changes, the adaptation subsystem chooses the configuration $K(t)$ of defense system that minimizes the effectiveness function:

$$\min_{S(t)} \sum_{i=1}^{n} C_i(S(t), K_D(t))$$

The following *adaptation criterion* is used in the experiments for defense team:

$$\min_{S(t)} \{C_{FP}(S(t), K_D(t)) + C_{FN}(S(t), K_D(t)) + C_T(S(t), K_D(t))\}$$

We define the following *parameters for attack team*:

- $E(t)$ – factor of defense effectiveness at time t. We define the defense factor as the amount of alive daemons (because one of the defense goals is to defeat attack agents);
- $K_A(t)= \{I_i, R_j\}$ – attack parameters at time t, I_i – attack intensity (defined by the malefactor), R_j – IP spoofing technique (no spoofing, constant, random, random real);
- $C_j(E(t),K_A(t))$ – the j-th component of attack cost (j=1,…,m).

We define the following *attack cost factors*:

- $C_P(E(t), K_A(t))$ – amount of sent packets;
- $C_D(E(t), K_A(t))$ – amount of defeated daemons.

The following *adaptation criterion* is used in the experiments for attack team:

$$\min_{E(t)} \{C_P(E(t), K_A(t)) + C_D(S(t), K_A(t))\}$$

Two general *attack detection techniques* can be used: *misuse detection and anomaly detection*. Both of them require preliminary learning.

In the first case the current data sets about the state of defended object are compared with the data sets that indicate the attack. For example, attack packets can be detected due to the values of their fields or their size when using generic attack realization software (exploits). The example of the other approach is to examine the whole cycle of DDoS attack and to detect the key parameters of attack traffic. Then they are used as signatures to detect the attack.

Anomaly detection is in comparison the current state of the system with the state when there were no security violations. The general approach to attack detection is in the following. Sensors collect data about normal traffic for the given network. Then analyzer-component compares current traffic with modeled in real time and detects anomalies. The paper considers the defense based on anomaly detection with the learning on the basis of thresholds, rules and probabilistic parameters of traffic.

The implemented defense mechanisms are based on two modes: (1) learning mode and (2) defense mode with data updating. The data collection using knowingly legitimate traffic is carried *in the learning mode*. The learning duration depends on network size, features of defense method and expected false positive and negative rates. Current network traffic is compared with modeled *in the defense mode*. The mismatch is considered as anomaly or attack and the countermeasures will be applied. If there are no anomalies or they are small then data are added to the model, i.e. it is updated.

Let us consider the features of learning in implementations of three defense methods used in experiments – SIPM [36], HCF [14] and BPS.

The database of IP addresses of legitimate clients is created in *the learning mode of SIPM method*. The following assumption is used: when attack starts there are a lot of new IP-addresses. This moment is detected due to CUSUM [36] algorithm. In defense mode the statistics on packets is collected in real time. It says about the amount of new (for defended system) IP addresses in the given intervals of time. If this value is less than given threshold, then the new addresses are written in the database; if this value is bigger, then the filtration is applied. Figure 4 shows the data acquired during learning mode – host addresses and time of registration and the graph of dependence of new hosts amount from the time of registration.

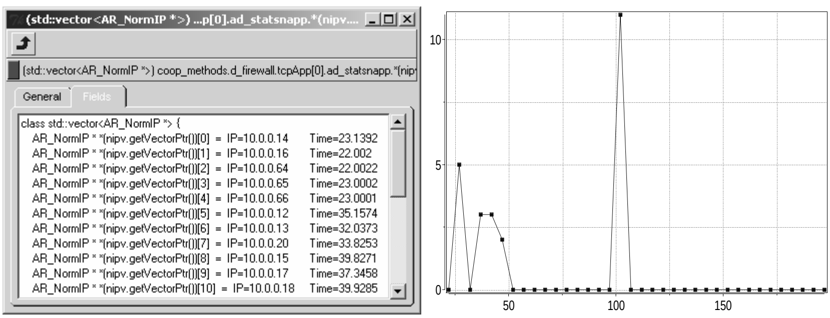


Fig. 4. SIPM learning data

HCF method uses the assumption that the packets from the same network travel from sender to receiver the same count of hops. The table of hops and addresses is created in learning mode. The hosts are grouped in subnets by the count of hops. The hop count packets traveled is evaluated due to TTL field of packet. Each router decrements this field by 1, when it transmits the packet. The table is updated periodically. In defense mode the system compares the hop count of the current packet with the table value on the basis of packet source address. If it differs, then the packet is considered as attack. Figure 5 shows the data acquired in learning mode – hosts addresses and hop counts.

In *the learning mode, BPS method* detects the maximum value of traffic from legitimate clients. Then, in the defense mode, traffic from hosts exceeded this value is dropped. Figure 6 shows the data collected during learning mode – hosts addresses and their traffic value and the graph of traffic change.

The main point of the learning mode is to create the model of typical traffic for the given network (without attack realization). The individual learning of the teams works in accordance with the described methods. The clients request data from the defended server and it replies in the learning simulation. Samplers register these requests and use them to compose the parameters of SIPM, HCF and BPS methods.

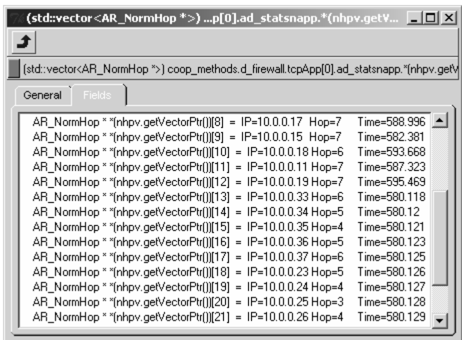


Fig. 5. HCF learning data

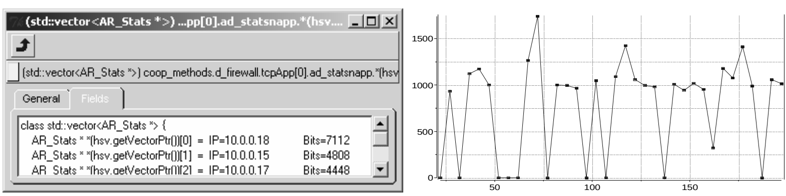


Fig. 6. BPS learning data

The defense teams use the data from other teams in their learning when working in sampler-level cooperation schema and in full cooperation. This is the main difference between *cooperative and individual learning*. The teams request data from the other samplers knowing their addresses, i.e. the data from the subnets of other teams. That gives the more complete network traffic model. There is no data exchange during learning mode in the other cooperation schemas since these schemas are aimed on co-operative defense only.

5 Experiments

It is proposed to investigate the following *adaptive schema* in the conducted experiments.

Attack team starts the attack in the given by malefactor time with given intensity and address spoofing technique. Master checks daemons periodically. If it finds that one of them is malfunctioned, then it does the following: redistributes the load in accordance with the given attack intensity; changes the address spoofing technique; sends these new parameters to daemons.

Defense team works initially using the least a resource-intensive defense method. When it detects the attack, it tries to block the attack packets, trace the attackers and defeat them. If after these actions the attack is still going on, then detector changes the defense method to the more complicated according to adaptation function and sends the command of method change to other agents.

Adaptation scheme works in the following way. Depending on attack state the defense team adapts the parameters of methods and cooperation reducing the defense cost. The most simple and not resource-intensive method is BPS. Defense team starts the defense implementing BPS. When attack is detected the team continues to use the same method, if it allows to neutralize the attack. If it fails, then defense team applies more complicated SIPM method. If it succeeds to stop the attack, then the defense team returns to BPS. If not – it will additionally use HCF.

Attack team redistributes the attack intensity between daemons and changes the address spoofing technique to minimize the amount of attack packets and reduce the probability of attack agents' exposure by defense agents. At first the team having many daemons distributes the load equal between them and does not use address spoofing (not to draw suspicion upon themselves from firewall in their subnet). If after the defense team actions some of the daemons will be defeated, the attack team will raise the load to the remaining daemons (to save the given attack intensity) and apply address spoofing technique to avoid detection by defense team. If the remaining daemons will not be defeated the team will continue the attack in former mode.

Adaptation schema is tested in various cooperation modes described below. Cooperative team learning is supposed in sampler-level and full cooperation.

The following *cooperation schemas* are supposed to be investigated:

- *no cooperation:* all teams work on their own;
- *filter-level cooperation:* team which network is the attack victim can apply the filtering rules on filters of other teams;
- *sampler-level cooperation:* team which network is the attack victim can receive traffic data from the samplers of other teams;
- *"poor" cooperation:* teams can receive traffic data from the samplers of some other teams and apply the filtering rules on filters of some other teams. Each team "knows" some amount of other teams depending on cooperation degree;
- *full cooperation:* team which network is the attack victim can receive traffic data from the samplers of other teams and apply the filtering rules on filters of other teams.

The investigation is supposed to be done on the basis of analysis of the following main parameters: the amount of incoming traffic before and after filter of team which network is the attack victim; false positive and false negative rates of defense team which network is the attack victim.

Let us describe *the procedures of experiments conducted and their results while using various cooperation schemas of defense teams*.

1. *No cooperation.* Attack begins at 300 seconds. Attack is executed without source address spoofing. Defense agents detect the attack and apply filtering rules due to BPS method. Because of this the traffic inside defended subnet reduces to acceptable level (figure 7, 350-600 seconds, black graph). Defense agents succeeded to defeat some attack agents and the traffic before defended subnet reduces also (figure 7, 350-600 seconds, gray graph). Master of attack team discovers that some of the daemons were defeated. It redistributes the load to the remaining daemons and changes address spoofing technique to “random”. This causes significant increase of traffic both before and inside attacked subnet (figure 7, 600-700 seconds). In response the defense team decides to use SIPM method. Traffic inside the subnet reduces to normal level (figure 7, after 700 seconds), but it stays high before the subnet.

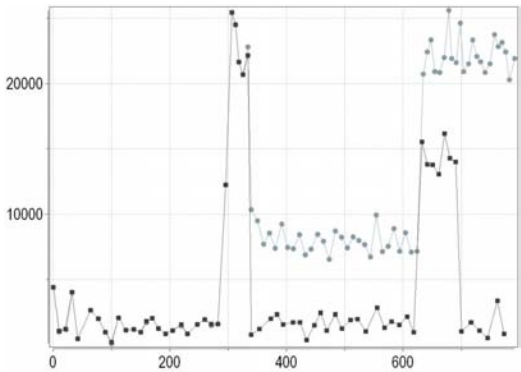


Fig. 7. Traffic before (gray) and inside (black) attacked subnet (*without team cooperation*)

2. *Filter-level cooperation.* The nature of traffic for adaptive scheme with filter-level cooperation is the same as for no-cooperation scheme right up to defense method change (figure 8). Defense team applies SIPM method. It receives filtering rules from other teams and the blocking of attack traffic happens much more quickly then without cooperation (figure 8, 600-630 seconds).

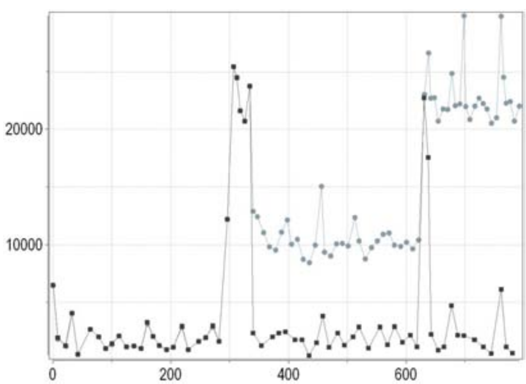


Fig. 8. Traffic before (gray) and inside (black) attacked subnet (*filter-level cooperation*)

3. *Sampler-level cooperation*. After attack starts in adaptive schema with sampler-level cooperation (figure 9), defense team succeeds to filter attack traffic and defeat some daemons (figure 9, 300-600 seconds, black and gray graphs accordingly). When attack method changes (figure 9, after 600 seconds), only traffic before defended subnet raises. Defense team applies SIPM method. Traffic inside the subnet stays at acceptable level. This is because defense teams have received data from samplers of other teams from other subnets during learning in cooperation. These data was enough for SIPM method to block attack traffic from spoofed addresses straight.

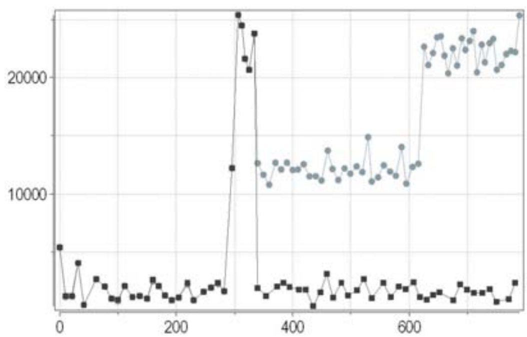


Fig. 9. Traffic before (gray) and inside (black) attacked subnet (*sampler-level cooperation*)

4. *Full cooperation*. Adaptive scheme with full cooperation unites advantages of all presented cooperation schemas (figure 10). Traffic here has the same nature as with sampler-level cooperation. One can say that sampler-level cooperation was determinative during defense actions.

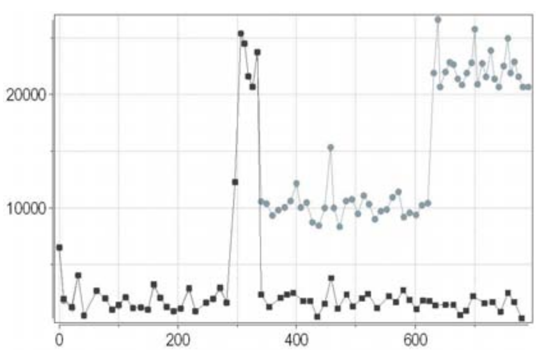


Fig. 10. Traffic before (gray) and inside (black) attacked subnet (*full cooperation*)

Conducted experiments showed that one can reach the best results in adaptive attack traffic blocking due to defense teams cooperation. The best adaptive schema is the sampler-level and full cooperation. Since sampler cooperation was the determinative in defense it can be used without applying full cooperation during which high teams interaction traffic is observed.

6 Conclusion

This paper proposed the approach to simulation of prospective adaptive and cooperative information security defense mechanisms in Internet. It is based on multi-agent simulation. Environment for the agent-oriented simulation was developed on the basis of OMNeT++ INET Framework. The example of simulation implemented is the adaptive cooperative strategies of realization DDoS attack and defense mechanisms.

Large amount of various experiments was carried out. The effectiveness parameters of adaptive cooperative defense mechanisms were studied. The conducted experiments showed the availability of proposed approach for simulation of complex adaptive cooperative defense mechanisms and security analysis of projected networks. The experiments showed also that the use of cooperation of several defense teams and combined adaptive application of various defense methods leads to the essential raise of defense effectiveness.

Future work is related to more deep investigation of cooperation mechanisms effectiveness of various teams and inter-team interaction of agents, the implementation of improved adaptation and self-learning agent mechanisms that are prone to manipulation by attackers, the expansion of attack and defense library, the investigation of new defense mechanisms. The important part of future research is providing numerous experiments to study various attacks and the effectiveness of prospective defense mechanisms (attack prevention, attack detection, tracing the source of attack and attack counteraction) and their combinations.

Acknowledgments

This research is being supported by grant of Russian Foundation of Basic Research, program of fundamental research of the Department for Informational Technologies and Computation Systems of the Russian Academy of Sciences (contract No 3.2/03), Russian Science Support Foundation and partly funded by the EU as part of the POSITIF project (contract No. IST-2002-002314) and RE-TRUST (contract No. 021186-2). Authors would like to thank the reviewers for their valuable comments to improve the quality of the paper and to advice prospective research directions.

References

1. Atighetchi, M., Pal, P.P., Jones, C.C., Rubel, P., Schantz, R.E., Loyall, J.P., Zinky, J.A.: Building Auto-Adaptive Distributed Applications: The QuO-APOD Experience. Proceedings of 3rd International Workshop Distributed Auto-adaptive and Reconfigurable Systems (DARES), IEEE CS Press (2003)
2. Atighetchi, M., Pal, P., Webber, F., Jones. C.: Adaptive Use of Network-Centric Mechanisms in Cyber-Defense. Proceedings of 6th IEEE International Symposium "Object-Oriented Real-Time Distributed Computing", IEEE CS Press (2003)
3. Atighetchi, M., Pal, P., Webber, F., Schantz, R., Jones, C., Loyall, J.: Adaptive Cyberdefense for Survival and Intrusion Tolerance. Internet Computing, Vol.8, No.6 (2004)
4. Chen, S., Song, Q.: Perimeter-Based Defense against High Bandwidth DDoS Attacks. IEEE Transactions on Parallel and Distributed Systems, Vol.16, No.7 (2005)

5. Cheng, S., Huang, A., Garlan, D., Schmerl, B., Steenkiste, P.: An Architecture for Coordinating Multiple Self-Management Systems. Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04) (2004)
6. Cohen, P., Levesque, H.J.: Teamwork, Nous, 35 (1991)
7. Combs, N., Vagel, J.: Adaptive mirroring of system of systems architectures. Proceedings of the First ACME SIGSOFT Workshop on Self-Healing Systems (WOSS ’02), ACM Press. Charleston, SC, USA (2002)
8. Dashofy, A. van der Hoek, Taylor, R.N.: Towards architecture-based self-healing systems. Proceedings of the First ACME SIGSOFT Workshop on Self-Healing Systems (WOSS ’02), ACM Press. Charleston, SC, USA (2002)
9. Gamer, T., Scholler, M., Bless, R.: A Granularity-adaptive System for in-Network Attack Detection. Proceedings of the IEEE / IST Workshop on Monitoring, Attack Detection and Mitigation (2006)
10. Gorodetski, V., Kotenko, I.: Conceptual foundations of stochastic simulation in the Internet. Proceedings of system analysis institute of RAS, Vol.9, Moscow, URSS (in Russian) (2005)
11. Gross, P.N., Gupta, S., Kaiser, G.E., Kc, G.S., Parekh, J.J.: An active events model for systems monitoring. Proceedings of theWorking Conference on Complex and Dynamic Systems Architecture, Brisbane, Australia (2001)
12. Grosz, B., Kraus, S.: Collaborative Plans for Complex Group Actions, Artificial Intelligence, Vol.86 (1996)
13. Horn, P.: Autonomic Computing: IBM's Perspective on the State of Information Technology. http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf (2001)
14. Jin, C., Wang, H, Shin, K.G.: Hop-count filtering: An effective defense against spoofed DDoS traffic. Proceedings of the 10th ACM Conference on Computer and Communications Security (2003)
15. Silva, F., Endler, M., Kon, F., Campbell, R.H., Mickunas, M.D.: Modeling Dynamic Adaptation of Distributed Systems. Technical Report UIUCDCS-R-2000-2196, Department of Computer Science, University of Illinois at Urbana-Champaign (2000)
16. Ishida, Y.: Immunity-Based Systems A Design Perspective. Springer Verlag (2004)
17. J-Sim homepage. http://www.j-sim.org
18. Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing. IEEE Computer Magazine, No.1 (2003)
19. Keromytis, A.D., Parekh, J., Gross, P.N., Kaiser, G., Misra, V., Nieh, J., Rubensteiny, D., Stolfo, S.: A Holistic Approach to Service Survivability. Proc. ACM Workshop on Survivable and Self-Regenerative Systems, ACM Press (2003)
20. Keromytis, A.D., Misra, V., Rubenstein, D.: SOS: An architecture for mitigating DDoS attacks. Journal on Selected Areas in Communications, Vol. 21 (2003)
21. Knight, J., Heimbigner, D., Wolf, A.L., Carzaniga, A., Hill, J., Devanbu, P., Gertz, M.: The Willow Architecture: Comprehensive Survivability for Large-Scale Distributed Applications. Proc. Of International Conf. Dependable Systems and Networks (DSN 02), supplemental vol., IEEE Press (2002)
22. Kotenko, I.V., Ulanov, A.V.: Agent-based simulation of DDOS attacks and defense mechanisms. Journal of Computing, Vol. 4, Issue 2 (2005)
23. Kotenko, I., Ulanov, A.: Agent-based modeling and simulation of network softbots’ competition. Knowledge-Based Software Engineering. Proceedings of the Seventh Joint Conference on Knowledge-Based Software Engineering (JCKBSE'06). Frontiers in Artificial Intelligence. IOS Press, (2006)

24. Kotenko, I., Ulanov, A.: Simulation of Internet DDoS Attacks and Defense. 9th Information Security Conference. ISC 2006. Proceedings. Lecture Notes in Computer Science, Vol. 4176 (2006)
25. Macal, C.M., North, M.J.: Tutorial on Agent-based Modeling and Simulation. Proceedings of the 2005 Winter Simulation Conference (2005)
26. Mahadevan, P., Krioukov, D., Fomenkov, M., Huffaker, B., Dimitropoulos, X., Claffy, K., Vahdat, A.: Lessons from Three Views of the Internet Topology. Technical Report. Cooperative Association for Internet Data Analysis (CAIDA) (2005)
27. Marietto, M., David, N., Sichman, J.S., Coelho, H.: Requirements Analysis of Agent-Based Simulaton Platforms: State of the Art and New Prospects. Multi-Agent-Based Simulation II, LNAI, Vol.2581, Springer-Verlag (2002)
28. Mirkovic, J., Dietrich, S., Dittrich, D., Reiher, P.: Internet Denial of Service: Attack and Defense Mechanisms. Prentice Hall PTR (2004)
29. Mirkovic, J., Robinson, M., Reiher, P., Oikonomou, G.: Distributed Defense Against DDOS Attacks. University of Delaware. Technical Report CIS-TR-2005-02 (2005)
30. Negoita, M., Neagu, D., Palade, V.: Computational Intelligence Engineering of Hybrid Systems. Springer Verlag (2005)
31. NS2 homepage. http://www.isi.edu/nsnam/ns/
32. OMNeT++ homepage. http://www.omnetpp.org/
33. Oreizy, P., Gorlick, M.M., Taylor, R.N., Heimbigner, D., Johnson, G., Medvidovic, N., Quilici, A., Rosenblum, D.S., Wolf, A.L.: An architecture-based approach to self-adaptative software. IEEE Intelligent Systems, Vol.14, No.3 (1999)
34. Papadopoulos, C., Lindell, R., Mehringer, I., Hussain, A., Govindan, R.: Cossack: Coordinated suppression of simultaneous attacks. Proceedings of DISCEX III (2003)
35. Paruchuri, P., Bowring, E., Nair, R., etc.: Mutiagent Teamwork: Hybrid Approaches. Computer society of India Communications (2006)
36. Peng, T., Leckie, C., Kotagiri, R.: Proactively Detecting DDoS Attack Using Source IP Address Monitoring. Networking 2004, Athens, Greece (2004)
37. Piszcz, A., Orlans, N., Eyler-Walker, Z., Moore, D.: Engineering Issues for an Adaptive Defense Network. MITRE Technical Report (2001)
38. SSF Net homepage. http://www.ssfnet.org
39. Tambe, M., Pynadath, D.V.: Towards Heterogeneous Agent Teams. Lecture Notes in Artificial Intelligence, Vol.2086 (2001)
40. Want, R., Pering, T., Tennenhouse D.: Comparing autonomic and proactive computing. IBM Systems Journal, Vol.42, No.1 (2003)
41. Webber, F., Pal, P.P., Schantz, R.E., Loyall, J.P.: Defense-Enabled Applications. Proc. DARPA Information Survivability Conf. (DISCEX II), Vol 2, IEEE CS Press (2001)
42. Xuan, D., Bettati, R., Zhao, W.: A gateway-based defense system for distributed dos attacks in high-speed networks. IEEE Transactions on Systems, Man, and Cybernetics (2002)
43. Zou, C.C., Duffield, N., Towsley, D., Gong, W.: Adaptive Defense against Various Network Attacks. IEEE Journal on Selected Areas in Communications: High-Speed Network Security (J-SAC), Vol.24, No.10 (2006)

On Competing Agents Consistent with Expert Knowledge

Edward Pogossian[1,3], Vachagan Vahradyan[1,2], and Arthur Grigoryan[1]

[1] Cognitive Algorithms and Models Laboratory at the Academy of Sciences of Armenia (IPIA)
[2] Russian and [3]State Engineering Universities of Armenia
epogossi@aua.am

Abstract. We aim to advance in constructing collaborative agents able to acquire the contents of human vocabulary associated with competitions. Refining the framework and criteria of performance of agents we project the study on the class of game tree represented competition problems. For known representative of the class – chess like combinatorial games, we categorize the contents of a comprehensive repository of units of chess vocabulary by formal structures of attributes, goals, strategies, plans, etc. We define Personalized Planning and Integrated Testing algorithms able to elaborate moves in target positions dependent on those categories of chess knowledge. We then demonstrate the effectiveness of the algorithms by experiments in acquisition the solutions of two top Botvinnik's tests – the Reti and Nodareishvili etudes. For min max game tree based search algorithms these etudes appears to be computationally hard due the depth of the required analysis and very dependence on the expert knowledge.

Keywords: competing, agents, knowledge, acquisition, consistent, measures.

1 Introduction

The concept of agents is based on the idea of assistance with tasks that require reciprocal and consistent human-agents knowledge processing.

We plan to advance in constructing collaborative agents able to improve their problem solving performance by acquiring, learning or inferring the contents of human vocabulary associated with competitions.

We develop the approach announced in [18] and study problem solving for the subclass of optimal strategy provision problems where the *space* of hypothesis of *solutions* can be specified by *game trees* (SSGT). Viability of the approach was demonstrated for the intrusion protection SSGT problems and the *Intermediate Goals At First* (IGAF) algorithms. The IGAF algorithms as well as [21] are based on a common knowledge planning and dynamic testing of the plans in the corresponding game trees. It was proven that the IGAF2 algorithms are able to acquire a range of expert knowledge in the form of goals or rules and to increase the efficiency of strategy formation with an increase in the amount of expert knowledge available to the algorithm.

The effectiveness of the IGAF2 algorithms was successfully tested for the network intrusion protection problems against representatives of four classes of attacks:

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 229–241, 2007.
© Springer-Verlag Berlin Heidelberg 2007

SYN-Flood, Fraggle, Smurf and Login-bomb, allowing to formulate, in particular, the following statements:

- Number of nodes searched by the IGAF2 algorithm for making decisions with all expert rules and subgoals is the smallest compared with the IGAF1 algorithm or with the minimax algorithm in which the depth of the search is increasing up to 13
- The recommended version of the IGAF2 algorithm with all expert rules and subgoals, for the depth of search 5 and 200 defending steps, is outperforming the productivity of the minmax algorithm by 14% while using 6 times less computing time and searching 27 times less nodes of the tree.

Knowledge based strategies for another SSGT problem where supply chain management agents compete in a market to gain max profit for different customer requests were studied in [20, 22]. For each request the agents generate possible offers and for each offer a game tree is constructed to allow simulation and tracing further actions with suppliers, production and delivery. In the mean time, strategic plans based on the handbook common expert knowledge are generated followed by the quantification of the plans for strategy analysis and their evaluation for final decision making.

In the paper regular improvement of programs by expert knowledge is studied for known representatives of the SSGT class – chess like combinatorial games [19], where the problems of knowledge representation and consistent inclusion into the programs stay central since the pioneering work by Shannon in 1950.

Players indicate and communicate chess knowledge by units of vocabulary and are able to form corresponding contents. Whether is it possible to form equal contents by computers remains under question.

A comprehensive overview of the state of the art in learning the contents of vocabulary in games is presented in [12]. Models of chess concepts based on the elements of chess contents – chunks, are discussed in [13, 14] to answer whether the chunks are common for all players or are individualized. These problems are analogous to other semantics related problems, particularly to understanding instructions by robots [25, 31], ontology formation, etc. A comprehensive review of ontology based studying is presented in [6,7].

The approaches to regular inclusion of chess knowledge into strategy formation process are described in [3, 4, 23, 30]. All of them try to bring common handbook knowledge to cut the search in the game tree. The frontiers of those approaches can be revealed by understanding the role and proportion of the personalized chess expertise compared with the common, communicable one.

In 1912 Zermelo proved that all chess positions are strongly divided into three classes: winning, losing or drawing [32]. In [16, 17] a correspondence was revealed between Zermelo's classes of winning chess positions and strategies and a typical units of chess vocabulary. While this correspondence supports the constructive nature of human models of chess realities it, at the same time, states that the real implementation of those models, in principle, can only approximate (to a specification) the winning game tree structures due to the irresistible complexity of computations required to prove the correctness of the models. Thus, it follows that

- for the same units of chess vocabulary both chess players and computers will, as a rule, use different models of chess realities essentially based on their individual experience
- any preferences between those models include uncertainty
- interpretations of the units of vocabulary have to rely on some personalized experience.

Therefore, strategy formation systems have to be able to integrate common chess knowledge with personal expertise of particular players and be able to improve their performance by regular acquisition, learning or inferring the contents of the units of chess vocabulary.

The paper reports our progress in programs acquiring typical categories of chess knowledge and is structured as follows. Sections 2 and 3 define the SSGT problems, refine the framework of study and measures of effectiveness of competing programs to induce corresponding requirements to them. Section 4 describes formal structures of the attributes, goals, strategies, plans, etc., and our approach to categorize a repository of chess vocabulary in about 300 units [24] by those types. Section 5 specifies the *Personalized Planning and Integrated Testing* (PPIT) algorithms able to elaborate moves in target positions dependent on typical categories of chess knowledge and demonstrates the effectiveness of PPIT algorithms by experiments in acquisition expert knowledge based solutions of two top Botvinnik's test etudes in strategy discovery, the Reti and Nodareishvili etudes. The Conclusion underlines the main findings of the research.

2 Optimal Strategy Provision Problems

We define the SSGT problems as a constructive subclass of a spacious class of *Optimal Strategy Provision* (OSP) problems [19,22]. The OSP problems, first of all, meet the following requirements:

- there are (a) interacting actors (players, competitors, etc.) performing (b) identified types of actions in the (c) specified moments of time and (d) specified types of situations
- there are identified benefits for each of the actors
- there are descriptions of the situations the actors act in and transformed after actions.

For such problems with given arbitrary situation x and actor A, who is going to act in x, we can generate corresponding *game tree* GT(x, A) comprising all *games* started from x.

The games represent all possible sequences of legal actions of the players and situations that they can create from given initial, or root situation x. In our consideration the games are finite and end by one of goal situations of the problem.

In chess, for example, the actors are white and black players with checkmate as the goal, chess piece moves are (contra) actions and compositions of chess pieces on the board determine specific game situations (positions) transformed by actions according to chess rules in the corresponding game tree.

Assuming A plays according to a deterministic program, a *strategy*, the GT(x, A) represents, in fact, all possible *performance trees* of the strategies from the x. In that sense the GT(x, A) determines the space of all possible solutions from the x situation.

Note, that performance trees of the strategies are described not on the level of detailed commands but by their compositions, i.e. the actions of the players.

Given criterion K to evaluate the quality of strategies we can define the best strategy S*(x, A) and corresponding best action of A from x.

In the OSP problem with criterion K of quality of strategies it is required to find the best action in respect to K for any given situation.

The OSP problems comprise chess-like combinatorial problems, security and competition problems, particularly, network intrusion protection and management in oligopoly competitions problems. Many other games represented security problems such as Computer Terrorism Countermeasures, Disaster Forecast and Prevention, Information Security, etc., announced by the NATO (http://www.nato.int/science/e/newinitiative.htm) seemingly can be reduced to OSP problems, as well.

The SSGT class comprises the OSP problems where the OSP requirement to have descriptions of situations after transforming them actions is replaced by the following stronger requirement:

- the situations the actors act in and transformed after actions can be *adequately simulated.*

Thus, for SSGT problems the game trees can be constructively simulated what allows to create a common theory and computer based game tree analysis methodology to find optimal solutions for corresponding problems.

In [19, 21, 22, 23] it was proved that chess and chess like combinatorial problems, intrusion protection and oligopoly competition are SSGT problems and common methodology can be effectively applied to find high quality solutions for them.

3 Refining Strategy Evaluation Measures

The framework of any study grounds on appropriate measures determined by the nature of studied problems and persuaded goals.

Concentration on the SSGT class of problems followed by focusing the chess permits to use measures of program performances induced by problems under consideration which, in contrast with universal ones, allow to control adequacy of measures by utilities of corresponding solutions.

We focus on measures of the *effectiveness* vs. *efficiency* and *On-the-job* (OJP) performance measures [20] vs. *Attributive Scoring of Preference* (ASP) based ones [9].

ASP measures evaluate performance of systems by aggregating the values of different attributes defined for different aspects of the performance into a global quality indicator (called the global preference).

Despite the fact that ASP measures are widely used in education, personal evaluation, systems and organizations evaluation, etc., they have a principle drawback in being very dependent on human subjective intervention in the measurement process. Particularly, they suppose a human choice of the base general criteria of functionality of examined systems, their partition on the subsystems with appropriate attributes and scales to evaluate them, and, finally, choice of a method to integrate measurements of the subsystems into global estimate of the system performance.

The OJP measures assume to be functions of quality of solutions of tasks comprising those problems. In other words, they have to be functions of “how“ the solutions perform their “job” for individual tasks of those problems.

We distinguish two types of OJP measures based either on *question-answers* or on *tasks solution values.*

The Turing Test (TT) [29] belongs to the first type and because it is not a formal procedure can cause uncertainties and drawbacks in its applications, like the following ones:

- TT procedure is very dependent on subjective characteristics of the judges and cannot be regularly reproduced
- TT in spite of measuring solutions of individual problems does not use their internal criteria and, instead, pretends to be a universal measure
- TT is based on local questions –answers to the solutions of problems and needs a framework of their correct integration into global estimates of the quality
- TT compares the “external” question-answers “shell” of the solutions and does not measure actions in the world caused by the solutions, or is not “grounded” [25, 27, 28].

The *tasks solution values* (TSV) criteria are long time used to measure human and program performance. Based on the analysis of performance trees TSV criteria are widely used for testing of programs [15].

In chess TSV criteria coincide with a widely applied tournament based ones [10, 26]. The OJP criteria for measuring performance of chess programs, management and intrusion protection skills were developed in [20-23]. For further studying we chose TSV OJP measures of effectiveness of solutions.

Finally, we need *human-computer consistent* measures, allowing to measure human and program performances in problem solving at the same scale. Particularly, to measure at the same performance scale an ability to incremental growth of the quality of performance by acquisition of expert knowledge.

At present, Elo’s rating system is in the use for chess players [10]. Therefore, computer performance measures consistent with Elo ratings are needed.

We argue that max Sum criteria of strategy performance can play that role.

Indeed, the vectors of the terminal nodes of examined strategies can be comprised into a matrix and optimal strategies can be determined by comparing those vectors by criteria that are functions of the results of the analysis of the values of components of those vectors. For max Sum criterion those values are summarized which makes the criterion equal to the robin, absolute tournament among all possible strategies and isomorphic to the ordering induced by Elo ratings. Thus, we can assume that max Sum ordering is sensitive to knowledge acquisition and changes of the players ranks in the ordering is directly proportional to changes of their Elo ratings.

But what are the relationships between changes in locations of programs in the max Sum ordering and inclusion of chess knowledge in their work?

We state the following:

- In contrast with chess players, rising the dispositions of programs in the max Sum orderings is not necessary caused by possessing more chess knowledge and skill than the lower positioned programs
- If changes of positions of programs in the max Sum ordering *ceteris paribus* are caused by knowledge acquired by programs from the hierarchies of common chess knowledge (e.g. from handbooks) then those changes are measuring the rising of quality of programs caused by that knowledge acquisition and can be compared with a human ones by the same max Sum criterion.

Therefore, if we test procedures of human knowledge acquisition and max Sum criterion indicates regular increasing of performance of programs simultaneous with the work of the procedures then the assumption that those procedures simulate human knowledge acquisition gets a valuable argument.

Concluding, we accept max Sum criterion as a common for chess players and programs measure to evaluate the quality of acquisition, learning or inferring chess knowledge.

4 Categorization of Chess Expert Knowledge

To represent chess knowledge we classify the units of chess repository (UCR) [24] in frame of known models and constructive views followed by assumptions on their origin .

The 1st step of UCR classification reveals elements of specification of chess as a game which in turn induces new game analysis based models as the following:

- game tree model adequately representing chess, all possible behaviors in it – strategies, concept of the best strategy, etc.
- strategy search methods and methods to enhance their efficiency
- Zermelo's statement on chess positions partition as winning, losing and drawing ones .

On the 2d step of UCR classification those models and their constituents were classified as conceptual and procedural types of strategy knowledge, represented, particularly, by attributes, classifiers and rules and with meanings referred to the elements of game tree, chess specification components or other not chess elements [16].

Attributes are defined as a kind of operators having values in a given set of realities. Particularly, they may be sets of numbers of some ranges.

We distinguish basic *attributes* and constructed ones that compose basic attributes to determine their values. Attributes with 0/1 or "false" /"true" values are named *classifiers* and correspond to concepts and goals.

Concepts are defined as a kind of knowledge to identify and recognize realities whereas goals are concepts with elements of intensions or requirements to achieve them.

Motives are attributes used to argue the preferences of some goals in analyzed situations.

Strategies, plans, goals and *rules* are defined as expert knowledge to specify compositions of his actions in time.

We name (c*omplete*) *S-strategy of C* the performance of the player/competitor C (i.e. the performance of corresponding program) in the game tree with starting position S (S-tree).

Rules are kind of if *x* / then *y* operators to specify a procedure *y* for the realities that fit to the *x* requirements. Any strategy may be determined as a composition of rules.

Given criteria F we say that *a strategy G achieves the goal F* if criteria F are satisfied for the set of terminal nodes of G.

The *strategy G will be called F-projected* if we are interested in whether the terminal nodes of G satisfy criteria F or not.

Any description of an F-projected strategy G aimed to make the search of G more efficient is named a *strategy plan for F* [19].

Descriptions of problems and strategies resolving them are extreme examples of strategy plans. A useful strategy plan would systematically identify the directions that are not promising and eliminate them for reducing the search space.

Strategy planning is a process of narrowing the search space for the target strategy which also reflects the specifics of the planner such as knowledge of language, the system of values and methods of search.

We identify the following two sources of strategy expertise represented by UCR:

- concepts and attributes determined by specification of the game tree
- winning strategies induced by examining the game tree.

Let us illustrate our reasoning to link chess concepts with classes winning by Zermelo (for the defeat and draw ones similar links can be induced by analogy).

In general, a position is winning if a winning strategy can be generated with the root in that position. The “Mat” positions are the only exclusion where winningness is determined by a few game specification attributes calculated either statically or by one-two plies. The attributes are the following:

K(ing) is under check, K can’t escape, K has not defenses.

Being constituents of the concept “Mat” those attributes get their own values of winningness induced by the value of “Mat” winning positions – the maximum value in chess.

It is evident, that any utility of winningness of positions provides not only two extreme “yes” or “not” utility values of those attributes but also values intermediate between those two poles.

Similar reasoning is natural to spread, analogically, to other than King pieces as well as to squares where pieces may be located.

Expanding the ranges of values and definitions of the attributes in a natural way we get the scales where not only their max and min values but intermediate ones, too, become meaningful respective to utilities they are associated with.

Another source for valued attributes and concepts are positions with known values, winning or not: endgames, combinations, etudes, beginnings, “tabbies” – typical analyzed positions, etc. [23] and, possibly,” chunks” [13, 14]. Partitioning and generalizing their constituents we get a set of basic concepts, attributes with associated estimates of their winningness. These attributes applied to constituents (or configurations of constituents) of analyzed positions create estimates of their utility in the context of winningness of those positions.

Analysis of chess concepts of the Repository allows to identify them as the descriptions of elements of positions having tangible winning utility. Hence, chess concepts become elements of specifications of winning by Zermelo positions what argues for possibility of their simulation. The players integrating in some reasonable way the utilities of constituents of positions get estimates of winningness of the positions.

What configurations of elements of positions are covered by concepts and how that is happening needs a special studying, particularly, in view of the theory of “chunks”.

The concepts can only indicate some possible utilities for further deeper analysis and strong estimate. An uncertainty is a priory in their nature caused partly by limitations of chess players’ static or quasi dynamic analysis of the winningness of positions and partly as a consequence of an individualized way in formation of concepts.

It is worth to accept that using the same names for the concepts chess players along with common strong meaning concepts use many other ones having, in general, not equal interpretations which coincide only in some “skeleton” parts. Those uncertain concepts, as a rule, don’t create big casualties due to their intermediate preliminary role of hypothesis of winningness of positions which later on are strongly tested by explicit game tree based analysis for final decisions.

5 Experiments in Acquisition Knowledge Intensive Solutions

Let’s describe the class of *Personalized Planning and Integrated Testing* (PPIT) programs aimed to acquire strategy expert knowledge to become comparable with a human in solving hard chess problems. In fact, the following two tasks of knowledge acquisition can be identified the process:

- construction of shells of the programs allowing to acquire the contents of units of chess vocabulary and
- construction of procedures for regular acquisition of the contents of the units by the shells.

We formulate the limitations in designing effective shells as following:

- be able to store typical categories of common chess knowledge as well as the personalized one and depend on them in strategy formation
- be able to test approximate knowledge based hypothesis on strategies in questioned positions by reliable means, for example, using game tree search techniques.

The second task we plan to solve in the following two stages:

- to prove that the shells, in principle, can acquire the contents of units of vocabulary used by chess players and allow to tune them properly to solve expert knowledge intensive chess problems
- to develop procedures for regular acquisition of the contents of those units .

The paper reports our progress in the first stage.

5.1 Personalized Planning and Integrated Testing Programs

We design shells of PPIT1 programs as a composition of the following basic units:

- Reducing Hopeless Plans (RHP)
- Choosing Plans with Max Utility (CPMU)
- Generating Moves by a Plan (GMP)

Given a questioned position P1 and a store of plans, RHP recommends to CPMU a list L1 of plans promising by some not necessary proved reasons to be analyzed in P1. The core of the unit is knowledge in classification of chess positions allowing to identify the niche in the store of knowledge the most relevant for analysis the position. If the store of knowledge is rich and P1 is identified properly it can provide a ready-to-use portion of knowledge to direct further game playing process by GMP unit. Otherwise, RHP, realizing a reduced version of CPMU, identifies L1 and passes the control to CPMU.

CPMU recommends to GMP to continue to play by current plan if L1 coincides with list L0 of plans formed in the previous position P0 and changes in P1 are not essential enough to influence on the utility of current plan.

If changes in P1 are essential, CPMU analyzes L1 completely to find a plan with max utility and to address it to GMP as a new current plan. Otherwise, CPMU forms new complementary list L1/ L1*L0 from the plans of L1 not analyzed, yet, in L0, finds a plan with the best utility in that list and comparing it with utility of current plan recommends the one of them with a higher utility.

To calculate utilities of the attribute, goal and plan type units of chess knowledge, we represent them as operators over corresponding arguments as follows:

- for basic attributes the arguments are characteristics of the states of squires in the questioned positions, including data on captures of pieces, threats, occupations, etc.
- for composed attributes, including concepts and goals, the arguments are subsets of values of basic attributes relevant to the analyzed positions
- for plans the arguments are utilities of the goals associated with realization of those plans.

Utilities of arguments of basic attributes are calculated by trajectory-zones based technique (TZT) [3, 18, 21] originally suggested to estimate utilities of only captures of the opponent pieces. For example, to chose capture with max utility TZT chains the moves to each piece of the opponent (trajectories) without accounting possible handicaps for real capturing then using all available knowledge "plays the zones" of the game tree induced by the trajectories followed by estimation of their values to choose the best.

The utility of units of knowledge the operators assemble from utilities of corresponding arguments in some predetermined ordering. Thus, each operator can provide by a request the arguments which are analyzed at the moment.

For example, realizing current plan the shell can determine the goal in the agenda which in turn determines basic attributes to be considered followed by indication of the arguments of those attributes.

Utility estimation operators rely on the principle of integration of all diversity of units of knowledge the shell possess at the moment. In fact, the operators represent a kind of expert knowledge with a variety of mechanisms and leverages to make them better. Along with dynamically changed parametric values of pieces they can include rules, positions with known values and strategies to realize them, other combinatorial structures. To estimate expected utilities the operators take into account the cost of resources necessary to get them.

In current C++ realization the units of knowledge are realized as OO classes with specialized interfaces for each type of knowledge and one common for the shell itself. We experiment in solving Reti and Nodareishvili etudes (Fig. 1, 2) requiring by Botvinnik [3, 4] intensive expert knowledge based analysis not available to conventional chess programs.

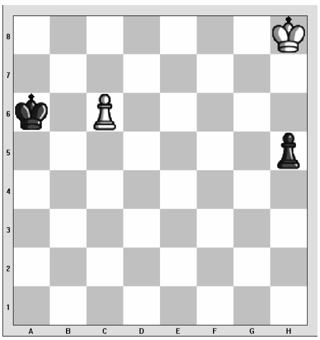


Fig. 1. Reti etude: draw

Fig. 2. Nodareishvili etude: winning

The following results can be reported at the moment.

5.2 Reti and Similar Etudes

Because Reti etude is a unique and handbook chess problem it is supposed to be in the knowledge store of the shell. RHP unit classifies it as a pawn endgame and suggests a known plan for the white king:

Realize G4 and if G1, or G2, or G3 realize them as well,

where goals G1,…,G4, listed by their priority, have the following contents:
G1. Capture the pawn
G2. Move forward the pawn at passant
G3.Protect your pawn
G4.Be max close to both: your and opponent pawns.

Because the plan is known the control is passed to GMP unit which acts as follows.

Based on the TZT the GMT unit determines utilities of attributes for each goal, finds the goal with max utility and makes the move induced by that goal.

Several other etudes induced by the original Reti one by adding new pawns in different parts of the chess board were considered.

In all experiments by adding new rules it was possible to focus the analysis of the program on the essential Reti part of the chessboard and successfully resolve the position.

5.3 Nodareishvili Etude

Nodareishvili etude is a unique and known chess problem and RHP classifies it as a pawn endgame as well but suggests four corresponding plans for different stages of the game.

The goals for the plans have the following contents:

G1.Move forward pawns at passant
G2. Protect your pieces
G3. Occupy or protect the most important squires
G4. Identify infinite check
G5. Avoid infinite check
G6. Decrease suppression of your king
G7. Capture the queen
G8. Find opponent pieces on the same line with its king
G9. Attack the king
G10. Attack the pieces on the line with the king
G11. Occupy the last horizontal line
G12. Approach king to your queen
G13. Be opposite to the pawn closest to opponent king.
The plans for experiments were presented as follows :

Plan1: Do G1and G6
Plan2: Do G1and if G4 do G5 and G6
Plan3: If G8 do G9 and G10
Plan4: Do G11 and G12 and G13.

After several corrections and tuning of utility estimation procedures the shell, in principle, was able to acquire the above units of knowledge, to choose corresponding plan for each stage and realize it.

Along with mentioned positive knowledge acquisition ability the efforts to formalize the units of knowledge for storing in the shell were too time consuming to rely only on those pure knowledge acquisition forms to solve knowledge intensive chess problems.

6 Conclusion

Currently the development of agents is facing a principal difficulty achieving a consistency and compatibility in the way agents and humans use and exchange knowledge.

In this paper we have reported our progress towards developing human knowledge acquisition by competitive agents that elaborate strategies.

We define a class of competitive problems with a wide range of applications. For chess like combinatorial games, which are long-recognized representatives of this class used for in-depth studies, we determine a measure common for chess players and programs – the so-called max Sum criterion. It can be used to evaluate the quality of acquisition, learning or inferring chess knowledge.

Classifying categories of chess knowledge according to the formal structures of attributes, goals, strategies, plans, etc., we define the class of Personalized Planning and Integrated Testing programs able to elaborate moves, in a given position, dependent on those categories of chess knowledge.

We demonstrate the effectiveness of PPIT1 programs by experiments in acquisition of knowledge for two etudes which are intractable for common chess programs — the Reti and Nadareishvili etudes suggested by Botvinnik to test the quality of knowledge intensive solutions in computer chess.

We find that PPIT1 programs, in principle, are able to acquire the contents (meanings) of units of vocabulary used by chess players and allow one to tune the contents properly to acquire the solutions of those etudes and play them properly.

In addition, we conclude that pure knowledge acquisition based approaches to solve problems consume too much time to formalize chess knowledge properly to be used in the programs and must be complemented by knowledge learning and inference based methods.

References

1. Adelman L.,Riedel S.: Handbook for Evaluating Knowledge–Based Systems, Kluwer Acad. Publ. (1997)
2. Atkinson G.: Chess and Machine Intuition. Ablex Publishing Corporation, Norwood, New Jersey (1993)
3. Botvinnik M.: About solving approximate problems, Sov. Radio, Moscow, (in Russian) (1979)
4. Botvinnik, M.: Computers in Chess: Solving Inexact Search Problems. Springer Series in Symbolic Computation, with Appendixes, Springer-Verlag: New York (1984)
5. Baghdasaryan T., Danielyan E, Pogossian E.: Supply Chain Management Strategy Provision by Game Tree Dynamic Analysis International Conference: Management of Small and Medium Business: Information Technologies (SBM2006), Sevastopol, Sept. 3-8, (2006) 37–41
6. Gorodetski V., Kotenko I.: Attacks against Computer Network: Formal Grammar Based Framework and Simulation Tool. Proc. of the 5 Intern. Conf. "Recent Advances in Intrusion Detection", Lecture Notes in Computer Science, v.2516, Springer Verlag (2002) 219–238
7. Gorodetski V., Kotenko I., Karsaev O.: Framework for ontology-based representation of distributed knowledge in multiagent network security system. Proc. of the 4-th World Multi-conf. on Systems, Cybernetics and Informatics (SCI-2000), vol. III, "Virtual Engineering and Emergent Computing", Orlando, USA, July (2000) 52-58
8. Dhar V, Stein R.: Seven Methods for Transforming Corporate Data into Business Intelligence, NJ, Prentice-Hall (1997)
9. Dujmovic J.: A Method for Evaluation and Selection of Complex Hardware and Software Systems. The 22nd Inter.Conf. for the Resource Management and Performance Evaluation of Enterprise Computing Systems. CMG 96 Proceedings, Vol. 1 (1996) 368–378
10. Elo A.E.: The Rating of Chess players, Past and Present. London, Batsford (1978)
11. Flavell J.: The Developmental Psychology of Jean Piaget, D.VanNostrand Company Inc., Princeton, New Jersey (1962)
12. Furnkranz J.: Machine Learning in Games: A Survey in “Machines that Learn to Play Games”, Nova Scientific (2001)
13. Gobet, F., Simon, H.A.: Expert chess memory: Revisiting the chunking hypothesis. Memory, 6 (1998) 225–255

14. Gobet, F.: Chunking mechanisms in human learning. Trends in Cognitive Sciences, 5, (2001) 236–243
15. Kaner C., Falk J., Nguen Q.N.: Testing Computer Software. International Thomson Computer Press (1993)
16. Pogossian E.: Specifying Personalized Expertise. International Association for Development of the Information Society (IADIS): International Conference Cognition and Exploratory Learning in Digital Age (CELDA 2006), 8-10 Dec., Barcelona, Spain (2006) 151–159
17. Pogossian E.: On Understanding Of Human Requests By Computers. The Twenty-Eighth Annual Conference of the Cognitive Science Society , July 26-29, 2006, Vancouver, Canada (2006) 221–223
18. Pogossian E., Javadyan A., Ivanyan E.: Effective Discovery of Intrusion Protection Strategies. The Intern. Workshop on Agents and Data Mining, St. Petersburg, Russia, Lecture Notes in Computer Science, Vol. 3505 (2005) 263–274
19. Pogossian E.: Combinatorial Game Models For Security Systems. NATO ARW on "Security and Embedded Systems", Porto Rio, Patras, Greece, Aug. (2005) 8–18
20. Pogossian E.: Business Measurements By On-The-Job Competition Scales. International Conference “Management of Small Business: Problems, Teaching, Future”, Sept. 20th - 24th , Sevastopol (2004) 31–35
21. Pogossian E. Djavadyan A.: A Game Model For Effective Counteraction Against Computer Attacks In Intrusion Detection Systems, NATO ASI, Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management", Tsahkadzor, Armenia, August 19-30 (2003) 823–851
22. Pogossian E.: Focusing Management Strategy Provision Simulation. Proceedings of the CSIT2001, 3d Inter. Conf. in Comp. Sci. and Inf. Technologies, Yerevan (2001) 37–42
23. Pogossian E.: Adaptation of Combinatorial Algorithms. Academy of Sciences of Armenia, Yerevan (1983) 1-.293 (in Russian)
24. Pogossian E., Hambartsumyan M., Harutunyan Y.: A Repository of Units of Chess Vocabulary Ordered by Complexity of their Interpretations. National Academy of Sciences of Armenia, IPIA, (research reports 1974-1980) (in Russian) 1–55
25. Roy D.: Grounding Language in the World: Signs, Schemas, and Meaning Cognitive Machines Group, The Media Laboratory, MIT http://www.media.mit.edu/cogmac/ projects . html) (2005) 1–27
26. Sadovski L.E., Sadovski A.L.: Mathematics and Sport. Moscow, Fiz-Mat Literature (1985)
27. Searle J.: Is the brain’s mind a computer program? Scientific American 262 (1990) 26–31
28. Searle J.: Minds, brains, and programs. Behavioral and Brain Sciences 3 (3) (1980) 417–457
29. Turing A.M.: Computing Machinery and Intelligence. Mind 49 (1950) [Reprinted in Minds and machines. A. Anderson (ed.), Engelwood Cliffs NJ, Prentice Hall (1964) 433–460]
30. Wilkins D.E.: Using Patterns and Plans in Chess. Artificial Intelligence, v.14 (1980) 165–203
31. Winograd T., Flores F.: Understanding Computers and Cognition. A new foundation for design Publishers, Huntington, NY (1986) 11–59
32. Zermelo E. : Uber eine Anwendung der Mengenlehre auf die theorie des Shachspiels. Proceedings of the fifth International Conference of Mathematicians, Cambridge, Cambridge University Press (1912) 501–504

On-Line Agent Teamwork Training Using Immunological Network Model

Lev Stankevich[1] and Denis Trotsky[2]

St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
39, 14-th liniya, Vasilievski Island, St. Petersburg, 199178, Russia
stank_lev@iias.spb.su, den_tr@mail.ru

Abstract. This paper describes a possibility of applying on-line learning techniques to train agents for teamwork. Special modules proposed based on immunological networks capable of on-line learning in dynamically changing environments. The above modules provide adaptive agents' behavior for teamwork after they are trained to select of primitive behaviors under variable environmental conditions. Reinforcement learning is considered to be the main method for training agents during a game. The special agent capable of on-line training for basketball competitions in RoboFIBA virtual system was developed and investigated. Examples of immunological networks for agent teamwork implementation are considered, and results of experiment with hard and training agents are described.

1 Introduction

Currently multi-agent systems are widely implemented in distributed control using a cooperating agents' set. The agents can use various functions of individual and collective behaviors. The individual behavior depends on solving a problem and the agent's role in a system. The collective behavior defines protocols of interacting between the agents and can be differed as cooperative, competitive, or teamwork one.

The teamwork is a special kind of collective behavior based on joint intentions or plans of the agents in achieving a common goal. A control for autonomous dynamical objects performing teamwork aimed at a common goal is one of important problems in this field. The considered objects can be mobile robots, participating in rescue and military operations, or unmanned vehicles performing various tasks in air, water and space, software agents solving industrial, economical, social tasks cooperatively, or robotic agents playing soccer or basketball and participating in rescue team competitions. Teamwork theory and examples of teamwork systems are discussed in detail by Cohen, P. and Tambe, M. [1, 2].

In preceding years various architectures for building teamwork systems have been developed. Logical, reactive, layered, and BDI (Belief-Desire-Intention) architectures [3] are well known. However, a problem of on-line agents' teamwork learning in dynamically changing environment was not thoroughly developed.

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 242–255, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Using on-line learning based data mining techniques allows developing information and control systems capable of learning from their environments [4]. The above can be used for creating intelligent systems and agents capable of solving complex control problems adaptively.

The data mining approach to training intelligent agents has been lately discussed in [5]. It has resulted in creating the powerful intelligent systems that can be defined as distributed systems capable of learning to implement mental reasoning for decision making in non-deterministic, dynamically changing environments.

In case of teamwork, where main problem is to provide dynamical interactions between agents, on-line training can offer significant advantages. The training allows implementing teamwork coordination mechanisms adaptively. It provides agents for capability of selecting primitive behaviors to be cooperated and/or competed in teamwork conditions. At the training reinforcement learning could be considered the main method for on-line learning collective behavior of the agent's teamwork.

The here presented paper also shows how the training agents in the teamwork using adaptive modules capable of fast learning can be implemented. The developed immunological network components can be specialized and adapted by the training through reinforcement learning. Using examples from RoboFIBA framework, it is possible to show that on-line adaptive cooperation of the basketball agents, based on a layered reactive architecture, as well as use of the immunological networks can provide for high efficiency of the teamwork in changing team tactics.

The current work is based on the author's personal experience in multi-agent systems' design of industrial and socio-technical object control [6], special cognitive soccer agents aimed at participation in RoboCup Simulation League competitions [7], basketball agents for RoboFIBA environment [8] and control systems for intelligent robots [9], [10].

2 On-Line Training Agents for Teamwork

On-line training agents for teamwork can be based on use adaptive modules, capable to fast learning. Well-known artificial neural networks can be used for implementing the adaptive modules. However, these networks have significant limitations related to kind of the mapping functions and learning rate. Special modules based on fast adaptive functional approximation and reinforcement learning were developed and examined. Among them cluster module based on neural and fuzzy logics was selected for practical usage and provided good results at training simple operations of soccer agents for RoboCup competitions [4]. However such module is difficultly used for training more complex scenario operations of teamwork.

Recently studying biological immune systems inspirited to development of immunity-based systems (artificial immune systems) that consist of immunological networks [11]. In our opinion, this approach can be effectively used for implementing the adaptive agent behavior. The immune systems are dedicated to self-preservation under hostile environment and have various interesting features such as immunological memory, immunological tolerance, pattern recognition, and so on.

There are several models of artificial immune systems such as models based on idiotypic network hypothesis [12], clonal-selection theory [13], and spatial immune network model [14].

In this work, the idiotypic network model is proposed to be used as basic one because it is most appropriate for selecting agent behavior under changing conditions of the environment. The model is updated by the use of fuzzy logic and named as immunological network model. The adaptive modules based on such model along with reinforcement learning can provide on-line training and data processing as it is described in next sections.

2.1 The Immunological Network Model

Jerne's idiotypic network hypothesis is based on the fact that each type of antibody also has its specific antigen determinant called an idiotope. This fact allowed Jerne to introduce concept of idiotypic network. In the network antibodies/lymphocytes are not just isolated, namely they are communicating to each other among different species of antibodies/lymphocytes. Each antibody has also paratope that able to recognize corresponding antigen. Idea of Jerne can be shortly described as following. Let's the idiotope 1 stimulates the B-lymphocyte 2, which attaches the antibody 2 to its surface, through the paratope of antibody 2. Here, the idiotope of antibody 2 works as an antigen. As a result, antibody 2 suppresses the B-lymphocyte 1 with antibody 1. On the other hand, the idiotope of antibody 3 stimulates antibody 1 since it works as an antigen in view of antibody 1. In this way, the stimulation and suppression chains among antibodies form a large-scaled network and work as a self and non-self recognizer. This regulation mechanism provides a new parallel decentralized processing mechanism.

The idiotypic network can be used to forming behavior selection mechanism of an agent. As it is described by Watanabe et al. [15], preliminary behavior primitive (competence modules) must be prepared. For example, if the agents (e.g. robots) work in the environment with obstacles, current situations (e.g. distance and directional to the obstacle, etc.) detected by installed sensors, work as multiple antigens. Each prepared competence module (e.g. simple behavior) is regarded as an antibody, while the interaction between modules is replaced by the stimulation and suppression between antibodies. The basis concept of the method is that idiotypic network selects a competence module (antibody) suitable for the detected current situation (antigens) in a bottom-up manner.

Dynamics of the idiotypic network can be described using main parameter of concentration of *i*-th antibody, which is denoted by A_i , across following equations:

$$\frac{dA_i(t)}{dt} = \{\alpha \sum_{j=1}^{N} m_{ji} a_j(t) - \alpha \sum_{k=1}^{N} m_{ik} a_k(t) + \beta m_i - k_i\} a_i(t),$$
$$a_i(t+1) = \frac{1}{1+\exp(0.5 - A_i(t))}, \quad (1)$$

Where in first equation, N – the number of antibodies; m_{ji} and m_i – denote affinities between antibody j and antibody i (i.e. the degree of disallowance), and between antibody i and the detected antigen, respectively. The first and second terms of the right hand of the equation denote the stimulation and suppression from other antibodies, respectively. The third term represents the stimulation from the antigen? And forth term the dissipation factor (i.e. natural death). Second equation is squashing function used to ensure the stability of the concentration. Selection of antibodies can be simply carried out on roulette-wheel manner according to the magnitude of concentrations of the antibodies. Note that only one antibody is allowed to activate and act its corresponding behavior to the world.

Using fuzzy-logical framework for representation of equations (1), fuzzy variable of Affinity having two terms such as Stimulation and Suppression can be introduced. Then m_{ji} and m_{ij} can be considered as degrees of stimulation and suppression accordingly. Thus, principles of the idiotypic networks and fuzzy logic are jointed and a concept of immunological networks is appeared.

2.2 Adjustment Mechanism for Training

The immunological network needs in adjustment mechanism that can be considered as the adaptation by changing parameters for prepared network. Such mechanism can be realized by the use special procedure of calculation of degrees of stimuli m_{ij}, which are described in each idiotope. The mechanism starts from the situation where idiotopes of the prepared antibodies are undefined, and then obtains idiotopes using reinforcement learning [16].

Reinforcement learning problem relates to learning from interaction of agent with the environment to achieve a goal. The agent interacts with the environment at each of a sequence of discrete time steps $t_k, k = 0,1,2,3,...$. At each time step, t, the agent receives some representation of the environment's *state*, $s_t \in S$, where S is the set of all possible states, and on that basis selects an *action*, $a_t \in A(s_t)$, where $A(s_t)$ is the set of actions available in state s_t. After each step with an action, the agent receives a numerical *reward*, $r_{t+1} \in R$, and finds itself in a new state, s_{t+1}.

Reinforcement signals of positive rewards and negative rewards (penalties) are used in order to calculate parameters m_{ji} and m_{ij} of each antibody. Let's assume that antigens 1 and 2 invade in immune network and each antigen simultaneously stimulates antibody 1 and 2. Consequently, the concentration of each antibody increases. However, since the priority between antibodies is unknown (because idiotopes are initially undefined), in this case either of them can be selected randomly. Now, let's assume that the network randomly selects antibody 1 and then receives a positive reinforcement signal as reward. To make the network tend to select antibody 1 under the same or similar antigens (situation), we record the number of antibody 1 (i.e. 1) in idiotope of antibody 2 and increase a degree of stimulation m_{12}. In generalized case, modify the degrees of stimulation and suppression can be used such equations:

$$m_{12} = \mu_{Af}^{St} = \max(\mu_{r1}^{p}, \mu_{r2}^{r}), \quad (2)$$

$$m_{21} = \mu_{Af}^{Sup} = \max(\mu_{r2}^{p}, \mu_{r1}^{r}), \quad (3)$$

where μ_{Af}^{St} and μ_{Af}^{Sup} are degrees of membership for terms of Stimulation (*St*) and Suppression (*Sup*) accordingly for fuzzy variable Affinity (*Af*); μ_{r1}^{p}, μ_{r2}^{p}, μ_{r1}^{r}, and μ_{r2}^{r} are degrees of membership for fuzzy terms of Penalty (*p*) and Reward (*r*) for reinforcement signal *R* (antibody 1 and antibody 2). Last four degrees of membership can be calculated using given membership functions for the terms *p* and *r* of the variable *R* through number of times of obtaining penalties and rewards when antibody 1 or antibody 2 are selected.

To investigate possibilities of the immunological networks along with the reinforcement learning procedure for on-line training agents for teamwork, RoboFIBA virtual system was used.

3 RoboFIBA Virtual System

RoboFIBA virtual system is client-server one that consists of server and two competitive teams of basketball agents [8]. The server was developed to be the heart of the client-server system for competition of agent teams. The server was implemented on Delphi 7 language. It is compatible with OS of 32bt Windows family. The basketball agents model players of teams and can be implemented on Delphi or C++ languages.

3.1 RoboFIBA Environment

The RoboFIBA environment is supported by the server. The environment consists of the court, two basket rings with backboard, ball and ten players. The size of the court is *xmax* by *ymax* (fig.1).

The position of each player and ball are defined in 3D space, and any two players can't be located in the same point.

The ball may have status FREE (the ball is free) and BUSY (the ball is taken by player). The player is identified by the number of its team as $TeamID = \{0,1\}$ and by player number in the team - $PlayerID = \{1,2,3,4,5\}$.

In the environment **action model** is used that updates time in discrete steps. A simulation step is 100 ms.

The server can process the limited actions that are defined as following commands sent by a player (one command of each player is executed for one step of time):

1) SHOOT (power *Pow*, direction *DirXY* and *DirZ*). The player shoots the ball with the power *Pow*, in direction of horizontal plane *DirXY*, and in direction of vertical plane *DirZ*.

2) PASS (power *Pow*, direction *DirXY* and *DirZ*). The player passes the ball with power *Pow*, in direction of horizontal planes *DirXY*, and in direction of vertical plane *DirZ*. The ball, moving with the power *Pow* in direction *DirXY* and *DirZ*, is switched in state «FREE».

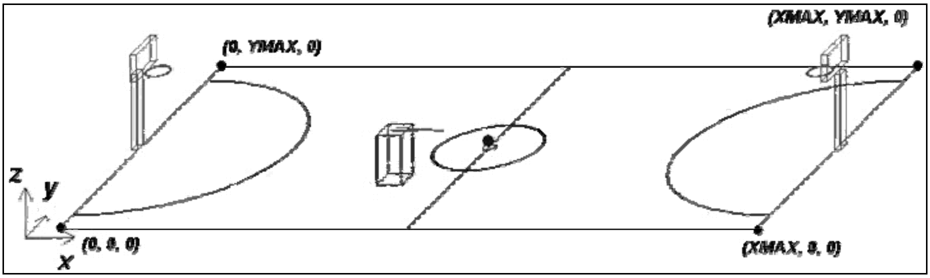


Fig. 1. The RoboFIBA environment

3) RUN (power *Pow*). The player runs with power *Pow* in current direction..

4) TURN_DIRECTION (direction *DirXY*). The player changes its body direction to *Dir XY*.

5) CATCH. The player captures the ball. If distance between the ball and the player is less than *CatchableDist*, the ball belongs to the player. If more then one player is within distance *CatchableDist* to the ball, the ball will belong to the nearest player. Catch action is executed only when the ball is free.

Sensor model implemented in the server allows to send the following information to each player:

1) Own position in 3D space $P = (x, y, z)$;
2) Coordinates, *TeamID*, *PlayerID*, *DirXY* parameters of all partners and opponents;
3) Coordinates and status of the ball, *TeamID* and *PlayerID* of player, who controls the ball, if status of the ball is BUSY.

3.2 Basketball Agent

To provide complex individual and collective behaviors at the teamwork in the RoboFIBA environment, basketball agents must be built using multi layered architecture. The basketball agent has three-level architecture that is similar to the soccer agent architecture used in [7].

The **structure of the agent** is presented in fig.2.

The low level of the basketball agent has several executive behaviors (skills). The middle level of the agent has set of production rules defining individual behavior of the agent. The rules use the conditions in form of situations that are known for the agent and allowed selecting skills that must be realized at low level of the agent. The upper level of the agent also has set of production rules that form corresponding collective agent's behavior related to teamwork. Thus the agent must take into account positions of partners and opponents. In case of arising conflicts the current collective behavior of the conflicting agents are formed.

Base of teamwork is **tactics of agent** defined by the second level. At the second level a player could make decision on attack, defense, or catch the ball. The goal of the team in the attack is to score the ball in a basket ring. The goal of the team in the defense is to not allow the opponents to finish the attack. There are some variants of the organization of the defense. In our case a personal marking of players is used.

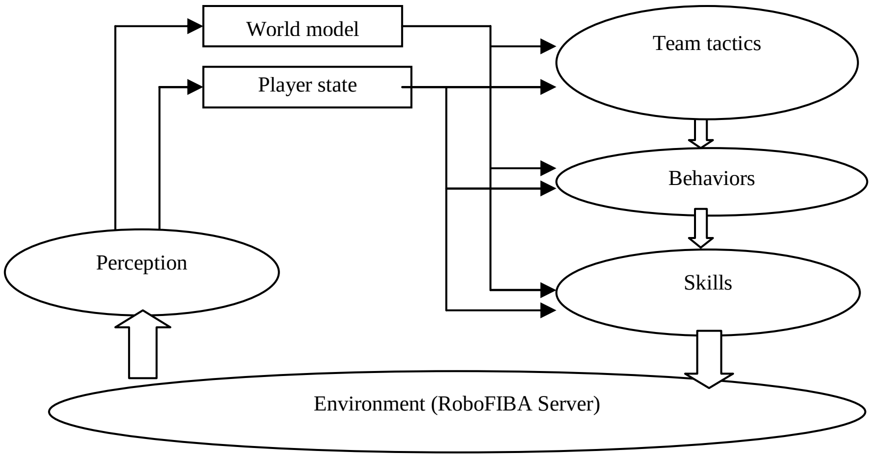


Fig. 2. Stricture of the basketball agent

At the second level (team tactics) the player makes a decision on current action in the team using the approach described in [17]. Typical production rule is described as following:

$$\begin{aligned} &((BALL = FREE) \wedge (t_0 \le t_1) \wedge (t_0 \le t_1)) \Rightarrow (Go\ to\ get\ the\ ball) \\ &else\ (t_1 \le t_2) \Rightarrow (Go\ to\ the\ attackhalf\,), \\ &else\,(Go\ back\ to\ the\ defense\ half\,) \end{aligned} \quad (4)$$

where t_0 is distance between the player and the ball, t_1 is distance between the ball and partner nearest to the ball, t_2 is distance between the ball and opponent nearest to ball.

If the ball is busy, the player uses rules such as:

$$\begin{aligned} &((BALL = BUSY) \wedge (BALLTeamId = PlayerTeamId)) \Rightarrow (Go\ to\ get\ the\ ball) \\ &((BALL = BUSY) \wedge (BALLTeamId \ne PlayerTeamId)) \Rightarrow (Go\ to\ the\ attackhalf), \end{aligned} \quad (5)$$

where $BALLTeamId$ is number of team, who controls the ball, $PlayerTeamId$ is number of player in the team.

Attack actions. The goal of the team in attack is to score the ball in the basket ring. First it is necessary to deliver the ball up to the basket of opponent. The ball can move on the court using two of the ways:

1) Player runs with ball;
2) Player passes the ball to partner;

Two tactics can be used such as selection of partner nearest to opponent basket ring and selection of partner, whose position is optimal. The algorithms implementing these strategies are described in detail in [17]. Optimality of the position of each of partners on the court is described by some value. This value is named as evaluation of the player. When a player needs to pass the ball, he selects teammate with the highest evaluation.

Evaluations of all partners are defined in following way:

$$t_1^n = f(d_1^n) + \sum_{\forall m, d_{2m}^n < \varepsilon} g(d_{2m}^n), \qquad (6)$$

where d_1^n is the distance between n-th teammate and basket; d_{2m}^n is the distance between n-th teammate and m-th opponent; $f(x)$ is a function that evaluates the goodness to shoot for the teammate; $g(x)$ is a function that evaluates the threat from opponents.

When partner is defined, ball's trajectory is calculated. Initial parameters of a pass are calculated such a way that the ball has not been intercepted by opponent and the given partner will be the first player who can intercept the ball.

Defense actions. As defensive strategy, the personal marking of players is used. The player i chooses of the opponent j for marking if the following condition is satisfied.

$$PlayerID_i = PlayerID_j. \qquad (7)$$

Further it is necessary to choose a position on a field, to which the player should move to mark the opponent. Coordinates of this position are calculated by the following formula:

$$P_i = (P_j + (P_j + CircleOwn)/2))/2, \qquad (8)$$

where P_j is coordinate of the j-th opponent; *CircleOwn* is coordinate of the own basket ring; P_i is required position.

P_i is the point located as a midpoint linking opponent and midpoint, linking opponent and the basket ring. If player is closer to the opponent, then marking player will be closer to him.

At the moment of a pass or a throw when the ball is switched in a status FREE, the marking player can make interception of the ball. If it was possible, then the player is switched to attack, else player is switched to defense.

4 Experiment on RoboFIBA Agent Teamwork Training

In previous section the tactics of the agents with two functions of selection of a partner for a pass were considered:

1) Tactics based on function of nearest partner (selection of partner nearest to opponent basket ring);

2) Tactics based on function of optimal position (selection of partner, whose position is optimal).

Efficiency of each system of a pass was estimated through tactical and technical parameters of each team. The tactical and technical parameters include: the score, 2 points throws (success/all), 3 points throws (success/all), pass (success/all), accuracy of a pass (%), and possession of a ball (%).

To investigate efficiency of each system was made match of two teams. Each of them used one of functions of a pass. Team A used the function of optimal position and was trained using on-line learning immunological networks and special coach agent as it is explained in section 4.2. Team B used the function of nearest partner implemented as a hard set of production rules described in section 3.2 by (4) and (5) expressions. In fig. 3 the snapshot of the game is shown.

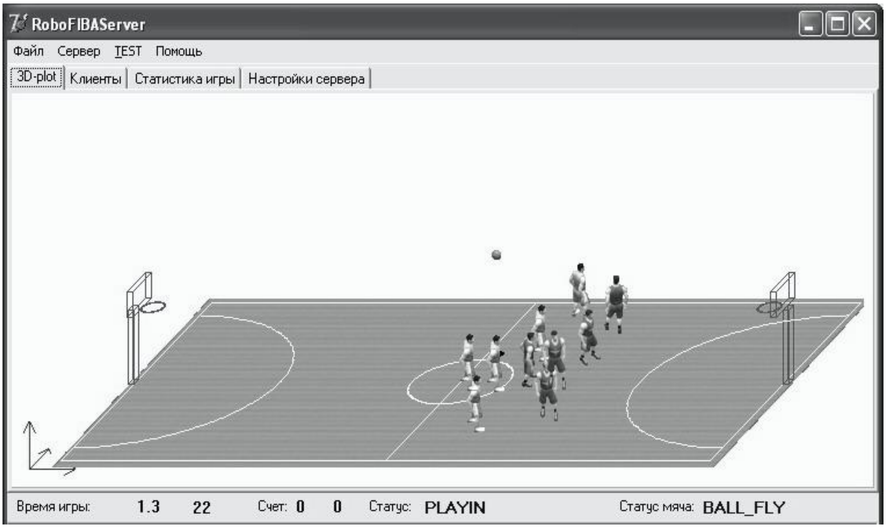


Fig. 3. Snapshot of game

4.1 Immunological Network for Simple Agent Behavior Selection

At first, in the aim of explanation, let's consider example of immunological network which must select behavior of the agent in **simple situation**: basketball agent tries to solve either to throw the ball to opponent ring, to catch ball, or to pass to other agent (partner) such a way, that it will not be intercepted by another agent (opponent).

The idea is calculation of the distances to ring, opponent, and ball positions. Agent calculates these parameters (conditions) that denotes as antigens.

All these conditions have resulted in applying immunological network represented in fig.4, which can learn from given examples, and then work even in these situations, which were not given to system at the learning time.

The network can be described as following. Four antibodies are prepared in advance that respond each to corresponding antigen. If antibody 1 is activated, it means that antigen 1 (Ring.Near) is detected and "Throw to Ring" behavior is selected. However, if opponent is near (Opp.Near), it antibody would give way to other antibodies represented by its idiotope (in this case, antibody 4) to make "Pass to Partner" behavior. Now assume that opponent is far (Opp.Far), in this case antibodies 1, 2, and 4 are

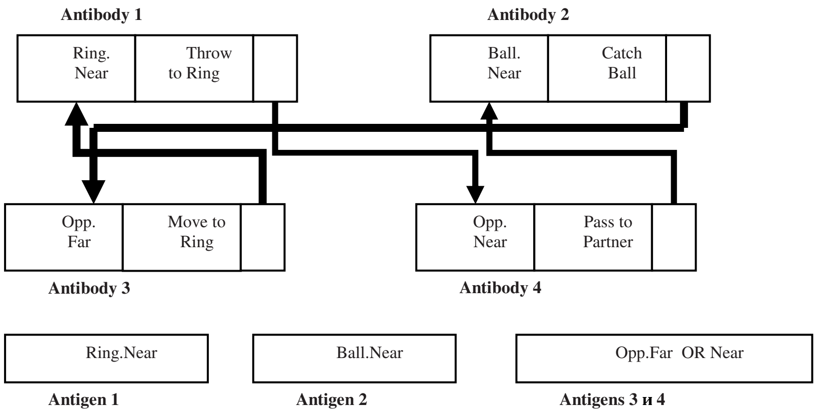


Fig. 4. Example of the simple network

simultaneously stimulated by the antigens. As a result, the concentration of these antibodies increases. However, due to the interactions indicated be arrows among the antibodies through their paratopes and idiotopes, the concentration of each antibody varies. Finally, antibody 2 will have the highest concentration, and then is allowed to be selected. This means that player do "Catches Ball" behavior. In the case where opponent is near (Opp.Near), antibody 1 tends to be selected in the same way. This means that player with the ball tries to do "Move to Ring" behavior. This example shows that interactions among the antibodies work as a priority adjustment mechanism.

4.2 Immunological Network for Scenario Agent Behavior Control

Real experiment supposes that one team uses the original rule-based control system of the basketball agent, while other team uses immunological network system after the training by the coach. The experiment relates to scenario behavior. Basketball agent can percept several situations (antigens) and form in respond corresponding behavior (if antibodies recognize these antigens via their paratopes).

Antigens/Preconditions of paratopes:

Time for attack remained is much or few (Time.Much/Time.Few);

Player is free or marked (Iam.Free/Iam.Marked);

Distance to ring is little, middle, or far (Ring.Near/Ring.Middle/Ring.Far);

Partner is free or marked (Part.Free/Part.Marked).

Behaviors of paratopes:

Player can select such **behaviors** as to throw to ring (Shoot), to dribble of ball (Dribble), to pass to partner (Pass), or to explore of situation (Explore).

Scenario of attack to ring of opponent is considered. Formally, scenario consists of two parts – conditions of initialization, and control system, or policy (the rules of acting in the scenario). For example, three players can be involved in this scenario – two teammates, trying to go to ring and throw ball to ring, and one opponent, defending the ring and trying to intercept ball.

On-line training of basketball agent was carried out using the special component of coach which conducts the training. Method of reinforcement learning is used that determinates of the policy, which allows agents to throw to ring as fast as possible. Only one agent learns to act, it is agent, possessing the ball (player). This agent starts the scenario. It must learn what to do in every step of the any episode: dribble to ring or pass ball directly to partner. The partner, having received the ball, just throws the ball into the ring along optimal throwing direction. The episode finishes successfully, if the ball goes into ring. Situation is considered as failing, if opponent catches the ball. All other possible terminal states are considered as errors, and aren't processed.

According to the reinforcement principle, following rewards and penalties were taken: for every episode step the reward is $r_{forEpisodeStep} = 0.001$, for successful end the reward is $r_{forSuccessEnd} = 1.0$, and for fail the penalty (negative reward) is $p_{forFailureEnd} = -1.0$. The rewards and penalty are taken to agents using the coach.

The immunological network obtained by the training is represented in fig. 5.

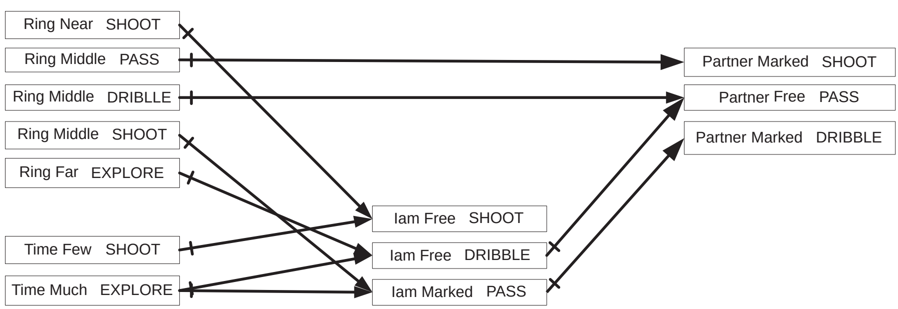


Fig. 5. Immunological network for experiment

In fig. 5 the interactions between antibodies with degrees of stimulations and suppressions equal or more than 0.6 are shown.

The network selects behaviors in according with following explanation. If Time for Attack is much (Time.Much) and Distance to Ring is big (Ring.Far), then player with ball can select "Explore" behavior but priority of behaviors "Pass" or "Dribble" is also defined from state of player (Free or Marked) and state of partner (Free or Marked). If Time for Attack is few (Time.Few) or Distance to Ring is little (Ring.Near), then always the highest priority has "Throw to Ring" behavior (Shoot).

Typical snapshots of the episode are shown in fig. 6.

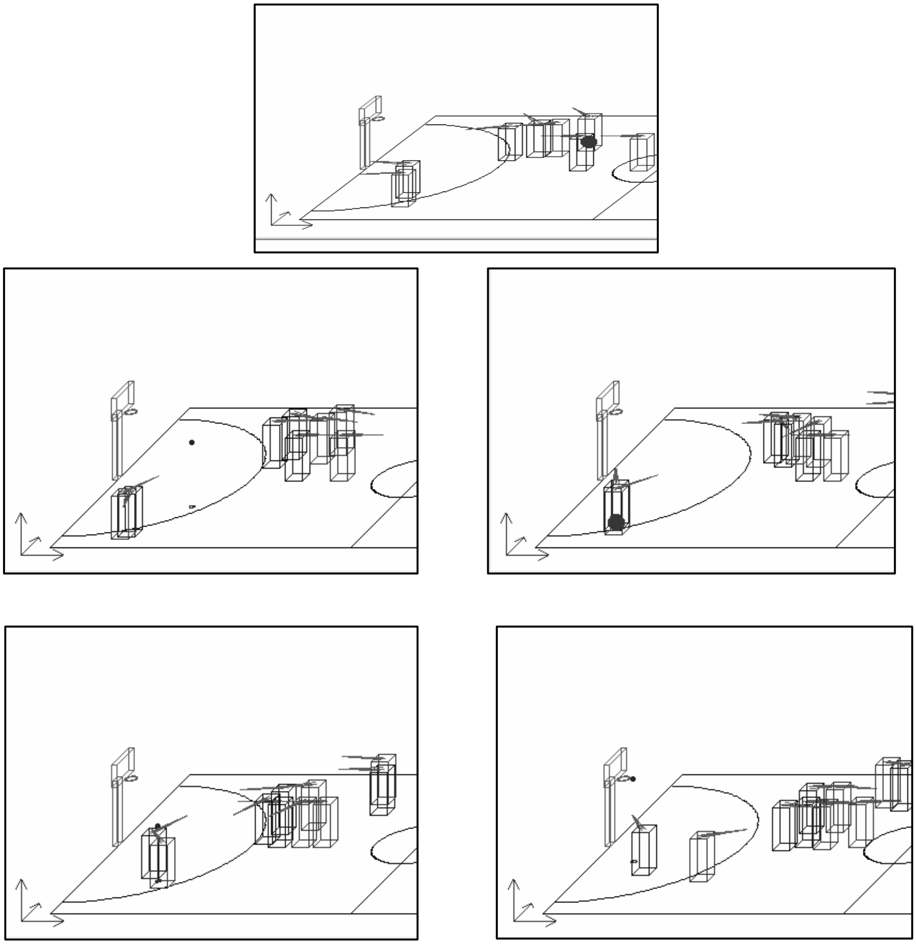


Fig. 6. Snapshots of ring attack episode

In the table the tactical and technical parameters of experiment are presented.

Table 1. Experiment results

Parameters	Team A	Team B
Score	10	5
2 points throws (success / all)	2 / 8	1 / 2
3 points throws (success / all)	2 / 7	1 / 1
Pass (success / all)	33 / 64	20 / 65
Accuracy of a pass (%)	52	31
Possession of a ball	52	48

Several evaluations can be concluded from this table.

Number of the passes and accuracy of the pass: The higher parameters in accuracy of a pass for the Team A are caused by using the function of optimal position; this function first of all prefers those players, who mark of the opponent less, and hence probability that the ball will be intercepted will be less.

Possession of the ball: The higher parameter in possession of the ball for the Team A is consequence of the superiority in accuracy of a pass. The team better plays a pass because it more holds the ball.

The score and throws: Team A has demonstrated better game in a pass; therefore it has advantage in the score as well as throws. The Team B only 3 times has finished the attack with a throw on a basket ring while the Team A has executed 15 throws. Low accuracy of a throw of the Team A has not allowed having considerable advantage in the score. The throws were not found effective since in order to obtain high accuracy of a throw, more long training of this skill is required.

Factor of adaptation: The experiment demonstrated that different tactics of the same team allow achieving different results. Therefore adaptation of the team to behavior of any other team is important factor of winning strategy. Such adaptation can be effectively performed using the immunological networks developed.

5 Conclusion

Layered reactive architecture and the special modules capable to be trained using on-line learning procedure for agent teamwork were developed. The modules based on immunological network model along with reinforcement learning have shown increasing effectiveness of agent teamwork in dynamically changing environments.

Experiment was run using Basketball server and agents that have been developed for competition of basketball agent teams in RoboFIBA virtual system. The experiment was made using typical three-layered agent architecture and the developed set of behavior rules for agents of one team and the immunological network modules capable of on-line training for the second team. The experimental game of the teams has shown that training agents forming immunological networks automatically can provide an increasing effectiveness of attack and defense of the team as compared with a previous version having a hard set of rules in the sense of adaptive possibilities under changing teams' tactics.

In the future it is intended to extend the applications' scope of the learning modules based upon the immunological networks for different types of simple and complex scenario behaviors (skills and scenarios). It is assumed to use the same modules to train agents for more complex 3D game environments such as RoboCup soccer 3D server and Rescue server. The agents capable of competing in such environments have to possess more complex individual and collective behaviors in order to win in games against serious opponent teams.

References

1. Cohen, P. R., Levesque, H. J.: Teamwork, Nous 25 (1991)
2. Tambe, M. Towards flexible teamwork, Journal of Artificial Intelligence Research (JAIR) 7, (1997)
3. Bratman, M.E., Israel, D.J., and Pollack, M.E.: Plans and resource-bounded practical reasoning. In Computational Intelligence, #4: 349–355 (1988)
4. Stankevich, L.A., Serebryakov, S.V., Ivanov, A.V.: Data Mining Techniques for RoboCup Soccer Agents. Autonomous Intelligent Systems^ Agents and Data Mining. Proceedings of International Workshop. AIS-ADM 2005 (St. Petersburg, Russia, June 6-8, 2005). Springer (2005)
5. Mitkas, P.: Knowledge Discovery for Training Intelligent Agents: Methodology, Tools, and Applications. Proceedings of International Workshop AIS-ADM 2005, St. Petersburg, Springer (2005)
6. Nurulin, Y.R., Stankevich, L.A.: Cognitive modeling and management of socio-technical systems. Proceeding of ICIL'99, S-Petersburg, Russia, 28 June-1 Jule 1999, Printed by Univ. of Southampton, UK (1999)
7. Stankevich, L.A.: A cognitive agent for soccer game. Proceeding of First Workshop of Central and Eastern Europe on Multi-agent Systems (CEEMAC'99), Printed by "Anatolyi", S-Petersburg (1999)
8. Stankevich, L., Trotsky, D.: 8. Humanoid-Robot Interaction: Group Behavioral Level. Proceedings of 11th International Conference “Speech and Computer” SPECOM’2006, S-Petersburg (2006) “Anatolya” Publishers, 616 p., p.571–576
9. Stankevich, L.A.: Multiagent technology in cognitive control systems for autonomous robots. «Extreme robotics. Proceeding of X-th Conference, St. Petersburg», St. Petersburg STUPress (1999) (in Russian).
10. Stankevich, L.A.: Cognitive structures and agents in intellectual robot control systems. AI News #1 (2004) 41–55 9 (in Russian).
11. Dasgupta, D., Attoh-Okine, N.: Immunity-Based Systems: a Survey. Workshop Notes-Workshop 4: Immunity-Based systems, International Conference on MAS, December, Keihanna Plasa, Tokyo, Japan (1996) 13–25
12. Jerne, N.K.: Towards a network theory of the immune system // Annales d’Immunologie (Institut Pasteur) (1974) 125C, 373–389
13. De Castro, L.N. and Timmis, J.: An artificial immune network for multimodal optimization // 2002 Congress on Evolutionary Computation. Part of the 2002 World Congress on Computational Intelligence, Honolulu, Hawaii, USA (2002) 699–707
14. Tarakanov, A.O., Kvachev, S.V., and Sukhorukov, A.V.: A Formal Immune Network and Its Implementation for On-Line Intrusion Detection // LNCS 3685: MMM-ACNS 2005 / Gorodetsky V., Kotenko I., and Skormin V. (Eds.), Springer-Verlag, Berlin Heidelberg (2005) 394–405
15. Watanabe, Y., Ishiguro, A., and Uchikawa, Y.: Decentralized Behavior Arbitration Mechanism for Autonomous Mobile Robot Using Immune Network // Artificial Immune Systems and Their Applications / Dasgupta D. (Ed.)”, Spring-Verlag, Berlin (1999)
16. Sutton, R.S. and Barto, A.G.: Reinforcement Learning: An introduction. MIT Press, Cambridge, MA (1998)
17. Bingchen Hu, Jiming Liu, Xiaolong Jin: From local behaviors to global performance in a multi-agent RoboNBA system. In Proceeding of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2004, Beijing) (2004) 309–314

Combination of Rough Sets and Genetic Algorithms for Text Classification

Rujiang Bai, Xiaoyue Wang, and Junhua Liao

Shandong University of Technology Library Zibo 255049, China
{brj,wangxy,ljhbrj}@sdut.edu.cn

Abstract. Automatic categorization of documents into pre-defined taxonomies is a crucial step in data mining and knowledge discovery. Standard machine learning techniques like support vector machines(SVM) and related large margin methods have been successfully applied for this task. Unfortunately, the high dimensionality of input feature vectors impacts on the classification speed. The kernel parameters setting for SVM in a training process impacts on the classification accuracy. Feature selection is another factor that impacts classification accuracy. The objective of this work is to reduce the dimension of feature vectors, optimizing the parameters to improve the SVM classification accuracy and speed. In order to improve classification speed we spent rough sets theory to reduce the feature vector space. We present a genetic algorithm approach for feature selection and parameters optimization to improve classification accuracy. Experimental results indicate our method is more effective than traditional SVM methods and other traditional methods.

Keywords: Document Classification; Support Vector Machine; Rough Sets; Genetic Algorithms.

1 Introduction

Due to the rapid growth in textual data, automatic methods for organizing the data are needed. Automatic document categorization is one of these methods. It automatically assigns the documents to a set of pre-defined classes based on their textual content. Document categorization is a crucial and well-proven instrument for organizing large volumes of textual information. There are many classification methods for textual data. A support vector machine, named SVM, was suggested by Vapnik (1995) and has recently been used in a range of problems including pattern recognition (Pontil and Verri, 1998), bioinformatics (Yu, Ostrouchov, Geist, & Samatova, 1999), and text categorization (Joachims, 1998).

When using SVM, three problems are confronted: (1) how to reduce the high dimension of feature vectors; (2) how to choose the optimal input feature subset for SVM, and (3) how to set the best kernel parameters. These three problems are crucial, because the feature subset choice influences the appropriate kernel parameters and vice versa (Frohlich and Chapelle, 2003). Therefore, obtaining the optimal feature subset and SVM parameters is important.

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 256–268, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In the literature, only a few algorithms have been proposed for SVM feature selection (Bradley, Mangasarian, & Street, 1998; Bradley and Mangasarian, 1998; Weston et al., 2001; Guyon, Weston, Barnhill, & Bapnik, 2002; Mao, 2004). Some other genetic algorithms(GA)-based feature selection methods were proposed (Raymer, Punch, Goodman, Kuhn, & Jain, 2000; Yang and Honavar, 1998; Salcedo-Sanz, Prado-Cumplido, Perez-Cruz, & Bousono-Calzon, 2002). However, these papers focused on feature selection and did not deal with attribute reduction and parameter optimization for the SVM classifier.

In addition to the feature selection, proper parameters setting can improve the SVM classification accuracy. The parameters that should be optimized include penalty parameter C and the kernel function parameters such as the gamma (γ) for the radial basis function (RBF) kernel. To design a SVM, one must choose a kernel function, set the kernel parameters and determine a soft margin constant C (penalty parameter). The Grid algorithm is an alternative to finding the best C and gamma when using the RBF kernel function. However, this method is time consuming and does not perform well (Hsu and Lin, 2002; LaValle and Branicky, 2002).

In order to improve SVM classification speed and accuracy, we proposed a new method. First, Rough Sets Theory (RST) is used to reduce feature vectors after data preprocess. Second, genetic algorithms are used to select feature and optimize the parameter for SVM.

This paper is organized as follows: a brief introduction to the SVM is given in Section 2. Section 3 describes Rough Sets Theory. Section 4 describes basic GA concepts. Section 5 describes the system overview. Including: algorithm of RST-based attribute reduction; and GA-based feature selection and parameter optimization. Section 6 presents the experimental results from using the proposed method to classify test datasets. Section 7 draws a general conclusion and describes future work.

2 Brief Introduction of Support Vector Machine[1~3]

The primary purpose of a support vector machine (SVM) is to use a high dimension space to find a hyper plane to do binary division, where the achieved error rate is minimum. An SVM can handle the problem of linear inseparability.

An SVM uses a portion of the data to train the system and finds several support vectors that represent training data. These support vectors will be formed into a model by the SVM, representing a category. According this model, the SVM will classify a given unknown document by the following classification decision formula

$$(x_i, y_i), \cdots, (x_n, y_n), x \in R^m, y \in \{+1, -1\} \tag{1}$$

Where $(x_i, y_i), \cdots, (x_n, y_n)$ are training samples, n is the number of samples, m is the input dimension, and y belongs to the category of +1 or -1, respectively.

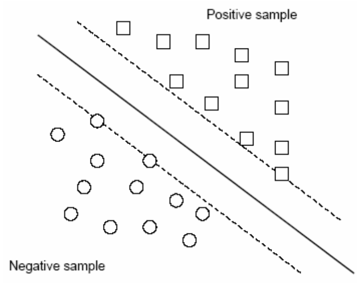


Fig. 1. The hyper plane of SVM

In a linear problem, a hyper plane is divided into two categories. Fig. 1 shows a high dimension space divided into two categories by a hyper plane. The hyper plane formula is: (w·x)+b=0.

The classification formula is:

$$(w \bullet x_i) + b > 0 \;\; if \;\; y_i = +1 \qquad (w \bullet x_i) + b < 0 \;\; if \;\; y_i = -1 \tag{2}$$

However, for many problems it is not easy to find a hyper plane to classify the data. The SVM has several kernel functions that users can apply to solve different problems, such as radial basis function, sigmoid, Polynomial etc.

Radial basis function kernel is:

$$k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \tag{3}$$

3 Rough Sets Theory

The rough sets theory has been developed for knowledge discovery in databases and experimental data sets [4~8]. The rough sets theory deals with information represented by a table called an information system. This table consists of objects (or cases) and attributes. The entries in the table are the categorical values of the features and possibly categories. It also denoted to attribute reduce.

An information system is a 4-tuple $S = \langle U, A, V, f \rangle$, where U is a finite set of objects, called the universe, A is a finite set of attributes. $V = U_{a \in A} V_a$ is a domain of attribute a, and $f := U \times A \to V$ is called an information function such that $f(x,a) \in V_a$,for $\forall a \in A, \forall x \in U$.

In the classification problems, an information system is also seen as a decision table assuming that $A = C \cup D$ and $C \cap D = \phi$, where C is a set of condition attributes and D is a set of decision attributes.

Let $S = \langle U, A, V, f \rangle$ be an information system, every $P \subseteq A$ generates a indiscernibility relation IND(P) on U, which is defined as follows:

$$\mathrm{IND(P)} = \{(\mathrm{x,y}) \in \mathrm{U} \times \mathrm{U} : \mathrm{f(x,a)} = \mathrm{f(y,a)}, \forall \mathrm{a} \in \mathrm{P}\} \tag{4}$$

$U / IND(P) = \{C_1, C_2, \cdots C_k,\}$ is a partition of U by P, every C_i is an equivalence class. For $\forall x \in U$, the equivalence class of x in relation $U / IND(P)$ is defined as follows:

$$[x]_{IND(P)} = \{y \in U : f(y,a) = f(x,a), \forall a \in P\} \tag{5}$$

Let $S = \langle U, C \cup D, V, f \rangle$ be a decision table, the set of attributes $P(P \subseteq C)$ is a reduction of attributes C, which satisfies the following conditions:

$$\gamma_P(D) = \gamma_C(D) \text{ and } \gamma_P(D) \neq \gamma_{P'}(D), \forall P' \subset P \tag{6}$$

A reduction of condition attributes C is a subset that can discern decision classes with the same discriminating capability as C, and none of the attributes in the reduction can be eliminated without decreasing its discriminating capability.

4 Genetic Algorithms[9~11]

GAs(Genetic Algorithms) are stochastic and evolutionary search techniques based on the principles of biological evolution, natural selection, and genetic recombination. They simulate the principle of 'survival of the fittest' in a population of potential solutions known as chromosomes. Each chromosome represents one possible solution to the problem or a rule in a classification. The population evolves over time through a process of competition whereby the fitness of each chromosome is evaluated using a fitness function. During each generation, a new population of chromosomes is formed in two steps. First, the chromosomes in the current population are selected to reproduce on the basis of their relative fitness. Second, the selected chromosomes are recombined using idealized genetic operators, namely crossover and mutation, to form a new set of chromosomes that are to be evaluated as the new solution of the problem. GAs are conceptually simple but computationally powerful. They are used to solve a wide variety of problems, particularly in the areas of optimization and machine learning (Davis, 1991; Grefenstette, 1994).

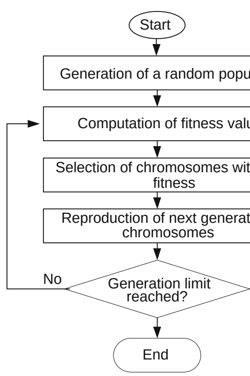


Fig. 2. A typical GA program flow

Fig. 2 shows the flow of a typical GA program. It begins with a population of chromosomes either generated randomly or gleaned from some known domain knowledge. Subsequently, it proceeds to evaluate the fitness of all the chromosomes, select good chromosomes for reproduction, and produce the next generation of chromosomes. More specifically, each chromosome is evaluated according to a given performance criterion or fitness function, and is assigned a fitness score. Using the fitness value attained by each chromosome, good chromosomes are selected to undergo reproduction. Reproduction involves the creation of offspring using two operators, namely crossover and mutation (Fig. 3). By randomly selecting a common crossover site on two parent chromosomes, two new chromosomes are produced. During the process of reproduction, mutation may take place. For example, the binary value of Bit 2 in Fig. 3 has been changed from 0 to 1. The above process of fitness evaluation, chromosome selection, and reproduction of the next generation of chromosomes continues for a predetermined number of generations or until an acceptable performance level is reached.

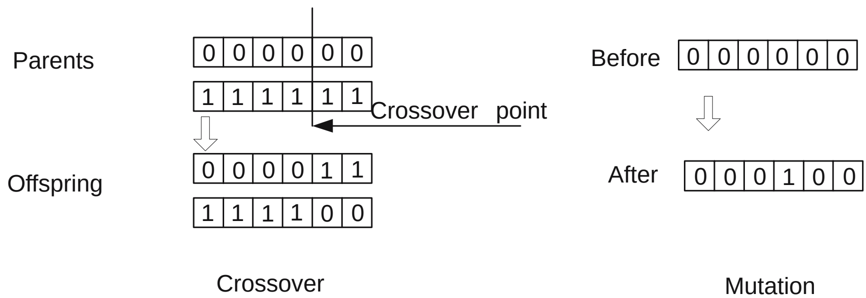


Fig. 3. Genetic crossover and mutation operation

5 System Overview

To improve classification accuracy and speed we proposed a hybrid solution called RGSC(Rough sets and Genetic algorithms for SVM classifier).The system architectures are shown in Fig. 4. The detailed explanation is as follows:

(1) Preprocessing: preprocessing includes removing HTML tags, segmenting words and constructing a Vector Space Model.
(2) Feature reduction by rough sets. Our objective is to find a reduction with a minimal number of attributes, as described in Alg. 1.
(3) Converting genotype to phenotype. This step will convert each parameter and feature chromosome from its genotype into a phenotype.
(4) Feature subset. After the genetic operation and conversion of each feature subset chromosome from the genotype into the phenotype, a feature subset can be determined.
(5) Fitness evaluation. For each chromosome representing C, γ, and selected features, training dataset is used to train the SVM classifier, while the testing dataset is used to calculate classification accuracy. When the classification accuracy is obtained, each chromosome is evaluated by fitness function— formula (8).
(6) Termination criteria. When the termination criteria are satisfied, the process ends; otherwise, we proceed with the next generation.
(7) Genetic operation. In this step, the system searches for better solutions by genetic operations, including selection, crossover, mutation, and replacement.
(8) Input the preprocessed data sets into the obtained optimized SVM classifier.

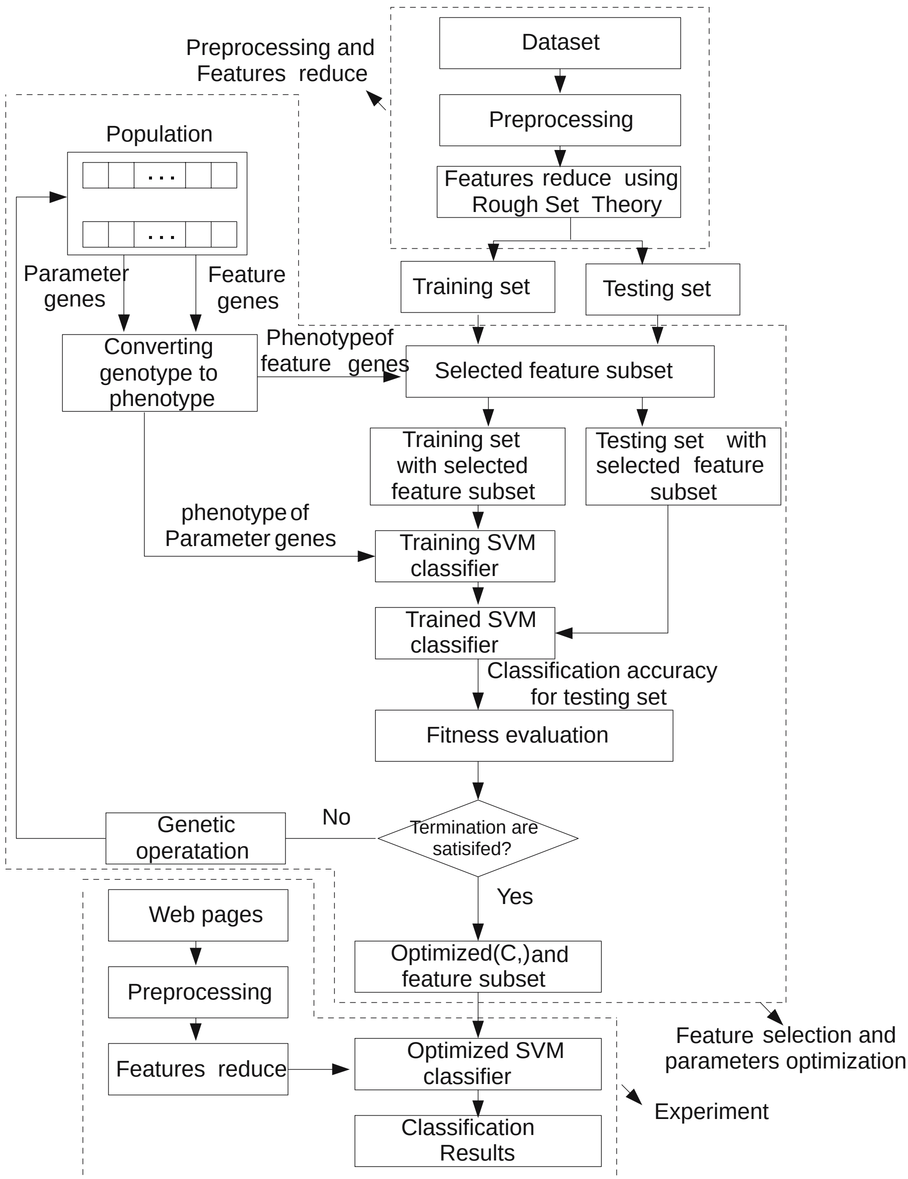


Fig. 4. System architectures

5.1 Algorithm of RST-Based Feature Reduction

Based on section 3, we proposed rough set feature reduction algorithm of finding a reduction of a decision table, which is outlined below.

Algorithm1.Rough Sets Attribute Reduction algorithm
Input: a decision table $T=\langle U, C \cup D, V, f \rangle$, $U=\{x_1, x_2, \cdots x_m\}$, $C=\{C_1, C_2, \cdots C_n\}$
Output: a reduction of *T* , denoted as *Redu*.
1: construct the binary discernibility matrix *M* of *T* ;
2: delete the rows in the M which are all 0's, Redu=ϕ
/* delete pairs of inconsistent objects*/
3: while $(M \neq \phi)$
4: {(1) select an attribute ci in the M with the highest discernibility degree (if there are several c_j (j=1,2,…,m) with the same highest discernibility degree, choose randomly an attribute from them);
5: (2) Redu$\leftarrow$ Redu$\cup\{c_i\}$;
6: (3) remove the rows which have ''1'' in the c_i column from M;
7: (4) remove the c_i column from M; }endwhile
/* the following steps remove redundant attributes from *Redu* */
8: suppose that Redu =$\{r_1, r_2, \cdots r_k\}$ contains *k* attributes which are sorted by the order of entering *Redu*, r_k is the first attributes chosen into *Redu*, r_1 is the last one chosen into *Redu*.
9: get the binary discernibility matrix *MR* of decision table *TR*=$\langle U, \mathrm{Re}du \cup \{d\}, V, f \rangle$;
10: delete the rows in the *MR* which are all 0's;
11: for i = 2 to k{
12: remove the r_i column from *MR*;
13: if (no row in the *MR* is all 0's){
14: Redu$\leftarrow$Redu – {r_i};
15: else
16: Put the r_i column back to MR;
17: Endif ;}
18: Endfor;}

5.2 Chromosome Design

To implement our proposed approach, this research used the RBF kernel function for the SVM classifier because the RBF kernel function can analyze higher-dimensional data and requires that only two parameters, C and γ be defined (Hsu, Chang, & Lin, 2003; Lin and Lin, 2003). When the RBF kernel is selected, the parameters (C and γ) and features used as input attributes must be optimized using our proposed GA-based system. Therefore, the chromosome comprises three parts, C, γ, and the features mask. However, these chromosomes have different parameters when other types of kernel functions are selected. The binary coding system was used to represent the chromosome.

g_C^1	$\cdots g_C^i \cdots$	$g_C^{n_C}$	g_γ^1	$\cdots g_\gamma^j \cdots$	$g_\gamma^{n_\gamma}$	g_f^1	$\cdots g_f^k \cdots$	$g_f^{n_f}$

Fig. 5. The chromosome comprises three parts, C, γ, and the features mask

Fig. 5 shows the binary chromosome representation of our design. In Fig. 5, $g_C^1 \sim g_C^{n_C}$ represents the value of parameter C, $g_\gamma^1 \sim g_\gamma^{n_\gamma}$ represents the parameter value γ, and $g_f^1 \sim g_f^{n_f}$ represents the feature mask. nc is the number of bits representing parameter C, nr is the number of bits representing parameter γ, and nf is the number of bits representing the features. Note that we can choose nc and nγ according to the calculation precision required, and that nγ equals the number of features varying from the different datasets.

In Fig. 5, the bit strings representing the genotype of parameter C and γ should be transformed into phenotype by Eq. (7). Note that the precision of representing parameter depends on the length of the bit string (nc and nγ); the minimum and maximum value of the parameter is determined by the user. For chromosome representing the feature mask, the bit with value '1' represents the feature is selected, and '0' indicates feature is not selected.

$$p = \min_p + \frac{\max_p + \min_p}{2^l - 1} \times d \qquad (7)$$

P phenotype of bit string
minp minimum value of the parameter
maxp maximum value of the parameter
d decimal value of bit string
l length of bit string

5.3 Fitness Function

Classification accuracy, the number of selected features, and the feature cost are the three criteria used to design a fitness function. Thus, for the individual (chromosome) with high classification accuracy, a small number of features, and low total feature cost produce a high fitness value. We solve the multiple criteria problem by creating a single objective fitness function that combines the three goals into one. As defined by formula (23), the fitness has two predefined weights: (i) WA for the classification accuracy; (ii) WF for the summation of the selected feature (with nonzero Fi) multiplying its cost. The weight accuracy can be adjusted to 100 % if accuracy is the most important. Generally, WA can be set from 75 to 100 % according to user's requirements. If we do not have the feature cost information, the cost Ci can be set to the same value, e.g. '1' or another number. The chromosome with high fitness value has high probability to be preserved to the next generation, so user should appropriately define these settings according to his requirements.

$$fitness = W_A \times SVM\_accuracy + W_F \times \left(\sum_{i=1}^{n_f} C_i \times F_i\right)^{-1} \qquad (8)$$

WA SVM classification accuracy weight
SVM_accuracy SVM classification accuracy
WF weight for the number of features

Ci cost of feature *i*
Fi ‘1’ represents that feature *i* is selected; ‘0’ represents that feature *i* is not selected.

6 Experiments

In this section, we designed an experiment to test the performance of the proposed RGSC. We also investigated k-NN and Decision tree to compare their classification performances. The experiments are described below.

6.1 Experimental Eenvironment

Our implementation was carried out on the YALE (Yet Another Learning Environment) 3.3 development environment (Available at:http://rapid-i.com/). Feature reduction by the Rough Sets Theory was carried out on ROSETTA(you can download it from http://rosetta.sourceforge.net/) .The empirical evaluation was performed on Intel Pentium IV CPU running at 3.0 GHz and 1GB RAM.

6.2 Data Set

To provide an overview on the base line accuracy of the classifiers and to compare them with various studies, the Reuters 21578 corpus was taken in our experiments (this collection is publicly available at: http://www.research.att.com/~lewis/reuters21578.html). These stories average about 200 words in length. Various splits of the Reuters 21578 can be used, whereas we followed the ModApte split in which 75 % of the stories (9603 stories) are used to build classifiers and the remaining 25 % (3299 stories) to test the accuracy of the resulting models in reproducing the manual category assignments. From this split, all categories (including the documents not assigned to any category), which have no training or test document were deleted. The resulting data set has 90 different categories and is the same as that used by Joachims (1998).

6.3 The Performance Measure

Given a binary-classification problem of topic versus not-topic, recall is the ratio of the correct topic cases to the total topic cases. Precision is the ratio of correct topic cases to the total predicted topic cases. The standard evaluation criterion for the Reuters benchmark is the breakeven point, at which precision equals recall, and the F1 measure, which is defined as (2×precision× recall)/(precision + recall).

6.4 Simulation

Fig. 6, 7 and 8 show the performance of our proposed method against the decision tree (Weiss et al., 1999) and the k-NN classifiers (Aas & Eikvil, 1999) for the ten most frequent categories.

The precision of the k-NN, Decision tree, and RGSC is shown in Fig. 6, the recall of the k-NN, Decision tree, and RGSC is shown in Fig. 7, the F1-value of the k-NN, Decision tree, and RGSC is shown in Fig. 8, and the speed of the k-NN, Decision tree, and RGSC is shown in Fig. 9.

The average precision for k-NN, Decision tree and RGSC are 75.6, 84.9 and 90.7% respectively. With the exception of categories Grain, Crude, Wheat, Corn, the precision of each category for RGSC is higher than other two methods. This indicates that the RGSC methods perform at generally high precision.

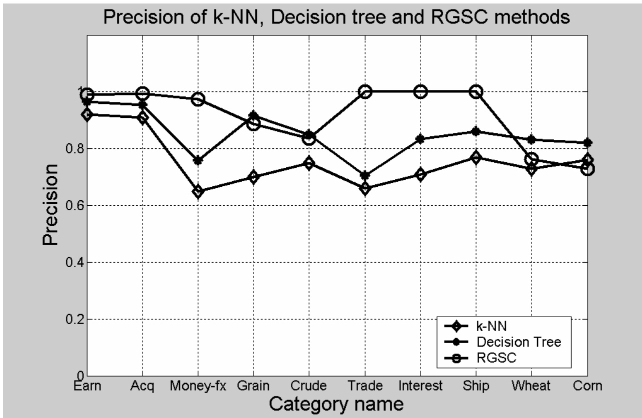


Fig. 6. The precision of the k-NN, Decision tree, and RGSC

The average recall results for the RGSC, k-NN and the Decision tree are 95.1, 82.3, and 87.8 %, respectively. The recall of k-NN and Decision tree are nearly the same and both are lower than the RGSC. The RGSC can classify documents into the correct category mapping to precision, with a high recall ratio (Fig. 6).

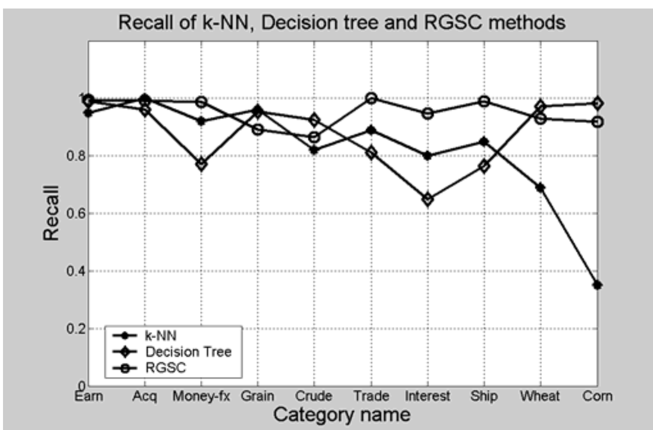


Fig. 7. The recall of the k-NN, Decision tree, and RGSC

The simulated result shows that the average F1-values of the RGSC, k-NN and the Decision tree are 92.5,78.8 and 87.9 %, respectively. This indicates that the RGSC yields a better classification result than the other two methods, but RGSC classification performs lower than Decision tree for the “Grain”, “Crude”, “Sheep” and "Wheat" categories. We will now try to explain why the performance of those four categories is poorer than other categories. We find that the "Grain", "Crude", "Sheep" and "Wheat" categories contain a smaller number of documents in Reuters 21578. This indicates that the RGSC is able to effectively process categories with large documents. But poorer with smaller documents. Fig. 7 shows that among the three methods, the RGSC has the average highest classification result.

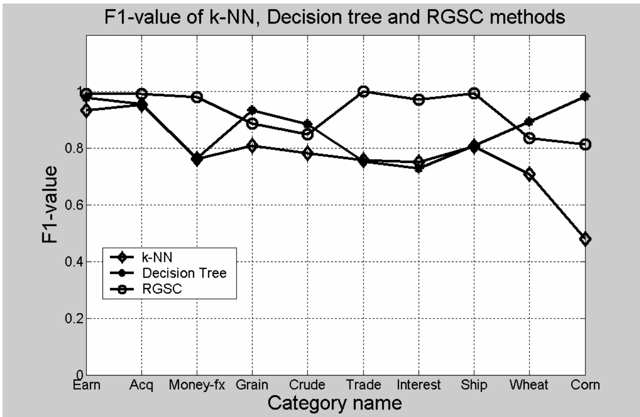


Fig. 8. The F1-value of the k-NN, Decision tree, and RGSC

The speed for RGSC, k-NN and Decision tree are 13.3, 20.5 and 35.7 seconds respectively. This indicates that the RGSC is more effective than the other two methods. It easily explain that the feature is reduced by Rough Set Theory before inputting the SVM classifier.

In general the performance of our approach is best in average.

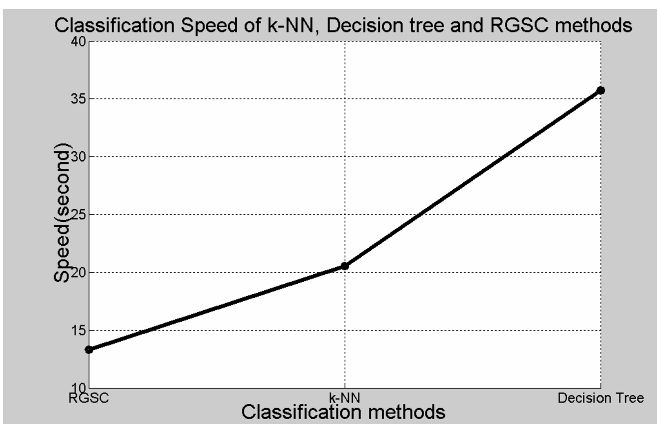


Fig. 9. The speed of the k-NN, Decision tree, and RGSC

7 Conclusion

In this paper, we have proposed a document classification method using an SVM based on the Rough Sets Theory and Genetic Algorithms. The feature vectors are reduced by the Rough Set Theory, and are selected and parameters are optimized by Genetic Algorithms. The experimental results show that the RGSC we proposed yields the best result of these three methods. The experiment also demonstrated that the RGSC yields better accuracy even with a large data set. When the larger category has more training data, the RGSC is able categorize documents with more accuracy. In future research, we will place more emphasis on the kernel function selection and parameters optimization for the Genetic Algorithms to improve the performance of RGSC.

References

1. Burges, C.A.: tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2) (1998) 121–167
2. Chang, C.C., & Lin, C.J. LIBSVM: A library for support vector machines. Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm (2001)
3. Cristianini, N., & Shawe-Taylor, J.: An introduction to support vector machines. Cambridge: Cambridge University Press (2000) 100–103
4. Pal, S.K., Skowron, A. (Eds.): Rough Fuzzy Hybridization: A New Trend in Decision-Making, Springer, Singapore (1983) 36–70
5. Pawlak, Z.: Rough sets, Int. J. Comput. Sci. 11 (1982) 341–356
6. Pawlak, Z., in: Rough Sets, Theoretical Aspects of Reasoning About Data, Kluwer, Dordrecht (1991) 10–50
7. Pawlak, Z., Skowron, A.: Rough membership functions, in: Yaeger R.R., Fedrizzi M., Kacprzyk J. (Eds.), Advances in the Dempster Shafer Theory of Evidence, Wiley Inc, New York, Chichester, Brisbane, Toronto, Singapore (1994) 251–271
8. Pawlak, Z., Wong, S.K.M., Ziarko, W.: Rough sets: probabilistic versus deterministic approach, Int. J. Man-Mach. Stud. 29 (1988) 81–85
9. Davis, L.: Handbook of genetic algorithms. New York, NY: Nostrand Reinhold. (1991) 55–61
10. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley (1989) 23–32
11. Grefenstette, J.J.: Genetic algorithms for machine learning. Boston, MA: Kluwer (1994) 100–106
12. Vapnik, V.N.: The nature of statistical learning theory. New York: Springer-Verlag. (1995).61–70
13. Frohlich, H., & Chapelle, O. Feature selection for support vector machines by means of genetic algorithms. Proceedings of the 15th IEEE international conference on tools with artificial intelligence, Sacramento, CA, USA (2003) 142–148
14. Yu, G.X., Ostrouchov, G., Geist, A., & Samatova, N.F. An SVMbased algorithm for identification of photosynthesis-specific genome features. Second IEEE computer society bioinformatics conference, CA, USA (2003) 235–243
15. Bradley, P.S., Mangasarian, O.L., & Street, W.N. Feature selection via mathematical programming. INFORMS Journal on Computing, 10 (1998) 209–217

16. Hsu, C.W., Chang, C.C., & Lin, C.J. A practical guide to support vector classification. Available at: http://www.csie.ntu.edu.tw/~cjlin/ papers/guide/guide.pdf (2003)
17. LaValle, S.M.,& Branicky, M.S. On the relationship between classical grid search and probabilistic roadmaps. International Journal of Robotics Research, 23 (2002) 673–692
18. Joachims, T. Text categorization with support vector machines. In Proceedings of European conference on machine learning (ECML) Chemintz, DE (1998) 137–142
19. Pontil, M., & Verri, A. Support vector machines for 3D object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(6) (1998) 637–646

Multi-agent Meta-search Engine Based on Domain Ontology

Marat Kanteev[2], Igor Minakov[1], George Rzevski[2],
Petr Skobelev[1], and Simon Volman[1]

[1]The Institute for the Control of Complex Systems, RAS,
Sadovaya Str., 61, 443020, Samara, Russia
{minakov,skobelev,volman}@magenta-technology.ru
[2]MAGENTA Technology, Gainsborough House, 59 - 60 Thames Street
Windsor, Berkshire, SL4 1TX, UK
{kanteev,george.rzevski}@magenta-technology.com

Abstract. This article describes a new approach of HTML pages search via Internet, which is based on the semantic understanding of pages content by means of multi-agent technology. Multi-agent text understanding system, which is the basis of the approach, converts an input query and pages, received from conventional search engines, to formalized semantic descriptors, and evaluates similarity of these descriptors. Both text understanding and descriptor comparison algorithms use the knowledge about problem domain, represented in open and easy-to-update form of ontology. The approach developed was applied to the analysis of web-pages related to car industry. As a result a meta-search engine was developed, capable of analyzing pages, retrieved from traditional search engines and sorting pages by their semantic relevance to the user request. In this article one will find description of the system, testing results and future perspectives.

Keywords: problem domain ontology, multi-agent text understanding, semantic networks comparison, web-based search, meta-search.

1 Introduction and Problem Definition

Today the World-Wide Web has become so popular that the Internet now is one of the main means of publishing information. To simplify the user access to required information, the Internet search engines were created that allowed getting a set of Internet pages on the basis of request keywords defined by the user.

In addition to conventional information search services, the search engines recently provide the users with a new service called “advertising links”. The key point of this service is as follows: large commercial companies provide the administration of this or that search engine with links to their web-site with detailed description of information it contains. As a rule this information is presented as a set of keywords. When the user specifies in his request a keyword that belongs to this set, he gets the reference to this company’s web page. Generally this reference occupies separate space of a page with found results, it is detached from a set of resultant links. Unlike

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 269–274, 2007.
© Springer-Verlag Berlin Heidelberg 2007

general network resources, the set of keywords for advertisement pages is created manually by the customer and search engine experts.

The main problem with using search engines is that the keywords specified by the user are generally too common and abundant. Thus the user gets thousands of useless pages but at the same time fails to find other pages with the same subject stated differently.

Therefore to make inexperienced user work efficiently with search engines, it is necessary to improve the algorithms of analyzing web-pages contents and the user queries in order to generate descriptors that contain monosemantic information on its concepts/relations. Besides comparison methods are required for these descriptors as such methods will allow evaluating similarity of two descriptors and using this evaluation to generate a set of links to the pages containing information required for the user.

2 Multi-agent Approach – Key Points

The proposed system is a semantic meta-search engine. It means that when processing the user query it addresses this query to conventional search engines, gets search results and then analyses the contents of found pages and prioritizes them in decreasing order according to their relevance to the user query.

The main difference between existing meta-search engines is the use of problem domain ontology, which helps to represent and reconstruct semantics of request and available pages. The intelligent multi-agent algorithm of analyzing texts in the natural language is used for this purpose.

Multi-agent approach is developing rapidly in the recent time in a number of applications such as distributed systems, logistic and supply chain scheduling, data mining etc., as described in [1]. New line of multi-agent systems investigation, which is studying open non-equilibrium systems with energy income, as described in [2], is also very important. It was shown in [3], that solution methods of the resource distribution task in open systems for logistic applications can be effectively used as well in the knowledge management, data clustering, natural language processing systems etc. For the text understanding systems, open non-equilibrium state means that the sense of previous words and phrases can be changed and revised during next phrases analysis.

The essence of the used algorithm is that each word of the text is assigned an autonomous software agent capable of negotiating with other similar agents about the meaning of each word in the sentence and its general meaning on the basis of domain ontology.

During negotiations the word agents can speculate about the possible word meanings and their semantic relations, find and resolve meaning conflicts, detect implicit information on the basis of domain knowledge, take into account the context of the word usage within one sentence and inter-phrasal context thus connecting the words of various sentences into one semantic network, as described in [4].

During negotiations the agents use the knowledge about the problem domain in use and language contained in ontology. As the result of analysis of text in the natural language we get text descriptor, which contains formal monosemantic description of the initial text meaning. Descriptor represents a semantic network that consists of ontology concepts and their relations. Unlike the initial text represented as a line it is rather simple to compare such descriptors on similarity of information they contain on the base of the ontology.

The algorithm of comparing semantic descriptors was developed for multi-agent meta-search engine. This algorithm is based on finding in one of the descriptors the sub-network which is close to the network of the other descriptor as much as possible. Similarity degree of two sub-networks is defined as similarity degree of their respective pairs of nodes. Similarity degree of two nodes depends on relative position of corresponding concepts in the ontology and on the values of attributes connected with nodes under comparison. Speed of this algorithm is about 2000 descriptors per second.

Thus the system allows solving such problems typical for traditional systems, as understanding of semantic context for coherent texts, dependency on text language, easy update of knowledge during the course of work.

3 Meta-search Engine Architecture

The general system structure is shown in Fig. 1.

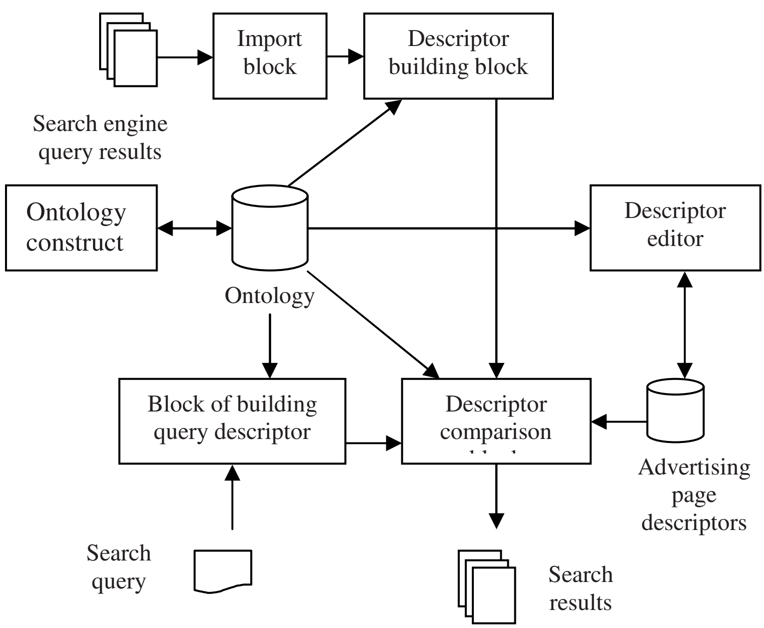


Fig. 1. Meta-search Engine General Structure

The system contains the following main components:

- **Import Block:** provides import of data into the system and transforms the data into the text.
- **Blocks of building page and query semantic descriptors:** automatically convert text information to semantic descriptor on the basis of domain-specific knowledge.
- **Block of comparing semantic descriptors:** compares two semantic descriptors built on the basis of domain-specific knowledge.
- **Ontology:** stores the system knowledge used when processing a query.
- **Ontology constructor:** provides possibilities of editing the system ontology.
- **Descriptors of advertising pages:** stores semantic descriptors of advertising pages.
- **Descriptor editor:** provides possibility of editing semantic descriptors on the basis of ontology.

In general the system workflow can be divided into 2 stages. At the first stage an expert user creates a domain ontology (or can use the existing one). This ontology serves the basis for automatic text analysis and contains semantic relations, syntactic rules and morphological tables of words. E.g., the ontology containing the concepts like “car”, “manufacturer”, “car body”, “engine”, etc. was created for the system under consideration (cars domain). This ontology included 60 objects, 40 attributes, 10 classes of relations and near 2000 instances of relations (including inherited ones). The ontology is created in semi-automated way, where system is provided by a set of documents from a selected problem domain, and all words, excluding non-meaningful, are presented to the user. Furthermore, for each word, basing on its frequency in texts and relative positions comparing to other words initial guess is made, whether it’s an object, relation or attribute value. After that user is able to manually assign each word as an object, relation, attribute or attribute value, and link concepts between themselves. Further on the ontology can be continuously extended with new common words and terms during the process of work.

At the second stage, when the user search query arrives, this query is redirected to conventional search engine. At the same time the module of text processing starts building semantic query descriptor. Currently a creation of semantic descriptor for average site takes approximately 2-3 seconds, but for the set of documents this process can be easily made parallel, as process is independent for each document.

After getting the resultant set of pages from a search engine the definite number of pages (10, 50, 100, etc.) that the search engine considers to match the query best of all is selected. Then the developed meta-search system builds semantic descriptors of these pages and compares them with semantic descriptors of query. As the result the pages are rearranged. According to comparison results, the pages that fit the query most of all, are placed on the top of the list.

Besides the query descriptor is compared to descriptors of advertising pages, stored in the system database. The appropriate pages are also shown to the user in a separate section of results window.

4 Analysis Method and Experiments Results

A number of test experiments were made to analyze the quality of this program.

We've taken most popular examples of requests in car industry domain, according to Google and Overture statistics. In average we analyzed approximately 100 typical requests, 100 pages per request per search engine. Below we are giving figures for most common and simple examples.

A list of 6 popular test queries is:

1. Car purchasing
2. Car rent
3. Car repairs
4. Audio systems
5. Car pictures
6. Car overview

Then each query was sent to Yahoo! search engine and the first 100 results were analyzed using multi-agent meta-search engine. After manual analysis of results the expert put into the table information concerning how many relevant documents contained the first 50 returned by traditional search engine and those returned by meta-search engine. The results are given in the table below:

Table 1. Comparing Effectiveness of Search Using Keywords and Semantic Descriptors

Query	Search using keywords	Search using semantic descriptors
Query #1	64%	74%
Query #2	70%	90%
Query #3	62%	84%
Query #4	60%	88%
Query #5	44%	68%
Query #6	52%	72%
Total	58.67%	79.33%

Next, advertising links returned by Google and meta-search engine for the same queries were also analyzed in terms of their relevance to the query. Each reference was considered by the expert to be either relevant or not relevant. Results are given in the table 2 (below).

To summarize all our analysis, meta-search system proved to be more effective than conventional search engines and traditional meta-search systems. During automatic generation of information about the page (usual search mode) the average number of relevant pages was 20-22 % higher than that of traditional search engines. During manual generation of information about the page (advertising links) the difference was a little bit lower – 15-16%, because of more thorough selection of keywords for advertising pages in traditional search engines.

Table 2. Comparing Search Modes Using Advertising Pages

Query	Search using keywords		Search using semantic descriptors	
	Quantity	Relevance	Quantity	Relevance
Query #1	8	75%	15	87%
Query #2	8	75%	4	100%
Query #3	2	50%	3	100%
Query #4	5	100%	3	100%
Query #5	2	100%	2	100%
Query #6	0		7	86%
Total	4.17	80.00%	5.67	95.50%

5 Conclusion

Therefore, the following results were achieved:

- Initial version of meta-search engine was developed providing interaction with the most popular search engines, storage and editing of its own set of advertising pages. It also provides debugging interface that allows to control the process of generating and comparing semantic descriptors and get information about decision making process of the system;
- Developed methods of representing knowledge of problem domain and algorithms for semantic representation of pages content, allow easy customization for any professional problem domain by common users, thus increasing quality of search and relevancy of advertising;
- Application of this system allows improving the effectiveness of searching text information in the network and other sources, increasing relevance of results, especially for those who doesn’t have enough experience of using advanced methods of search engines.

References

1. Wooldridge, M.: An Introduction to MultiAgent Systems: New-York, Jonh Wiley & Sons (2002)
2. Nicolis, G., Prigogine, I.: Self-Organization in Nonequilibrium Systems: New-York, Jonh Wiley & Sons (1977)
3. Skobelev, P.: The Theoretical Basis of the Open MAS Development for Operational Data Processing in the Decision Making Process. In 5th International Conference ‘Complex Systems: Control and Modelling Problems’, Samara, Russia (2003) 295–303
4. Andreev V., Iwkushkin K., Karyagin D., Minakov I., Rzevski G., Skobelev, P., Tomin, M.:: Development of the Multi-Agent System for Text Understanding. In 3rd International Conference ‘Complex Systems: Control and Modelling Problems’. Samara, Russia (2001) 489–495

Efficient Search Technique for Agent-Based P2P Information Retrieval

Byungryong Kim[1] and Kichang Kim[2]

[1] Department of Computer and Science Engineering, Inha University, Incheon, Korea
[2] School of Information and Communication Engineering, Inha University, Incheon, Korea
{doolyn,kchang}@inha.ac.kr

Abstract. There have been many studies on the design of P2P systems for effective keyword search. This paper proposes and tests technique to reduce network traffic due to many inverted lists forwarded in carrying out query containing multi-keywords in DHT base structured p2p network. Many of inverted lists, forwarded inter-medium, are discarded regardless of search result. This paper proposes Distance and Smart-bloom filter to diminish those unrelated inverted lists. Distance can correctly distinguish document not containing a certain keyword. Smart-bloom filter with low false positive rate can sort out document with a high chance of including certain keyword among inverted lists selected by distance. Ultimately large amount of unrelated inverted lists can be diminished. The performance of Distance and Smart-bloom filter was tested through simulation and the traffic was decreased by 67%.

Keywords: Smart-Bloom Filter, Distance-Based Pruning, DHT, P2P.

1 Introduction

In early P2P systems, documents are serviced directly by the servers which own them, and a query asking the location of some document has to be broadcasted to all agents. This early technique is called Partition-by-Document and is still being used in Gnutella [1], Google [2], Yahoo [3], KaZaA [4], etc. The query could contain the title of the file or keywords in it. For the latter case, the serving agents manage a local index table that maps keywords to documents.

It is obvious that we need more structured way of managing files to avoid the expensive query broadcasting. A Distributed Hash Table DHT) is used to map a keyword to the peer responsible for holding the corresponding reverse index. Partition-by-Keyword uses DHT to store the indices of files, and uses these tables to find the location of a target document. It hashes the keyword of a document and stores the index of the document in a agent which controls the hash table corresponding to the keyword's hash value. Since a document can contain several keywords, the index of this document can be stored in a number of agents. To find the location of a document that contains a keyword x, the client peer simply hashes it and send a query to a agent which controls the corresponding hash table.

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 275–286, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Partition-by-Keyword is much superior to Partition-by-Document in finding a document quickly with given keywords. However it still poses some performance problems, especially when the query contains multiple keywords. Multiple keyword search is common, and it is known that 71.5% of search queries in Internet contains at least two keywords [5]. For multiple keywords, the query should be sent to all agents that control these keywords. Since the inverted lists (lists mapping the keyword to a list of documents that contain it) from the relevant agents need to be combined through JOIN operation (intersection), the query is passed around with the inverted list being updated at each agent. If the first keyword in the query was a common keyword, the starting inverted list would be huge, and transmitting this huge list to the next agent would cause a heavy traffic. Usually the keywords in a particular query consist of a set of common words and another set of specific words that occur only in a limited number of documents. Most of the indices of documents in the starting inverted list, therefore, will most likely be dropped through JOIN operation sooner or later.

It would be beneficial if we can cut down these irrelevant indices from the inverted list before transmitting it to the next agent. But how can we know which indices will eventually be dropped? In this paper, we suggest a technique based on the concept of distance and smart-bloom filter between keywords that can remove irrelevant indices from the list. Our technique is explained in detail later in the paper, and the preliminary result shows the technique is very promising. The rest of the paper is organized as follows: Section 2 explains the basic operations of DHT-based P2P system and surveys searching techniques for multiple keywords; Section 3 explains the proposed technique in detail; Section 4 evaluates its performance; and Section 5 gives a conclusion.

2 Related Works

DHT-based P2P system [6, 7, 8, 9] stores the document IDs and the addresses of the owner agents, represented by a 2-tuple, {document ID, URI of agent ID}, in a distributed way at each agent. The document ID is obtained by hashing some unique information of each document such as URI, while agent ID is computed through hashing its IP address. To ensure an even distribution of the identification numbers, a consistent hashing technique is used [10]. Since document IDs and agent IDs do not always match one-to-one, we order the agent IDs in an increasing order and store a document ID to the first agent whose agent ID is greater than or equal to the document ID. If we know all the agents that are participating in the current P2P system, it is simple to compute the location of the controlling agent that has the target document ID -- it is simply the first agent in the ordered agent list. However, a typical P2P system contains a huge number of participating agents and many of the agents constantly enter or exit the system. In this large changing network of agents, it is very hard to keep track of all active agents correctly. Instead, many DHT-based search techniques adopt a strategy where only some reference agents are remembered at each agent, and use them to hunt down the location of the target document ID. Which agents are selected as reference agents varies with the search technique.

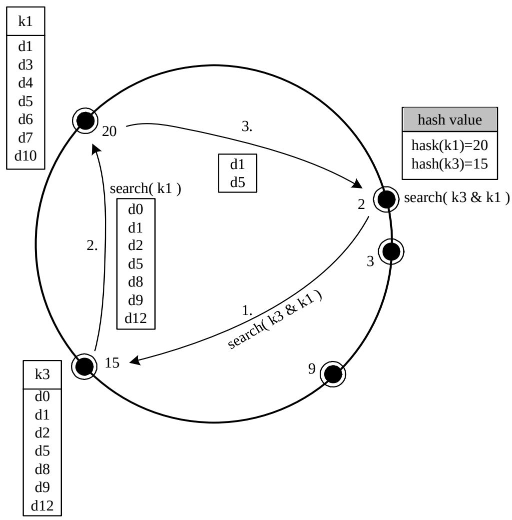


Fig. 1. Keyword search in Chord-ring with 5-bit id space

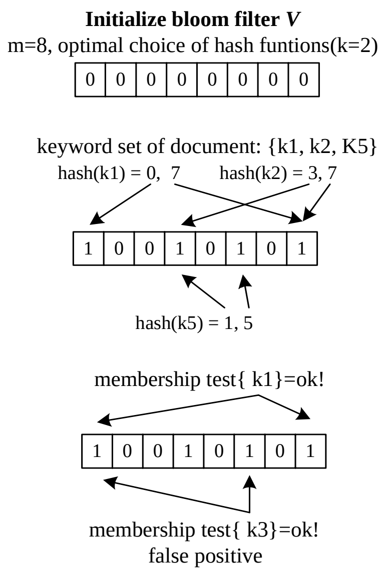


Fig. 2. 8bit bloom filter and membership test with 8 bit size

Fig. 1 is the example of search in Chord-ring. Query started search(k3 & k1) at node 2. This query is sent to node 15 responsible for keyword k3. Node 15 searches all inverted lists including k3 and sends them(d0, d1, d2, d5, d8, d9, d12) to node 20 responsible for k1. Node 20 intersects inverted list sent by inverted list of k1 and node 15 and the final values, d1 and d5 are sent to node 2, then the search is completed. After all, d0, d2, d8, d9, d12 among the inverted lists sent by node 15 are useless and they just caused network traffic. If k3 is very common keyword, it will cause huge traffic. Distance and smart-bloom filter are proposed to improve such traffic in this study. Even though smart-bloom filter is similar to the bloom filter of [5], [10], [22], there is distinct difference in terms of the performance. Fig. 2 shows the example of bloom filter. In the figure the size of bloom filter(V) is 8 in size bit with 2 optimal hash functions, and the document includes keyword k1, k2, and k5. Because $hash_0$(k1)=0, and $hash_1$(k1)=7, the 0th and 7th bit of V is 1. k2 and k5 are computed likewise. Whether k1 is included in this document or not can be found through membership test. It can be known that k1 is included in the document as a result of membership test of k1. Bloom filter is a good method to sort out keyword with a high chance of inclusion in document and there have been many studies on Bloom filter [5, 22, 23, 24, 25]. However in Fig. 2 k3 is not included in the document but it is false positive with the success of the membership test since $hash_0$(k3)=0, $hash_1$(k3)=5. With the smart-bloom filter proposed in this study false positive can be decreased further and the filter plays an effective role to make up for the distance performance.

Numerous researches have been performed to reduce the network traffic. Multi-level partitioning (MLP) transmits the query only to some limited number of peers to decrease the traffic. In order to select appropriate peers, the technique divides the agent space into k logical groups [11]. [5, 12] uses Bloom Filter and previous search results to compress the intermediate URI list. Bhattacharjee proposed another technique to improve efficiency: result caching [13]. Keyword Fusion [14] uses Fusion Dictionary, which contains a set of common keywords, to achieve an efficient searching. SCALLOP [15] utilizes a balanced lookup tree to avoid the hot spot problem. In order to reduce query overhead, KSS (Keyword Set Search) by Gnawali partitions the index by a set of keywords [16]. Hybrid-indexing [17] extracts a set of important keywords from the document and utilizes it together with the inverted URI list. mSearch [18] also employs Hybrid-indexing, but it defines a multicast tree for each keyword in order to multicast the query only to some relevant agents. pSearch [19] reduces the number of visiting agents for a given query and still achieves high accuracy.

The above techniques have been successful in reducing the network traffic in some aspect, but either the reduction rate is not enough, or they require another type of system resource such as memory space. MLP introduces additional cost for communication between agents to maintain the grouping information. Using Bloom Filter can cause the problem of false positive in which the hit rate varies considerably depending on the number of hash functions and the size of the bit vector. The dictionary in Keyword Fusion takes time to build and needs to be updated frequently, which costs additional traffic. SCALLOP requires additional storage for lookup table. KSS also causes increasing storage overhead as the number of keyword combination increases. The multicast tree used in mSearch demands additional space overhead, and Hybrid-indexing requires additional space to store major keywords for each document.

3 S-BF(Smart-Bloom Filter) and Distance-Based Pruning

To resolve a multiple-keyword query in DHT-based structured P2P networks, several agents have to be visited, and for each visit an inverted list containing the URI list have to be transmitted for the JOIN operation. The problem is that this list may contain a significant number of irrelevant URI's (ones which will not show up in the final result). We focus in reducing the size of the inverted URI list by predicting the irrelevant entries in the list and removing them before sending to the next agent. We can predict the irrelevant ones with the concept of distance. We define the distance of a key, k_n, as

$$\text{distance}(k_n) = \text{hash}(k_n) - \text{hash}(k_{n-1}), \text{ for } n > 1$$
$$\text{distance}(k_n) = -1, \text{ for } n = 0$$

where k_1, k_2, ..., k_n are keys sorted in increasing order of hash values, and hash(k_n) is the hash value of key k_n. According to this definition, the distance of a key in a document shows how far this key is separated from the previous key in the sorted key list. And more importantly it shows that this document does not contain a key whose hash value falls between these two keys.

We utilize this fact to identify and remove the documents that will be dropped eventually from the inverted list. Suppose a query contains three key words, k_a, k_b, and k_c, where hash(k_a) > hash(k_b) > hash(k_c). We collect an inverted list with k_a by sending the query to the controlling agent for k_a. The list will be huge (suppose k_a is a common word), and we want to remove irrelevant documents from this list before we sending it to the next controlling agent for k_b. Irrelevant documents are ones that do not contain key k_b at all. We can identify those documents by comparing the distance(k_a) of each document with the difference between the hash values of two keys, that is hash(k_a) - hash(k_b). If we find a document for which distance(k_a) is greater than the difference between hash(k_a) and hash(k_b), it means this document does not have key k_b as explained above, and we can remove it from the inverted list.

In addition smart-bloom filter is used together with the distance to enhance the performance of the distance. It is similar to the bloom filter described in chapter 2 but the performance is greatly different. Smart-bloom filter is used to find out inverted list which cannot be removed by distance. If the assumption is made that the number of keywords in a document is n and the relation is hash(k_0) < hash(k_1) <hash(k_2)… <hash(k_n), then Smart-Bloom Filter S-BF(k_i) of random keyword k_i from k_0 to k_n is defined as follows:

S-BF(k_i)= BF(k_0, k_1, k_2, … , k_{i-2}) for n >= i >= 2,
S-BF(k_i)= S-BF(0) , for i=0 and 1 (with every bit set as 0).

i-th keyword creates BF(bloom filer) using only 0 to i-2'th keyword and without computing S-BF(k_0) and S-BF(k_1) of the two keywords with the smallest hash value every bit equals to 0. S-BF is created this way for the following reasons. Let's assume that there is search query, search(k_a & k_b) and hash(k_a) > hash(k_b). As mentioned earlier, node responsible for k_a, keyword with the largest hash value included in search query is to be firstly visited. Then entry with larger distance(k_a) than (hash(k_a) - hash(k_b)) is excluded from the search. If distance(k_a) equals to -1, according to the definition of distance it means the document does not include keyword having smaller hash value than hash(k_a)(when i=0) so it is entry unrelated to search. Therefore it is no need to create S-BF(k_0) (with every bit set as 0). In a document which includes k_a when there is only one keyword having smaller hash value than hash(k_a), (when i=1), it can be exactly known that whether this document includes k_b or not according to the sameness of distance (k_a) and (hash(k_a) - hash(k_b)) Therefore in this case also it does not need to create S-BF(k_1) either. Let's assume that there are 2 keywords with smaller hash value than hash(k_a) in a document, the keywords are k_0, k_1 and hash(k_1) > hash(k_0). If hash(k_b) > hash(k_1), it means that the relationship is distance(k_a) > (hash(k_a) - hash(k_b)) so this entry is excluded. Reversely if hash(k_b) < hash(k_1), it means distance(k_a) < (hash(k_a) - hash(k_b)) so it is determined whether this entry is something to do with the search or not according to the type of keyword k_0. Therefore it needs only to guess what kind of keyword k_0 is. Therefore S-BF(k_2) = BF(k_0). In the end when i=0, because it can be clearly known that which keyword is included in document using distance, smart-bloom filter is not needed and only in case that i is 2 or more, applying distance and smart-bloom filter altogether it can be found whether certain keyword is included or not.

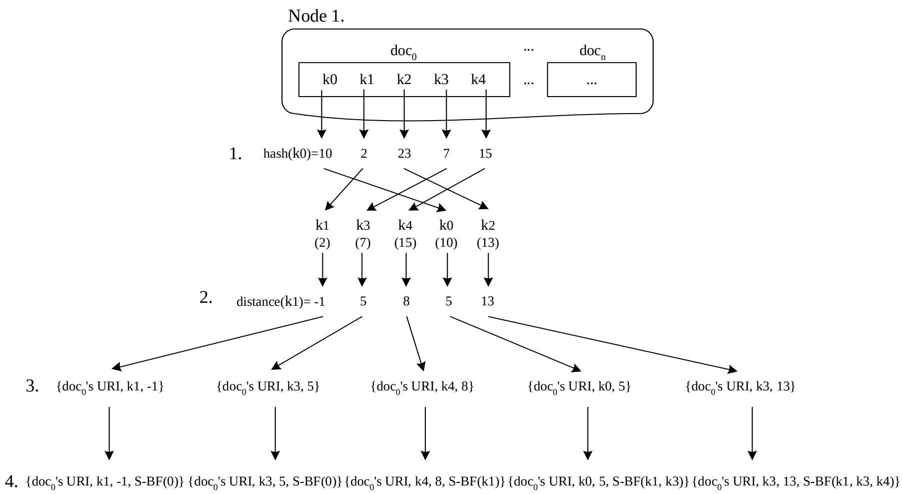


Fig. 3. 8 Four steps to add distance and smart-bloom filter to inverted list of keywords included in doc0

[5, 22, 23, 24, 25] create bloom filter using all the keywords in a document. With Bloom filters, all the entries of the index would have to be scanned, thereby incurring a high processing cost per query [26]. The difference between the proposed smart-bloom filter and bloom filter lies in that all the keywords in the document are not used. When bloom filter is m-bit and the number of keywords is n, choosing the optimal hash function the probability of false positive $P_{fp} = 0.6185^{m/n}$ [5]. However with proposed smart-bloom filter technique, $S\text{-}P_{fp} = 0.6185^{m/i}$. According to i, the $S\text{-}P_{fp}$ is equal to or far smaller than P_{fp}. Therefore false positives can be effectively reduced thereby dropping network traffic.

Fig. 3 illustrates the process to add distance and smart-bloom filter explained above to inverted list. In step 1 each keyword is hashed and in step 2 the values are arranged in descending order and then the distance of each keyword is computed. In step 3 the computed distance is added, then in step 4 smart-bloom filter is created and it is added to inverted list together with distance. Finally this created inverted list is stored to node responsible for each keyword.

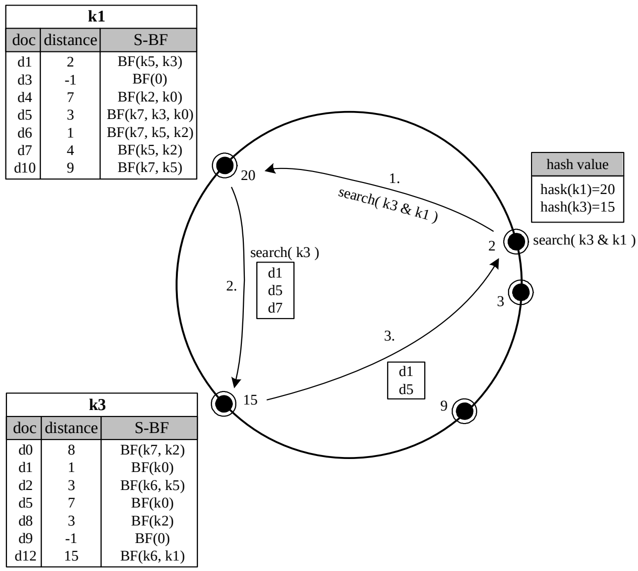


Fig. 4. Keyword search in chord-ring with smart-bloom filter and distance

Fig. 4 illustrates inverted list stored to each node. Search (k3 & k1) query is started at node 2. Because hash(k1) > hash(k3) node responsible for k1 is visited first contrary to the search in Fig. 1. Node 20 responsible for inverted list of k1 ignores entry with larger distance than 5. Since hash(k1) - hash(k3) equals to 5 entry with larger distance than 5 means document not containing k3 according to the definition of distance. Therefore there remain only {d1, 2, S-BF(k5, k3)}, {d5, 3, S-BF(k7, k3, k0)}, {d6, 1, S-BF(k7, k5, k2)}, {d7, 4, S-BF(k5, k2)}. Then out of the four remained, bloom filter of inverted list is checked. Here {d6, 1, S-BF(k7, k3, k0)} is removed so only remaining three inverted lists are sent to node 15. Sending {d1}, {d5} to node 2 after intersection is performed at node 15 brings an end to the search. In the end using the distance three unrelated lists are removed among the total 7 inverted lists. Smart-bloom filter removed one unrelated entry out of four entries searched with distance.

As shown on Fig. 4 since distance can find out, for certain, document not containing certain keyword, many inverted lists nothing to do with search result can be reduced. In addition smart-bloom filter creates bloom filter using smaller number of keyword than bloom filter so the false positive rate is lower and filters again entry with a high chance to include keyword only out of filtered inverted list. Distance function is supplemented by smart-bloom filter. Finally using distance and smart-bloom filter can reduce network traffic. Chapter 4 tests and evaluates this.

4 Experiments

Experiments were performed to measure the reduction rate in network traffic. We have modified Chord Simulator [20] for this purpose. The number of agents was

1000, and the average number of keywords per document was 12 [21]. A query can contain maximum 5 keywords. Fig. 5 shows the processing flow in Chord Simulator. The traffic generator generates a traffic file to build a Chord network, and the resulting traffic is fed into the Chord Simulator.

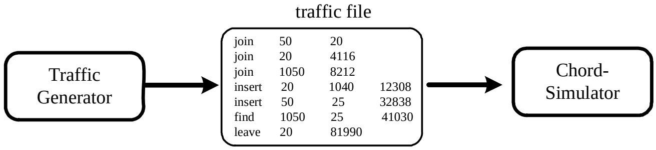


Fig. 5. Processing flow in Chord Simulator

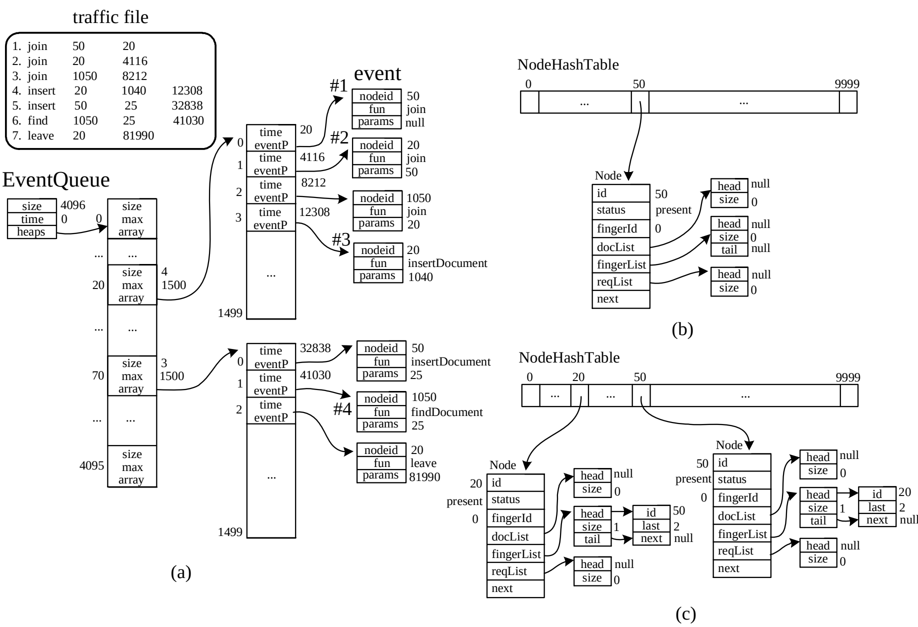


Fig. 6. The inner working of Chord Simulator for creation of events and node joining

Fig. 6 and 7 shows the inner working of the simulator. In Fig. 6, the simulator reads the traffic file and turns each line into an event. These events are inserted into EventQueue. For example, the first line in the file tells node 50 will join the Chord network in 20ms. The 4th line tells node 20 will insert key 1040 in 32838ms to the network. The 6th line tells node 1050 will query key 25 in 41030ms. After reading the traffic file, an EventQueue as shown in Fig. 6(a) will be formed. Each event is positioned to one of the buckets through "time mod 4096" operation. Now each event will be processed along the time line. Fig. 6(b) and 6(c) show the results after the first and second event were processed, respectively.

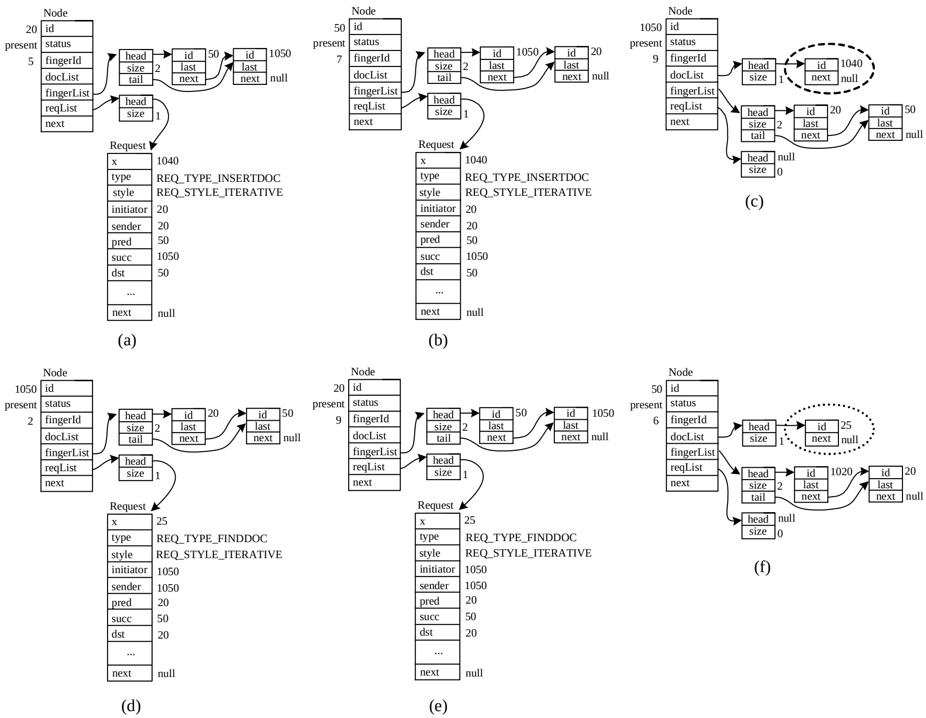


Fig. 7. The inner working of Chord Simulator for node insertion and searching

Fig. 7 shows the insertion and search of a key. The processing of the 3rd event in Fig. 6(a) will raise a request for insertion of key 1040 to the network at node 20. The figure shows the routing process to find the controlling node for key 1040. In Fig. 7(a), node 20 uses its own Finger Table to find the predecessor (which is node 50) and the successor (which is node 1050) for key 1040 and move the key to node 50, the predecessor. Node 50 performs a similar process as shown in Fig. 7(b). But this time, node 50 finds out that key 1040 should be positioned between itself and node 1050, the successor. At this time, the movement is stopped, and key 1040 is stored at node 1050. Fig. 7(d), 7(e), and 7(f) shows the search process. The search request has been raised from the 4th event in Fig. 6(a). Node 1050 has requested to search for key 25. To find out the key, node 1050 moves the request to node 20, the predecessor. Following the same process as in key insertion, node 50, the controlling node for key 25, will answer the request.

To implement our technique, we have modified the traffic generator (to generate multiple-key search traffic), Chord-Simulator (to support multiple key search), Smart-Bloom filter routine, request format, and DocumentList format.

To carry out test, the number of nodes was 1000 and the number of keywords included in query was 2 to 5 [5]. With bloom filter in m size and n keywords, m/n rate was 20.48 and optimal false positive rate $(0.6185)^{20.48}$ = 0.0053% [23, 27, 28]. Fig. 8

shows the comparison of the total numbers of inverted lists caused by getting search result according to the number of keywords in query. With distance only traffic was reduced by 41% in total and in case of 4 or 5 keywords bloom filter showed a bit more improved performance. However in use of both distance and smart-bloom filter, network traffic could be reduced by approx. 67%, and showed further more improved performance than using distance or bloom filter only. As proved through test result, distance and smart-bloom filter can effectively reduce large amount of inverted lists nothing to do with search result.

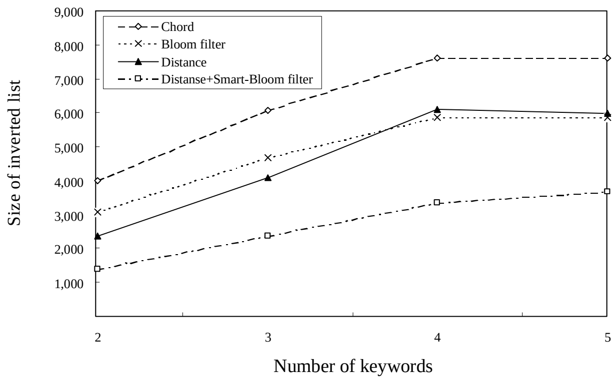


Fig. 8. The size of inverted list transmitted during query processing

5 Conclusion

DHT base structured p2p network is structured so as to perform keyword search faster than unstructured p2p. However in case of performing query containing multi-keyword it causes network traffic due to many inverted lists forwarded among nodes. Furthermore many of those inverted lists are nothing to do with search result. So this paper proposed distance and smart-bloom filter technique to reduce those many forwarded inverted lists unrelated to the final search. Distance accurately enables discerning document not containing specific keyword. Smart-bloom filter has lower false positive rate than bloom filter and filters entry with the biggest chance to include specific keyword out of inverted lists discerned by distance. Test results show that traffic was reduced by approx. 67 % in the application of both distance and smart-bloom filter. Further study will be necessary to vary m value by obtaining optimal m of bit number of smart-bloom filter according to the location of keywords included in document in the application of smart-bloom filter.

Acknowledgements

This work was supported by INHA UNIVERSITY Research Grant.

References

1. The Gnutella Protocol Specification v0.41 Document Revision 1.2., http://rfc-gnutella.sourceforge.net/developer/stable/index.html/
2. Google. http://www.google.com
3. Yahoo. http://www.yahoo.com
4. Kazza. http://www.kazza.com.
5. Patrick Reynolds and Amin Vahdat. Efficient Peer-to-Peer Keyword Searching. In Middleware (2003)
6. Stoica, I., Morris, R., Karger, D., Kaashoek, M., and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In ACM SIGCOMM (2001)
7. Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S.: A scalable content-addressable network. In SIGCOMM'01 (2001)
8. Rowstron, A. and Druschel, P.: Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems, Middleware (2001)
9. Zhao, B.Y., Kubatowicz, J., Joseph, A.: Tapestry: an infrastructure for fault-tolerant wide-area location and routing, Tech. Rep. UCB/CSD-01-1141, University of California at Berkeley, April 2001
10. Karger, D., Lehman, E., Leighton, T., Levine, M., Lewin, D., and Panigrahy, R.: Consistent Hashing and Random Trees: Tools for Relieving Hot Spots on the World Wide Web. In Proceedings STOC, pages 654.663, May 1997
11. Shuming Shi, Guangwen Yang, Dingxing Wang, Jin Yu, Shaogang Qu, and Ming Chen: Making Peer-to-Peer Keyword Searching Feasible Using Multi-Level Partitioning. In IPTPS (2004)
12. Bloom, B.: Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 13(7) (1970) 422–426
13. Bhattacharjee, B., Chawathe, S., Gopalakrishnan, V., Keleher, P., Silaghi, B.: Efficient peer-to-peer searches using result-caching, in The 2nd International Workshop on Peer-to-Peer Systems(IPTPS'03) (2003)
14. Lintao Liu, Kyung Dong Ryu, Kang-Won Lee: Keyword fusion to support efficient keyword-based search in peer-to-peer file sharing, CCGRID 2004 (2004) 269–276
15. Jerry C. Y. Chou, Tai-Yi Huang, Kuang-Li Huang, Tsung-Yen Chen. SCALLOP: A Scalable and Load-Balanced Peer-to-Peer Lookup Protocol. IEEE Trans. Parallel Distrib. Syst. 17(5) (2006) 419–433
16. Gnawali, O.: A Keyword Set Search System for Peer-to-Peer Networks, Master's thesis, Massachusetts Institute of Technology (2002)
17. Tang, C. and Dwarkadas, S.: Hybrid Global-Local Indexing for Efficient Peer-to-Peer Information Retrieval, In Proceedings of the Symposium on Networked Systems Design and Implementation (NSDI), June 2004
18. Ajay Gulati and Supranamaya Ranjan: Efficient Keyword Search using Multicast Trees in Structured p2p Networks submitted to Middleware 2005
19. Tang, C., Xu, Z., and Dwarkadas, S.: Peer-to-Peer Information Retrieval Using Self-organizing Semantic Overlay Networks. In Proceedings of SIGCOMM 2003
20. The Chord Simulator, http://pdos.csail.mit.edu/chord/sim.html, and http://cvs.pdos.csail .mit.edu/cvs/~checkout~/sfsnet/
21. Lintao Liu, Kyung Dong Ryu, Kang-Won Lee: Keyword fusion to support efficient keyword-based search in peer-to-peer file sharing, CCGRID 2004 (2004) 274

22. Kai-Hsiang Yang and Jan-Ming Ho: Proof: A Novel DHT-based Peer-to-Peer Search Engine, accepted and will appear in IEICE Transactions on Communications Special Section on Networks Software, Vol. E90-B, No. 4, February ,2007
23. Lu, T., Sinha, S., and Sudam, A.: Panache: A scalable distributed index for keyword search. Technical report, MIT (2002)
24. Ahmed, R. and Boutaba, R.: Distributed Pattern Matching: A Key to Flexible and Efficient P2P Search. To appear in IEEE Journal on Selected Areas in Communications (JSAC) issue on Peer-to-Peer Communications and Applications, 1st Quarter 2007
25. Xing Jin, Wai-Pun Ken Yiu, and Shueng-Han Gary Chan: Supporting Multiple-keyword Search in a Hybrid Structured Peer-to-Peer Network, in Proceedings of IEEE International Conference on Communications (ICC), Istanbul, Turkey, June 11-15, 2006
26. Gopalakrishnan, Vijay, System support for keyword-based search in structured Peer-to-Peer systems, Dissertation, University of Maryland, August, 2006
27. Andrei Broder and Michael Mitzenmacher: Network Applications of Bloom Filters: A Survey, Internet Mathematics Vol. 1, No. 4 (2004) 485–509
28. Mitzenmacher, Michael: Presentation: \Codes, Bloom Filters, and Overlay Networks. 2002, Cambridge, MA.14
29. Yong Yang, Rocky Dunlap, Michael Rexroad and Brian F. Cooper. Performance of Full Text Search in Structured and Unstructured Peer-to-Peer Systems. IEEE INFOCOM, Barcelona (2006)

Classification of Web Documents Using Concept Extraction from Ontologies

Marina Litvak, Mark Last, and Slava Kisilevich

Department of Information Systems Engineering, Ben-Gurion University of the Negev,Beer-Sheva 84105, Israel
{litvakm,mlast,slaks}@bgu.ac.il

Abstract. In this paper, we deal with the problem of analyzing and classifying web documents in a given domain by information filtering agents. We present the ontology-based web content mining methodology that contains such main stages as creation of ontology for the specified domain, collecting a training set of labeled documents, building a classification model in this domain using the constructed ontology and a classification algorithm, and classification of new documents by information agents via the induced model. We evaluated the proposed methodology in two specific domains: the chemical domain (web pages containing information about production of certain chemicals), and Yahoo! collection of web news documents divided into several categories. Our system receives as input the domain-specific ontology, and a set of categorized web documents, and then perfroms concept generalization on these documents. We use a key-phrase extractor with integrated ontology parser for creating a database from input documents and use it as a training set for the classification algorithm. The system classification accuracy is estimated using various levels of ontology.

1 Introduction

To meet our information needs today, we need more intelligent agent systems to gather the useful information from the huge amount of data sources available on the Web. Web Content Mining uses the ideas and principles of data mining to screen the web data. One of the problems in the web content mining area is to represent web documents as a meaningful, informative input for data mining algorithms, and then to interpret the mining results in a meaningful and useful way.

Over last couple of years, a new, promising area of research has emerged in web content mining - the usage of *domain ontology*[1]. In this paper, we introduce an ontology-based web content mining application for analyzing and classifying web documents in a given domain. The proposed classification methodology can be used by intelligent information agents for retrieving the relevant documents

[1] According to the most cited definition in the literature ([4]), ontology is an explicit specification of a domain conceptualization. It denotes and organizes entities that do exist in a domain of interest using a formal declarative language. It accumulates and organizes knowledge in a machine-processable and human-readable way.

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 287–292, 2007.
© Springer-Verlag Berlin Heidelberg 2007

from the Web. We use *domain ontology*, which organizes concepts, relations and instances into a domain [5], for the purpose of enriching the term vectors representing documents with concepts. This approach has two benefits: first, it resolves synonyms; and second, it introduces more general concepts.

One of the first prototypes of an ontology-based system for information retrieval on the web was introduced by authors of [2]. The domain ontology was characterized by a set of relevant concepts and relationships between them. The global relevance grade of a given page was computed as a combination of a syntactic grade (based on page ranking by a search engine), a semantic-syntactic grade (based on the presence of domain-related words), and a semantic grade (based on the domain-specific semantic network). In another publication ([6]) Hotho et al. use ontologies to improve text document clustering.

Contrary to these papers, we deal here with a classification task. Our system builds the decision model after training on a set of documents introduced by a domain expert. Instead of building a model based on some pre-specified words, our model uses an ontology built for a specific domain. We use some of the strategies proposed in [6] to improve web document classification.

The rest of this paper is organized in the following way. Section 2 describes the methodology and Ontology-based Phrase Extractor. In Section 3, we present the results of initial experiments. Finally, in the last section we outline the conclusions and the future work.

2 Methodology

The goal of our system is to build a compact model (profile) of the pages collected from the web so that new unlabeled pages can be reliably categorized later on. To generate a collection of training documents for a given domain, a user can initiate the system operation by submitting a set of domain-related keyword queries to a search engine (such as Google^{TM}). A domain expert reads the retrieved documents and labels them as belonging to a specific category based on their content. We induce a classification model from a training collection that includes a mix of labeled pages from multiple categories. Each page is represented as a vector of $< term_i, \ weight_i >$ pairs, received from Ontology-based Phrase Extractor module, described in the sub-section below. The phrases are extracted using a list of domain-specific terms and ontology information. The term-frequency (*tf*) $weight_i$ indicates the frequency of a $term_i$ in the observed document.

2.1 Ontology Specification

In this paper, an ontology represents the conceptual information describing the domain of interest (see Section 3) by hierarchy of domain concepts with multiple inheritance. We use such ontology for the purpose of conceptual document representation, extraction of more meaningful and relevant (even abstract) information from text of documents, and, as a result, building more accurate classification models. Of course, in case of some ontology updates, the system should be retrained.

WordNet^{TM} is one of the famous examples of ontology widely used for experimental evaluations. Although not explicitly designed as an ontology, WordNet [8] largely fits into the ontology definitions given above. The WordNet database organizes simple words and multi-word expressions of different syntactic categories into the so called synonym sets (synsets), each of which represents an underlying concept and links these through semantic relations. The current version of WordNet comprises a total of 115,424 synsets and 144,309 lexical index terms. Its hierarchical structure is not necessarily a tree structure. It may also be a directed acyclic graph possibly linking concepts to multiple superconcepts at the same time.

2.2 Ontology-Based Phrase Extractor

The Ontology-based Phrase Extractor receives as input web documents, a domain ontology and a user-specified abstraction level (k) and creates concept vectors. At the first stage, the Extractor prepares phrase collection – XML file including all general thing names (instances in ontology) as phrases with their associated classes at the k^{th} level of hierarchy (bottom-up) as related concepts. In case of abstraction level equal to 0, the collection does not include any related concepts.

Note, that fusing of instances from ontology (as phrases) during the phrase collection can be replaced or complemented by extraction of significant phrases from text of documents, that is beneficial in case of general, not domain-specific ontologies like WordNet. Therefore, when we are working with WordNet, the phrase collection contains significant phrases with their associated super-concepts of ontology. For the purpose of extracting the associated super-concepts from WordNet, we utilize the disambigation function $dis(t)$ (see [6]) that returns the semantically closest concept for the term t based on the context of document. Since this function presents some type of *semantic distance measure* (see [1]), the depth of hierarchy can be ignored.

The Phrase Extractor scans the phrases included in the collection, and every time it finds the name of a thing (in a domain-specific ontology) or a significant noun (in case of WordNet) it refers to the related concepts. **Add Concepts** ("**add**") strategy ([6]) extends each term vector t_d by new entries for Wordnet concepts c appearing in the document set, while the **Replace Terms by Concepts** strategy expels all terms from the vector representations for which at least one corresponding concept exists.

For the purpose of disambiguation we also borrow from [6] two strategies: **All Concepts** ("**all**") (used in the chemical domain) and **Disambiguation by Context** ("**context**") (used in Yahoo! collections). **All Concepts** strategy does not do anything about disambigation and considers all concepts for augmenting the text document representation, while the **Disambiguation by Context** chooses the most appropriate concept by the document context using the disambiguation function (see dis function in [6]).

Note that different methods should be used for different datasets and ontologies while trying to adapt the methodology to their specific nature. For example, in the chemical ontology, the synsets are disjoint, while the assignment of terms to concepts in WordNet is ambiguous. Therefore, when we are using WordNet, we should perform the word sense disambiguation for the purpose of prevention of extra noise in the dataset resulting from extending or replacing terms by concepts. For the same reason, extending the terms by the concepts instead of replacing them in the concept vectors is much more appropriate for an ambiguous ontology.

3 Experiments

The main goal of this research is increasing the classification accuracy of an information agent through maintaining an ontology. We tested our system on two domains: chemical domain, and Yahoo!^{TM} benchmark collection of web documents, called F-series. The experiments in the chemical domain have shown that most classification algorithms, except the Naive Bayes, are significantly improved by the synonyms handling. Conceptualization had an opposite effect on the same algorithms. The full description of these experiments as well as depiction of a domain-specific ontology, created for performing the experments are contained in [7]. For the F-series we used WordNet ontology [8].

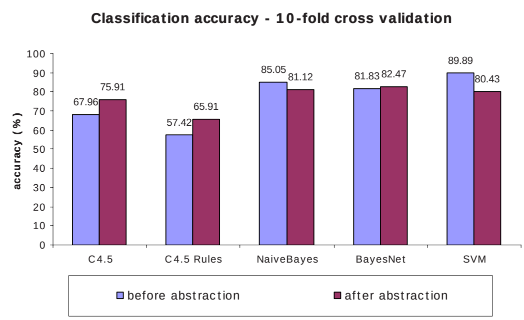


Fig. 1. Classification Accuracy with/without Abstraction – F-Series

The F-series contains 98 HTML documents belonging to one of four major category areas: manufacturing, labor, business & finance and electronic communication & networking. The results of experiments are presented in Fig. 1. The results of the *t-test* are presented in Table 1. As we can see, accuracy of decision trees models was improved after conceptualization, while NaiveBayes and SVM, conversely, got worse results than on usual term vectors.

We can explain such conflicting behaviour of these algorithms by their characteristics. During the abstraction (concept vectors building) we extend the term vectors by concepts. The relevance of these concepts is dependent on the disambiguation method. "Bad" methods insert a lot of noise to the data. Decision trees handle this problem by using the feature selection procedure that filters out the irrelevant features while building the model. NaiveBayes and SVM, contrarily,

Table 1. Results of t-test – F-Series

C4.5	C4.5 Rules	Naive Bayes	Bayes Network	SVM
↑ 0.0009*	↑ 0.0012*	↓ 0.0232*	↑ 0.6012	↓ 0.0001*

consider all features, that in case of noise, distort the results. Based on some current publications (see [9]) and our experience we can conclude that using the WordNet is not sufficient to reliably disambiguate word senses in text.

In our opinion, one of the promising approaches to solution of this problem is utilization of *domain-specific* ontology. Today, there are hundreds of ontologies available from the Internet, that usually cover very specific domain areas. A researcher can find something suitable for the processed domain, update it, merge several ontologies or build a new one based on domain knowledge as we did for the chemical domain. We need the knowledge of a qualified domain expert to express the domain very accurately via relations among instances and classes. A dictionary of key phrases for different categories of a given domain has proved to be useful too.

4 Conclusions and Future Work

In this paper we presented a new ontology-based methodology for automated classification of web documents by an information agent. The main contribution of this work is using domain-based ontology in the conceptual representation of documents. In contrast to the results on the chemicals domain reported in [7], where the synonyms and the taxonomic relationships were handled, the results received on the Yahoo! collections did not demonstrate such significant improvements. We explain such results by an insufficiency of the WordNet ontology to disambiguate word senses and generalize text representation properly as well as the experimental data specifications. According to different experimental results we can expect that the textual data with extensive number of synonyms (like documents describing the chemicals or food, etc.) will produce the best accuracy.

We intend to enhance our methodology via utilizing some tools (like GATE [3]) for automatic construction of ontologies. Such update will enable us to make our system completely domain-independent. We can also extend an existing ontology by several ontologies for the same set of documents. Also, we would like to use hyperlinks as well as the text from the linked pages to improve the classification accuracy.

References

1. Budanitsky, A., & Hirst, G.: Semantic Distance in WordNet: An Experimental, Application-oriented Evaluation of Five Measures. Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2000), Pittsburgh, PA

2. Cesarano, C., d'Acerno, A., & Picariello, A.: An Intelligent Search Agent System for Semantic Information Retrieval on the Internet. Proc. of the Fifth ACM International Workshop on the Web Information and Data Management, New Orleans, Louisiana, USA (2003) 111–117
3. GATE - General Architecture for Text Engineering, The Natural Language Processing Research Group, Department of Computer Science, University of Sheffield [http://gate.ac.uk/]
4. Gruber, T. R.: A translation approach to portable ontologies. Knowledge Acquisition, 5 (2) (1993) 199–220
5. van Heijst, G., Schreiber, A.Th., & Wielinga, B.J.: Using explicit ontologies in KBs development. IJHCS (1997) 183–291
6. Hotho, A., Staab, S., & Stumme, G.: Ontologies Improve Text Document Clustering. In Proc. of ICDM-03 (2003)
7. Litvak, M., Last, M., & Kisilevich, S.: Improving Classification of Multi-Lingual Web Documents using Domain Ontologies. ECML/PKDD-2005, October, 2005, Porto, Portugal
8. Miller, G.A., Beckwith, R.T., Fellbaum, C.D., Gross, D., & Miller, K.: Wordnet: An Online Lexical Database. International Journal of Lexicography, 3 (4) (1990) 235–244
9. Voorhees, E.: Using WordNet^{TM} to disambiguate word senses for text retrieval. Proc. of the 16th annual international ACM SIGIR conference, Pittsburgh, PA, (1993)
10. Witten, I.H., Frank, E., Trigg, L., Hall, M., Holmes, G., Cunningham, S.J.: Weka: Practical machine learning tools and techniques with java implementations. In Proc. of ICONIP/ANZIIS/ANNES'99 Int. Workshop on Emerging Knowledge Engineering and Connectionist-Based Info. Systems (1999) 192–196
11. Yao, Y.Y., Zhong, N., Liu, J., & Ohsuga, S.: Web Intelligence (WI): research challenges and trends in the new information age. In Zhong et al., eds., Web Intelligence: research and development, LNAI 2198, Springer-Verlag (2001) 1–17
12. Zhong, N., Liu, J., Yao, Y. Y.: In search of the wisdom Web. IEEE Computer, 35 (11) (2002) 27–31

Emotional Cognitive Agents with Adaptive Ontologies

Leonid I. Perlovsky

Visiting Scholar at Harvard University and
Technical Advisor at the Air Force Research Laboratory,
Sensors Directorate/SNHE, 80 Scott Drive, Hanscom AFB, MA 01731
leonid@harvard.deas.edu

Abstract. An unsolved problem in AIS is adaptive ontologies. Semantic Web requires flexible ontologies adaptive to user needs and to Web contents. The paper describes emotional intelligent agents with adaptive ontologies. Their emotions are essential part of intelligence and inseparable from cognition. Particular emotions important for adaptation are aesthetic emotions related to the knowledge instinct. We describe mathematical mechanisms involved, and analyze the role of emotions in cognition, language, and adaptive ontologies.

Keywords: Agents, Emotion, Cognition, Differentiation, Synthesis, Evolution, Language, Culture, Adaptive Ontologies, Dynamic Logic.

1 Emotions, Cognition, and Adaptation

Emotions are sometimes regarded as the antithesis of reason; as is suggested by phrases such as “appeal to emotion” or “don't let your emotions take over.” A distinctive and challenging fact about human beings is a potential for both opposition and entanglement between will, emotion, and reason. Emotions refer to both exaggeratedly expressive communications and to internal states related to feelings. Love, hate, courage, fear, joy, sadness, pleasure, and disgust can all be described in both psychological and physiological terms. Emotion is the realm where thought and physiology are inextricably entwined, and where the self is inseparable from individual perceptions of value and judgment toward others and ourselves. This paper studies emotions as an inseparable part of intelligence and cognition and present arguments that adaptive ontologies requires emotional agents.

Emotions as essential mechanism of the mind were analyzed in philosophy [1,2,3], psychology [4], neuro-psychology [5,6,7], linguistics [8], neuro-physiology [9], and from the learning and cognition perspective by the author [10,11,12]. Jung analyzed conscious and unconscious aspects of emotions. He emphasized undifferentiated status of primitive fused emotion-concept-behavior psychic states in everyday functioning and their role in neuroses. He also emphasized rational aspect of conscious differentiated emotions. According to Damasio, emotions are primarily bodily perceptions, and feelings of emotions in the brain invoke “bodily markers.” Grossberg and Levine consider emotions as neural signals that relate instinctual and conceptual brain centers. In processes of perception and cognition, emotions evaluate concept-models of objects and situations for satisfaction or dissatisfaction of instinctual needs.

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 293–304, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Some aspects of the mathematical theory developed in the following sections closely follow ideas of Kant, Jung, Grossberg and Levine. The main mechanisms of this theory include the following. Concepts are mathematically equivalent to ontologies-models of the objects and situations in the world. They evolved for satisfaction of the basic instincts, which have emerged as survival mechanisms long before concepts. Mathematically instincts are described as internal sensors indicating the basic needs: for example, when a sugar level in blood goes below a certain level an instinct "tells us" to eat. Below we mathematically describe specific instinct driving knowledge and adaptation. Instincts are connected to cognition and behavior by emotional neural signals. Whereas in colloquial usage, emotions are often understood as facial expressions, higher voice pitch, exaggerated gesticulation, these are outward signs of emotions, serving for communication. A more fundamental role of emotions is that emotional signals evaluate concepts for the purpose of instinct satisfaction.

Cognition mechanisms are partly conscious, partly unconscious. They often proceed independently from any conscious desire "to think." They are "moved" by the knowledge instinct, an inborn mechanism responsible for cognition and adaptation [13,14]. Biologists and psychologists discussed various aspects of this drive [15,16,17]. Until recently, however, it was not mentioned among 'basic instincts' on a par with instincts for food and procreation. The fundamental nature of this mechanism is related to the fact that our knowledge always has to be modified to fit the current situations. Mathematical modeling of the mind makes obvious the fundamental nature of our desire for knowledge. Virtually all learning and adaptive algorithms (thousands of publications) maximize correspondence between the algorithm internal structure (knowledge in a wide sense) and objects of recognition. Knowledge is not just a static state; it is in a constant process of adaptation and learning. Therefore, we have an inborn need, a drive, an instinct to improve our knowledge. I call it *the knowledge instinct*. Mathematically it is described as a maximization of a similarity measure between concept-models and the world.

Satisfaction or dissatisfaction of the knowledge instinct is not directly related to bodily needs. Therefore, emotions evaluating satisfaction of the knowledge instinct are 'spiritual' or aesthetic. Of fourteen contemporary psychologists [6] only three authors mentioned emotions that I consider aesthetic. Despite of the fact that most of us experience an infinite manifold of emotions every time we listen to songs and music, psychologists have not recognized this and have no words for this sea of emotions. Already Spinoza mentioned that every emotion is different, depending on object it is associated with [18]. But contemporary psychologists do not recognize this basic fact of our psychology. And engineers did not have basic knowledge to develop emotional agents. Relationship between cognition and emotion, leads to the diversity of aesthetic emotions are necessary for adaptive ontologies.

2 Neural Modeling Fields (NMF)

This paper continues mathematical development of the theory of neural modeling fields (NMF) [19], briefly summarized below. The mind involves a hierarchy of multiple levels of concept-model-ontologies, from simple perceptual elements (like edges, or moving dots), to models-ontologies representing objects, or complex scenes, and up the hierarchy… toward the concept-models of the unified meaning of the knowledge for instinctual needs. Ontologies represent not only objects in the world, but also language information about these objects. At every level of the hierarchy NMF associates lower-level signals with higher-level models-ontologies; a result is an understanding of signals as concepts. NMF is a multi-level, hetero-hierarchical system. Bottom-up signals $\{\mathbf{X}(n)\}$ is a neuronal field of input synapse activations. In the process of cognition they are matched to top-down signals generated by models-ontologies $\{\mathbf{M}_h(n)\}$. Computationally, it maximizes a similarity measure between the sets of models and incoming signals, $L(\{\mathbf{X}(n)\},\{\mathbf{M}_h(n)\})$, over the model parameters, $\{\mathbf{S}_h\}$,

$$L(\{\mathbf{X}\},\{\mathbf{M}\}) = \prod_{n\in N} \sum_{h\in H} r(h)\, l(\mathbf{X}(n) \mid \mathbf{M}_h(n)); \quad (1)$$

here, $l(\mathbf{X}(n)|\mathbf{M}_h(n))$ (or simply $l(n|h)$) is a conditional partial similarity between one signal $\mathbf{X}(n)$ and one model $\mathbf{M}_h(n)$; (1) accounts for all possible combinations of signals and models. Parameters $r(h)$, the "weights" of concepts h, are proportional to the number of signals $\{n\}$ associated with the model h. The similarity maximization is a mathematical description of the knowledge instinct. Changes in similarity values are aesthetic emotional signals.

Understanding of the world, or cognition consists in associating signals with concepts-ontologies and estimating model parameters $\mathbf{S}_h$ by maximizing similarity (1). Note, that (1) contains a large number of combinations of models and signals, a total of H^N items; this was a cause for the combinatorial complexity of the algorithms and neural network training procedures in the past. NMF solves this problem using the mechanism of fuzzy dynamic logic (DL) [20,19]. A fundamental aspect of DL is that the initial state of model similarities, $l(n|h)$, is fuzzy and corresponds to uncertainty in the knowledge of model parameters. In the course of learning, knowledge improves, and similarities converge to low-fuzzy, probabilistic, or crisp functions. When new data appear, which do not correspond well to existing models, similarity (1) becomes low and the esthetic emotion is negative. During learning, knowledge improves, similarity (1) increases, and the aesthetic emotion is positive. Emotional signals constitute essential aspect of NMF and DL. They overcome difficulty of combinatorial complexity, which prevented in the past developing adaptive ontologies.

Language in NMF is modeled similarly to cognition [21]. During language learning, input signals are sounds of language, and language models are models of words as composed of phonemes, or models of phrases as composed of words, and similarly up the hierarchy of the mind to models of paragraphs composed of phrases... to the models of text, like "Romeo and Juliet", or "Anna Karenina."

Language and cognition are integrated using a mechanism of dual models with cognitive and language parts; so that

$$\mathbf{M}_h(n) = \{\mathbf{MC}_h(n), \mathbf{ML}_h(n)\}. \quad (2)$$

Model **MC** stands for cognitive and **ML** for language models. It is not necessary to consider combinations of MC and ML, because initially all models are same, fuzzy blobs, just placeholders for future knowledge.

NMF system learns similarly to human, in parallel in three realms: (1) language and cognitive models are learned jointly, when language data are present in association with perception signals, like during mother talking to a baby: “this is a car” (perception-models and word-models), (2) language models are learned independently from cognition, when language data are encountered for the first time with no association with perception and cognition (most of language learning during the age 2 to 7); (3) similarly, cognitive models are learned independently from language, when perception signal data are encountered for the first time without association with linguistic data. Cognitive and language learning always depend on each other to some extent. The original, inborn models are fuzzy structures equally and poorly matching any sensory or language data. In the process of learning fuzziness decreases, crisp models get associated with specific situations and phrases, and cognitive models always remain associated with language models. Because the integrated (cognitive, language)-model structures are inborn, association between language and cognition begins at a “pre-conceptual” fuzzy level, inaccessible to consciousness. Child learns a large number of language models; their association with real life is vague; later in life they facilitate learning of corresponding cognitive models; similarly, cognitive models facilitate learning of language models; eventually cognitive and language models are learned to be associated. Similarly, NMF/DL agents adaptively develop ontologies representing contents of the surrounding world as well as its usage and language description.

3 Hierarchy of the Mind

Dual integrated models (2) combined with hierarchical NMF organization lead to integrated hierarchies as illustrated in Fig. 1. An amazing aspect of the human mind is that these two hierarchies are integrated in such a way that relationships among constituent models are preserved. This seems so obvious, that complexity of the required mechanism was noticed only recently by T. Deacon [22].

Consider an example. A dog can learn to bring shoes on command. The dog can associate shoes with a word “shoes.” Does it mean dog’s mind possesses models (2)? Try to teach a meaning of a word “rational” to a dog. Apparently, a dog can associate sounds with objects, which it sees in the world. A dog treats sounds just like other objects. But it does not possess a hierarchy of integrated models. In dog’s mind, cognitive models are “grounded” in objects and situations in the world. But abstract concepts require grounding in other concepts, a hierarchy of concepts is required. According to [22], smartest apes after years of training, could possibly learn 2 levels of a hierarchy. Why is it so difficult? Higher levels of a hierarchy in the ape mind have no “ground.” In the human mind, higher level language models are grounded in conversations with other people: Mutual understanding “assures” our mind of the reality of language hierarchy. A cognitive hierarchy is supported by a language hierarchy. Possibly, an essential inborn difference between human and animal minds is that we possess structures modeled by eq. (2) and Fig. 1. This might be sufficient for evolution of symbolic culture. Using mathematical description of human mind we will make a step toward adaptive human-computer systems, in which agents will learn from human and interact human similarly to human-human interactions.

Cultural evolution is much faster than genetic one. This is because concept-models of the mind evolve much faster than genes. Dawkins [23] called concepts "memes" and emphasized that model selection will overtake gene selection because models are more efficient replicators. A mathematical description of this process is a subject of this paper. Cognitive models that proved useful in life and evolution cannot be directly transferred to the minds of the next generation. Only language models are transferred to the next generation. This separation between cognitive models and language models can be compared to separation between phenotypes and genotypes. In some ways this comparison could be deep and inspiring, in other ways, it is superficial and wrong. According to the current knowledge of genetics, acquired properties of phenotypes are not incorporated into genetic information and are not transfer directly to the next generation (as proposed by Lamarck). Nobody knows why this is so, why genetic mechanisms avoid using a potentially more efficient Lamarckian accumulation of experience. One hypothesis is that it would result in a too fast adaptation to a local environment, so that if environment changes, Lamarckian species would not survive. I will attempt to identify mechanisms speeding and decelerating cultural evolution, and identify potential benefits and dangers of these mechanisms.

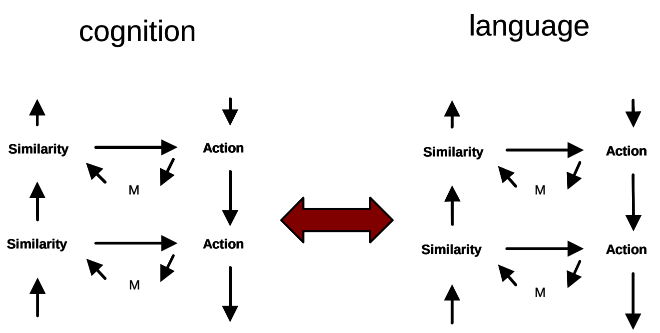


Fig. 1. Hierarchical integrated language-cognition MF system. At each level in a hierarchy there are integrated language and cognition models. Similarities are integrated as products of language and cognition similarities. Initial models are fuzzy placeholders, so integration of language and cognition is sub-conscious. Associations depend on both language and cognitive models and signals. Therefore language model learning helps cognitive model learning and v.v. Abstract cognitive concepts are grounded in abstract language concepts.

Conceptual information, cognitive models created by each generation are accumulated in culture due to language. Cultural evolution selects useful models. Language accumulates cultural knowledge at all levels in a hierarchy of the mind. Due to integration of language and cognition, language provides a foundation for developing abstract high-level cognitive models in every human being. But, this requires that individual minds in each generation connect language and cognitive models. Every generation has to learn differentiated conscious cognitive models corresponding to the level of differentiation accumulated in language and culture.

4 Differentiation and Synthesis

An important unsolved problem in developing adaptive hierarchical systems is that of *grounding* high-level hierarchical ontologies, so that results of adaptation are adequate to the surrounding world and relevant to users. Aesthetic emotions play crucial role in solving this problem, in computer agents as well as in human cognition (as discussed in section 2). To develop agents with hierarchical adaptive ontologies, we need to analyze the corresponding mechanisms in the human mind. Our minds are capable of separating other emotions related to bodily instincts from thinking processes. We can discuss dangerous situations without fear, we can discuss food without being hungry. This ability for differentiating concepts and emotions is closely related to our ability for deliberate thinking. It is uniquely human. Animals cannot separate conceptual thinking about food from the emotion of hunger, from the instinctual need to eat.

Similarly, animal communications cannot be separated from emotions. Animals cannot deliberately control their vocal tract. Even our closest relatives, chimpanzees, cannot separate vocalizations and emotions [24]. Voice tract in animals is governed from an ancient emotional center in the limbic system. Conceptual and emotional systems in animals are less differentiated than in humans. Sounds of animal cries engage the entire psyche, rather than concepts and emotions separately. An ape or bird seeing danger does not think about what to say to its fellows. A cry of danger is *inseparably* fused with recognition of a dangerous situation, and with a command to oneself and to the entire flock: “Fly!” [25]. An evaluation (emotion of fear), understanding of situation (concept of danger), and behavior (vocalization and wing sweep) – are not differentiated. Conscious and unconscious are not separated: Recognizing danger, crying, and flying away is a unified situational-behavioral fused form of thought-action. Animals can not control their larynx muscles *voluntarily*.

Two emotional centers govern human vocal tract, ancient less-conscious and less-voluntary, in limbic system, and recent cortical emotional center, more conscious and voluntary. Language evolved toward differentiation of psyche. Language differentiates concepts, as well as concepts from emotions [12]. Differentiation between emotions and concepts, as mentioned, is a foundation for our thinking ability. This differentiation, however, is not entirely “good” for cognition. The *meaning* of concepts is not limited to relationships among words and phrases in language, but requires connections between language and cognitive models, eq. (2), connections among cognitive models within the cognitive hierarchy, Fig. 1, and connections of cognitive models to instincts, including the instinct for knowledge, eq. (1). Animal vocalizations are not as differentiated as human, but directly wired to their instinctual needs. Animal cries, therefore, are directly meaningful to animals. Not so for humans. Human languages give us an ability to talk conceptually about tens of thousands or millions of various topics, but conversations and texts are not automatically related to our instinctual needs, *meanings* of language expressions requires instinctual grounding, which mechanism includes emotional signals. Jung called this mechanism *synthesis* [26,11]. Although our ability for differentiated conceptual thinking is enabled by differentiation between concepts and emotions, the other side of this differentiation is that the meaning of even highly differentiated conceptual thoughts might disappear. Dispassionate arguments are important for scientific discourse, but dispassionate thinking may lead to an entire culture loosing its meanings. This might have been the reason for the death of many old civilizations [12,27].

Evolution of cultures requires connecting language and cognitive concepts. Words passed from generation to generation have to be connected to their meanings in, culture. This requires emotional connections to instinctual needs. Let us look into the nature of these emotions preserving synthesis or unity of psyche in the face of differentiated knowledge. The knowledge instinct, as formulated in section 2, results in a single aesthetic emotion, related to a single measure of similarity between all models and all sensor data (all experience). This is certainly a great simplification. When trying to understand the complexity of surrounding world we do not maximize a single measure uniformly over all conceptual knowledge and all experiences. This would not do justice to the diversity of our experiences. Certain concepts are more important for us than others. This is true even within purely scientific domain; for example, the law of energy conservation will not be casually questioned based on some unproven measurements; a first reaction would be to question measurements. Certain moral, political, or religious concepts accumulate cultural experience of many generations and become more important than individual percepts of the reality. Strong emotional feelings usually intervene, when high value concepts are questioned. Autonomous intelligent agents developed using computers should implement similar mechanisms. They should be able to learn emotions evaluating relative importance of various ontologies and use these emotions for meaningful adaptations.

Our survival directly involves emotions; this is relatively simple to understand due to direct involvement of bodily instincts. This understanding of emotions was discussed in section 1: Emotions evaluate concepts with respect to satisfaction or dissatisfaction of instincts. But when emotions involve correspondence between different abstract concepts, a new explanation is required. Let us look from this vantage point at the hierarchical structure in Fig. 1. Every level in the hierarchy involves a similarity measure, the knowledge instinct operating at this level. Only at the lowest level, the knowledge instinct involves sensor signals and models. At each higher level, a similarity is between bottom-up sets of models and top-down sets of models. What is the source of emotions involved with high value models? We have to understand this to develop agents with adaptive high-level ontologies.

A single aesthetic emotion related to correspondence between top-down and bottom-up signals is not rich enough to account for the discussed emotions and their role in cognition. Each valuable concept-ontology emotionally affects our recognition and understanding of many other concepts. In other words, adaptive concept-ontologies act similarly to instincts: They emotionally evaluate other concepts. Therefore, the cognitive hierarchy in the left of Fig.1 involves multiple sub-hierarchies associated with valuable concepts. Each sub-hierarchy produces its own emotional signals, which measure correspondence of the sub-hierarchy to the valuable concept that spawns it. Since the designation “valuable concept” is a matter of degree, all concepts are interrelated by a web of mutual emotional correspondence-evaluations. A similar view on emotions was formulated by Spinoza [18]; he emphasized that every emotion is different depending on the involved object.

These diverse emotions unify the total knowledge within mutual interrelationships. The number of aesthetic emotions is combinatorial in terms of the number of concepts, resulting in practically uncountable manifold of emotions. As discussed later, we hear these emotions in music, which makes us aware of many of them. Most of these emotions are below the threshold of awareness, unconscious. The more these emotions become conscious the more a person is aware of the diversity of knowledge, while preserving synthesis or the unity of the psyche. Similar mechanisms should be used in AIS to ensure system stability and purposeful actions, while enabling diversity of knowledge.

Therefore, an essential aspect of meaning in natural and engineered systems is synthesis. If psyche is torn apart by diversity of knowledge, if an agent lose its bearing because of emerging contradictions among autonomous ontologies, the meaning disappears and purposefulness of actions is threatened. One aspect of creating meanings is the discussed synthesis of language and cognition. Another aspect is synthesis of knowledge at the top levels of the hierarchy. General and abstract models near the top of the hierarchy encompass all the diversity of knowledge at the intermediate and lower levels. The gain is synthesis, the "price" is that general models are less specific, vague-fuzzy, and unconscious. Their conscious aspects are developed in cultural evolution, their unconscious aspects are fused with the instinctual bases of the psyche, archetypes, fuzzy models which conceptual and emotional contents are undifferentiated. In engineered systems, adaptive ontologies near the top of the hierarchy cannot be crystal clear, they have to be fuzzy. The more specific are top ontologies, the less adaptive and more brittle is the system. This is a lesson hard learned over the last decades. Its solution offered here by analogy with human psychology is a new direction in engineering.

We lack psychological or neural data at this time to determine uniquely the mathematical structure of sub-hierarchies. They could be similar to eq. (1) with every sub-hierarchy endowed with its own measures of similarity, or they all might share the same structure (Fig. 1), with concept values being represented by weight-parameters, r(h), in eq. (1). These parameters acquire the meaning of emotional signals; and their values are not determined simply by empirical evidence (the number of signals corresponding to concept h in the experience of a single agent or person), but are also influenced by culturally accumulated experience preserved in sound of language [21]. In engineered systems they have to be learned from humans.

5 Language, Cultural Evolution, and Evolving Ontologies

Concept-models at higher level of the mind hierarchy, abstract cognitive models are grounded in language. The language provides conceptual structures for the models, but also their emotional connections to emotional centers in the brain. Evolutionary recent emotional centers in the cortex are responsible for differentiation of emotional and conceptual contents in languages. Evolutionary old emotional centers in the limbic system are responsible for synthesis. As we discussed, mechanisms connecting language to old emotional centers are closely related to the sound of language. Since Saussure [28] many linguists subscribe to the view that sounds of languages are arbitrary notations for meanings. Different languages use different sounds for the words with similar meanings. Nevertheless, recent results in language evolution suggest that meaning-sound pairing are not arbitrary [29], especially, when we consider evolution of the entire culture [30].

Languages change with time. The same meanings are expressed by different sound. English today sounds differently from English of Chaucer. Before Great Vowel Shift in the 15th and 16th century English sounded similar to continental German languages and words were pronounced as spelled. Today English is a partly hieroglyphic language, sound departed from writing. This change in sound followed changes in English grammar. In Old English nouns had complex declensions, with affixes for different numbers and 4 cases (nominative, accusative, genitive, dative), personal pronouns also changed by gender and had a dual number, word order was not fixed, verbs were conjugated. Words were pronounced as spelled. Middle English, nouns had 3 cases (nom., gen., dat.), personal pronouns kept 4 cases; verbs were conjugated by person and number. Words were still pronounced as spelled, and vowel pronunciation was similar to other Germanic languages.

Evolutionary stability of word sounds depends on grammar, on inflectional affixes. Sounds of affixes is “a tail that wags the dog.” Pronunciation of inflectional structures is fixed by grammar, which usually remain stable over many generations; stable sounds of the word endings, to an extent, stabilize the sound of the entire word. This property of inflectional languages is responsible for what Humboldt called “inner firmness of words” [31]. In highly inflectional languages (like German or Russian) sounds and therefore emotions are closely related to meanings. In modern English, sounds and therefore emotions are dissociated from meanings. It would be an interesting topic for psycholinguistic research to compare emotionality of speech of native speakers in various Indo-European languages. We predict positive correlation between degree of language inflection on the one hand, and on the other, connection between emotions and cognitive content.

English grammar is simple, its advantage is an ease of forming new conceptual structures. Virtually any two words can be combined to form a new meaning. This is not so in highly inflective languages, like Russian. To form a new meaning, a combination of words should “sound right”; that is, an emotional content should correspond to a conceptual content. Because of that, forming new meanings in Russian is much more difficult than in English. English is a pragmatically powerful language. Advancement of pragmatic capitalistic culture and democracy, which pushed aside many old cultural emotional encumbrances, occurred soon after English changed from Middle to Modern.

This powerful differentiating capacity of English has other side, synthesis lags behind differentiation. This might be the cause of some crisis-like phenomena in English-speaking countries. Weakened emotional connections between language and meaning may lead to a loss of synthesis, weakened self identity. Popular songs, by connecting words with emotional sounds, restore synthesis in Western psyche [32]. The reason for explosion in consumption of popular songs is a need for synthesis in our differentiated culture.

This connection between language sounds and emotionality on the one hand, and differentiation vs. synthesis on the other, was studied for Indo-European (IE) languages [33]. It might seem amazing that these connections also explain the well-known differences between English-speaking and Arab-speaking cultures. Arab language is an inflectional language, but the structure of inflections is different from IE languages. Whereas in IE, word affixes are changed according to situations, in Arabic, the entire word sound changes. Sounds are therefore much more closely fused with meanings than in IE. It follows that the emotionality of Arabic is much stronger than in IE. Culture is less pragmatic, development of new meanings is difficult, synthesis and a feel of self-identity is strong. Of course, these general characterization of cultures do not necessarily apply to individuals. Less educated and less conscious part of population is more affected by general properties of languages. An individual person striving for conscious cognition of the world can use advantages offered by his or her language and overcome language limitations.

This novel understanding of different emotionality among languages and cultures is crucial for the success of engineering collaborative AIS-ADM systems for multi-national collaborations. Recognizing the fundamental importance of this new dimension of engineering for the future of the world AFOSR conducts new programs addressing this topic.

Paths of cultural evolutions are governed by differentiation and synthesis [27], and these connections can be verified in social and psychological studies. Theoretical connections between sound of language (prosody) and emotions discussed in this paper are studied in psycholinguistic laboratories [34,35]. These studies are being extended to multiple languages to test the proposed effects of grammar. Current research in evolution of languages and cultures uses mathematical simulations of communities of interacting agents and related statistical models [36,37,38]. Joint evolution of cognition and language was initiated in [39,40,41]. This research is essential for the success of future Sematic World Wide Web. The proposed mechanisms of integrating language and cognition, their affects on differentiation and synthesis, their interactions with emotionality of language and their effects in cultural evolution will be incorporated into future studies of evolution of languages and cultures.

6 Conclusion

Future Semantic Web will need adaptive ontologies capable of learning new types of Web contents and new user requirements. Functionality and mechanisms of these adaptive ontologies will resemble human language. Agents utilizing these ontologies will acquire elements of human intelligence. Mathematical description of the knowledge instinct that drives human intelligence and is suitable for intelligent agents is presented in the paper. It includes mechanisms of cognition, emotions, language, and their interactions.

Adaptation and evolution of these abilities involve two aspects of the knowledge instinct. Differentiation that creates detailed ontologies at every hierarchical level, and synthesis that works among multiple hierarchical levels and emotionally relates ontologies to system-level goals (instincts). Our analysis emphasized emotional differences among languages and cultures. Agents utilizing these emotional adaptive capabilities will be important for building future systems for multi-national collaborations.

References

1. Descartes, R.: The Passions of the Soul: Les Passions De Lame, 1646. Hackett Publishing Company, Indianapolis IN (1989)
2. Kant, I.: Critique of Judgment, 1790. Tr. J.H. Bernard, Macmillan & Co., London (1914)

3. Sartre, J.P.: Emotions, 1948. See J. P. Sartre, Existentialism and Human Emotions, Citadel Press; Reissue edition (1984)
4. Jung, C.G.: Psychological Types, 1921. In: the Collected Works, v.6, Bollingen Series XX. Princeton University Press, Princeton, NJ (1971)
5. Grossberg, S. & Levine, D.S.: Neural dynamics of attentionally modulated Pavlovian conditioning: blocking, inter-stimulus interval, and secondary reinforcement. Psychobiology, 15(3) (1987) 195–240
6. Ortony, A., & Turner, T.J.: What's Basic about Basic Emotions? Psychological Review, 97 (1990) 315–331
7. Ledoux, J.: The Emotional Brain: The Mysterious Underpinnings of Emotional Life. Simon & Schuster, New York, NY (1998)
8. Griffiths, P. E.: What Emotions Really Are: The Problem of Psychological Categories. University Of Chicago Press, Chicago, IL (1998)
9. Damasio, A.R.: Descartes' Error: Emotion, Reason, and the Human Brain. Avon, NY, NY (1995)
10. Perlovsky, L.I.: Emotions, Learning, and Control. In: Proc. International Symp. Intelligent Control, Intelligent Systems & Semiotics, Cambridge MA (1999) 131–137
11. Perlovsky, L.I.: Modeling Field Theory of Higher Cognitive Functions. Chapter in: Artificial Cognition Systems, Loula, A., Gudwin, R., Queiroz., J. (eds.): Idea Group, Hershey, PA (2006) 64–105
12. Perlovsky, L.I.: The Knowledge Instinct. Basic Books. New York, NY
13. Perlovsky, L.I.:& McManus, M.M. Maximum Likelihood Neural Networks for Sensor Fusion and Adaptive Classification. Neural Networks, 4(1) (1991) 89–102
14. Perlovsky, L.I.: Toward Physics of the Mind: Concepts, Emotions, Consciousness, and Symbols. Phys. Life Rev. 3(1) (2006) 22–55
15. Harlow, H.F., & Mears, C.: The Human Model: Primate Perspectives. V. H. Winston and Sons, Washington, DC (1979)
16. Berlyne, D. E.: Conflict, Arousal, And Curiosity, McGraw-Hill, New York, NY (1973); Berlyne, D. E. Pleasure, Reward, Preference: Their Nature, Determinants, and Role in Behavior. Academic Press, New York, NY (1960)
17. Festinger, L.: A Theory of Cognitive Dissonance. Stanford University Press, Stanford, CA (1957)
18. Spinoza, B: Ethics, 1677. Oxford Univ. Press, New York, NY (2000)
19. Perlovsky, L.I.: Neural Networks and Intellect. Oxford University Press, New York, NY (2001)
20. Perlovsky, L.I.: Physical Concepts of Intellect. Proc. Russian Academy of Sciences, 354(3) (1997) 320–323
21. Perlovsky, L.I.: Symbols: Integrated Cognition and Language. Chapter in: Gudwin, R., Queiroz, J. (eds.): Semiotics and Intelligent Systems Development. Idea Group, Hershey, PA (2006) 121–151
22. Deacon, T. W.: The Symbolic Species: The Co-Evolution of Language and the Brain. Norton, New York, NY (1998)
23. Dawkins, R.: The Selfish Gene, Oxford Univ. Press, Oxford, GB (1976)
24. Lieberman, P.: Human language and our reptilian brain. Harvard Univ. Press, Cambridge, MA (2000)
25. Animals can use deceptive behavior, including deceptive vocalizations (e.g. Lorenz, K.: The Foundations of Ethology, Springer Verlag, New York, NY (1981)); this does not change the main argument here about fused emotion-concept-behavior

26. Jung, C.G.: Psychological Types. In: the Collected Works, v.6, Bollingen Series XX, 1971, Princeton University Press, Princeton, NJ (1921)
27. Perlovsky, L.I.: Neural Dynamic Logic of Consciousness: the Knowledge Instinct. Chapter in: Perlovsky, L.I.:, Kozma, R.. (eds.): Neurodynamics of Higher-Level Cognition and Consciousness, Springer, to be published (2007)
28. Saussure, F. De: Course in General Linguistics, 1916. McGraw-Hill, New York, NY (1965)
29. Ruhlen M.: The Origin of Language. J. Wiley & Sons, USA (1994)
30. Fontanari, J.F., Perlovsky, L.I.: Evolving Compositionality in Evolutionary Language Games. IEEE Transactions on Evolutionary Computations, in print (2007)
31. Humboldt, W. von: Über die Verschiedenheit des menschlichen Sprachbaues und ihren Einfluss auf die geistige Entwickelung des Menschengeschlechts, Berlin: F. Dummler (1836). Also in W. P. Lehmann (ed.): A Reader in Nineteenth Century Historical Indo-European Linguistics. Indiana University Press, Bloomington, IN (1967)
32. Perlovsky, L.: Evolution of Consciousness and Music. Zvezda, 8 (Russian) (2005) 192–223; http://www.ceo.spb.ru/libretto/kon_lan/ogl.shtml, Music—The First Principle (English) (2005)
33. Perlovsky, L.: Consciousness, Language, and Mathematics. Zvezda, 2 (Russian) (2003) 204–228
34. Harris C.L., Ayçiçegi A., Gleason, J.B. (2003). Taboo words and reprimands elicit greater autonomic reactivity in a first language than in a second language. Applied Psycholinguistics, 24: 561–579
35. Immordino-Yang, M.H. (2004). A Tale of Two Cases: Emotion and Prosody after Hemispherectomy. American Educational Research Association 2004 Annual Meeting
36. Nowak, M.A., Komarova, N.L. & Niyogi, P.: Computational and evolutionary aspects of language. Nature, 417(6) (2002) 611–617
37. Nowak, M.A. and Krakauer, D.C. The evolution of language. Proc. Natl. Acad. Sci. USA, 96, 8028–8033, Evolution (1999)
38. Brighton et al.: Language as an evolutionary system. Physcs of Life Reviews, 2(3) (2005) 177–226
39. Fontanari, J.F. and Perlovsky, L.I.: Evolution of communication in a community of simple-minded agents. IEEE Int. Conf. Integration of Knowledge Intensive Multi-Agent Sys., Waltham, MA (2005)
40. Fontanari, J.F. and Perlovsky, L.I.: Meaning Creation and Modeling Field Theory. IEEE Int. Conf. Integration of Knowledge Intensive Multi-Agent Sys., Waltham, MA (2005)
41. Perlovsky, L.I. & Fontanari, J.F.: How Language Can Guide Intelligence. EVOLANG'06, Rome Italy (2006)

Viral Knowledge Acquisition Through Social Networks

Dmitri Soshnikov and Mikhail Chernomordikov

Moscow Aviation Technical University
{dsh,mixen}@mailabs.ru

Abstract. In this paper, we present an approach for semi-structured knowledge acquisition through the concept of Structured Semantic Wiki, based on social virus spreading in the internet-based community. This approach allows harnessing collective intelligence of a community and inducing structured annotated knowledgebase of community relations by viral-driven actions of community participants.

Keywords: Semantic Web, Semantic Wiki, Social Virus, Online Community, Social Networks, Knowledge acquisition.

1 Introduction

Nowadays, there are two approaches to deal with Internet complexity: full-text search in unstructured data, and explicit knowledge representation / resource annotation for automatic machine-based reasoning. The main problem with the latter approach, which forms the basis for Semantic Web, is the creation of *resource annotations* – in order for this approach to be really beneficial with respect to automatic methods those annotations have to be created manually.

This paper presents an approach for collective development of semantic web knowledgebase by a community of users driven by a social virus. The key elements of this approach are:

- Use of *Structured Semantic Wiki* as a backend for knowledge representation, that serves as a common denominator for unified knowledge organization
- Community-based *collaborative knowledge acquisition* that allows harnessing collective intelligence of a community and exhibits additional self-organization
- Usage of *social virus* mechanism for engaging users in knowledge acquisition process

2 Structured Semantic Wiki

Many modern web systems that belong to so-called Web 2.0 generation are based on the idea of self-organization and *collaborative intelligence* of users. Good example of collaborative web solution is Wiki, one of the main examples being Wikipedia, free web encyclopedia. There users can create or modify content themselves, having only some very limited guidelines on how the content should be organized.

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 305–308, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Today Wikipedia is one of the most popular sites worldwide. It includes huge amount of articles in many different languages. But from knowledge management point of view it has one serious shortage – it is not suitable for automatic processing and reasoning. All information in Wikipedia is accessible only by users, not computers. The only *explicit semantics* lies in the links between pages and in flexonomic tagging, the major part being hidden in natural text.

Semantic Wiki [2] usually refers to a wiki with semantic extensions, allowing, for example, to semantically annotate pages using ontologies based on Semantic Web standards such as RDF and OWL. There are many different implementations and flavours of Semantic Wikis, most of them relying on some additional format and syntax conventions for creating semantically rich objects such as pages and links.

The notion of *Structured Semantic Wiki* presented in this paper is slightly different, in a sense that it does not allow creating arbitrary page structures, but rather constraints the user by requiring him to comply to some *ontology*. For example, we could pre-define the set of possible page or object relations, and force the author to provide a link type from this set for each link (eg. by choosing it from drop-down menu). We could assign a set of required properties for a page, and force the author to fill in those properties, which would establish required ontology bindings for that page. Taking this concept one step further we can provide some mechanism for embedding structured production rules and/or some other dynamic knowledge representations on top of structured semantic wiki, thus making this a platform for *collaborative knowledgebase development*. Semantic Wiki can be considered to be an extension on top of Semantic Platform for knowledge-based site development [3].

Additional ontology-based constraints take away some of the flexibility, but result in a more structured knowledgebase which is bound to ontology and suitable for reasoning. Also, it is theoretically possible to embed some dictionaries and/or entire ontology structure into the wiki itself, making ontology *self-contained* and *modifiable* by authors. Still, it is important that the ontology remains logically separated from the main content. In this case collaborative knowledge management becomes efficient.

3 Community-Based Viral Mining

Collaborative projects rely on the joint activities by their users. Large number of users has a potential to dramatically increase collaborative knowledge accumulation. The idea presented in this paper is to use so called ‘social virus’ to drive users to knowledge acquisition activities.

Social viruses live in so-called *communities* or *social networks*. This idea is crucial for the 2.0 generation web. By community $C=<U, R>$, we mean a set of users U with an acquaintance relation R. A social virus is a phenomenon when community users “spread the word” through their acquaintances, asking them to perform some action and pass the information along. We end up with the proper subgraph of affected participants $A \subseteq C$, which is typically connected (if we start spreading the virus from a connected set of vertices).

Social network itself has some knowledge structure about the relationship of participants. When we perform *social viral mining*, this knowledge is transferred to the underlying knowledgebase, and we ask each participant to augment it by some additional knowledge through the interface of a structured semantic wiki. In this manner, we are inducing an explicit knowledgebase about the subset of the original community, enriched with the knowledge extracted from individuals.

There are many cases where this approach can be applied. In the most obvious way, we can seek to create advanced knowledgebase about community participants, capturing their interests with relation to some ontology, as well as their relationships with each other and with other objects such as organizations. On top of such knowledgebase it would be easy to build reasoning service that would recommend a person a proper way to get in touch with interested parties based on the similarity of ontological descriptions [1], performing ontology-based community restructuring. Another useful example would be to ask users to populate certain areas of semantic wiki site, thus performing collective knowledge extraction.

The effectiveness of *viral-driven mining* lies in the fact that users like to take part in mass actions like ‘flash mobs’. Social virus should include some useful message to motivate users to perform the action. This work should be somehow motivated on one hand and useful to the whole system on the other hand. The concrete architecture of a social virus is not universal – it depends on the actual system and the user community. Virus should be tuned to concrete aims and concrete target audience.

4 Shoqeri: Social Semantic Super-Network Populated by Viral Merging

Proposed approach can also be used to unify several existing social networks by inducing combined knowledgebase on top of them. In this process, the main problems are those of identity and relationship.

Identity problem is related to people having different accounts on different social networks, and thus different profiles and different behaviors. Relationship problem is caused by users having different links and contacts in different communities. Separation of communities does not allow creating cross-community links, and it is not always feasible to reflect entire relationship picture from real life. Using one account identity system on top of other networks one can reduce these problems.

Shoqeri is a research project at MAILabs laboratory of Moscow Aviation Technical University that tries to implement the approach described above. The aim of this project is to build a new generation of *semantic social network* that would bridge different existing networks and services and provide semantically annotated knowledge-based relationship and identity model on top of the existing social networks. By running different social viruses in different networks we can drive users to the main semantic wiki that will allow user to describe his ontology-based account, providing both *annotation* and *identity mapping* between different social networks.

As a research project Shoqeri consists of different stages. During the first stage we are interested to investigate how effective social virus approach would be on the student audience. According to Yandex research, LiveJournal is currently the most popular and widespread social and blog system in Russia. That is why current aim of Shoqeri project is to drive students from LiveJournal and induce knowledge about their relationships and fields of interest to form ontological profile.

The implementation of Shoqeri project is ASP.NET 2.0 website with Microsoft AJAX extensions. Users do not have to create new account on this site - instead they use existing OpenID from LiveJournal. It is a serious step to lower ‘entry level’ for users. On the site users answer different questions about themselves, their friends and colleagues and about the university. All this information is linked to pre-defined ontology. We use LiveJournal’s community that allows us to automatically determine user’s friends and use that info for ontology-based annotation.

To motivate users to add information we use social virus in MAI LiveJournal communities, student forums and mailing lists. To provide incentives we designed some rating mechanism, where users can increase their rating by answering questions, filling information and inviting new users. Also they can put special buttons with their rating and link to the project on their websites, blogs, forums, etc.

First phase of Shoqeri project will allow us to test the effectiveness of social-based viral mining in the closed community. It will allow to research and improve social mechanisms for collaborative knowledge management.

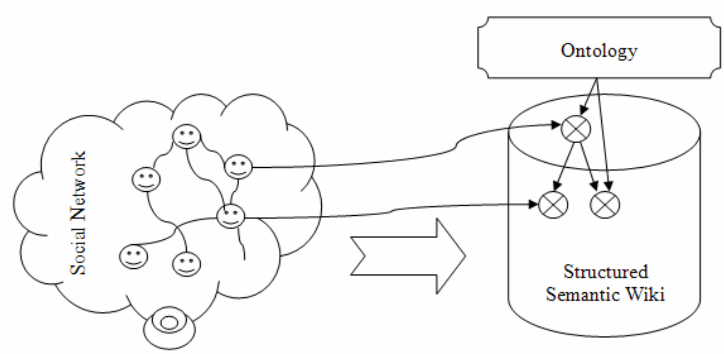


Fig. 1. Explicating Social Network Structure through Viral Mining using Semantic Wiki

References

1. Soshnikov, D.: Multi-agent Approach for Community Clustering Based on Individual Ontology Annotations. In V. Gorodetsky, J. Liu and V. Skormin (eds): Autonomous Intelligent Systems: Agents and Data Mining. Lecture Notes on Computer Science, Vol. 3505. Springer-Verlag (2005) 247–262
2. R. Tazzoli, P. Castagna, and S. E. Campanini: Towards a semantic wiki web. Proceedings of the International Semantic Web Conference (2004)
3. Tivelkov, A.: Building Problem-Oriented Meta-Interactive Web Resources Based on Frame Knowledge Representation. In Proc. of Conference Microsoft Technologies in Theory and Practice of Programming. MSU Publishing (2006)

Chinese Weblog Pages Classification Based on Folksonomy and Support Vector Machines

Xiaoyue Wang, Rujiang Bai, and Junhua Liao

Shandong University of Technology Library Zibo 255049, China
{wangxy,brj,ljhbrj}@sdut.edu.cn

Abstract. For centuries, classification has been used to provide context and direction in any aspect of human knowledge. Standard machine learning techniques like support vector machines and related large margin methods have been successfully applied for this task. Unfortunately, automatic classifiers often conduct misclassifications. Folksonomy, a new manual classification scheme based on tagging efforts of users with freely chosen keywords can effective resolve this problem. In folksonomy, a user attaches tags to an item for their own classification, and they reflect many one's viewpoints. Since tags are chosen from users' vocabulary and contain many one's viewpoints, classification results are easy to understand for ordinary users. Even though the scalability of folksonomy is much higher than the other manual classification schemes, the method cannot deal with tremendous number of items such as whole weblog articles on the Internet. For the purpose of solving this problem, we propose a new classification method FSVMC (folisonomy and support vector machine classifier). The FSVMC uses support vector machines as a Tag-agent which is a program to determine whether a particular tag should be attached to a weblog page and Folksonomy dedicates to categorize the weblog articles. In addition, we propose a method to create a candidate tag database which is a list of tags that may be attached to weblog pages. Experimental results indicate our method is more flexible and effective than traditional methods.

Keywords: Classification, Folksonomy, Support Vector Machines, Tag-agent, Weblog.

1 Introduction

According to Google index, the number of web pages now exceeds 8 billion, and is increasing by 1.5 million per day. The global population of internet users is also growing rapidly. For example, according to the China Internet Network Information Center (CNNIC) statistics, to the end of December of 2005, the population of internet users in China rose by 111 million, while the total number of broadband internet users has reached 64.3 million.

Users can find the web pages they want in the enormous database of web pages represented by the Internet, and many search engines are available to users, including the popular Yahoo!, Baidu, and Google. Typical search engines work through

V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476, pp. 309–321, 2007.
© Springer-Verlag Berlin Heidelberg 2007

keyword inputs. However, pages retrieved in this manner usually include invalid links and irrelevant web pages. A good web page classification method is thus an urgent need in facilitating user searches.

There are many classification methods for web pages. A decision tree Apte et al. (1998) is a general data classification method [1]. Its two major advantages are (1) it is faster; and, (2) the classification result can be transformed into an IF-THEN relation that the user can easily understand. Common decision tree methods include ID3 Mitchell (1997) and C4.5 Quinlan (1993). The disadvantage is that when categories are more numerous, it makes mistakes more easily. Mccallum and Nigam (1998) transform the frequency of keywords to conditional probabilities in which Bayesian probability is used to calculate the probability value between every document and category. Under this system, the category with highest probability is the one the document belongs to. The advantage is that the correlation between two documents can be represented by a probability. However, the processing load is higher. A support vector machine, named SVM, is a supervised method Cortes and Vapnik (1995), Gunn (1998) and Joachims (1998) that uses a portion of the data to train the system and then forms a learning model that can predict the category of documents [2]. k-NN method is often used in text document classification Tan (2005). Woog and Lee (2003) use a k-nearest neighbor (k-NN) approach to calculate the likelihood of a category and relevant web page. In order to improve performance, they add a feature selection, HTML tags, and a new similarity measure and evaluation. Selamat and Omatu (2004) use a training sample to do the stemming and remove stop words, then the feature vector dimensions for a portion are reduced, while another portion is used for each category extraction of the keyword and to assign the weight value [3]. Two types of feature vectors are then combined and inputted to a neural network for training. The system can then classify the web pages into the desired categorization. Unfortunately, a long time is required for training, and the convergence speed is low.

The main problem of automatic classification is hard to classify the big messy web pages such as whole weblog articles on the Internet. Because it will waste a long time and storage space to learning and classing the web articles through automatic classification methods. Another problem is that the machine learning method is hard to realize the community, extendable and feedback from the users. Thus the common users are hard to understand the labels defined by the professors. A new social classification method Folsonomy may solve this problem, but its weakness is imprecision.

For the purpose of solving above problems, we propose a new classification method, which uses a support vector machine combining Folksonomy. We name it FSVMC (Folisonomy and Support Vector Machine Classifier). First we preprocess the weblog articles. Extract key words , tags, and the classification by authors from the weblog articles. Second we construct a candidate tag database. Third the tag-agent created, which determined a particular tag is appropriate for a weblog article or not. At last the web pages are categorized through Folksonomy.

The rest of this paper is structured as follows. In section 2 and 3 we give a brief review of the Folksonomy and SVM method, respectively. The classification method that uses FSVMC is described in section 4. We provide our experimental methodology and results in section 5. In section 6, we make conclusions.

2 Folksonomy

2.1 What Is a Folksonomy?

Folksonomy is a neologism for a practice of collaborative categorization using simple tags. It is a word combination of “folk” and “taxonomy”. A folksonomy is an Internet-based information retrieval methodology consisting of collaboratively generated, open-ended labels that categorize content such as Web pages, online photographs, and Web links. A folksonomy is most notably contrasted from a taxonomy in that the authors of the labeling system are often the main users (and sometimes originators) of the content to which the labels are applied. The labels are commonly known as tags and the labeling process is called tagging.

The process of folksonomic tagging is intended to make a body of information increasingly easier to search, discover, and navigate over time. A well-developed folksonomy is ideally accessible as a shared vocabulary that is both originated by, and familiar to its primary users. Two widely cited examples of websites using folksonomic tagging are Flickr and Del.icio.us. In 2004, Flickr (an online photo sharing service) and del.icio.us (an online bookmarking service) employed folksonomy as a classification method, and they became successful. The success of these services made folksonomy as one of major players of browsing information on the web.

Because folksonomies are developed in Internet-mediated social environments, users can discover (generally) who created a given folksonomy tag, and see the other tags that this person created. In this way, folksonomy users often discover the tag sets of another user who tends to interpret and tag content in a way that makes sense to them. The result, often, is an immediate and rewarding gain in the user's capacity to find related content.

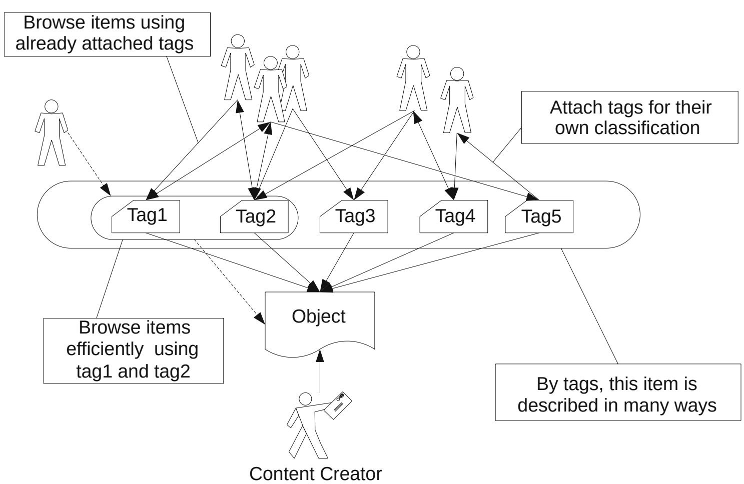


Fig. 1. Folksonomy

2.2 Strengths and Weaknesses of Folksonomy

Robin Good is quoted as saying that “a folksonomy represents simultaneously some of the best and worst in the organization of information.” There is clearly a lot to be learnt from this new method of classification as long as you remain aware of the strengths and weaknesses.

Here are some advantages of folksonomy. First, folksonomy can provide serendipity. There are two methods for getting the information you want. These were represented as:

1. searching to find relevant items in a query, and
2. browsing to find interesting items.

Recently, the first way is popular. Google and other full text search engines are good examples of the first approach. The latter approach was popular in the Internet of 1990s. The hierarchal Yahoo directory was developed for the purpose of browsing, but categorization by a limited number of professionals cannot practically deal with the huge number of web pages. Looking at the cost and limitation of human power, folksonomy is a new promising method for the classification method.

The second advantage of folksonomy is that the classification results are familiar to ordinary people. What should be noted is that taggings are not done by professionals but by ordinary users with their everyday languages. It results in tags that are keywords familiar to ordinary users.

In addition, the vocabulary used by ordinary people is changing everyday. For example, a web page concerning “using xHttpRequest JavaScript” should be classified in “JavaScript” in 2003. In 2005, however, it should be classified in “Ajax”. Looking at the changing vocabulary, folksonomy has a big advantage to other methods. In the case of categorization by professionals or by authors, re-classification requires much effort. However, with folksonomy, when the vocabulary changes, new users attach tags to old articles based on the new vocabulary.

The weaknesses of folksonomy are imprecision of terms and uncontrollable. Folksonomy terms are added by users, which means that they are often ambiguous, too personalized and imprecise. Some sites only allow for single word metadata resulting in many compound terms; many tags are of single use and at present there is little or no synonym or homonym control. The uncontrolled and chaotic set of terms created means that folksonomies do not support searching as well as controlled vocabularies.

3 Support Vector Machine

The primary idea of support vector machine (SVM) is using a high dimensional space to find a hyper plane to do binary division, where the achieved error rate is minimal. An SVM can handle the problem of linear inseparability.

An SVM uses a portion of the data to train the system and finds several support vectors that represent training data. These support vectors will be formed into a model by the SVM, representing a category. According to this model, the SVM will classify a given unknown document by the following classification decision formula:

$$(x_i, y_i), \cdots, (x_n, y_n), x \in R^m, y \in \{+1, -1\}.$$

Here $(x_i, y_i), \cdots, (x_n, y_n)$ are training samples, *n* is the number of samples, *m* is the input dimension, and *y* belongs to the category of +1 or –1, respectively.

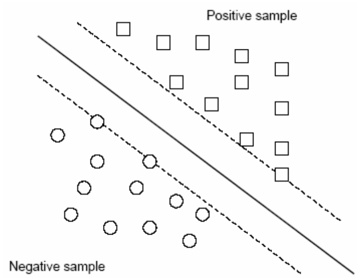


Fig. 2. The hyper plane of SVM

In a linear problem, a hyper plane is divided into two categories. Fig. 2 shows a high dimension space divided into two categories by a hyper plane. The hyper plane formula is $(w \bullet x_i) + b = 0$.

The classification formula is:

$$(w \bullet x_i) + b > 0 \;\; if \;\; y_i = +1 \qquad (w \bullet x_i) + b < 0 \;\; if \;\; y_i = -1$$

However, for many problems it is not easy to find a hyper plane to classify the data. The SVM has several kernel functions that users can apply to solve different problems. Selecting the appropriate kernel function can solve the problem of linear inseparability.

Another important capability of the SVM is that it can deal with linear inseparable problems. Internal product operations will affect the classification function. A suitable inner product function K(xi·yj)can solve certain linear inseparable problems without increasing the complexity of the calculation. Table 1 lists four kernel functions that are often used. The different kernel functions are suited to different problem types.

Table 1. Four kernel functions

Kernel	Kernel function	Parameter
Dot	k(x,y)=x • y	None
Polynomial	k(x,y)=(x • y+1)d	d(degree)<integer>
Neural	k(x,y)=tanh(ax • y+b)	a,b<float>
Anova	k(x,y)=(Σ iexp(-γ(xi-yi))	γ(gamma),d(degree)<integer>

4 Combination of Folksonomy and Support Vector Machines for Web Pages Classification

4.1 System Overview

We proposed a new classification method called FSVMC (folisonomy and support vector machine classifier). After preprocessing the weblog articles, Support vector machines (SVM) act as a Tag-agent which is a program to determine whether a particular tag should be attached to an article. In addition, we propose a method to create a candidate tag database which is a list of tags that may be attached to articles.

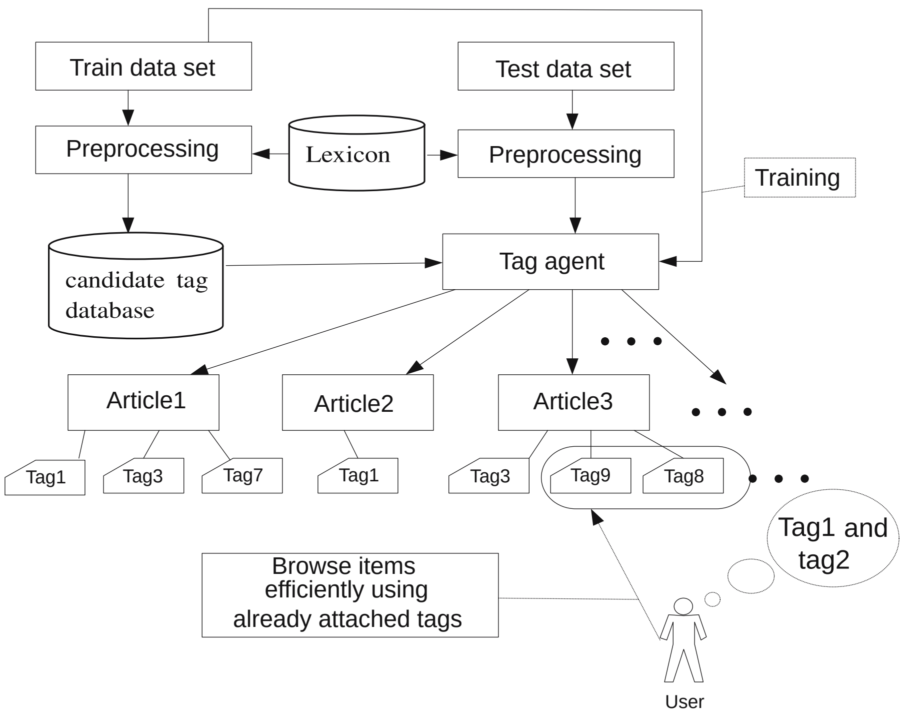


Fig. 3. System Overview

The overview of FSVMC is shown in Fig. 3, described as follows:

(1) Preprocessing: preprocessing includes removal of HTML tags and Chinese word segmentation. HTML tags are removed but the text is retained, to prevent interference. Then, the text is compared with a Chinese lexicon to extract Chinese keywords for word segmentation.

(2) Constructing candidate tag database: after segmentation of the Chinese word, the system extracts the web page text features.

(3) Creating Tag-agent: attach tags to the web pages through Tag-agent.

The detail description as follows:

4.2 Preprocessing

4.2.1 Removal of HTML Tags

Most web pages are written in HTML at present. HTML uses open/closed tags to indicate web page commands, represented by '!' and '/O', respectively. Since content is not marked by these tags, we remove the HTML tags to reduce the burden of analysis.

4.2.2 Segmentation of the Chinese Words

English sentences have spaces between the words, but Chinese sentences do not. Therefore, computers find it difficult to analyze exactly how many Chinese words comprise a term. Different methods of extracting Chinese terms lead to different semantic meanings. We used the Institute of Computing Technology Chinese Lexical Analysis System (ICTCLAS), developed by Institute of Computing Technology of Chinese Academy of Sciences, for that purpose.

ICTCLAS divides the inputted Chinese text into several terms and marks word classes such as noun, verb, or adjective. Hence, we used the word class to select terms. After that we remove the stop word.

4.3 Constructing Candidate Tag Database

We create a candidate tag database by selecting category names and tags used on all the train weblog data set. This is because most weblog services allow users to construct their own category systems and attaché tags. The important point to note is that these category names depend on the weblog authors' vocabulary. Taking these situations into account, we chose popular and descriptive category names, and used them as the candidate tag set. It should be noted that the selection should be repeatedly performed for reflecting new category names. Some examples of these category names are shown in Table 2.

Table 2. Category Names and Their Goodness as a Tag

Good Tag	Lack descriptiveness	Lack popularity
"Music" (26316; 80%)	"DIARY" (35630; 62%)	"Blogurmet" (1;-)
"Movie" (17476; 88%)	"Daily Life" (63857; 66%)	"Comparison of Cleaners" (1;-)
"Football" (392; 90%)	"Others" (33252; 60%)	"The Wing Goes to the Sky" (1; -)
"Economy" (2502; 88%)	"Murmur" (13115; 71%)	"Contents of Subjects" (1;-)

First, we measured the popularity of a category name by the number of weblog sites containing the category name in its categorization system. If the category name is a popular word (or a popular short phrase), they should be used by multiple bloggers. We experimentally employed twenty bloggers as the threshold for the popularity.

Second, we measured the descriptiveness of each category name. We prepared a classifier for each category name. These are SVM classifier used in the Tag-agent. If a category name is not descriptive, the classification accuracy of the corresponding classifier should be low.

Two conditions are required to the process of deciding whether a category name is descriptive or not. First, unsuitable tags in a candidate tag set should be minimized. Second, the required sample data to make a judgment should be kept minimal.

To fulfill these two requirements, we have to estimate the proper classification accuracy using a small number of articles. And then, we can see a correct classification as a probability event, and by defining p as the proper accuracy of the classifier, the probability of the event is p. Since we do not have a priori knowledge about this distribution, it is natural that the distribution of the classification result is assumed to be a binomial distribution. In addition, according to the central limit theorem, the distribution can be approximated by the normal distribution when n is not too small.

As a result, we can calculate the confidence interval [10]. For example, setting the confidence rate at 99.5%, and the lower bound of confidence interval of the proper accuracy can be calculated as follows:

$$\frac{c}{n} - 2.58 \times \sqrt{\frac{c}{n}(1.0 - \frac{c}{n}) / n}$$

When the lower bound of the confidence interval exceeds 80% (the threshold will be chosen in Section 5.2), we can judge that the category name is suitable for the candidate tag database.

4.4 Creating Tag-Agent

Tags are attached to each article by a Tag-agent. Our Tag-agent is an array of taggers that determine whether to assign a particular tag or not (see Figure 4). Each tagger is a classifier that determines whether a particular tag is appropriate for an article or not. For each tag of the candidate tag database, the corresponding tagger is created.

A tagger should be periodically trained. We should use the latest weblog articles from train data set as the training data, because the tagger should take latest trends into consideration. For the purpose of multi-tagging, the following conditions are required for a classifier in each tagger. (1) Fast classification speed, (2) low memory consumption (during classification), (3) over fitting avoidance, (4) high classification accuracy. Taking into account of these four conditions, we compared three popular text classification algorithms: k-Nearest Neighbor, Naive Bayes, and SVM, and we finally selected SVM.

For training each classifier, we used collected latest weblog articles from train data set. We used articles which were categorized into “A” by those authors, as positive examples for a classifier corresponding to the tag “A”. If there are more than 2,000 articles, we used the latest 2,000 articles. As negative examples, we use the same number of randomly selected articles categorized into other than “A”. The threshold “2,000” is decided for reducing the time consumption of the experiment, in spite of the fact that we know more articles make the classification accuracy better. As usual, we converted each weblog article into a vector using the vector space model.

For dealing with the changing vocabulary of ordinary people, every tagger is retrained regularly (e.g. once a day).

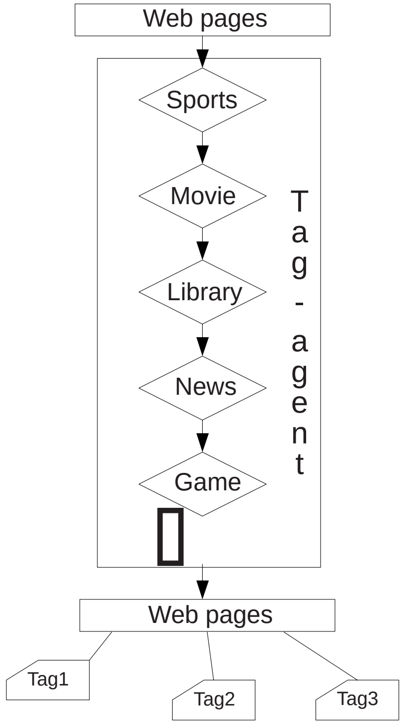


Fig. 4. Tag-agent

5 Experiment

5.1 Experimental Data Set

Because weblog has original tags and categorized by authors. We used real Chinese weblog articles as our experimental data set. We collected the data set from http://blog.sina.com.cn (14,910 articles) as train data set. There are 74,339,392 articles from http://blog.sohu.com/ as test data set.

We treat the description section of an article as its contents, and class name by authors or tags of an article as its category.

Articles categorized in "Uncategorized", "Diary","News"(Since our experimental data are Chinese weblog articles. In this paper, for convenience, all category names are translated into English), were removed from the experimental data set, because weblog articles of these categories are too many, and they prolong our experimental time. Investigating these articles, we can see that most of the categories have small numbers of weblog articles, and only a small portion of categories have more than 1,000 articles.

5.2 Classification Accuracy

We measured the classification accuracy of the classifiers for more than 40 tags. In this experiment, the test set to measure the accuracy is increasing according to the number of articles. A part of the results is shown in Figure 5. It is obvious that there is a dichotomy of the classification accuracies for the categories.

The category names in a lower group, say “DIARY” and “Daily Life” in Figure 5, are ambiguous and do not specify their contents. Taking Figure 5 and the same experiments on the other categories into consideration, we chose 80% as the classification accuracy threshold of candidate tag set selection.

5.3 Constructing Candidate Tag Database

The candidate tag set selected by our program is compared to a candidate tag set selected by subjects. We applied the algorithm described in the previous section to category names in our experimental data, and compared the output of our program with manually selected tags.

The resulting accuracies and precisions are shown in Table3. The experimental result implies that our algorithm can select the appropriate candidate tag set. In particular, the precisions of the results are very high. Even when the training has done with about 100 articles, the precision rate is kept high.

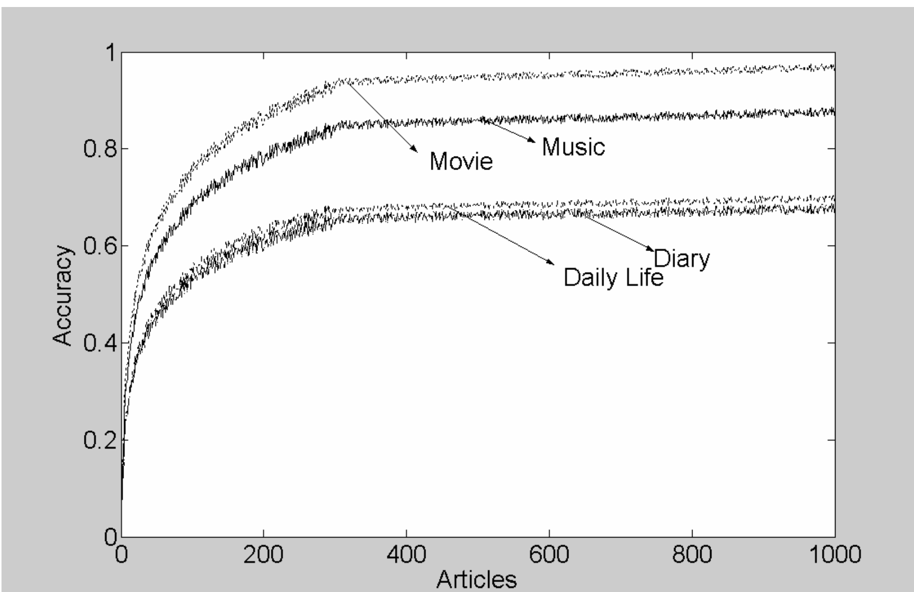


Fig. 5. Classification Accuracy of Descriptive and not Descriptive Category Names

Table 3. Evaluation of Candidate Tag Selection

articles	tp	fp	fn	tn	accuracy	precision
>=100	91	3	113	135	66.0%	96.8%
>=200	76	3	67	107	72.3%	96.2%
>=400	41	2	24	62	79.8%	95.3%
>=800	27	1	5	34	91.0%	96.4%
>=1600	11	1	0	16	96.4%	91.6%

[articles] Condition of applied category names. “>= 100” means “category names which have more than 100 articles.”

[tp] Category names where both the algorithm and human answered it is suitable as a candidate tag.

[fp] Category names where our algorithm indicates that it is appropriate for candidate tag set, but a human do not agree with this.

[fn] Category names where our algorithm indicatess that it is not appropriate for candidate tag set, but a human do not agree with this.

[tn] Category names where both the algorithm and human answered it is not suitable as a candidate tag.

[accuracy] Accuracy of our algorithm compared to human selection. (tp + tn)/(tp + tn + fp + fn).

[precision] Precision of our algorithm compared to human selection. tp/(tp + fp).

5.4 Tag-Agent

We performed experiments on tagging by measuring the precision and recall of our taggers in Tag-agent, which are limited to categories that have more than 1100 articles.

The results of the experiment are shown in Figure 6. Most of the taggers have high precisions. However, recalls of these taggers vary from 0.1 to 0.8. By the way, looking at the situation where users browse weblog articles guided by tags, precision is very important. From this point of view, our tagger is proven to work well.

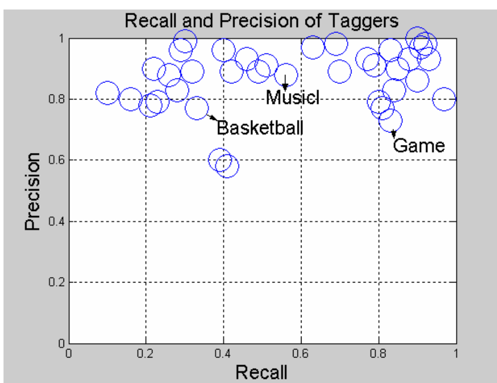


Fig. 6. Recall and Precision of Taggers

5.5 Demo

Here, we demonstrate our system. Our system is implemented on a web server. On the main page of our system, tags are listed on the right (see Figure 7). If you click the "足球"(Football) tag, the system displays listed weblog articles to which the "足球" tags are attached by our system. These articles are related to football, not only articles categorized into the "football" by these authors but also the other categories.

Fig. 7. Screen shot of Folksonomy and SVM based classifier

6 Conclusion

6.1 Advantages and Limitations

First of all, we emphasize the scalability of our method. Since all taggers are independent, the automated folksonomy can be implemented with paralleled machines. In our experiments, we used six (virtual) processors on two machines that do not share their memories.

Through our method users can find their interesting and easy understandable information.

Since our proposed method is a replacement of the tagging by users, our method and most of the existing methods related to folksonomy can be used together. Looking at the existing services with folksonomy, many useful techniques are available. These methods can be used in coordinated with our automated tagging. In addition, our method can be used even in a user-tagged folksonomy system. Using our method for tag suggestion, better suggestions can be expected.

However, compared to folksonomy based on user's tagging, our method attaches too many tags. If an article is about economics and it contains a bit of content related to a game as an example, our system attaches the "Game" tag to this article. When a user who seeks game articles uses our system, the above article is shown. Although the system can determine if a particular concept is contained or not, it cannot determine whether that concept is a peripheral one.

6.2 Different from Other Works

Looking at Tag-agent, many approaches have already been proposed ([11] etc). However, our proposed methods are totally different from most of them. Most studies concerning train their taggers with multitagged documents. In contrast, our method learns tags and their concepts from categorized documents. The reason why we employ this approach is that the majority of weblog articles are assigned to one category and hard understand. Moreover, [12] showed the importance of the centralized topic-centric view of the weblog sphere. Our approach aims at the same objective but these two approaches are different. Their approach treats a category as a unit and the relations between the categories are managed. In contrast, our approach treats a weblog article as a unit and each article is classified by its topic.

6.3 Conclusion

In this paper we proposed an effective flexible classifier, called FSVMC (folisonomy and support vector machine classifier). The experiments on test data set showed that FSVMC is effective for the weblog documents classification, and has the better performance in classification precision, stability and flexible comparing with the traditional classification methods. The results reported here are not necessarily the best that can be achieved. To seek new techniques to enhance the performance of FSVMC is an important issue in future study.

References

1. Apte, C., Damerau, F., & Weiss, S. M.: Text mining with decision trees and decision rule. Proceeding of the automated learning and discovery conference. Carnegie-Mellon University (1998) 99–103
2. Gunn, S. R.: Support vector machines for classification and regression. ISIS technical report. Image speech and intelligent systems group of University of Southampton. (1998) 31–36
3. Tan, S.: Neighbor-weighted k-nearest neighbor for unbalanced text corpus. Expert Systems with Applications (2005) 1–5
4. Salton&McGill, Introduction to modern information retrieval, New York, McGraw-Hill, USA (1983) 13–17
5. Schapire, R. E. and Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine Learning (2000) 135–168
6. Paolo Avesani, Marco Cova, Conor Hayes, and Paolo Massa: Learning contextualised weblog topics. In WWW 2005 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics (2005) 20–33

Author Index