THE IEEE
Computational Intelligence
BULLETIN

IEEE COMPUTER SOCIETY

IEEE Computer Society
Technical Committee
on Computational Intelligence

February 2004 Vol. 3 No. 1 (ISSN 1727-5997)

Profile

Conference Reports

Feature Articles

Announcements

On-line version: http://www.comp.hkbu.edu.hk/~cib (ISSN 1727-6004)

The IEEE Computational Intelligence Bulletin

Aims and Scope

The IEEE Computational Intelligence Bulletin is the official publication of the Technical Committee on Computational Intelligence (TCCI) of the IEEE Computer Society, which is published twice a year in both hardcopies and electronic copies. The contents of the Bulletin include (but may not be limited to):

1) Letters and Communications of the TCCI Executive Committee
2) Feature Articles
3) R & D Profiles (R & D organizations, interview profiles on individuals, and projects etc.)
4) Book Reviews
5) News, Reports, and Announcements (TCCI sponsored or important/related activities)

Materials suitable for publication at the IEEE Computational Intelligence Bulletin should be sent directly to the Associate Editors of respective sections.

Technical or survey articles are subject to peer reviews, and their scope may include the theories, methods, tools, techniques, systems, and experiences for/in developing and applying biologically and linguistically motivated computational paradigms, such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, and self-organization in the research and application domains, such as data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality.

Publisher: *The IEEE Computer Society Technical Committee on Computational Intelligence*
Address: *Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (Attention: Dr. Jiming Liu; Email: jiming@comp.hkbu.edu.hk)*
ISSN Number: *1727-5997 (printed) 1727-6004 (on-line)*
Abstracting and Indexing: *All the published articles will be submitted to the following on-line search engines and bibliographies databases for indexing — Google (www.google.com), The ResearchIndex (citeseer.nj.nec.com), The Collection of Computer Science Bibliographies (liinwww.ira.uka.de/bibliography/index.html), and DBLP Computer Science Bibliography (www.informatik.uni-trier.de/∼ley/db/index.html).*

Spike-Based Sensing and Processing

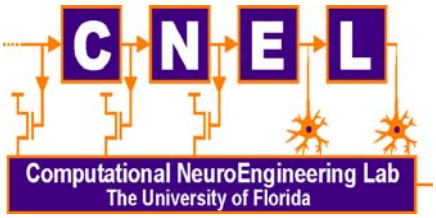
AT THE COMPUTATIONAL NEUROENGINEERING LAB AT THE UNIVERSITY OF FLORIDA

I. INTRODUCTION

Dr. John G. Harris co-directs the Computational NeuralEngineering Lab (CNEL) at the University of Florida, together with its founder: Dr. Jose C. Principe. CNEL seeks to advance the theory and applications of adaptive systems using mathematics and anthropomorphic principles. This work is highly multidisciplinary and of broad impact since it is geared to provide new engineering design principles. Analogies from biology are expressed in appropriate mathematical frameworks and implemented in digital algorithms or directly in analog VLSI chips. Since its inception in 1992, the CNEL has created an international reputation in the areas of adaptive filtering theory, artificial neural networks, nonlinear dynamics, neuromorphic engineering, and more recently in brain machine interfaces and information theoretic learning.

Fig. 1. PhD students Vishnu Ravinthula, Dazhi Wei and Xiaoxiang Gong with Dr. Harris.

Within the CNEL Lab, Dr. Harris and his students are engineering sensors and signal processing systems that use biologically-inspired algorithms and custom analog VLSI circuits. There are many aspects of the brain that are desirable to emulate in engineering systems in the long term, including the following notable performance metrics:



1. **Incredible fault tolerance:** the brain loses an average of 10,000 neurons per day without requiring any sort of explicit reconfiguration or rewiring.
2. **Ultra-low power consumption:** The brain consumes an average of 12 Watts, much less than a typical Pentium computer performing much less computation.
3. **Phenomenal performance:** The best man-made engineered solutions pale in comparison to human performance in common sensory processing tasks such as the recognition of faces or speech.

Unfortunately, it is not well understood how the brain achieves its amazing performance but a more immediate advantage of bio-inspired computation is currently being expoited in the CNEL lab: spiking representations. The brain represents signals using the timing of discrete spikes (or pulses) which is a hybrid of traditional analog and digital computation. The pulses are digital in that the amplitude and width of the pulse do not contain information but the timing of the event is asynchronous, and therefore analog. As humans have learned through the years with such systems as digital cellular phones and digital TV, it is much more efficient to transmit digital signals than to transmit continuous analog voltages due to the improved noise immunity and less cross talk susceptibility. The resulting spike-based engineering systems enjoy reduced power consumption and enhanced dynamic range.

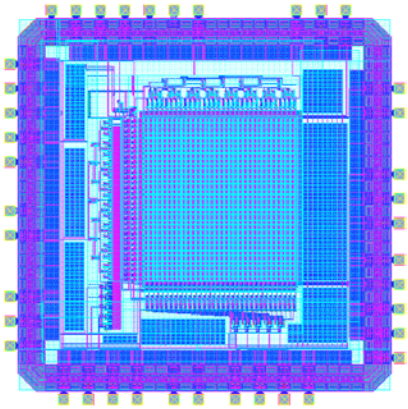


Fig. 2. Experimental 32x32 pixel time-to-first-spike imager.

Through the electronics revolution over the past decades, CMOS process technology is shrinking the usable voltage swing, wreaking havoc on traditional analog circuit design. However, the faster "digital" transistors are better able to process timing signals leading researchers to consider analog computation more similar to that of the brain. This trend will likely continue with nanotechnology since even smaller voltage ranges and even faster devices are promised. Of course, CMOS processes are primarily scaling in favor of faster and faster digital devices, however power consumption is beginning to limit how far these digital circuits can scale.

II. SENSORS

Together with his students, Dr. Harris is developing novel VLSI sensors using this pulse-based methodology. A sensor can typically be designed with a wider dynamic range when time is used to encode the measured signal instead of a voltage, as is the case for typical engineering systems. Graduate students Xiaochuan Guo and Xin Qi have developed a novel time-to-first spike imager using this strategy (see Figure 2).

Fig. 3. PhD students Xin Qi and Harpreet Narula are developing novel spike-based sensors.

Conventional CMOS imagers must choose a single integration time for each pixel which limits the dynamic range to 60-70 dB. On the other hand, each pixel in the time-to-first-spike imager outputs a single spike at a time inversely proportional to pixel intensity. Each pixel therefore chooses a suitable integration time resulting in a greatly enhanced dynamic range of 140dB.

Harpreet Narula has designed a low-power, spike-based potentiostat that can measure currents as low as 1pA. Potentiostats are used to measure electrochemical activity (as a current) for such applications as blood analyzers, food control and glucose sensors.

Du Chen is designing a spike-based neuro-amplifier suitable for implantation. Typical extracellular neural signals have amplitudes of 10-100uV with DC offsets ranging up to 200mV and frequencies ranging from below 1Hz up to 6KHz. A low-noise amplifier was designed to provide a gain of 40dB before translating the output to a series of pulses for efficient transmission.

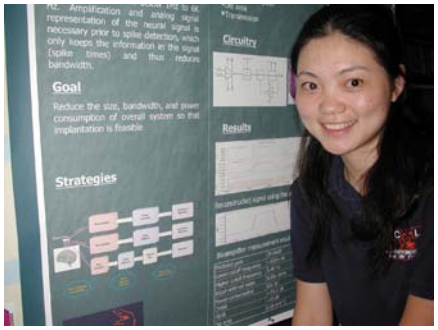


Fig. 5. PhD Student Du Chen is developing spike-based bioamplifiers suitable for implantation.

III. Spike-Based Processing

Rather than convert the spike outputs from the sensors into an analog voltage or a digital signal, the sensor outputs can be processed directly in the spike domain. Time-based signal representations have been in use for many years, including such standard techniques as pulse-width modulation and sigma-delta converters but temporal codes are becoming more and more common with the rising popularity of such techniques as class D amplifiers, spike-based sensors and even ultra-wideband (UWB) signal transmission. However, these temporal codes are typically used as temporary representations and computation is only performed after translation to a traditional analog or digital form.

Xiaoxiang Gong is developing a novel spike-based adaptive filter that processes spike signals as the input and desired signals. Much like traditional adaptive filters, this new class of adaptive filter has applications in areas such as system identification, signal prediction, noise cancellation and channel equalization.

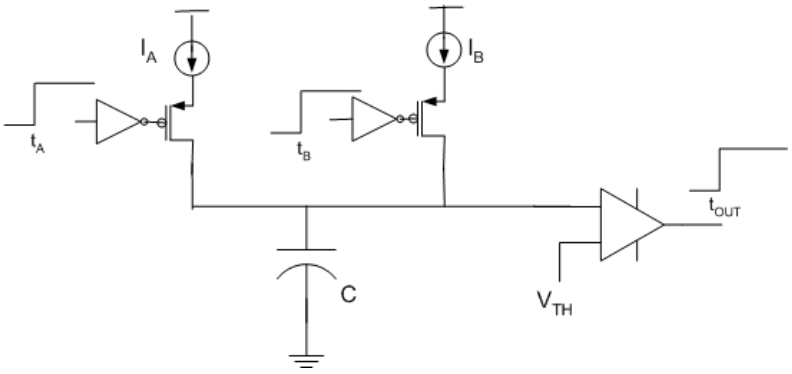


Fig. 4. An arithmetic circuit using the timing of step functions.

Vishnu Ravinthula has developed time-based arithmetic circuits that can perform weighted addition or subtraction in the time domain. One such circuit, shown in Figure 4, computes the following function:

$$t_{out} = \frac{I_A t_A + I_B t_B}{I_A + I_B} + \frac{CV_{TH}}{I_A + I_B}$$

where t_A and t_B are the rise times of the two input step waveforms and t_{out} is the timing of the output step. The circuit computes a fully continuous analog function using only current sources, digital switches and a comparator.

IV. Conclusion

As has been shown, spike-based processing shows great promise for many engineering applications in terms of improved dynamic range and lower power consumption. Nanoscale implementations of these ideas are being considered in collaboration with Dr. Jose Fortes, also at the University of Florida.

Another direction of interest is to explore the use of these circuits to better understand the biological systems that originally inspired them. An understanding of how nervous systems attain their incredible fault-tolerant performance will lead to further improved engineering systems.

Ultimately it is hoped that future generations of biologically-inspired circuits can be directly interfaced to the brain since they will share similar signal representations and organization. Advanced treatments for such disorders as Alzheimer's, strokes and some kinds of paralysis could become feasible.

Contact Information

John G. Harris
Computational NeuroEngineering Lab
PO Box 116130
University of Florida
Gainesville, FL 32611

Email: harris@cnel.ufl.edu
Phone: (352) 392-2652
Website: www.cnel.ufl.edu

2003 IEEE/WIC International Joint Conference on Web Intelligence and Intelligent Agent Technology

Yuefeng Li, Publicity Chair of IEEE/WIC/ACM WI-IAT 2004

The IEEE/WIC International Joint Conference on Web Intelligence and Intelligent Agent Technology was held in Halifax, Canada from 13th to 16th of October 2003. The two proceedings of WI and IAT (including main track regular/short papers and industry track papers) were published by the IEEE Computer Society Press.

This year's officials were: Ning Zhong (Conference Chair), Nick Cercone, Ruqian Lu, and Toyoaki Nishida (Conference Co-Chairs), Jiming Liu (Program Chair), Boi Faltings, Matthias Klusch and Chunnian Liu (Program Co-Chairs), Jianchang Mao, Yiming Ye and Lizhu Zhou (Industry Track Chairs), Cory Butz, Zhongzhi Shi and Yiyu Yao (Workshop Chairs), Jeffrey Bradshaw and Jinglong Wu (Tutorial Chairs), and Yiu-Ming Cheung (Publicity and Web Chair).

I. WEB INTELLIGENCE

Web Intelligence (WI) is a new direction for scientific research and development that explores the fundamental roles as well as practical impacts of Artificial Intelligence (AI) (e.g., knowledge representation, planning, knowledge discovery and data mining, intelligent agents, and social network intelligence) and advanced Information Technology (IT) (e.g., wireless networks, ubiquitous devices, social networks, wisdom Web, and data/knowledge grids) on the next generation of Web-empowered products, systems, services, and activities. It is one of the most important as well as promising IT research fields in the era of Web and agent intelligence. The IEEE/WIC International Conference on Web Intelligence (WI 2003) (http://www.comp.hkbu.edu.hk/WI03/) was a high quality and impact conference, which was sponsored and organized by IEEE Computer Society Technical Committee on Computational Intelligence (TCCI) and by Web Intelligence Consortium (WIC).

Following the great success of WI 2001 held in Maebashi City, Japan in 2001 (http://kis.maebashi-it.ac.jp/wi01/), WI 2003 provided a leading international forum for researchers and practitioners (1) to present the state-of-the-art WI technologies; (2) to examine performance characteristics of various approaches in Web-based intelligent information technology; and (3) to cross-fertilize ideas on the development of Web-based intelligent information systems among different domains.

By idea-sharing and discussions on the underlying foundations and the enabling technologies of Web intelligence, WI 2003 has captured current important developments of new models, new methodologies and new tools for building a variety of embodiments of Web-based intelligent information systems.

II. INTELLIGENT AGENT TECHNOLOGY

The IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003) (http://www.comp.hkbu.edu.hk/IAT03/) was also sponsored and organized by TCCI and WIC.

The upcoming meeting in this conference series follows the great success of IAT-99 held in Hong Kong in 1999 (http://www.comp.hkbu.edu.hk/IAT99/) and IAT-01 held in Maebashi City, Japan in 2001 (http://kis.maebashi-it.ac.jp/iat01/). The aim of IAT 2003 was to bring together researchers and practitioners from diverse fields, such as computer science, information technology, business, education, human factors, systems engineering, and robotics to (1) examine the design principles and performance characteristics of various approaches in intelligent agent technology, and (2) increase the cross-fertilization of ideas on the development of autonomous agents and multi-agent systems among different domains.

By encouraging idea-sharing and discussions on the underlying logical, cognitive, physical, and biological foundations as well as the enabling technologies of intelligent agents, IAT 2003 has demonstrated a lot of new results for building a variety of embodiments of agent-based systems.

II. TUTORIAL & WORKSHOPS

This year, the conferences accepted two tutorials: "A Glimpse at the Future of Agent Technology" by Jeffrey M. Bradshaw at the Institute for Human and Machine Cognition, USA, and "Adaptive Web-Based Systems: Technologies and Examples" by Peter Brusilovsky at University of Pittsburgh, USA.

The conference also accepted 3 workshops on "Knowledge Grid and Grid Intelligence", "Applications, Products and Services of Web-based Support Systems", and "Collaboration Agents: Autonomous Agents for Collaborative Environments".

IV. KEYNOTES/INVITED SPEAKERS

This year, the keynote/invited speakers discussed the following issues about WI and IAT: "Web Intelligence and Fuzzy Logic - The Concept of Web

IQ (WIQ)" (Professor Lotfi A. Zadeh, the slides of this talk can be found from the White Papers Session at WIC home page http://wi-consortium.org/), "Mining and Monitoring Data Streams" (Dr. Philip S. Yu), "Reasoning about Cooperation" (Professor Michael Wooldridge), "Web Information Extraction with Lixto: Visual Logic and Expressive Power" (Professor Georg Gottlob), and "Grid Research in China and the Vega Grid Project at ICT" (Professor Zhiwei Xu).

V. PAPER SUBMISSIONS

WI 2003 and IAT 2003 have received an overwhelming number of paper submissions, more than 592 papers (350 for WI 2003 and 242 for IAT) from over 48 countries and regions: Australia, Austria, Belgium, Brazil, Canada, Chile, China, Colombia, Croatia, Cuba, Czech Republic, Denmark, Egypt, Finland, France, Germany, Greece, Hong Kong, India, Iran, Ireland, Israel, Italy, Japan, Korea, Kuwait, Malaysia, Mexico, New Zealand, Norway, Poland, Portugal, Russia, Saudi Arabia, Singapore, Slovenia, Spain, Sweden, Switzerland, Taiwan, Thailand, The Netherlands, Tunisia, Turkey, UAE, UK, Uruguay, and USA.

It was about 16% of the 350 WI 2003 submissions were accepted as regular papers and 21% of the 350 were accepted as short papers. For IAT 2003, around 24% of the 242 submissions were accepted as regular papers and 21% of the 242 were accepted as short papers.

Figure 1 shows the paper submissions and the number of their countries or regions in 2001 and 2003 for WI and ITA, respectively. This figure depicts that the number of paper submission on WI from 2001 to 2003 have increased significantly.

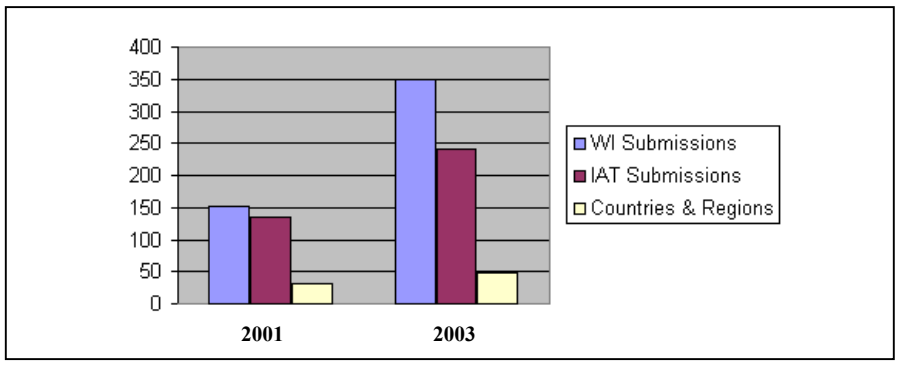


Fig. 1. Paper Submissions for WI and IAT

VI. PRESENTATION SESSIONS

There were 11 technical sessions for WI 2003. They were: Web mining and data engineering, Web topology and social networks, Web prefetching, ontology engineering, context-aware computing, collaborative filtering and recommendation, categorization and ranking, Web services, Web information search and retrieval, e-business and e-technology, and Web information extraction and management.

For IAT 2003, there were 13 technical sessions: agent behaviours and reinforcement learning, distributed problem solving, task-oriented agents, autonomy-oriented computing, autonomous pricing and negotiation, autonomous information services, embodies agents and agent-based system applications, multi-agent systems, modelling and methodology, knowledge discovery and data mining agents, mobil agents, agent-based simulation, and autonomous auctions.

VII. SPECIAL EVENTS

The very exciting thing for the conferences was the lobster banquet in a historic warehouse near the Halifax harbour. The reception was held in the Atrium of the Computer Science Building at Dalhousie University. Apart from the delicious food, another interesting thing is that the reception was held after the lobster banquet. The reason was that the conferences were held just several days after a hurricane, what an excellent schedule!

This year, the conference committee and chairs selected two best papers: "Dynamic Stochastic Capacity Pricing for Resource Allocation" (by Alain G. Njimolu Anyouzoa, Theo D'Hondt, D.C. Akoa, and Mamour Ba), and "Exploiting a Search Engine to Develop More Flexible Web Agents" (by Shou-de Lin and Craig A. Knoblock). We can find such reports from WIC home page (http://wi-consortium.org/) and the News and Events Session at University of Southern California's Information Sciences Institute (http://www.isi.edu).

In the prize competition, the WI 2003 and IAT 2003 conference program committees selected eight papers, respectively, and forwarded them to the conference chairs. The chairs then selected three papers for each conference. The best one was decided according to the author's presentations.

VIII. WI 2004 & IAT 2004

WI 2004 and IAT 2004 will take place in Beijing, China (home pages: http://www.maebashi-it.org/WI04 and http://www.maebashi-it.org/IAT04; also mirrored at http://www.comp.hkbu.edu.hk/WI04 and http://www.comp.hkbu.edu.hk/IAT04) during September 20-24, 2004. The conferences are sponsored and organized by IEEE Computer Society Technical Committee on Computational Intelligence (TCCI), Web Intelligence Consortium (WIC), as well as ACM-SIGART.

The conference will be held in the best season (autumn) in Beijing. It is also one of the best months to visit some famous places in Beijing, such as the Great Wall.

The important dates are as follows: Electronic submission of full papers: 4 April 2004; Notification of paper acceptance: 10 June 2004; Workshop and tutorial proposals: 10 June 2004; Camera-ready of accepted papers: 5 July 2004; Workshops/Tutorials: 20 September 2004; and Conference: 21-24 September 2004.

Dr Yuefeng Li is a Lecturer in School of Software Engineering and Data Communications at Queensland University of Technology. His research interests are Web Intelligence, Data Mining and Reasoning, and Multi-Agent Systems (Email: y2.li@qut.edu.au).

2003 AAAI Robot Competition and Exhibition

I. Overview

The Twelfth Annual AAAI Robot Competition and Exhibition was held in Acapulco, Mexico in conjunction with the 2003 Int'l Joint Conf. on Artificial Intelligence. The events included the Robot Host and Urban Search and Rescue competitions, the AAAI Robot Challenge, and the Robot Exhibition. Three-days of events were capped by the two robots participating in the Challenge giving talks and answering questions from the audience.

The purpose of the Robot Competition and Exhibition is to bring together teams from colleges, universities, and research laboratories to share experiences, compete, and demonstrate state-of-the-art robot capabilities. Of interest this year is that some of the prizes for the competition events were iRobot Roomba robot vacuum cleaners. Six years ago, at the 6th AAAI Robot Competition, one of the events challenged teams to develop a vacuum cleaning robot [1]. This year, that event came back full circle, and people can now buy robot vacuum cleaners for their homes at a price similar to that of a non-robotic vacuum. Thus, progress continues, and the highlights of this year's competition could be a window into consumer robots of the next decade.

II. Robot Host: Robots Helping People

This year the two competition events–Robot Host and Urban Search and Rescue [USR]–focused on helping people, albeit in very different situations.

For the Robot Host event, the teams had two tasks: mobile information server, and robot guide. The primary task was to interact with people and provide information to them about the conference–talks and exhibit locations, for example. The secondary task was to act as a guide for conference attendees, guiding them either to specific talk rooms or exhibition booths. Other than outlining the mission, and requiring a safety qualifying round, the task contained no specific restrictions or constraints on the environment or the robots. The robots performed their duties in the middle of the main lobby of the conference center, navigating around people and natural obstacles.

Fig. 1. University of Rochester's robot Mabel in the 2003 Robot Host Competition.

This year two teams participated: the University of Rochester and Stony Brook University. Both incorporated speech recognition, a visual interface, vision capability, and synthetic speech on a mobile platform. Figure 1 shows one of the robots interacting with conference attendees.

First place this year went to the University of Rochester, and second place went to the State University of New York, Stony Brook. Both the first and second place teams won an iRobot Roomba and a $1000 certificate towards the purchase of an ActivMedia robot.

III. Urban Search and Rescue

The goal of the IJCAI/AAAI Rescue Robot Competition is to increase awareness of the challenges involved in search and rescue applications, provide objective evaluation of robotic implementations in representative environments, and promote collaboration between researchers. It requires robots to demonstrate their capabilities in mobility, sensory perception, planning, mapping, and practical operator interfaces, while searching for simulated victims in a maze of increasingly difficult obstacles.

The competition encourages participants to contribute to the field of urban search and rescue (USAR) robotics and provides the competitors with a sense of what a real USAR situation involves. Six teams competed this year: Idaho National Engineering and Environmental Laboratory [INEEL] (USA), Swarthmore College (USA), University of Manitoba (Canada), University of New Orleans (USA), University of Rochester (USA), and Utah State University (USA).

Two place awards and a technical award were presented at this year's competition. The place awards are based solely on the teams' performances during the competition missions. The technical award is given to the team exhibiting novel artificial intelligence applications and technical innovations.

INEEL won the first place award and Swarthmore College won the second place award. These two teams had the highest cumulative scores from four (of five total) missions. Both teams performed well, but INEEL was able to find victims in both the yellow arena and the orange arena, which contains more significant obstacles, even negotiating the ramp at one point to find a number of victims on the elevated floor. They also showed 100% reliability by scoring points in every mission. Swarthmore attempted the more advanced arenas but their robots were not able to move over the uneven flooring and score points, which hurt their overall reliability (60%). By staying mainly in the yellow arena with its reduced arena weighting, and avoiding costly penalties, Swarthmore's high score was 12.5, with an average score of 6.1.

The University of New Orleans earned a technical award for their innovative

attempt at collaborative mapping. However, their reliance on multiple operators to control several robots generally lowered their overall scores. The University of Rochester also performed well during particular missions. Meanwhile, the University of Manitoba and the Utah State University demonstrated fully autonomous custom-made robots with varying degrees of success in negotiating the simplest arena, but didn't attempt to produce maps of the arenas with victim identified–a key element in scoring.

IV. The Robot Challenge

The Robot Challenge, first dreamed up at the 1998 AAAI Robot Competition, entered its fifth year. The Challenge is for a robot to successfully attend the National Conference, which includes finding the registration desk, registering for the conference, navigating to a talk venue, giving a talk, and answering questions. Other possible tasks include acting as a conference volunteer, and talking with conference attendees during coffee breaks.

This year, for the first time, two teams–the GRACE team and Lewis, from Washington University, St. Louis–completed the main Challenge tasks. The GRACE team consisted of Carnegie Mellon University, the Naval Research Laboratory, Metrica Labs, Northwestern University, and Swarthmore College. Both teams were successful at getting their robots to a faux registration booth, registering, going to the talk venue and giving a talk. Each of the aspects of the challenge were addressed with varying levels of success. None of the robots could attempt the trek to the real registration booth as it was on the second floor, and, more importantly, the convention center had no elevators. The GRACE team actually brought two robots, GRACE and George, both of which independently undertook the challenge, demonstrating slightly different capabilities. Figure 2 shows both GRACE and George giving their talk at the end of the Challenge event.

Fig. 2. GRACE and George giving their talk as part of the 2003 Robot Challenge.

Washington University received the title of Challenge Champion for 2003, and an iRobot Roomba, and the GRACE team received the "Grace Under Fire" award for success in spite of tremendous challenges and hardware difficulties. The GRACE team also received a technical award for integration, integration, integration.

This year the Ben Wegbreit Award for Integration of AI Technologies, which includes a $1000 prize, went to the Washington University for Lewis' smooth run in the Challenge Event.

V. Summary

The Twelfth AAAI Robot Competition and Exhibition continued the tradition of demonstrating state-of-the-art research in robotics. Many of the improvements this year were largely invisible to those watching the robots, but improvements in integrating systems and vision capabilities will eventually make the robots more robust, more adaptable, and better able to succeed in their challenging tasks. Without progress in these invisible areas, progress in the more visible robot capabilities will be slow.

The challenge of making robots that can navigate and successfully complete tasks in the real world was the focus of all the events this year, and that is a great advance over the events of a decade ago that required special arenas and brightly colored objects. Where are we going next?

In 2004, it will be the AAAI National Conference in San Jose. Bill Smart and Shiela Tejada will be co-chairing the event. We invite everyone in robotics to participate and demonstrate their current research. For more information, see http://palantir.swarthmore.edu/aaai04.

References

[1] R. Arkin. The 1997 aaai mobile robot competition and exhibition. *AI Magazine*, 19(3):13–17, 1998.

Proteus, a Grid based Problem Solving Environment for Bioinformatics: Architecture and Experiments

Mario Cannataro[1], Carmela Comito[2], Filippo Lo Schiavo[1], and Pierangelo Veltri[1]

***Abstract*— Bioinformatics can be considered as a bridge between life science and computer science. Biology requires high and large computing power to performance biological applications and to access huge number of distributed and (often) heterogeneous databases. Computer scientists and database communities have expertises in high performance algorithms computation and in data management. Considering bioinformatics requirements, in this paper we present PROTEUS, a Grid-based Problem Solving Environment for bioinformatics applications. PROTEUS uses ontology to enhance composition of bioinformatics applications. Architecture and preliminary experimental results are reported.**

***Index Terms*— Bioinformatics, Grid, Ontology, Problem Solving Environment (PSE).**

I. INTRODUCTION

RESEARCH in biological and medical areas (also known as *biomedicine*), requires high performance computing power and sophisticated software tools to treat the increasing amount of data derived by always more accurate experiments in biomedicine. The emerging bioinformatics area involves an increasing number of computer scientists studying new algorithms and designing powerful computational platforms to bring computer science in biomedical research. According to [5], Bioinformatics can thus be considered as *a bridge between life science and computer science*.

Biologists and computer scientists are working in designing data structure and in implementing software tools to support biomedicine in decoding the entire human genetic information sequencing (i.e. DNA), also known as *genome.* Even if many issues are still unsolved, (i.e., such as heterogeneous data sets integration and metadata definitions), the attention is now focused on new topics related to genomics. Today, the new challenge is studying the *proteome*, i.e. the set of *proteins* encoded by the genome, to define models representing and analyzing the structure of the proteins contained in each cell, and (eventually) to prevent and cure any possible cell-mutation generating human diseases such that producing cancer-hill cells [15].

Proteins characteristics can be simply represented by strings sequences encoding *amino acids* that are the basic building blocks composing proteins. Nevertheless, the high number of possible combinations of amino acids composing proteins, as well as the huge number of possible cell-mutation, require a huge effort in designing software and environments able to treat generic micro-biology problems. Moreover, proteins present spatial (i.e., three dimensional) structure that (partially) depends on amino acids composition: 3D protein structure predictions and folding are other important issues interesting medicine and drug discovery. Pattern matching algorithms and tools have to be combined with high performance multidimensional and imaging software tools to analyze and eventually prevent proteins behaviors.

Proteomics data sets in applications can be produced by experiments, or can be extracted from publicly available databases as those produced and maintained by research community: e.g. Protein Data Bank (PDB) [22], the SWISS-PROT protein database [29], the GenBank DNA sequences collections [21]. Optimized data models are required to represent protein structures as well as ”ad hoc” software tools are necessary to integrate and combine data obtained from experiments or from querying protein database and to extract information understandable by biomedical researchers. Nevertheless, heterogeneity both in data format and database access policy justify the interest of bioinformaticians for (biomedical-) data models, specialized software for protein searching and combinations, as well as data mining tools for information extraction from datasets. On the other hand, data and software distribution requires high performance computational platforms to execute distributed bioinformatics applications.

Computational Grids (or simply Grid) are geographically distributed environments for high performance computation [27]. In a Grid environment is possible to manage heterogeneous and independent computational resources offering powerful services able to manage huge volumes of data [28]. Grid community [14] recognized both bioinformatics and postgenomic as an opportunity for distributed high performance computing and collaboration applications. The Life Science Grid Research Group [24] established under the Global Grid Forum, believes bioinformatics requirements can be fitted and satisfied by Grid services and standards, and is interested in what new services should Grids provide to bioinformatics applications. In particular, given the number of applications requiring ability in reading large and heterogeneous datasets (e.g. protein databases) or in creating new datasets (e.g. mass spectrometry proteomic data [15]), a large number of biologist projects are investing in Grid environments as well as many computer scientists are investing in developing Bioinformatics applications on Grid (also known as *BioGrids*). E.g., the Asia Pacific BioGRID [4] is attempting to build a customized, self-installing version of the Globus Toolkit [32], a diffused environment for designing and managing Grid, comprising well tested installation scripts, avoiding dealing with Globus details. In the European Community Grid Project [31], whose

[1]University of Magna Graecia of Catanzaro, Italy surname@unicz.it
[2]University of Calabria, Italy surname@si.deis.unical.it

aim is funding Grid applications in selected scientific and industrial communities, the Bio-GRID work group is developing an access portal for biomolecular modeling resources [18]. The project develops various interfaces for biomolecular applications and databases that will allow chemists and biologists to submit work to high performance computing facilities, hiding Grid programming details. Finally, myGrid is a large United Kingdom e-Science project to develop open source data-intensive bioinformatics application on the Grid [30]. The emphasis is on data integration, workflow, personalization and provenance. Database integration is obtained both by dynamic distributed query processing, and by creating virtual databases through federations of local databases.

In this paper we consider a world where biomedical software modules and data can be detected and composed to define problem-dependent applications. We wish to provide an environment allowing biomedical researchers to search and compose bioinformatics software modules for solving biomedical problems. We focus on semantic modelling of the goals and requirements of bioinformatics applications using *ontologies*, and we employ tools for designing, scheduling and controlling bioinformatics applications. Such ideas are combined together using the Problem Solving Environment (PSE) software development approach [23]. A Problem Solving Environment is an integrated computing environment for composing, compiling, and running applications in a specific area [34], leaving the user free to work on application and not on software programming [9]. Grid-based PSEs are related to distributed and parallel computing and leverages basic Grid services and functionalities. E.g., the KNOWLEDGE GRID [13], based on the Globus Toolkit [32], is a Grid-based problem solving environment providing a visual environment (i.e., called VEGA) to design and execute distributed data mining applications on the Grid [12].

We present PROTEUS , a software architecture allowing to build and execute bioinformatics applications on Computational Grids [27]. The proposed system is a Grid-based Problem Solving Environment (PSE) for bioinformatics applications. We define an ontology-based methodology to describe bioinformatics applications as distributed workflows of software components. The architecture and first implementation of PROTEUS based on the KNOWLEDGE GRID [13], are presented. Also, we present use of PROTEUS to implement an application of human protein clustering. A preliminary version of this work can be found in [11].

The paper is organized as follows. Section II report biological data characterisics and environment requirements for bioinformatics applications. Section III presents a first implementation of PROTEUS based on KNOLEDGE GRID, reporting PROTEUS architecture and software modules. Section IV presents the ontology based processing to design bioinformatics applications with PROTEUS . Section V reports experiences on designing and running a simple case study of clustering human proteins using PROTEUS , and finally Section VI concludes the paper and outlines future works.

II. BIOINFORMATICS ISSUES

Bioinformatics involves the design and development of advanced algorithms and computational platforms to solve problems in biomedicine. Applications deal with biological data obtained by experiments, or by querying heterogeneous and distributed databases. Methods for acquiring, storing, retrieving and analyzing such data are also necessary. In this section we sketch some characteristics of biological data, with particular emphasis to proteins data, and present some available biological databases. We then discuss about requirements of biological applications.

A. Biological Data and Databases

Handling biological data has to deal with exponentially growing sets of highly inter-related data rapidly evolving in type and contents. Designers of biological databases and querying engines have to consider some data management issues well known to database community. Biological data are often obtained combining data produced by experiments, or extracted by common databases. Data are thus often heterogeneous both in structure and content. Combining data coming from different sources requires human expertise to interact with different data format and query engines: e.g., data can be reported in text files or in relational tables or in HTML documents, while query interfaces may be textual or graphical (e.g., SQL-like, or query by example). Moreover, databases need to react to frequent data update: new data emerge regularly from new experimental results, thus databases must be updated and refreshed accordingly.

Biological data are often represented as string sequences and described using natural language. Most of the existing biological data represent data as flat file structured as a set of field/value pairs, weakly interconnected with indexing systems such as the Sequence Retrieval System (SRS) [7] (see below). Even 3D protein structures are often represented as raster images which content cannot be captured by any automatic query engine (e.g., based on similarity image matching), and need human interaction.

Biological data in bioinformatics comprise sequences of nucleotides (i.e., DNA) and sequences of amino acids (i.e., proteins). There are four different type of nucleotides, distinguished by the four bases: adenine (A), cytosine (C), guanine (G) and thymine (T), thus a single strand of DNA can be represented as a string composed of the four letters: A, C, G, T. A *triple* of nucleotides encodes an amino acid, while amino acids form proteins. Although there are $4^3 = 64$ different triples of nucleotides, in nature there exists only 20 different amino acids that can compose a protein. Each protein can be thus represented as a string composed by a 20-character alphabet, where each character represents an amino acid (e.g., G for glycine, A for alanine, V for valine, etc.). Since nucleotides and amino acids are represented with alphabet letters, the natural representation of a biological element (genes sequence or proteins sequence) is a string of characters. Data models are then based on string structures. To represent both nucleotides and amino acid chains, flat non-structured files as well as files enriched by field/value pairs structures can be used.

Structured data models (e.g., object oriented or relational [33]) are useful for data retrieval. Nevertheless, most of the useful biological databases are populated gathering data from different and often heterogeneous sources each providing its own database structure and query search engine. The data integration topic and the effort of defining uniform data model and query engine is another important issue that has been interesting computer scientists, for all kind of data. E.g., XML (eXtensible Mark up Language), the language for data exchange on the Web, has been attracting bioinformaticians. Thanks to its semi-structured nature [1], in XML it is possible to represent both data and (when present) structure in a single paradigm. XML query engine can filter data using their structure (if presents) and finally extract data using key-word based queries. Where still documents exists in different databases, XML "abstract" documents [2] can be used to integrate heterogeneous data sources or as exchange mechanism (data mediator) between different databases. Moreover, *ontologies* can also be used for data integration. An Ontology is a system to share standard and unambiguous information about an observed domain. Ontologies are used to realize semantic tools to retrieve and analyze biological data coming from different data sources, using a given set of similar terminology. As we will see, PROTEUS utilizes ontologies to leverage users from knowing exactly all applications specifications and data locations and structures.

The existent biological databases contain protein and DNA sequences, 3D structures of protein sequences (i.e., images and description) and relationships between different sequences. They are mainly public available through the Web and offer database query interfaces and information retrieval tool to catch data coming from different databases. Most of them are produced and maintained by the research community; e.g., European Molecular Biology Laboratory (EMBL) [29] and American National Center for Biotechnology Information (NCBI) [21] give access to nucleotide and protein sequence databases. The former gives access to SWISS-PROT, a database of protein sequences obtained from translations of DNA sequences or collected from the scientific literature or applications. The latter maintains GenBank, a collection of all known DNA sequences. Moreover, a useful protein database is the Protein Data Bank (PDB) [22], that is a database of 3D-coordinates of macromolecular structures. Moreover two Web publicly available databases are the *Sequence Retrieval System (SRS)* and the *Entrez system*. SRS [7] is a Web-based retrieval system for biological data. It accesses to different available web databases and builds an index of URLs to integrate them. The index is used as a database view on different databases, providing a single interface allowing users to formulate queries on different databases. SRS provides the user with transparency from communication with sources (i.e. location, connection protocols and query language), but it does not provide guidance about source relevance for a given query, and no data integration is provided in the query results. Entrez [20] is the NCBI text-based search interface on the major biological databases (e.g., nucleotide database, protein sequence databases, structure databases, etc). Query results are obtained by combining data coming from different databases, using a proximity score grouping sequences and references based on similarity characteristics. Queries can be built using a "query by example" based interface.

B. Biological Application Requirements

Novel Bioinformatics applications and in particular Proteomics ones, involve different data sets either produced in a given experiment, or available as public databases or different software tools and algorithms. Applications deal with (i) data sources, i.e. local and/or remote databases, and (ii) specialized services, algorithms and software components: e.g., pattern matching algorithms to match protein sequences in protein databases. From a computational point of view, it is necessary consider that Bioinformatics applications:

- are naturally distributed, due to the high number of involved data sets;
- require high computing power, due to the large size of data sets and the complexity of basic computations;
- access heterogeneous and distributed data, e.g. answering queries may require accessing several databases;
- need secure software infrastructures to manage private data.

Computational requirements have to deal with the sharing of computational resources, the integrated access to biological databases, as well as an efficient, large-scale data movement and replication. High performance requirements and distribution of software and data in Bioinformatics created a great interests in the Grid community.

Finally, software tools, data sources and Grid computational nodes, can be glued by using knowledge representation and management techniques. Defining semantic representation of data is one of the last challenge of the computer science community [26]. A possibility is using *ontologies* to build Knowledge Bases modeling knowledge about bioinformatics resources and processes. Basic retrieval techniques, as well as querying tools, can be used to extract knowledge by ontology databases.

III. PROTEUS: ARCHITECTURE AND SOFTWARE MODULES

This Section presents PROTEUS , a Grid-based Problem Solving Environment for composing, compiling, and running Bioinformatics applications on the Grid. To fulfill bioinformatics application requirements and to help biologists in their applications, PROTEUS introduces semantic modeling of Bioinformatics processes and resources, following an emergent trend in Semantic Grids and Knowledge Grids.

To fulfill bioinformatics application requirements, we propose a framework based on:

- Grids, with their security, distribution, service orientation, and computational power;
- Problem Solving Environment approach, useful to define, describe and execute (i.e. control) such applications;
- Ontologies, Web (Grid) Services, and Workflows technologies, at an inner level, to describe, respectively, the semantics of data sources, software components with their interfaces, and performances and bioinformatics tasks.

With the first item PROTEUS satisfies the high powerful computational requirements of bioinformatics applications. Moreover Grid environment is composed of distributed computational nodes, and fulfill the distributed nature of bioinformatics applications and data management.

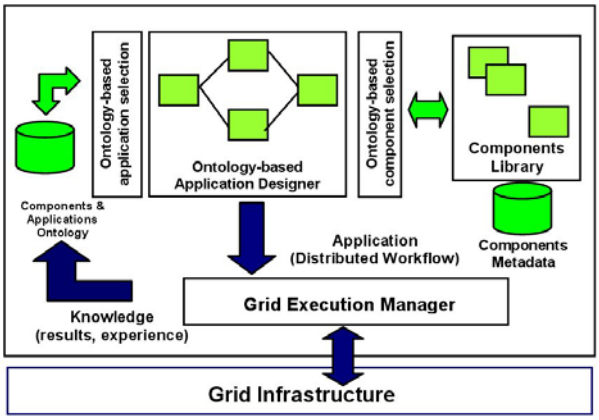


Fig. 1. PROTEUS General Architecture

PSE provide a dictionary of data and tools locations allowing users to build their applications disposing of all necessary tools. We imagine a world where biologists want to access a single tools and data virtual store where they may compose their applications. In particular, PROTEUS modules uses and combines open source bioinformatics software, and public-available biological databases. Private databases (i.e. databases accessible with registration via Web) can be also considered. Drawback in using open source packages (i.e., often defined in research environments) and in providing software tools, is that users have to know the nature of their data (i.e. their semantic) and details of software components, while they have to concentrate on biological domain and attended results. Moreover, the access to such components is often available by command line only. To overcome such problems, PROTEUS simplifies the use of software tools by adding metadata to available software and modelling applications through *ontology*. Ontologies are used to build PROTEUS Knowledge Base, modeling knowledge about bioinformatics resources and processes.

PROTEUS can be used to assist users in:

- formulating problems, allowing to compare different available applications (and choosing among them) to solve a given problem, or to define a new application as composition of available software components;
- running an application on the Grid, using the resources available in a given moment thus leveraging the Grid scheduling and load balancing services;
- viewing and analyzing results, by using high level graphic libraries, steering interfaces (that allow to interactively change the way a computation is conducted), and accessing the past history of executions, i.e. the past results, that form a knowledge base.

In the following, we present the PROTEUS overall architecture, while the next subsection describes a first implementation of the system and its main software modules.

A. Architecture

A main goal of PROTEUS is to leverage existing software easing the user work by: (i) adding metadata to software, (ii) modeling application through ontology, (iii) offering pre-packaged bioinformatics applications in different fields (e.g. proteomics), (iv) using the computational power of Grids. PROTEUS extends the basic PSE architecture and is based on the KNOWLEDGE GRID approach [13]. Main components of PROTEUS (see Figure 1) are:

- **Metadata repository** about software components and data sources (i.e. software tools, databases and data sources). It contains information about specific installed resources.
- **Ontologies**. We have two kinds of ontology in our system: a domain ontology and an application ontology. The domain ontology describes and classifies biological concepts and their use in bioinformatics as well as bioinformatics resources spanning from software tools (e.g. EMBOSS) to data sources (biological databases such as SWISS-PROT). The application ontology describes and classifies main bioinformatics applications, represented as workflows. Moreover it contains information about application's results and comments about user experience. Both ontologies contain references to data in metadata repository.
- **Ontology-based application designer**. An ontology-based assistant will either suggest the user the available applications for a given bioinformatics problem/task, or will guide the application design through a concept-based search of basic components (software and databases) into the knowledge base. Selected software components will be composed as workflows through graphic facilities.
- **Workflow-based Grid execution manager**. Graphic representations of applications are translated into Grid execution scripts for Grid submission, execution and management.

Ontologies and metadata are organized in a hierarchical schema: at the top layer ontologies are used to model the rationale of bioinformatics applications and software components, whereas at the bottom layer specific metadata about available (i.e. installed) bioinformatics software and data sources are provided. Ontology guides the user in the choice of the available software components or complete applications on the basis of her/his requirements (ontology-based application design) [8], whereas the low layer metadata will be used to really access software tools and databases, providing information like installed version, format of input and output data, parameters, constraints on execution, etc. When the application requires an installed tool, i.e. the ontology-based application design module issues a (resource) request, an ontology-based match-making algorithm finds the best match between the request and the available resources.

The ontology will be updated whenever new software tools or data sources are added to the system, or new applications are developed (i.e. designed through composition of software components). This enables the realization of a Knowledge Base of applications/results, which is enriched whenever new applications are developed or new results are obtained. Thus, new

users may gain knowledge about pre-existing experiments.

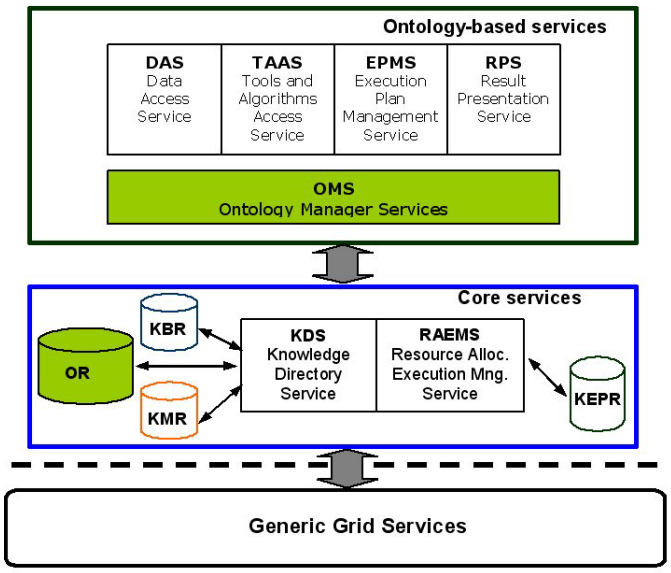


Fig. 2. Software Modules of PROTEUS

B. A First Implementation

The current implementation of PROTEUS is based on the KNOWLEDGE GRID, a joint research project of ICAR-CNR, University of Calabria, and University of Catanzaro, aiming at the development of an environment for geographically distributed high-performance knowledge discovery applications [13]. PROTEUS system modules are described in Figure 2. The ontology modules represent the main innovation with respect to the KNOWLEDGE GRID. It allows to describe bioinformatics resources (i.e. the Ontology Repository) offering new ontology-based services (i.e. the Ontology Management Services) to search and find the most appropriate software components needed to solve a bioinformatics task. We are working on PROTEUS implementation based on a new architecture specialized to support the complex workflows of bioinformatics applications on Grid [10].

Similarly to the KNOWLEDGE GRID, PROTEUS is built as a bag of services divided in two layers: the Core services that interface the basic Grid middleware and the Ontology-based services that interface the user by offering a set of services for the design and execution of bioinformatics applications.

The Core services allow the submission, execution, and control of a distributed computation over the Grid. Main services include the management of ontologies and metadata describing features of software components, applications and data sources. Moreover, this layer coordinates the application execution by attempting to fulfill the application requirements and the available grid resources. The Core services comprise:

- The Knowledge Directory Service (KDS) offers a uniform access to ontologies and metadata stored in the following repositories: resource ontology (OR), resource metadata (KMR), execution plans, i.e., application workflows (KEPR), and results of bioinformatics applications (KBR). The ontology is represented by a DAML+OIL [16] document stored in the Ontology Repository (OR), whereas metadata are represented as XML documents.
- The Resource Allocation and Execution Management Service (RAEMS) is used to find the best mapping between an execution plan and available Grid resources, with the goal of satisfying the application requirements and Grid constraints.

The Ontology-based services allow to compose, validate, and execute a parallel and distributed computation, and to store and analyze its results. The Ontology-based services comprise:

- The Ontology Management Services (OMS) offer a graphical tool for the ontology browsing, a set of utilities for the updating of the ontology, and a set of APIs for accessing and querying the ontology by means of a set of object-oriented abstractions of ontology elements. These services are used to enhance the following services.
- The Data Access Service (DAS) allows to search, select, extract, transform and delivery data to be analyzed.
- The Tools and Algorithms Access Service (TAAS) allows to search and select bioinformatics tools and algorithms.
- The Execution Plan Management Service (EPMS) is a semi-automatic tool that takes data and programs selected by the user and generates a set of different, possible execution plans (workflows) that meet user, data and algorithms requirements and constraints. Execution plans are stored into the KEPR.
- The Results Presentation Service (RPS) allows to visualize the results produced by a bioinformatics applications. The result metadata are stored in the KMR and managed by the KDS.

The design and execution of an application using PROTEUS run through the following steps:

1) Ontology-based resources selection. The search, location and selection of the resources to be used in the applications are executed by using the DAS and TAAS tools that invoke the OMS. Using the OMS the design process is composed of two phases:
   - Software tools and data sources selection. Browsing and searching the ontology allow a user to locate the more appropriate component to be used in a certain phase of the application.
   - XML metadata access. The ontology gives the URLs of all instances of the selected resources available on the grid nodes, i.e. the URLs of the relevant metadata files stored in the KMRs.
2) Visual application composition, through a graphical model that represents the involved resources and their relations.
3) Abstract execution plan generation, corresponding to the graphical model of the application. The plan is generated by using the EPMS services and then is stored into the KEPR.
4) Application execution on the Grid. The abstract execution plan is translated into a source Globus RSL (Resource Specification Language) script by the RAEMS module, then this script is submitted to the GRAM (Globus Resource Allocation Manager) service.

5) Results visualization and storing, by using the RPS services.

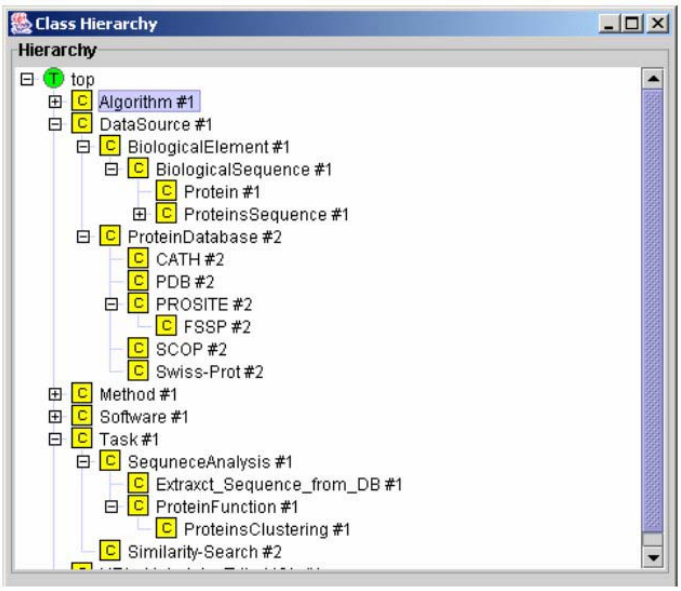


Fig. 3. Some Taxonomies of the Bioinformatics Ontology

IV. Ontologies in PROTEUS

Ontologies are used in PROTEUS to describe the semantics of the components and data resources involved in applications. In this section we describe a first Bioinformatics Ontology, and its management using the Ontology Management Services.

A. An Ontolgy on Bioinformatics Domain

Currently PROTEUS presents an ontology on bioinformatics domain that tries to integrate different aspects of bioinformatics, including computational biology, molecular biology and computer science. In such ontology we classify the following bioinformatics resources:

1) biological data sources, such as protein databases (e.g., SwissProt, PDB);
2) bioinformatics software components, such as tools for retrieving and managing biological data (e.g., SRS, Entrez, BLAST, EMBOSS);
3) bioinformatics processes/tasks (e.g. sequence alignment, similarity search, etc.).

The modelling of the above cited bioinformatics resources, has been made on the basis of classification parameters that will guide users in the composition of the application and in the choosing of the most suitable resources to use.

Biological data sources have been classified on the basis of the following features:

- the kind of biological data (e.g., proteins, genes, DNA);
- the format in which the data is stored (e.g., sequence, BLAST proteins sequence);
- the type of data source (e.g., flat file, relational database, etc);
- the annotations specifying the biological attributes of a database element.

Bioinformatics processes and software components have been organized in the ontological model on the basis of the following parameters:

- the *task* performed by the software components; that is the typology of the bioinformatics process (e.g., sequence analysis, secondary structure prediction, etc);
- the steps composing the task and the order in which the steps should be executed;
- the methodology (*method*) that the software uses to perform a bioinformatics task;
- the *algorithm* implemented by the software;
- the *data source* on which the software works on;
- the kind of *output* produced by the software;
- the *software* components used to perform a task (e.g. BLAST, EMBOSS, etc.).

Taxonomies that specialize each of those classification parameters have been partially implemented. Every taxonomy specializes the concept of interest using two kinds of relationships through which simple/multiple inheritance could be applied: the first kind of relationship is the *specialisation/generalisation* ("is-a") relationship that specialises/generalises general/specific concepts in more specific/general ones; and the *part of/has part* relationship that defines a partition as subclass of a class. Figure 3 shows some taxonomies of the ontology by using the OilEd ontology editor [6].

Thus we have organized our ontological model in such a way to have a large number of small local taxonomies that may be linked together via non-taxonomic relations. As an example, since every software performs a task, the *Software* taxonomy is linked to the *Task* taxonomy through the *PerformsTask* relation. The ontology can be explored by choosing one of the previous classification parameters. For example, exploring the *Task* taxonomy it is possible to determine for a given task what are the available algorithms performing it and then which software implements the chosen algorithm. Moreover it is possible to find the data sources and the biological elements involved in that task. On the other hand, exploring the *Algorithm* taxonomy it is possible to find out the biological function behind an algorithm, the software implementing it, the kind of data source on which it works.

B. The Ontology Management Services

PROTEUS offers ontology-based services and as such it needs a means through which manipulate and access ontologies stored in the *Ontology Repository* (see Figure 2). To this aim we introduced in the architecture shown in Figure 2 the *Ontology Management Services* (OMS). The OMS provides a set of high-level services for managing ontologies such as utilities for browsing and querying them. These utilities are supplied both as graphical tools as well as a set of Java APIs.

The API implementation is realized for accessing and querying the ontology: the API will provide a set of object-oriented abstractions of ontology elements such as Concept, Relation, Properties, and Instance objects providing query facilities.

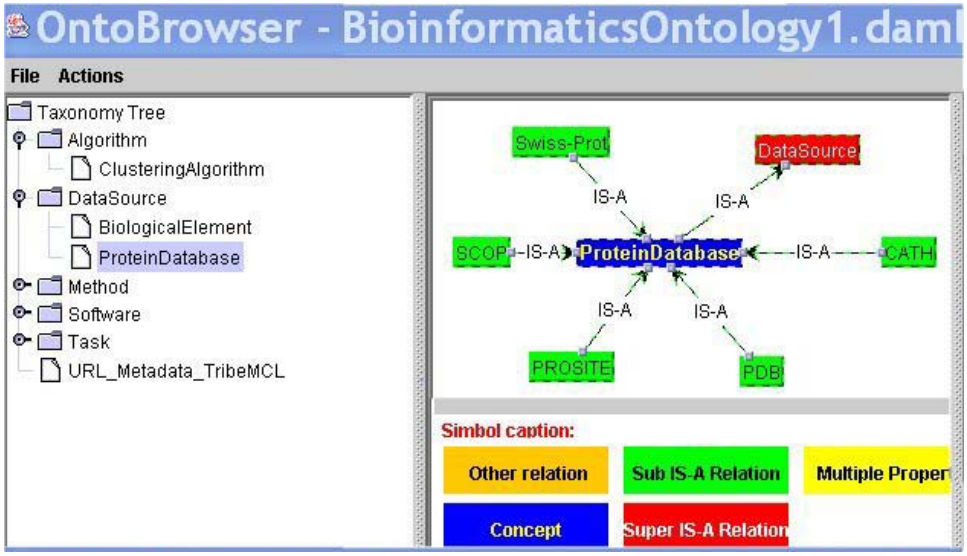


Fig. 5. Snapshot of the Ontology Browser

The graphical tool provides a combined search and browse facility over the ontology:

- *Ontology querying*. Through the ontology-based search engine offered by the OMS, user can find detailed information about domain resources modeled in the ontology. The result set is accurate, because the semantic of the target terms is indicated by concepts from the underlying ontology. Our ontology-based search engine supports several kinds of simple inference that can serve to broaden queries including equivalence (to restate queries that differ only in form), inversion, generalization, and specialization to find matches or more general or more specific classes and relations. If the result set of a query is empty, the user can at least find objects that partially satisfy the query: some classes can be replaced by their superclasses or subclasses. Both narrowing and broadening the scope of the query are possible due to the ontological nature of the domain description.
- *Ontology browsing*. The ontology browser is a navigation facility that presents an overview of the whole data set: it shows the classes, their relations and instances. The browser gradually presents deeper levels of the ontology: the user starts at the top of the ontology and can navigate towards more specific topics by clicking the classes of interest (diving into the information).

Since we have implemented the ontology in the DAML+OIL ontology language, the services offered by the OMS allow support only for DAML+OIL [16] encoded ontologies. At this time we have implemented a graphical tool for the browsing of ontologies (see Figure 5); using such tool the user browses the ontology choosing one of the input point (left panel of the frame) representing the taxonomies of the ontology and navigates visiting the sub tree topics until reaching a concept of interest. The concept of interest is shown in the middle of the right panel of the frame and related concepts are displayed around it. The ontology may be browsed by promoting any of the related concepts to be the central concept. The new central concept is then linked to all its related concepts.

V. A CASE STUDY: CLUSTERING OF HUMAN PROTEINS

This Section presents some first experimental results obtained implementing a simple bioinformatics application. We first present the overall application workflow, and then we discuss the design of such application. Currently, the application is first designed by using the Ontology Management Services described in the previous section, and then the selected resources are composed into a Data Flow Diagram by using VEGA (*Visual Environment for Grid Applications*) [12], the KNOWLEDGE GRID user interface.

Protein function prediction uses database searches to find proteins similar to a new protein, thus inferring the protein function. This method is generalized by protein clustering, where databases of proteins are organized into homogeneous families to capture protein similarity. We implemented a simple application for the clustering of human proteins sequences using the TribeMCL method [3]. TribeMCL is a clustering method through which it is possible to cluster correlated proteins into groups termed ”protein family”. This clustering is achieved by analysing similarity patterns between proteins in a given dataset, and using these patterns to assign proteins into related groups. In many cases, proteins in the same protein family will have similar functional properties. TribeMCL uses the Markov Clustering (MCL) algorithm [17].

We organized the application (see Figure 4) into four phases: the *Data Selection phase* extracts sequences from the database, the *Data Preprocessing phase* prepares the selected data to the clustering operation, the *Clustering phase* performs the Markov Clustering algorithm to obtain a set of protein clusters, and finally the *Results Visualization phase* displays the obtained results.

In the Data Selection phase all the human protein sequences are extracted from the Swiss-Prot database using the `seqret` program of the EMBOSS suite. EMBOSS is a package

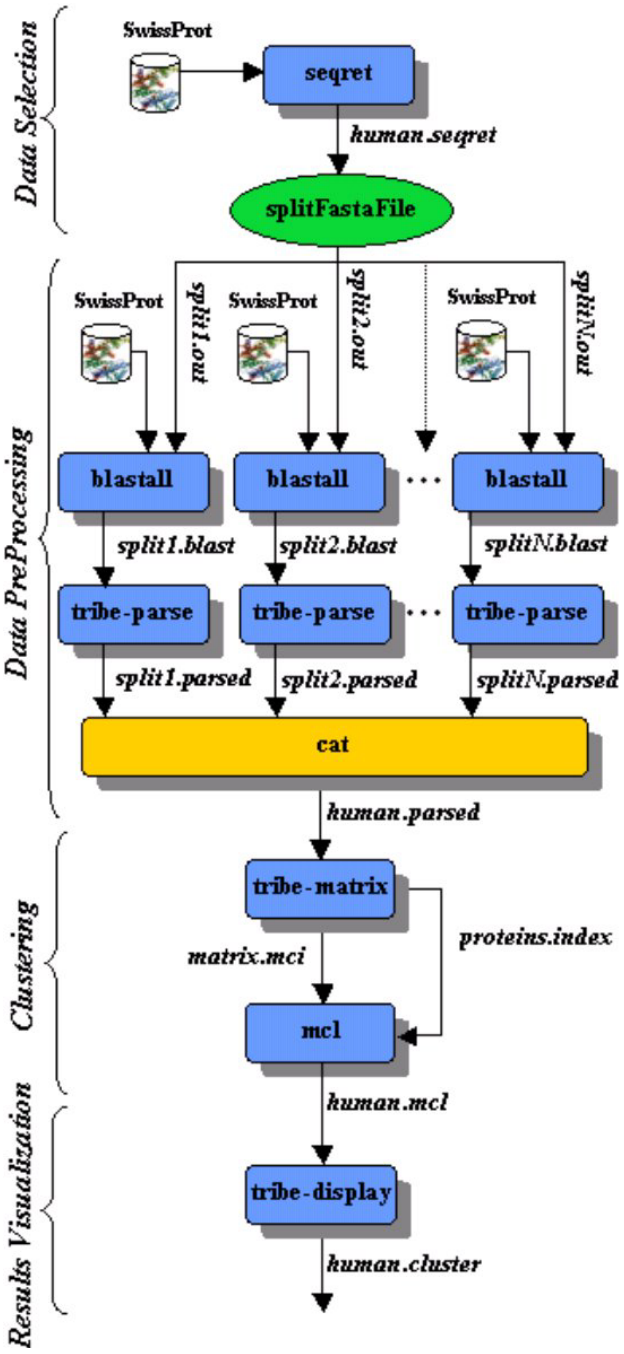


Fig. 4. Human Protein Clustering Workflow

of high-quality Open Source software for sequence analysis [25]. `seqret` is a program for extracting sequences from databases: in our application this program reads sequences from the database and then write them to a file.

TribeMCL needs a BLAST comparison on its input data. BLAST is a *similarity search* tool based on string matching algorithm [19]. Given a string it finds string sequences or sub-sequences matching with some of the proteins in a given database (*alignment*). BLAST carries out *local alignments* between sequences or between a sequence and protein database. Local alignment algorithms look for protein string matching between protein subsequences. It ranks the subsequence results using an expectation value (e-value). Given a sequence, it is able to return the probability of a particular alignment to occur. E.g., an e-value equal to zero means that the probability for a given alignment to occur by chance is zero. In particular, TribeMCL uses an *all against all* BLAST comparison as input to the clustering process, thus once the protein sequences have been extracted from the database, a BLAST computation has to be performed.

The Data Preprocessing phase comprises the following steps. To speed up the similarity search activity we partitioned the `seqret` output in three smaller files; in this way three BLAST computations can be run in parallel. The obtained raw NCBI BLAST outputs are converted in the format required to create the Markov Matrix used in the clustering phase by TribeMCL. The parsing has been executed by using `tribe-parse` program. Finally, the files obtained in the `tribe-parse` steps are concatenated by using the `cat` program.

In the Clustering phase, the Markov Matrix is built by using the `tribe-matrix` program that produces the `matrix.mci` and `proteins.index` files. Then the clustering program `mcl` is executed using the file `matrix.mci`.

Finally, in the Results Visualization phase the clustered data are arranged in an opportune visualization format.

A. Application Development on PROTEUS

In VEGA the resources are just described by basic metadata about technical details, and it does not provide any semantic modelling. Moreover, users have to browse metadata on each Grid node to search and select the resources needed in an application.

In order to overcome these limitations, we have supplied the VEGA environment with an ontological modelling of the bioinformatics resources and an ontologies mananging tool.

The proposed Ontology Management Services can be used both to enhance the application formulation and design, and to help users to select and configure available resources (software components and data sources).

The first step in the development of bioinformatics applications on PROTEUS is the *Ontology-based resource selection* in which the user browses the ontology locating the more appropriate components to use in the application. Next, the selected resources are composed through the graphical model of VEGA (*Visual application composition*).

The application workflow shown in Figure 4 has been modelled as a set of VEGA workspaces [12]. We briefly remind that a computation in VEGA is organized in *workspaces*. The jobs of a given workspace are executed concurrently; whereas workspaces are executed sequentially. The implementation of our application required the development of 13 workspaces grouped into the four different phases of the application: *Data Selection*, *Data Preprocessing*, *Clustering* and *Results Visualization*.

Consider the following scenario: a PROTEUS user logged on the host `minos` wants to define and execute the clustering of human proteins. He/she only knows that needs a protein sequences database from which to retrieve the sequences and a software tool performing the clustering process. Moreover,

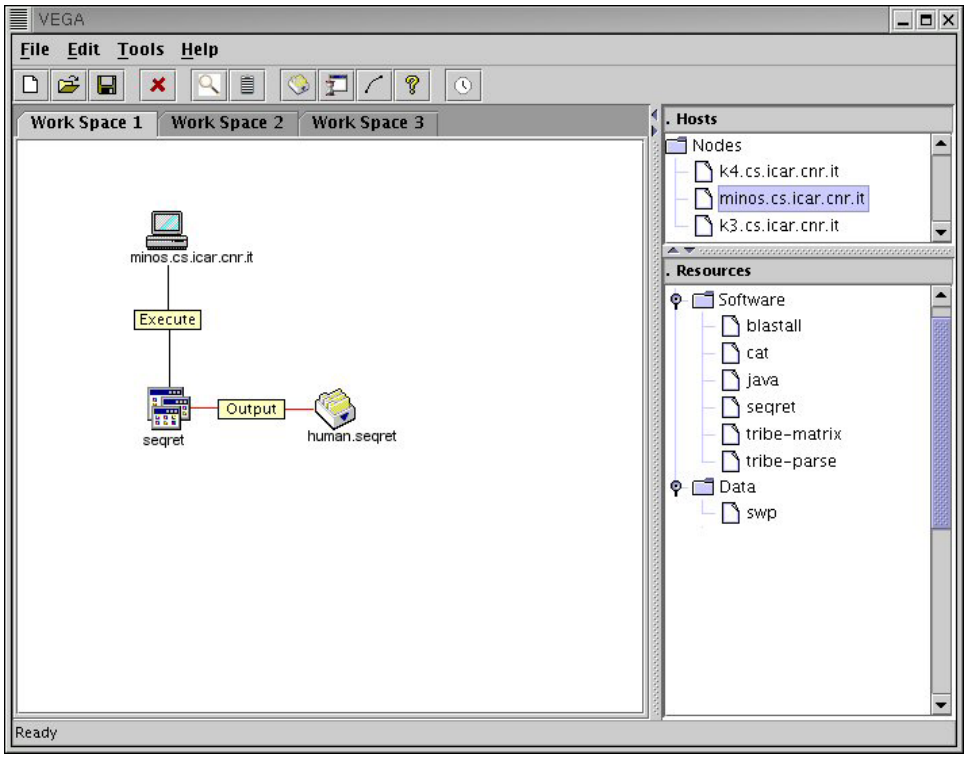


Fig. 6. Snapshot of VEGA: Workspace 1 of the Data Selection Phase

let suppose that Grid nodes are configured as shown in Table I and the `Swiss-Prot` database is replicated on each of them.

As a first step of the application formulation, the user browses the *Data Source* taxonomy (see Figure 5) of the domain ontology to locate the `Swiss-Prot` database. After that he/she searches software for extracting sequences from the database. Thus the user starts the ontology browsing from the *Task* taxonomy and identifies the `Extracting-sequences-from-DB` concept. From there following the `performed-by` label the user finds the seqret program (see Figure 7) and through its metadata file he/she locates the software on the `minos` node.

Software Components	Grid Nodes		
	minos	k3	k4
seqret	•		
splitFasta	•		
blastall	•	•	•
cat	•	•	•
tribe-parse	•	•	•
tribe-matrix	•		
mcl		•	
tribe-families			•

TABLE I
SOFTWARE INSTALLED ON THE EXAMPLE GRID

At this point the user is ready to design the *Data Selection* phase through VEGA constructing the following three work spaces:

1) Workspace 1. The human protein sequences are extracted from the `SwissProt` database using the `seqret` program on `minos` (see Figure 6).
2) Workspace 2. The file obtained as result of the `seqret` execution is partitioned in three smaller files using the `splitFasta` java utility class available on `minos` producing the files `split1.out`, `split2.out` and `split3.out`.
3) Workspace 3. `split2.out` and `split3.out` files are transferred respectively on `k3` and `k4` nodes.

The next step in the application design is to identify the tool performing the clustering process. To this aim the user starts the ontology browsing from the *Task* taxonomy (see Figure 7) and identifies the `proteins-clustering concept` (see Figure 8). From this point following the `performedBySoftware` property, the user finds out that `TribeMCL Tool` is a software used for the clustering of proteins (see Figures 8, 9). The *HasInput* property specifies that TribeMCL takes as input the results of a BLAST computation, and the *producesOutput* property states that output is a clustering of protein families.

Following the `HasMetadata` link the user finds the URL of the software metadata file. This file other than locating on which Grid nodes the tool is installed, contains information about how to access and use the tool, e.g. TribeMCL tool uses an *all against all* BLAST comparison as input to the clustering computation. Once again the user traverses the ontology to search the opportune version of the BLAST software needed in the process. This time the user explores the *Software Tool* taxonomy in the direction of the `similarity-search-sw` concept and from here identifies the BLAST tool and thus the `blastp` program needed.

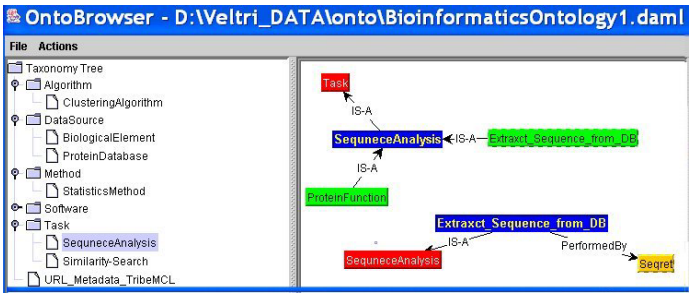


Fig. 7. Snapshot of the ontology browser

The Data Preprocessing phase consists of four VEGA work spaces:

1) Workspace 1. The BLAST computation is performed on the three nodes involved in the application containing the output files of the first phase (see Figure 10).
2) Workspace 2. The sequence similarity search output files are parsed using the `tribe-parse` software installed on three nodes.
3) Workspace 3. The files created on the nodes `k3` and `k4` in the Workspace 2 are transferred to the `minos` node where the software necessary to construct the Markov matrix is available.
4) Workspace 4. `cat` execution to concatenate the files.

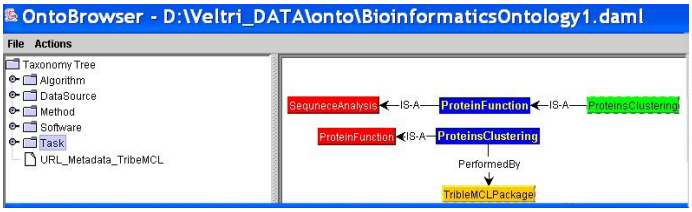


Fig. 8. Snapshot of the Ontology Browser

Once the files have been parsed using `tribe-parse`, it is possible to build the Markov matrix using the `tribe-matrix` program and perform the clustering operation. To this aim we have organized the Clustering phase into three VEGA workspaces:

1) Workspace 1. The Markov matrix is built using the `tribe-matrix` program installed on `minos`
2) Workspace 2. The `matrix.mci` file is transferred to `k3` where the clustering program `mcl` is available.
3) Workspace 3. `mcl` execution producing the `human.mcl` file.

Finally the Result Visualization phase has been organized in three VEGA workspaces:

1) Workspace 1. The `human.mcl` and the `protein.index` files are transferred on `k4` node
2) Workspace 2. The `tribe-families` program is executed on `k4` producing the file `human.cluster`.
3) Workspace 3. The final result, `human.cluster`, is transferred on `minos` to make it available to the user.

B. Experimental Results

The measurement of the execution times has been done in two different cases: a) we considered only 30 human proteins, and b) all the human proteins in the Swiss-Prot database (see Table II). Comparing the execution times shown in Table II we note that:

- The Data Selection and Results Visualization phases take the same time for the two cases, meaning that sequences extraction, file transfers and results displaying do not depend on the proteins number to be analyzed.
- In the Pre-processing phase there is a huge difference between the execution times of the two cases: the BLAST computations considering all the proteins are computationally intensive, so we have *8h50'13"* in the all proteins case compared to *2'50"* of the 30 proteins case.
- The execution of the mcl clustering program in the Clustering phase is a computationally intensive operation and consequently takes much more time when all the proteins have to be analyzed (*2h50'28"* versus *1'40"*). Note that the matrix file transferring time is the same for both applications.

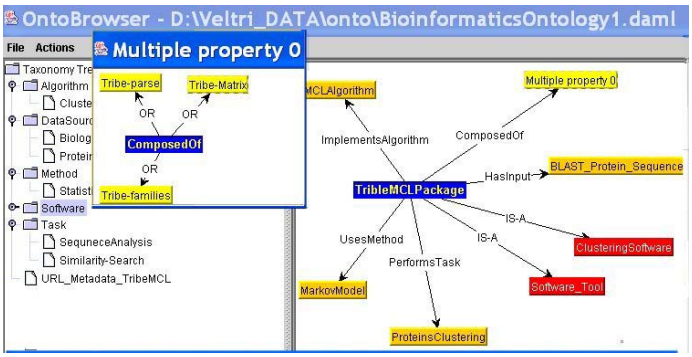


Fig. 9. Snapshot of the Ontology Browser

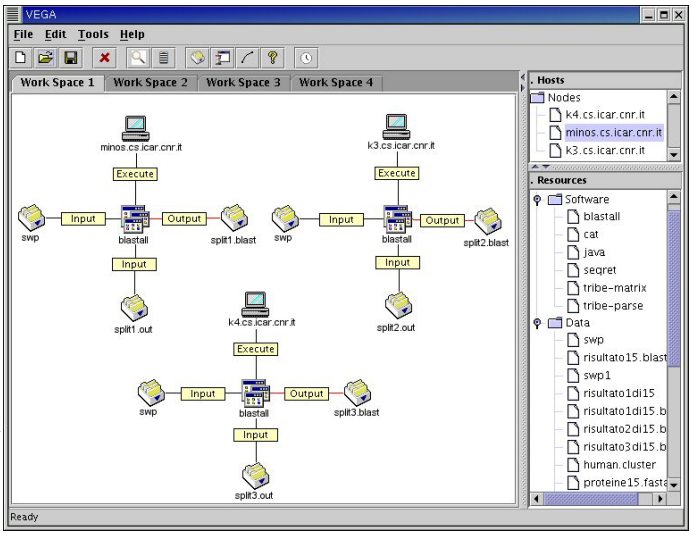


Fig. 10. Snapshot of VEGA: Workspace 1 of the Pre-processing Phase

Finally, a sequential version of the application, all human proteins case, has been executed on the minos host. This computation has taken a total execution time of *26h48'26"* compared to the *11h50'53"* of the parallel version. Moreover, some problems occurred in the management of the BLAST output file by the tribe-parsing program due to the high dimension of the file (about 2GB).

VI. Conclusion and Future Work

Novel Bioinformatics applications, and in particular Proteomics applications, will involve different software tools and various data sets, either produced in a given experiment, or available as public databases. Such applications will need a lot of semantic modeling of their basic components and will require large computational power.

In this paper we presented the design and implementation of PROTEUS , a Grid-based Problem Solving Environment for Bioinformatics applications. PROTEUS uses an ontology-based methodology to model semantics of bioinformatics applications. The current implementation of PROTEUS , based on the KNOWLEDGE GRID, has been successfully used to implement an application of human protein clustering.

We are improving PROTEUS architecture and functionalities by adding workflows methodologies for designing and monitoring applications [10]. Future works will regard the full implementation of PROTEUS and its use for the advanced analysis of proteomic data produced by mass spectrometry, for the early detection of inherited cancer [15].

TribeMCL Application	**Execution Time**	
Data Selection	30 proteins	*1'44"*
	All proteins	*1'41"*
Pre-processing	30 proteins	*2'50"*
	All proteins	*8h50'13"*
Clustering	30 proteins	*1'40"*
	All proteins	*2h50'28"*
Results Visualization	30 proteins	*1'14"*
	All proteins	*1'42"*
Total Execution Time	30 proteins	*7'28"*
	All proteins	*11h50'53"*

TABLE II
EXECUTION TIMES OF THE APPLICATION

Acknowledgment

This work has been partially supported by Project "FIRB GRID.IT" funded by MIUR. Authors are grateful to Domenico Talia for several suggestions on the main topic of this paper: we owe him many ideas on Grid use and applications. Authors also thank Antonio Massara, for support on the DAM+OIL ontolgy browser. Finally, authors are particularly grateful to Antonio Congiusta for discussion and contributions on the first implementation of PROTEUS on Vega System.

References

[1] S. Abiteboul and P. Buneman D. Suciu. *Data on the Web*. Morgan Kauffman, 2000.
[2] Vincent Aguilra, Sophie Cluet, Tova Milo, Pierangelo Veltri, and Dan Vodislav. Views in a Large Scale XML Repository. *VLDB Journal*, 11(3), November 2002.
[3] Enright A.J., Van Dongen S., and Ouzounis C.A. Tribemcl: An efficient algorithm for large scale detection of protein families. http://www.ebi.ac.uk/research/cgg/tribe/.
[4] ApBIONet.org. Asia pacific biogrid initiative. http://www.ncbi.nlm.nih.gov/.
[5] P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach*. MIT Press, 1998.
[6] S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens. OilEd: a reasonable ontology Editor for the Semantic Web. In *Artificial Intelligence Conference*. Springer Verlag, September 2001.
[7] LION bioscience AG. Srs search data bank system. http://srs.ebi.ac.uk/.
[8] M. Cannataro and C. Comito. A DataMining Ontology for Grid Programming. In *Workshop on Semantics in Peer-to-Peer and Grid Computing (in conj. with WWW2003)*, Budapest-Hungary, 2003.
[9] M. Cannataro, C. Comito, A. Congiusta, G. Folino, C. Mastroianni, A. Pugliese, G. Spezzano, D. Talia, and P. Veltri. Grid-based PSE Toolkits for Multidisciplinary Applications. FIRB "Grid.it" WP8 Working Paper 2003/10, ICAR-CNR, December 2003.
[10] M. Cannataro, C. Comito, A. Guzzo, and P. Veltri. Integrating Ontology and Workflow in PROTEUS, a Grid-Based Problem Solving Environment for Bioinformatics. Technical report, Univ. of Catanzaro, 2003.
[11] M. Cannataro, C. Comito, F. Lo Schiavo, and P. Veltri. PROTEUS: a Grid Based Problem Solving Environment for Bionformatics. In ISBN 0-9734039-0-X, editor, *Workshop on DataMining Ontology for Grid Programming (KGGI 03)*, Halifax-canada, 2003.
[12] M. Cannataro, A. Congiusta, D. Talia, and P. Trunfio. A Data Mining Toolset for Distributed High-performance Platforms. In Wessex Inst. Press, editor, *Data Mining Conference*, 2002. Bologna, Italy.
[13] M. Cannataro and D. Talia. KNOWLEDGE GRID An Architecture for Distributed Knowledge Discovery. *Communication of ACM*, 46(1), 2003.
[14] Grid Community. Global grid forum. http://www.gridforum.org/.
[15] G. Cuda, M.Cannataro, B. Quaresima, F. Baudi, R. Casadonte, M.C. Faniello, P. Tagliaferri, P. Veltri, F.Costanzo, and S. Venuta. Proteomic Profiling of Inherited Breast Cancer: Identification of Molecular Targets for Early Detection, Prognosis and Treatment, and Related Bioinformatics Tools. In *WIRN 2003*, *LNCS*, volume 2859 of *Neural Nets*, Vietri sul Mare, 2003. Springer Verlag.
[16] Daml.org. Daml+oil language. http://www.daml.org/2001/03/daml+oil-index.html.
[17] A.J. Enright, S. Van Dongen, and C.A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids*, 30(7), 2002.
[18] EUROGRID. Biogrid. http://biogrid.icm.edu.pl/.
[19] NCBI-National Cancer for Biotechnology Information. Blast database. http://www.ncbi.nih.gov/BLAST/.
[20] NCBI-National Cancer for Biotechnology Information. Entrez, the life science search engine. http://www.ncbi.nlm.nih.gov/Entrez/Index.html.
[21] NCBI-National Cancer for Biotechnology Information. Genbank dna sequences. http://www.ncbi.nlm.nih.gov/.
[22] Research Collaboratory for Structural Bioinformatics (RCSB). Protein data bank (pdb). http://www.rcsb.org/pdb/.
[23] S. Gallopoulos, E.N. Houstis, and J. Rice. Computer as Thinker/Doer: Problem-Solving Environments for Computational Science. In *Computational Science and Engineering*. IEEE, 1994.
[24] Grid.org. Grid life science group. http://forge.gridforum.org/projects/lsg-rg.
[25] EMBOSS Group. The european molecular biology open software suite. http://www.emboss.org.
[26] WWW Semantic Group. World wide web semantic group. http://www.w3c.org/semantic/.
[27] Foster I. and Kesselman C. *The Grid: Blueprint for a Future ComputingInfrastructure*. Morgan Kaufmann Publishers, 1999.

[28] W. E. Johnston. Computational and Data Grids in Large-Scale Science and Engineering. *Future Generation Computer Systems*, 18, 2002.
[29] EMBL-European Molecular Biology Laboratory. The swiss-prot protein database. http://www.embl-heidelberg.de/.
[30] University of Manchester. mygrid. http://mygrid.man.ac.uk/.
[31] Research and Technology Development project (RTD)-granted by the European Commission. Eurogrid- application testbed for european grid computing. http://www.eurogrid.org/.
[32] The globus project. http://www.globus.org/.
[33] J. D. Ullman. *Principles of Database and Knowledge-Base Systems*, volume I. Computer Science Press, 1988.
[34] D. Walker, O. F. Rana, M. Li, M. S. Shields, and Y. Huang. The Software Architecture of a Distributed Problem-Solving Environment. *Concurrency: Practice and Experience*, 12(15), December 2000.

Identifying Global Exceptional Patterns in Multi-database Mining

Chengqi Zhang[1], Meiling Liu[2], Wenlong Nie[3], and Shichao Zhang[1,2]

***Abstract*— In multi-database mining, there can be many local patterns (frequent itemsets or association rules) in each database. At the end of multi-database mining, it is necessary to analyze these local patterns to gain global patterns, when putting all the data from the databases into a single dataset can destroy important information that reflect the distribution of global patterns. This paper develops an algorithm for synthesizing local patterns in multi-database is proposed. This approach is particularly fit to find potentially useful exceptions. The proposed method has been evaluated experimentally. The experimental results have shown that this method is efficient and appropriate to identifying exceptional patterns.**

***Index Terms*— multi-database mining; local pattern evaluation; local pattern; global pattern; exceptional pattern**

I. Introduction

With the increasing development and application of distributed database technique and computer network, there exist many distributed databases in a business or financial organization. For example, a large company has many subsidiary companies, and each subsidiary company has its own database, all of the databases from each subsidiary company are relevant or irrelevant in logic, but they are distributed in different places. Different subsidiary company has different functions in helping the head company to make decisions. To make decisions for the development of company, the decision maker of the head company needs to know every database's interesting pattern or regulation and then synthetically evaluate these local patterns to generate global patterns.

It would appear to be unrealistic to collect data from different branches for centralized processing because of the potentially volume of data [20]. For example, different branches of Wal-Mart collect 20 million transactions per day. This is more than the rate at which data can feasibly be collected and analyzed by using today's computing power.

On the other hand, because of data privacy and related issues, it is possible that some databases of an organization can share their association rules but not their original data. Therefore, mining association rules from different databases and forwarding the rules (rather than the original raw data) to the central company headquarter provides a feasible way dealing with multiple database problems [19].

However, current data mining researches focus on mining in mono-database, but mono-database mining is different from multi-database mining because of their different data structure. So we need to come up with other solutions to analyze the data in multi-databases instead of using the technique in mono-database mining. This paper mainly discusses the pattern evaluation process at the end of data mining process and presents a method for identifying exceptional patterns.

[1]Faculty of Information Technology, University of Technology Sydney, PO Box 123, Broadway NSW 2007, Australia. {chengqi, zhangsc}@it.uts.edu.au

[2]Department of Computer Science, Guangxi Normal University, Guilin, 541004, P. C. China. hsp01wl@zsu.edu.cn

[3]The Institute of Logic and Cognition, Zhongshan University, Guangzhou, 510275, P. C. China. hsp01wl@zsu.edu.cn

This paper is organized as follows. Section II describes the process of multi-database mining and the patterns that exist in multi-database. Section III proposes a model for identifying exceptional patterns. Section IV designs an algorithm for identifying global exceptional patterns. In Section V, several experiments have be conducted for evaluating the proposed approach. In the last section we conclude this paper.

II. Description of Multi-Database Mining Problem

For description, this section states multi-database mining problem in a simple way.

Multi-database mining is the process of analyzing the data in multi-databases, and finding useful and novel knowledge, which is highly supported by most of databases or individual databases. Different from mono-database mining, there maybe exist semantic conflicts in multi-databases. The conflicts consist of synonym and homonyms. Synonym means that different field names in different databases denote the same data object. The head company must observe the semantic equivalence of the fields and translate the different local fields into a single global field name. Another kind of conflict is homonym, which means different data objects have the same name in different databases. The head company must recognize the semantic difference between the same field names and translate the same name into different field names. Because of these conflicts in different databases, the preprocessing in multi-database mining is very important. If we combine all the data in different databases into a single one and mine the large single database, then it may hide some features in a database and lose some useful pattern, moreover the techniques for integrating multi-databases is not perfect and it will take a large amount of efforts to integrate all the databases. The huge dataset after integrating will be difficult to deal with and its data may not be stored into memory at a time. So we cannot use traditional multi-database mining technique to analyze the data in multi databases.

The existing techniques for dealing with multi-databases first will classify all the databases into several group, databases in each group are relevant [23]. The classification of database is necessary, if not, the mined patterns may not be understood because there are some irrelevant data. For example, a large chain store has 10 subsidiary stores, some of them mainly sell groceries, and others sell electrical appliances. When mining these databases, one should classify the transaction databases in order to find out the databases that are relevant in the product categories. If we integrate all the transaction databases into a single one, at last perhaps we will not find rules because integrating all the databases involves in some irrelevant information. For example, the data in the food transaction database and the electrical appliances transaction database are put together, and then the association between the two kinds of product is difficult to understood by user because these product are not sold together and are distributed in different places. So it is very necessary to classify all the databases before mining data in multi-databases. After classifying, we can apply the techniques for mining mono-database to multi-database, and then find all the local patterns in every database. At last, all the local patterns will be analyzed and evaluated in order to find out the valuable information.

Patterns in multi-database can be divided into 4 categories [20], [21]:

(1) Local patterns. In a large interstate company, its branches has its own databases, it is impossible for the head company to analyze all its branches' database, so the branches need to analyze its data in its own local database and submit the mined patterns to head company. The mined patterns from each branch is called local patterns. Their function is to provide local databases' data features to the head company to make decision.
(2) High-voting patterns [21]. This kind of patterns is supported by most subsidiary companies. They reflect the common features among subsidiary companies. According to these patterns, the head company can make decisions for the common profits of branches.
(3) Exceptional patterns. These patterns are highly supported by only a few branches, that is to say, these patterns have very high support in these branches and zero support in other branches. They reflect the individuality of branches. And according to these patterns, the head company can adjust measures to local conditions and make special policies for these branches.
(4) Suggesting patterns. These patterns have fewer votes than the minimal vote but are very close to minimal vote. The minimal vote is given by users or experts. If a local pattern has votes equal to or greater than minimal vote, the local pattern is said to be a global pattern, called as high-voting pattern. If a local pattern has votes less than the minimal votes but are very close to the minimal vote, it is called suggesting patterns and sometimes it is useful for decision making.

The definitions of these patterns in multi-database indicate that there are differences between multi-database and mono-database mining. The final purpose of multi-database is to analyze and evaluate the common or special features in all the databases. In this paper, we only describe the exceptional patterns.

A. Related Work

Data mining techniques (see [1], [16], [22]) have been successfully used in many diverse applications. These include medical diagnosis and risk prediction, credit-card fraud detection, computer security break-in and misuse detection, computer user identity verification, aluminum and steel smelting control, pollution control in power plants and fraudulent income tax return detection. Developed techniques are oriented towards mono-databases.

Multi-database mining has been recently recognized as an important research topic in the KDD community. One article [24] proposed a means of searching for interesting knowledge in multiple databases according to a user query. The process involves selecting all interesting information from many databases by retrieval. Mining only works on the selected data.

Liu, Lu and Yao [10] proposed another mining technique in which relevant databases are identified. Their work has focused on the first step in multi-database mining, which is the identification of databases that are most relevant to an application. A relevance measure was proposed to identify relevant databases for mining with an objective to find patterns or regularity within certain attributes. This can overcome the drawbacks that are the result of forcedly joining all databases into a single very large database upon which existing data mining techniques or tools are applied. However, this database classification is typically database-dependent. Therefore, Zhang and Zhang have proposed a database-independent database classification in [23], which is useful for general-purpose multi-database mining.

Zhong et al [25] proposed a method of mining peculiarity rules from multiple statistical and transaction databases based on previous work. A peculiarity rule is discovered from peculiar data by searching the relevance among the peculiar data. Roughly speaking, data is peculiar if it represents a peculiar case described by a relatively small number of objects and is very different from other objects in a data set. Although it appears to be similar to the exception rule from the viewpoint of describing a relatively small number of objects, the peculiarity rule represents the well-known fact with common sense, which is a feature of the general rule.

Other related research projects are now briefly described. Wu and Zhang advocated an approach for identifying patterns in multi-database by weighting [19]. Ribeiro et al. [14] described a way of extending the INLEN system for multi-database mining by incorporating primary and foreign keys as well as developing and processing knowledge segments. Wrobel [18] extended the concept of foreign keys to include foreign links since multi-database mining also involves accessing non-key attributes. Aronis et al. [4] introduced a system called WoRLD that uses spreading activation to enable inductive learning from multiple tables in multiple databases spread across the network. Existing parallel mining techniques can also be used to deal with multi-databases [2], [6], [7], [12], [13], [15].

The above efforts provide a good insight into multi-database mining. However, there are still some limitations in traditional multi-database mining that are discussed in next subsection.

B. Limitations of Previous Multi-database Mining

As have seen, traditional multi-database mining is fascinated with mono-database mining techniques. It consists of a two-step approach. The first step is to select the databases most relevant to an application. All the data is then pooled together from these databases to amass a huge dataset for discovery upon mono-database mining techniques that can be used. However, there are still some limitations discussed below.

1) Putting all the data from relevant databases into a single database can destroy some important information that reflect the distributions of patterns. The statement "85% of the branches within a company agree that a customer usually purchases sugar if he/she purchases coffee" is an example of such a piece of information. These patterns may be more important than the patterns present in the mono-database in terms of global decision-making within a company. Hence, existing techniques for multi-databases mining are inadequate for applications.

In some contexts, each branch of an interstate company, large or small, has equal power in voting patterns for global decisions. For global applications, it is natural for the company headquarters to be interested in the patterns voted for by most of his/her branches. It is therefore inadequate in multi-database mining to utilize existing techniques for mono-databases mining.

2) Collecting all data from multi-databases can amass a huge database for centralized processing using parallel mining techniques.

It may be an unrealistic proposition to collect data from different branches for centralized processing because of the huge data volume. For example, different branches of Wal-Mart receive 20 million transactions a day. This is more than the rate at which data can be feasibly collected and analyzed using today's computing power.

3) Because of data privacy and related issues, it is possible that some databases of an organization may share their patterns but not their original databases.

Privacy is a very sensitive issue, and safeguarding its protection in a multi-database is of extreme importance. Most multi-database designers take privacy very seriously, and allow some protection facility. For source sharing in real-world applications, sharing patterns is a feasible way of achieving this.

From the above observations, it is clear that traditional multi-database mining is inadequate to serve two-level applications of an interstate company. This prompts the need to develop new techniques for multi-database mining.

Based on the above analysis, the problem for our research can be formulated as follows.

> Let $D_1, D_2, \cdots, D_m$ be m databases in the m branches $B_1, B_2, \cdots, B_m$ of a company, respectively; and LI_i be the set of local patterns (local instances) from D_i ($i = 1, 2, \cdots, m$). We are interested in the development of new techniques for identifying global exceptional patterns of interest in the local patterns.

III. Identifying Exceptional Patterns of Interest

Given n databases $D_1, D_2, \cdots, D_n$, they represent the databases from n branches of a large company. Let $LP_1, LP_2, \cdots, LP_n$ be the corresponding local patterns which are mined from every database; And $minsupp_i$ be the user specified minimal support in the database D_i $(i = 1, 2, \cdots, n)$. For each pattern P, its support in D_i is denoted by $Supp_i(P)$. We define the average vote of local patterns in the databases as follows.

Formula 1: $$AverVotes = \frac{\sum_{i=1}^{Num(GP)} Num(P_i)}{Num(GP)}$$

Where GP means the Global Patterns, it is the set of all patterns from each database, that is $GP = \{LP_1 \cup LP_2 \cup \cdots \cup LP_n\}$, and $Num(GP)$ is the number of patterns in GP. We regard the $AverVotes$ as a boundary to identify exceptional patterns and high-voting patterns. If a pattern's votes is less than the $AverVotes$, then it will be considered as an candidate exceptional pattern, otherwise as an high-voting pattern. We use CEP to denote the set of Candidate Exceptional Patterns and define the the global support of a pattern as follows.

Formula 2: $$Supp_G(P) = \frac{\sum_{i=1}^{Num(P)} \frac{Supp_i(P) - minsup_i}{1 - minsup_i}}{Num(P)}$$

where, $Supp_G(P)$ means the global support of a pattern; $Num(P)$ is the number of databases which support the pattern P. In this formula, we assume that the n databases play the same role in helping the head company to make decisions, that is to say that they have the same authority in providing their patterns to the head company. So we don't consider the weight of every database. Because

$$\frac{Supp_i(P) - minsup_i}{1 - minsup_i} \leq 1$$

therefore,

$$\sum_{i=1}^{Num(P)} \frac{Supp_i(P) - minsup_i}{1 - minsup_i} \leq Num(P)$$

The value $Supp_G(P)$ will be equal to or less than 1, as a result the closer $Supp_G(P)$ is to 1, the more significant the pattern will be.

The formula gives a method to compute a pattern's significance value. It uses the distance between a pattern's support and the corresponding database's minimal support as a measure. Because different database have different data information, We cannot simply say that 0.5 is greater than 0.22 in two databases whose minimal support is 0.48 and 0.13 respectively; This is because the two databases' minimal supports are different. According to the formula, we can obtain the significance of a pattern P in D_i. The greater the value $\frac{Supp_i(P) - minsup_i}{1 - minsup_i}$ is, the more significant the pattern P will be in D_i.

We only need to calculate the $Supp_G(P)$ values of patterns in CEP because these patterns' votes are less than the $AverVotes$, they will possibly be exceptional patterns. If a pattern has very high support in few databases and zero support in other databases, then its global support will be high. This pattern is referred to an exceptional pattern defined in Section 2. To evaluate the highness of the support of a pattern P in a database, we define a metrics as follows.

Formula 3: $$S(P) = \frac{Supp_i(P) - minsup_i}{1 - minsup_i}$$

where, $S(P)$ is the highness of the support of P in the database D_i, $Supp_i(P)$ is the support of P in D_i, $minsup_i$ is the user-specified minimum support for mining D_i.

This formula means, the higher the support of a pattern in a subsidiary company, the more interesting the pattern will be. We define the formula to compute the deviation of a pattern from the corresponding minimal support $minsup_i$, The value will be used to draw plots to show how far the patterns deviate from the same level.

IV. Algorithm Design

Exceptional patterns reflect the individuality of branches within an interstate company. This section presents an algorithm, *Identify-ExPattern*, for identifying exceptional patterns.

Algorithm 1: IdentifyExPattern

Input: LP_i: set of local patterns; $minsup_i$: minimal support threshold in D_i $(i = 1, 2, \cdots, n)$;

Output:EP: the set of exceptional patterns;

begin

(1) $GP \leftarrow \{LP_1 \cup LP_2 \cup \cdots \cup LP_n\}$; CEP=Ø;

(2) For each pattern P in GP do
Count P's votes, $Num(P)$; And Record which database support it, using $from$ to note them.
Calculate the average votes using Formula 1: $AverVotes = \frac{\sum_{i=1}^{Num(GP)} Num(P_i)}{Num(GP)}$

(3) For each pattern P in GP do
if $(Num(P) < AverVotes)$ $CEP = CEP \cup P$

(4) For each candidate exceptional pattern P in CEP do
$$Supp_G(P) \leftarrow \frac{\sum_{i=1}^{Num(P)} \frac{Supp_i(P) - minsup_i}{1 - minsup_i}}{Num(P)}$$

(5) Rank all the patterns P in CEP by their $Supp_G(P)$;

(6) Output the high rank patterns in CEP and the databases which support them;

End.

The algorithm $IdentifyExPattern$ is to search all the significant exceptional patterns from the given n local patterns.

Step (1) generates the set of patterns from each database. Step (2) counts each pattern's votes, and the average votes of patterns $AverVotes$. Step (3) generates the candidate exceptional patterns. Step (4) is to calculate all the candidate exceptional patterns's $Supp_G(P)$ values. Step (5) ranks the candidate exceptional patterns by their $Supp_G(P)$. Step (6) outputs all the exceptional patterns which satisfy the user's requirement and have high rank.

Example 1: Consider 5 databases $D_1, D_2, \cdots, D_5$, their corresponding patterns is in the following. Patterns are denoted by A-F, the value after each colon is the pattern's support; $minsup_1 = 0.49$, $minsup_2 = 0.48$, $minsupp_3 = 0.82$, $minsupp_4 = 0.20$, $minsupp_5 = 0.13$ are 5 databases' minimal support respectively.

$$\begin{aligned}
LP_1 &= \{\{A:0.69\}; \{C:0.68\}; \{F:0.52\}\} \\
LP_2 &= \{\{A:0.50\}; \{B:0.62\}; \{C:0.91\}; \{E:0.82\}; \\
&\quad \{F:0.76\}; \{G:0.86\}\} \\
LP_3 &= \{\{A:0.87\}; \{C:0.85\}; \{D:0.86\}; \{E:0.86\}; \\
&\quad \{F:0.95\}\} \\
LP_4 &= \{\{B:0.36\}; \{C:0.31\}; \{E:0.28\}\} \\
LP_5 &= \{\{E:0.22\}\}
\end{aligned}$$

We now use the algorithm $IdentifyExPattern$ to search all the exceptional patterns from the given local patterns. According to the Step (1) and Step (2), we can get $GP = \{A, B, C, D, E, F, G\}$,and the $AverVotes = \frac{18}{7} = 2.57$. Because Pattern B, D and G have less votes than the $AverVotes$. After pruning by $AverVotes$, $CEP = \{B, D, G\}$. The $Supp_G(P)$ value of each pattern in CEP are shown as follows.

$Supp_G(B) = 0.235$, Pattern B comes from $\{D_2, D_4\}$
$Supp_G(D) = 0.222$, Pattern D comes from $\{D_3\}$
$Supp_G(G) = 0.73$, Patterns G comes from $\{D_2\}$

After rank the patterns in CEP by their $Supp_G(P)$, the order will be $\{G, B, D\}$. It is obvious that pattern {G} has the highest global

support and it is supported by only a database. So it can be regarded as an exceptional pattern. After finding such exceptional patterns, the head company can use the patterns to assist making special decision for the corresponding subsidiary company.

From the example, we can see that this approach is reasonable and when the manager of head company makes decisions for the development of his company, he can not consider only the number that supported a certain pattern but also the pattern's support value in these databases.

In the practical application of multiple database, such as chain stores and interstate company , because it maybe generate large amount of patterns, it is necessary to find an approach to evaluate all the patterns.

V. Experiments

In this section, we evaluate the function of the approach. The following experiments were conducted on Pentium 4 personal computer with 256 MB main memory running Microsoft Windows 2000. Our intention is not to evaluate the running time of our approach, So the experiment environment is not important. The dataset used in one experiment were generated randomly, we considered the data as mined patterns. And the other experiment was conducted on real dataset downloaded from the Internet (http://www.ics.uci.edu/ mlearn/MLSummary.html).

A. Random Patterns

First, we present our experiment on the randomly generated patterns. These patterns were generated randomly by our patterns generator and were assigned certain support. In addition, the minimal support of each database was also assigned randomly. Of course, when designing the algorithm for generating patterns' support, we assigned that each pattern's support must equal to or greater than the corresponding database's minimal support because we considered these patterns were pruned by minimal support threshold. Table 1 shows the parameter setting in Experiment 1. And Table 2 shows the number of patterns in each database and the minimal support of each database.

Table 1: Parameters setting in Experiments

Number of datasets	10
Average number of patterns in all datasets	10
Patterns Symbols	1-15

Table 2: Number of patterns and the minimal support in each dataset

Dataset	Number of patterns	Patterns	minsupp
D0	3	{11}:0.57 {5}:0.63 {4}:0.21	0.19
D1	9	{6}:0.87 {13}:0.77 {12}:0.80 {3}:0.75 {7}:0.78 {10}:0.82 {4}:0.75 {8}:0.88 {5}:0.82	0.74
D2	5	{7}:0.46 {4}:0.47 {15}:0.49 {2}:0.54 {14}:0.51	0.45
D3	11	{5}:0.80 {7}:0.85 {14}:0.81 {6}:0.87 {13}:0.81 {2}:0.84 {3}:0.81 {1}:0.88 {10}:0.81 {9}:0.83 {12}:0.81	0.80
D4	2	{10}:0.50 {14}:0.50	0.05
D5	2	{13}:0.22 {12}:0.40	0.10
D6	5	{4}:0.89 {1}:0.88 {12}:0.88 {7}:0.88 {6}:0.89	0.88
D7	10	{10}:0.39 {4}:0.52 {13}:0.71 {1}:0.88 {7}:0.27 {3}:0.38 {5}:0.86 {8}:0.81 {11}:0.74 {12}:0.74	0.22
D8	3	{3}:0.74 {4}:0.85 {15}:0.86	0.54
D9	2	{4}:0.61 {2}:0.49	0.38

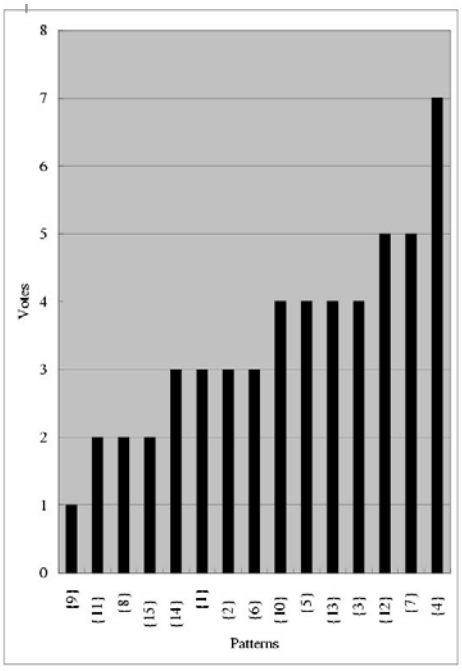


Fig. 1. Votes of patterns

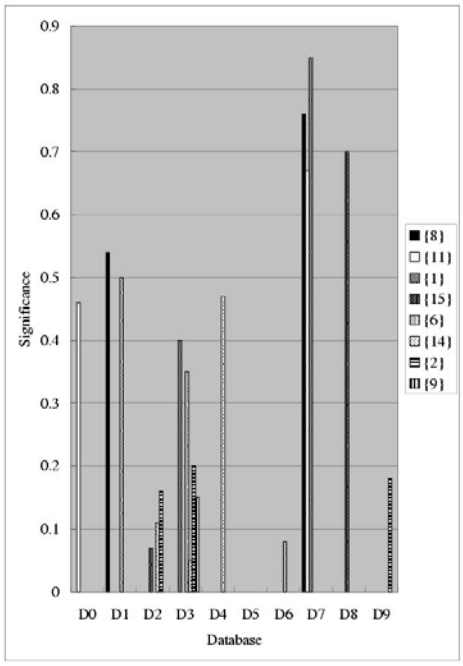


Fig. 2. Each pattern's significance calculated by formula 3.

The distributions of patterns in GP are shown in Figure 1. X-coordinate denotes the patterns in GP, and Y-coordinate are the patterns' votes. In this experiment, $averVotes = 3.3$, from Figure 1, we can see that pattern $\{\{9\}, \{11\}, \{8\}, \{15\}, \{14\}, \{1\}, \{2\}, \{6\}\}$ are candidate exceptional patterns because their votes are less than 3.3.

After executing the $IdentifyExPattern$ algorithm, we can get the global support of all candidate exceptional patterns. The $Supp_G(P)$ values are shown in Table 3.

Table 3 shows that the candidate exceptional pattern $\{8\}$ has the highest global support and it is supported by $D1$ and $D7$. When searching Table 2, we find that pattern $\{8\}$ has the highest support in $D1$ and the second highest support in $D7$ comparing to their corresponding minimal support. The experiment results show that this method can be used to find exceptional patterns when there exist exceptional patterns in Multiple databases. In section 5.2, we will present an experiment in which we can not find any exceptional patterns because there doesn't exist any exceptional patterns in the specified multi-database.

B. Real Datasets

For real-life applications, we have also evaluated our approach using the database downloaded from the Internet (please see http://www.ics.uci.edu/ mlearn/MLSummary.html/). We choose the Zoo Database containing 101 instances and 18 attributes (animal name, 15 boolean attributes, 2 numerics). The boolean attributes are "hair", "feathers", "eggs", "milk", "airborne", "aquatic", "predator", "toothed", "backbone", "breathes", "venomous", "fins", "tail", "domestic" and "catsize". And the numeric attributes are "legs" and "type", where the "type" attribute appears to be the class attribute. All the instances are classified into 7 classes.

To obtain multiple and relevant databases, we vertically partitioned the Zoo Database into 7 subset datasets according to the "type" attribute. Each dataset contained 18 attributes. When preprocessing, we used different number to denote different attribute values. After preprocessing, we mined the 7 datasets respectively and obtained their own frequent itemsets. Table 4 shows the 7 datasets' corresponding information.

Because of the large amount of frequent itemsets, to better illustrate the efficiency of our approach, we only selected some special frequent itemsets which were relevant to the specified attribute. We selected 97 frequent itemsets and their votes are shown in Figure 3. In this experiment, the $AverVotes = 2.4$.

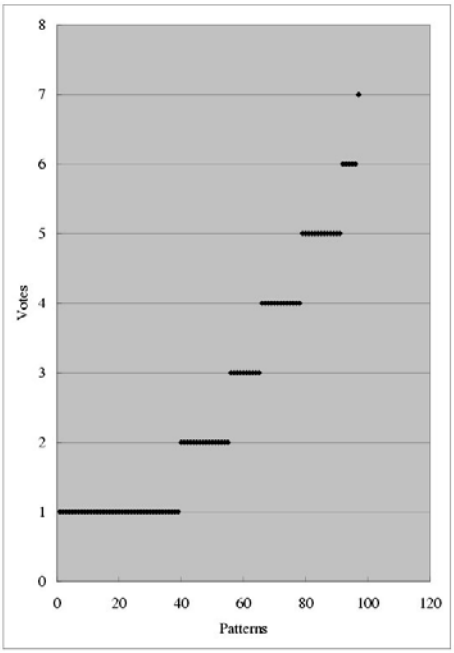


Fig. 3. Votes of 97 frequent itemsets

From Figure 3, we can see that there are about 55 frequent itemsets whose votes are less than the $AverVotes$. Table 5 shows the mined typical exceptional patterns using our approach.

From Table 5, we can easily see that the animals in D1 are characteristic with Pattern P1 and those animals in other datasets have no the character. So it can be regarded as an exceptional pattern owned by D1. This is because the animals in D1 are mammals which are different from other datasets. And D4's instances are all fish, only fish have fins, so the results is reasonable. For other patterns showed in Table 5, they are also considered as exceptional patterns. In this experiment, we partitioned the original database into 7 datasets by their "type" attribute. This partition makes that each database belongs to a certain class, So we can find the potential exceptional patterns. From the experiment, we can draw a conclusion that our approach is useful to identify exceptional patterns.

At last, we simply presented another experiment. In this experiment, we only selected 3 datasets (D4, D5, D6, D7) and 3 attributes in the Zoo Database ("feathers", "eggs", "milk"). The experiment result shows that most of the animals in the 4 datasets have the common features: most of them have no feathers, and can lay eggs but have no milk. That is to say, there doesn't exist potential exceptional patterns. As a result, we can not find any exceptional patterns.

Table 3: Exceptional patterns analysis

Pattern	Patterns	Supported by which dataset	Patterns' Meaning
P1	{hair=1}	D1	The animals in D1 usually have hair
P2	{eggs=0}	D1	The animals in D1 usually can not lay eggs
P3	{milk=1}	D1	The animals in D1 usually have milk
P4	{legs=4}	D1	The animals in D1 usually have 4 legs
P5	{feathers=1}	D2	The animals in D2 usually have feathers
P6	{legs=2}	D2	The animals in D2 usually have 2 legs
P7	{fins=1}	D4	The animals in D4 usually have fins
P8	{legs=0}	D4	The animals in D4 usually have no legs
P9	{hair=0 and legs=4}	D5	The animals in D5 usually have 4 legs, but no hair. These characters are different from those in D1, in D1, the animals also have 4 legs but they have hair.
P10	{predator=0}	D6	The animals in D6 are not predators.
P11	{legs=6}	D6	The animals in D6 usually have 6 legs.
P12	{hair=0 and backbone=0}	D7	The animals in D7 usually have no hair and no backbones.

VI. Summary

In this paper, we studied an approach for identifying exceptional patterns from multiple databases. It can be considered as an postprocessing work after mining multiple, relevant databases. We conducted several experimental studies, one was experimented on patterns which were generated randomly and the other was experimented on real Zoo Database. We found that our approach can identify potential exceptional patterns from multiple databases' patterns. On one hand, if there exists potential exceptional patterns in multiple databases, the approach can be used to find them out. On the other hand, if there does not exist any potential exceptional patterns in multiple databases, no exceptional patterns can be found. Therefore, the approach is fit to find potential exceptional patterns. It seems that the datasets used in the experiments are not relevant to the business data, but our intention is to illustrate the function of our approach. In the practical application, when faced with the patterns of multiple databases, we can use the method to find exceptional patterns from the multiple databases and make special decisions.

However, if more information about the multiple databases can be considered, the experiment results will be more perfect. There are one direction for ongoing work by weighting. If each subsidiary company plays different roles in assisting making decision for the head company. We can assign weights for each database.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments on the first version of this paper.

References

[1] R. Agrawal, T. Imielinski, and A. Swami, Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, 6(1993): 914-925.

[2] R. Agrawal, J. Shafer: Parallel Mining of Association Rules. *IEEE Transactions on Knowledge and Data Engineering*, 8(6) (1996): 962-969.

[3] J. Albert, Theoretical Foundations of Schema Restructuring in Heterogeneous Multidatabase Systems. In: *Proceedings of International Conference on Information and Knowledge Management*, 2000: 461-470.

[4] J. Aronis et al, The WoRLD: Knowledge discovery from multiple distributed databases. *Proceedings of 10th International Florida AI Research Symposium*, 1997: 337-341.

[5] P. Chan, An Extensible Meta-Learning Approach for Scalable and Accurate Inductive Learning. *PhD Dissertation*, Dept of Computer Science, Columbia University, New York, 1996.

[6] J. Chattratichat, et al., Large scale data mining: challenges and responses. In: *Proceedings of Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, (KDD-97), Newport Beach, California, USA, AAAI Press, August 14-17, 1997: 143-146.

[7] D. Cheung, V. Ng, A. Fu and Y. Fu, Efficient Mining of Association Rules in Distributed Databases, *IEEE Transactions on Knowledge and Data Engineering*, 8(1996), 6: 911–922.

[8] E. Han, G. Karypis and V. Kumar, Scalable Parallel Data Mining for association rules. In: *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1997: 277-288.

[9] A. Hurson, M. Bright, and S. Pakzad, *Multidatabase systems: an advanced solution for global information sharing.* IEEEComputer Society Press, 1994.

[10] H. Liu, H. Lu, and J. Yao, Identifying Relevant Databases for Multidatabase Mining. In: *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1998: 210–221.

[11] J. Park, M. Chen, P. Yu: Efficient Parallel and Data Mining for Association Rules. In: *Proceedings of International Conference on Information and Knowledge Management*, 1995: 31-36.

[12] A. Prodromidis, S. Stolfo. Pruning meta-classifiers in a distributed data mining system. In: *Proceedings of the First National Conference on New Information Technologies*, 1998: 151-160.

[13] A. Prodromidis, P. Chan, and S. Stolfo, Meta-learning in distributed data mining systems: Issues and approaches, In *Advances in Distributed and Parallel Knowledge Discovery*, H. Kargupta and P. Chan (editors), AAAI/MIT Press, 2000.

[14] J. Ribeiro, K. Kaufman, and L. Kerschberg, Knowledge discovery from multiple databases. In: *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (KDD-95), Montreal, Canada, AAAI Press, August 20-21, 1995: 240-245.

[15] T. Shintani and M. Kitsuregawa, Parallel mining algorithms for generalized association rules with classification hierarchy. In: *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1998: 25-36.

[16] G. Webb, Efficient search for association rules. In: *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*, 2000: 99-107.

[17] D. Wolpert, Stacked Generalization. *Neural Networks*, 5(1992): 241–259.

[18] S. Wrobel, An algorithm for multi-relational discovery of subgroups. In: J. Komorowski and J. Zytkow (eds.) *Principles of Data Mining and Knowledge Discovery*, 1997: 367-375.

[19] Xindong Wu and Shichao Zhang, Synthesizing High-Frequency Rules from Different Data Sources. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 2, March/April 2003: 353-367.

[20] Shichao Zhang, Xindong Wu and Chengqi Zhang, Multi-Database Mining. *IEEE Computational Intelligence Bulletin*, Vol. 2, No. 1, June 2003: 5-13.

[21] Shichao Zhang, Chengqi Zhang and Xindong Wu, *Knowledge Discovery in Multiple Databases*. Springer, 2004.

[22] Shichao Zhang and Chengqi Zhang, Anytime Mining for Multi-User Applications. *IEEE Transactions on Systems, Man and Cybernetics (Part A)*, Vol. 32 No. 4(2002): 515-521.

[23] Chengqi Zhang and Shichao Zhang, Database Clustering for Mining Multi-Databases. In: *Proceedings of the 11th IEEE International Conference on Fuzzy Systems*, Honolulu, Hawaii, USA, May 2002.

[24] J. Yao and H. Liu, Searching Multiple Databases for Interesting Complexes. In: *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1997: 198-210.

[25] N. Zhong, Y. Yao, and S. Ohsuga, Peculiarity oriented multi-database mining. In: *Principles of Data Mining and Knowledge Discovery*, 1999: 136-146.

A Support Environment for Domain Ontology Development with General Ontologies and Text Corpus

Naoki Sugiura[1], Noriaki Izumi[2], and Takahira Yamaguchi[1]

***Abstract*—For constructing semantically rich service descriptions in Grid services, emerging ontologies are being used. To generate ontologies, an issue named "ontology bottleneck", the lack of efficient ways to build ontologies, has been coming up. Therefore, it is an urgent task to improve the methodology for rapid development of more detailed and specialized domain ontologies. However, it has been a hard task because domain concepts have highly-specialized semantics and the number of concepts is fairly large. In order to reduce the cost, DODDLE II (a domain ontology rapid development environment II) has been developed in our research group. In this paper, we confirm the significance of DODDLE II. In addition, we introduce our plan for further extension for the Semantic Web as a future work.**

***Index Terms*—Ontology Development, Knowledge Engineering, Grid services**

I. Introduction

WHILE Grid services deliver dynamic and relevant applications, a key remaining challenge is supporting automated interoperability without human intervention. Although ontologies are being used in many application areas to improve interoperability, we still face the problem of high cost associated with building up ontologies manually. In particular, since domain ontologies have the meaning specific to application domains, human experts have to make huge efforts for constructing them entirely by hand. In order to reduce the costs, automatic or semi-automatic methods have been proposed using knowledge engineering techniques and natural language processing ones [1]. However, most of these environments facilitate the construction of only a hierarchically-structured set of domain concepts, in other words, taxonomic conceptual relationships. For example, DODDLE [2] developed by us uses a machine-readable dictionary (MRD) to support a user in constructing concept hierarchy only.

In this paper, we extend DODDLE into DODDLE II that constructs both taxonomic and non-taxonomic conceptual relationships, exploiting WordNet [4] and domain specific text corpus with the automatic analysis of lexical co-occurrence statistics based on WordSpace [3] and an association rule algorithm [5]. Furthermore, we evaluate how DODDLE II works in the field of business, xCBL (XML Common Business Library)[6]. The empirical results show us that DODDLE II can support a domain expert in constructing domain ontologies.

[1]Department of Computer Science, Shizuoka University, 3-5-1, Johoku, Hamamatsu, Shizuoka, 432-8011, Japan (phone: +81-53-478-1473; fax: +81-53-473 –6421). {sugiura, yamaguti}@ks.cs.inf.shizuoka.ac.jp

[2]Cyber Assist Research Center, National Institute of Advanced Industrial Science and Technology, 2-41-6, Aomi, Koto-ku Tokyo, Japan. niz@ni.aist.go.jp

II. DODDLE II: A DOMAIN ONTOLOGY RAPID DEVELOPMENT ENVIRONMENT

A. Overview

Fig. 1 describes the system overview of DODDLE II. We can build concept specification templates by putting together taxonomic and non-taxonomic relationships for the input domain terms. The relationships should be identified in the interaction with a human expert.

B. Taxonomic Relationship Acquisition

First of all, TRA module does "spell match" between input domain terms and WordNet. The "spell match" links these terms to WordNet. Thus the initial model from the "spell match" results is a hierarchically structured set of all the nodes on the path from these terms to the root of WordNet. However, the initial model has unnecessary internal terms (nodes) and

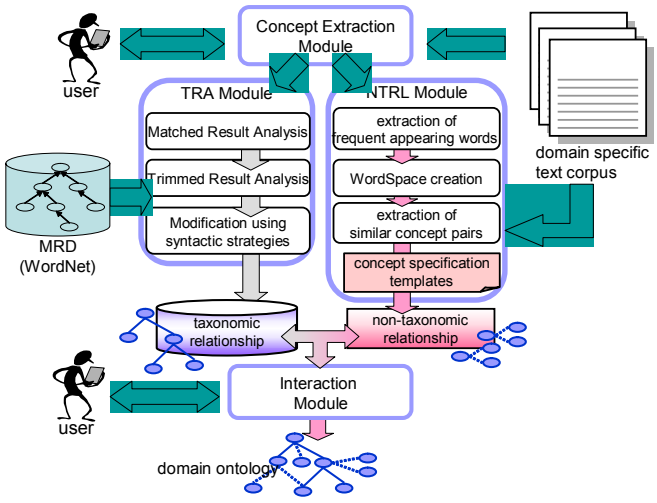


Fig. 1. DODDLE II overview

they do not contribute to keep topological relationships among matched nodes, such as parent-child relationship and sibling relationship. So we get a trimmed model by trimming the unnecessary internal nodes from the initial model (see Fig. 2). After getting the trimmed model, TRA module refines it by interaction with a domain expert, using Matched result analysis (see Fig. 3) and Trimmed result analysis (see Fig. 4). TRA module divides the trimmed model into a PAB (a PAth including only Best spell-matched nodes) and an STM (a Subtree that includes best spell-matched nodes and other nodes and so can be Moved) based on the distribution of best-matched nodes. A PAB is a path that includes only best-matched nodes that have the senses good for given domain specificity.
Because all nodes have already been adjusted to the domain in PABs, PABs can stay in the trimmed model. An STM is such a subtree that an internal node is a root and the subordinates are only best-matched nodes. Because internal nodes have not been confirmed to have the senses good for a given domain, an STM can be moved in the trimmed model.

In order to refine the trimmed model, DODDLE II can use trimmed result analysis. Taking some sibling nodes with the same parent node, there may be big differences about the number of trimmed nodes between them and the parent node. When such a big difference comes up on a subtree in the trimmed model, it is better to change the structure of it. DODDLE II asks a human expert whether the subtree should be reconstructed. Based on the empirical analysis, the subtrees with two or more differences may be reconstructed.
Finally, DODDLE II completes taxonomic relationships of the input domain terms manually from the user.

C. *Non-Taxonomic Relationship Learning*

NTRL module almost comes from WordSpace, which derives lexical co-occurrence information from a large text corpus and is a multi-dimension vector space (a set of vectors). The inner product between two word vectors works as the measure of their semantic relatedness. When two words' inner product is beyond some upper bound, there are possibilities to have some non-taxonomic relationship between them. NTRL module also uses an association rule algorithm to find associations between terms in text corpus. When an association rule between terms exceeds user-defined thresholds, there are possibilities to have some non-taxonomic relationships between them.

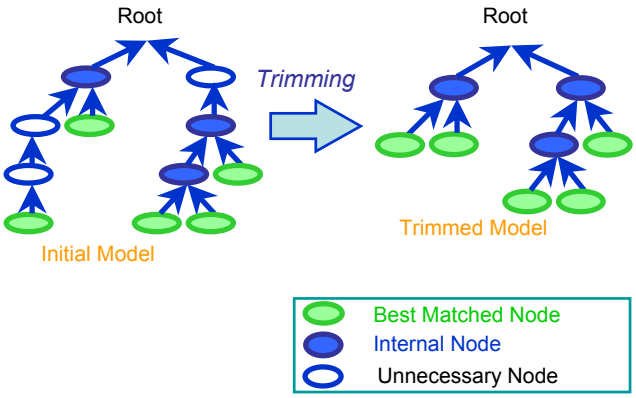


Fig. 2. Trimming Process

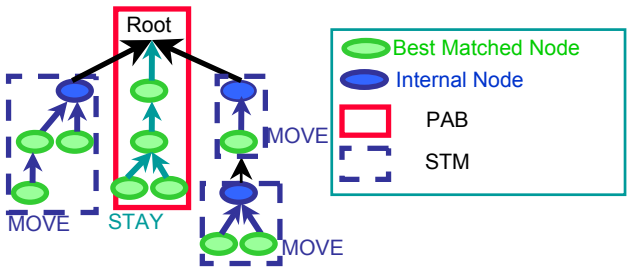


Fig. 3. Matched Result Analysis

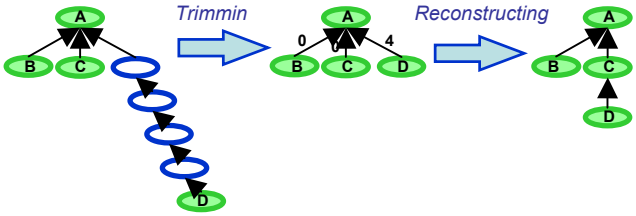


Fig. 4. Trimmed Result Analysis

D. *Construction of WordSpace*

WordSpace is constructed as shown in Fig. 5.
1. *Extraction of high-frequency 4-grams* Since letter-by-letter co-occurrence information becomes too much and so often irrelevant, we take term-by-term co-occurrence information in four words (4-gram) as the primitive to make up co-occurrence matrix useful to represent context of a text based on experimented results. We take high frequency 4-grams in order to make up WordSpace.
2. *Construction of collocation matrix* A collocation matrix is constructed in order to compare the context of two 4-grams. Element $a_{i,j}$ in this matrix is the number of 4-gram f_i which comes up just before 4-gram f_i (called collocation area). The collocation matrix counts how many other 4-grams come up before the target 4-gram. Each column of this matrix is the 4-gram vector of the 4-gram f.
3. *Construction of context vectors* A context vector represents context of a word or phrase in a text. A sum of 4-gram vectors

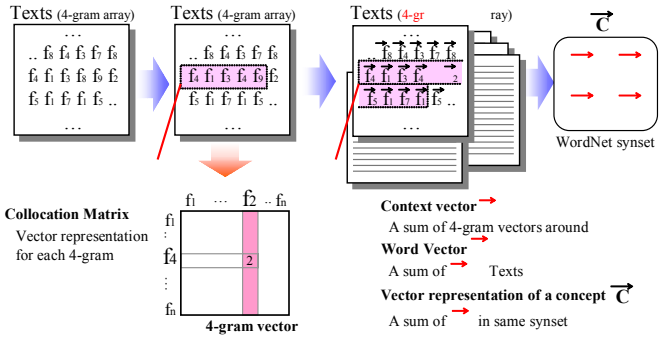


Fig. 5. Construction Flow of WordSpace

around appearance place of a word or phrase (called context area) is a context vector of a word or phrase in the place.

4. *Construction of word vectors* A word vector is a sum of context vectors at all appearance places of a word or phrase within texts, and can be expressed with Eq.1. Here, is a vector representation of a word or phrase *w*, *C(w)* is appearance places of a word or phrase *w* in a text, and *φ(f)* is a 4-gram vector of a 4-gram *f*. A set of vector *τ(w)* is WordSpace.

$$\tau(w) = \sum_{i \in C(w)} \left(\sum_{f\ closetoi} \varphi(f) \right) \quad (1)$$

5. *Construction of vector representations of all concepts* The best matched "synset" of each input terms in WordNet is already specified, and a sum of the word vector contained in these synsets is set to the vector representation of a concept corresponding to an input term. The concept label is the input term.

6. *Construction of a set of similar concept pairs* Vector representations of all concepts are obtained by constructing WordSpace. Similarity between concepts is obtained from inner products in all the combination of these vectors. Then we define certain threshold for this similarity. A concept pair with similarity beyond the threshold is extracted as a similar concept pair.

Finding Association Rules between Input Terms The basic association rule algorithm is provided with a set of transactions, $T := \{t_i \mid i = 1..n\}$, where each transaction t_i consists of a set of items, $t_i = \{a_{i,j} \mid j = 1..m_i, a_{i,j} \in C\}$ and each item $a_{i,j}$ is form a set of concepts C. The algorithm finds association rules $X_k \Rightarrow Y_k : (X_k, Y_k \subset C, X_k \cap Y_k = \{\})$ such that measures for support and confidence exceed user-defined thresholds. Thereby, support of a rule $X_k \Rightarrow Y_k$ is the percentage of transactions that contain $X_k \cup Y_k$ as a subset (Eq.2)and confidence for the rule is defined as the percentage of transactions that Y_k is seen when X_k appears in a transaction (Eq.3).

$$support\,(X_k \Rightarrow Y_k) = \frac{\left|\left\{t_i \middle| X_k \cup Y_k \subseteq t_i\right\}\right|}{n} \quad (2)$$

$$confidence\,(X_k \Rightarrow Y_k) = \frac{\left|\left\{t_i \middle| X_k \cup Y_k \subseteq t_i\right\}\right|}{\left|\left\{t_i \middle| X_k \subseteq t_i\right\}\right|} \quad (3)$$

As we regard input terms as items and sentences in text corpus as transactions, DODDLE II finds associations between terms in text corpus. Based on experimented results, we define the threshold of support as 0.4% and the threshold of confidence as 80%. When an association rule between terms exceeds both thresholds, the pair of terms is extracted as candidates for non-taxonomic relationships.

E. Constructing and Modifying Concept Specification Templates

A set of similar concept pairs from WordSpace and term pairs from the association rule algorithm becomes concept specification templates. Both of the concept pairs, whose meaning is similar (with taxonomic relation), and has something relevant to each other (with non-taxonomic relation), are extracted as concept pairs with above-mentioned methods. However, by using taxonomic information from TRA module with co-occurrence information, DODDLE II distinguishes the concept pairs which are hierarchically close to each other from the other pairs as TAXONOMY. A user constructs a domain ontology by considering the relation with each concept pair in the concept specification templates, and deleting unnecessary concept pairs.

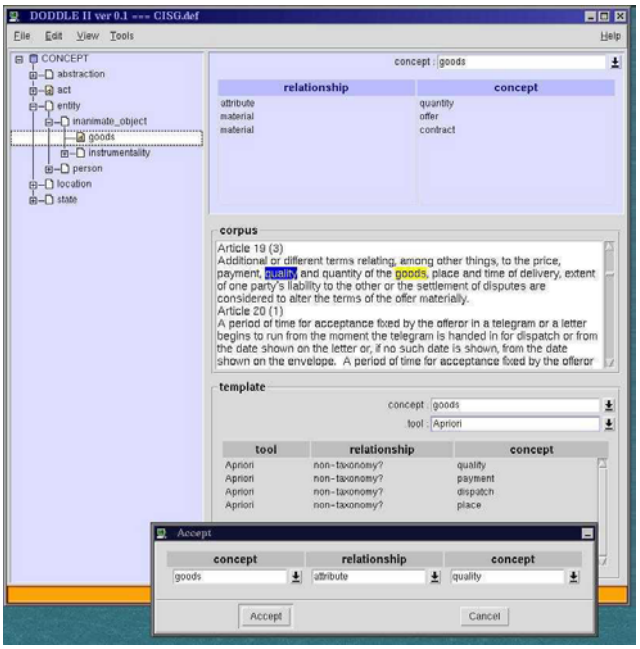


Fig. 6. The Ontology Editor

III. CASE STUDY

In order to evaluate how DODDLE II is going in a practical field, a case study has been done in particular field of business called xCBL (XML Common Business Library) [6]. DODDLE II has been implemented on Perl/Tk. Fig. shows the typical screen of DODDLE II.

A. Input terms

Table 1 shows input terms in this case study. They are 57 business terms extracted by a user from xCBL Document Reference. The user is not an expert but has business knowledge.

B. Taxonomic Relationship Acquisition

Table 2 shows the number of concept pairs in each model under taxonomic relationship acquisition and

Table 3 shows the evaluation of two strategies by the user. The recall per subtree is more than 0.5 and is good. The precision and the recall per path are less than 0.3 and are not so good, but about 80 % portion of taxonomic relationships were constructed with TRA module support. We evaluated TRA module worked well in this case study.

TABLE 1
SIGNIFICANT 57 CONCEPTS IN XCBL

acceptance	agreement	auction	availability	business
buyer	change	contract	customer	data
date	delivery	document	Exchange rate	financial institution
foreign exchange	goods	information	invoice	item
Line item	location	marketplace	message	money
order	organization	partner	Party	payee
payer	payment	period of time	Price	process
product	purchase	Purchase agreement	Purchase order	quantity
quotation	quote	receipt	rejection	request
resource	response	schedule	seller	service
shipper	status	supplier	system	third party
transaction	user			

TABLE 2
THE CHANGE OF THE NUMBER OF CONCEPTS UNDER TAXONOMIC RELATIONSHIP ACQUISITION

Model	Input Terms	Initial Model	Trimmed Model	Concept Hierarchy
# Concept	57	152	83	82

TABLE 3
PRECISION AND RECALL IN THE CASE STUDY WITH XCBL

	Precision	Recall per Path	Recall per Subtree
Matched Result	0.2(5/25)	0.29(5/17)	0.71(5/7)
Trimmed Result	0.22(2/9)	0.13(2/15)	0.5(2/4)

C. Non-Taxonomic Relationship Learning

1) Construction of WordSpace

High-frequency 4-grams were extracted from xCBL Document Description (about 2,500 words), and 1240 kinds of 4-grams were obtained. In order to keep density of a collocation matrix high, the extraction frequency of 4-grams must be adjusted according to the scale of text corpus. As xCBL text is relatively short, the extraction frequency was set as 2 times this case. In order to construct a context vector, a sum of 4-gram vectors around appearance place circumference of each of 57 concepts was calculated. In order to construct a context scope from some 4-grams, it consists of putting together 10 4-grams before the 4-gram and 10 4-grams after the 4-grams independently of length of a sentence. For each of 57 concepts, the sum of context vectors in all the appearance places of the concept in xCBL was calculated, and the vector representations of the concepts were obtained. The set of these vectors is used as WordSpace to extract concept pairs with context similarity. Having calculated the similarity from the inner product for concept pairs which is all the combination of 57 concepts, 40 concept pairs were extracted.

2) Finding Associations between Input Terms

DODDLE II extracted 39 pairs of terms from text corpus using the above-mentioned association rule algorithm. There are 13 pairs out of them in a set of similar concept pairs extracted using WordSpace. Then, DODDLE II constructed concept specification templates from two sets of concept pairs extracted by WordSpace and Associated Rule algorithm. However, the user didn't have enough time to modify them and didn't finish modifying them.

3) Evaluation of Results of NTRL module

The user evaluated the following two sets of concept pairs: one is extracted by WordSpace (WS) and the other is extracted by Association Rule algorithm (AR). Fig. 5 shows two different sets of concept pairs from WS and AR. It also shows portion of extracted concept pairs that were accepted by the user. Table 4 shows the details of evaluation by the user, computing precision only. Because the user didn't define concept definition in advance, we can not compute recall. Looking at the field of precision in Table 4, the precision from WS is higher than others. Most of concept pairs which have relationships were extracted by WS. The percentage is about 77% (30/39). But there are some concept pairs which were not extracted by WS. Therefore taking the join of WS and AR is the best method to support a user to construct non-taxonomic relationships.

TABLE 4
EVALUATION BY THE USER WITH XCBL DEFINITION

	WordSpace (WS)	Association Rules (AR)	The Join of WS and AR
# Extracted concept pairs	40	39	66
# Accepted concept pairs	30	20	39
# Rejected concept pairs	10	19	27
Precision	0.75(30/40)	0.51(20/30)	0.59(39/66)

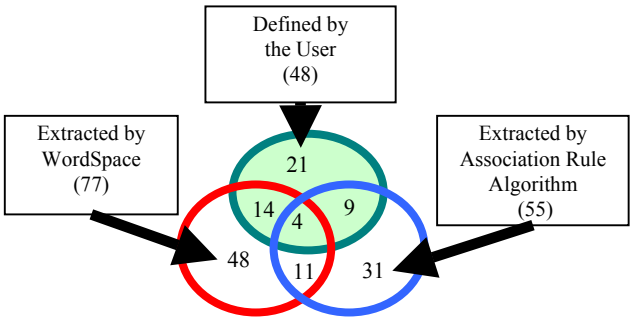


Fig. 5. Two Different Sets of Concept Pairs from WS and AR and Concept Sets have Relationships

D. *Results and Evaluation of the Case Study*

In regards to support in constructing taxonomic relationships, the precision and recall are less than 0.3 in the case study. Generally, 70 % or more support comes from TRA module. About more than half portion of the final domain ontology results in the information extracted from WordNet. Because the two strategies just imply the part where concept drift may come up, the part generated by them has about 30 % hit rate. So one out of three indications based on the two strategies work well in order to manage concept drift. Since the two strategies use matched and trimmed results, based on structural information of an MRD only, the hit rates are not so bad. In order to manage concept drift smartly, we may need to use more semantic information that is not easy to come up in advance in the strategies, and we also may need to use domain specific text corpus and other information resource to improve supporting a user in constructing taxonomic relationships.

In regards to construction of non-taxonomic relationships, the precision in the case study with xCBL is good. Generating non-taxonomic relationships of concepts is harder than modifying and deleting them. Therefore, DODDLE II supports the user in constructing non-taxonomic relationships.

After analyzing results of the case study, we have the following problems:

- Determination of a Threshold: Threshold of the context similarity changes in effective value with domain. It is hard to set up the most effective value in advance.
- Specification of a Concept Relation: Concept specification templates have only concept pairs based on the context similarity, it still requires high cost to specify relationships between them. It is needed to support specification of concept relationships on this system in the future work.
- Ambiguity of Multiple Terminologies: For example, the term “transmission” is used in two meanings, “transmission (of goods)” and “transmission (of communication)”, in the xCBL document. However, DODDLE II considers these terms as the same and creates WordSpace as it is. Therefore constructed vector expression may not be exact. In order to extract more useful concept pairs, semantic specialization of a multi-sense word is necessary, and it should be considered that the 4-grams with same appearance and different meaning are different 4-grams.

IV. Conclusions and Future Work

In this paper, we discussed how to construct domain ontologies using an existing MRD and text corpus. In order to acquire taxonomic relationships, two strategies have been proposed: matched result analysis and trimmed result analysis. Furthermore, to learn non-taxonomic relationships, concept pairs may be related to concept definition, extracted on the basis of the co-occurrence information in text corpus, and a domain ontology is developed by the modification and specification of concept relations with concept specification templates. It serves as the guideline for narrowing down huge space of concept pairs to construct domain ontologies.

It is almost craft-work to construct domain ontologies, and still difficult to obtain the high support rate on the system. DODDLE II mainly supports for construction of a concept hierarchy with taxonomic relationships and extraction of concept pairs with non-taxonomic relationships. However, a support for specification concept relationship is indispensable.

As a future work, we are trying to find out the way to extend DODDLE II into DODDLE-R (DODDLE RDF model extension). In the recent stream of ontology engineering towards the Semantic Web, the relation between meta-models of Web resources represented in RDF (Resource Description Framework) [7] and RDFS (RDF Vocabulary Description Language) [8] (as a kind of ontology for particular Web resources) are gathering more attention than before.

Fig. 8 shows the general procedure of DODDLE-R. In

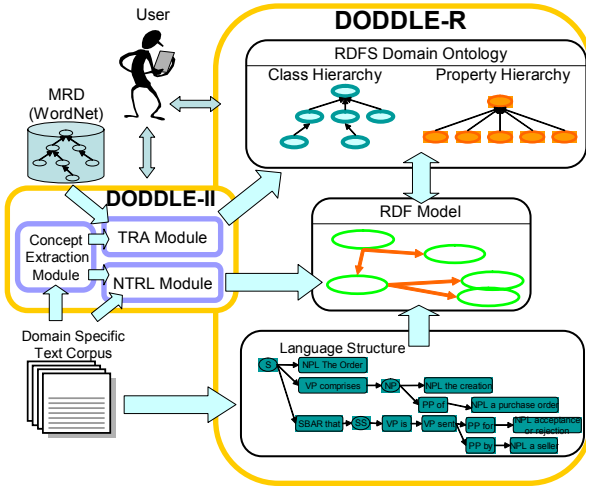


Fig. 6. General Procedure of DODDLE-R

addition to DODDLE-II, DODDLE-R generates natural language structures from text corpus. Then, based on the structures and non-taxonomic relationships produced by NTRL, the prototype of RDF model is built up. Also taxonomic relationships are constructed by using TRA and they become the basis of RDFS class hierarchy. After that, to build up and improve the RDF model and RDFS class hierarchy based on the prototypes as mentioned above, it is necessary to manage their relation. To do that, and also to improve the interaction process with users, the combination with MR3 [9], a state-of-the-art

RDF(S) management tool, must be essential. Furthermore, the strategy to manage the semantic and granularity gaps between RDFS class hierarchy, RDF model and natural language structures would be the key issue of this research work.

ACKNOWLEDGEMENT

This work was supported by Masaki Kurematsu (Iwate Prefectual University, Japan), Naomi Nakaya and Takamasa Iwade (former students of Shizuoka University, Japan).

REFERENCES

[1] Y. Ding and S.Foo, "Ontology Research and Development, Part 1 – A Review of Ontology", Journal of Information Science, Vol.28, No2, 123 – 136 (2002)

[2] Rieko Sekiuchi, Chizuru Aoki, Masaki Kurematsu and Takahira Yamaguchi, "DODDLE: A Domain Ontology Rapid Development Environment", PRICAI98, 1998

[3] Marti A. Hearst, Hirsch Schutze, "Customizing a Lexicon to Better Suit a Computational Task", in Corpus Processing for Lexical Acquisition edited by Branimir Boguraev & James Pustejovsky, 77–96

[4] C.Fellbaum ed, "WordNet", The MIT Press, 1998. See also URL: http://www.cogsci.princeton.edu/~wn/

[5] Rakesh Agrawal, Ramakrishnan Srikant, "Fast algorithms for mining association rules," Proc. of VLDB Conference, 487–499 (1994)

[6] xCBL.org, http://www.xcbl.org/xcbl40/documentation/listofdocuments.html

[7] Resource Description Framework (RDF) , http://www.w3.org/RDF/

[8] RDF Vocabulary Description Language 1.0 RDF Schema, http://www.w3.org/TR/rdf-schema/

[9] Noriaki Izumi, Takeshi Morita, Naoki Fukuta and Takahira Yamaguchi, "RDF-based Meta-Model Management Environment", Sanken (ISIR) International Symposium, 2003

Classification Rule Discovery with Ant Colony Optimization

Bo Liu[1], Hussein A. Abbass[2], and Bob McKay[2]

Abstract—**Ant-based algorithms or ant colony optimization (ACO) algorithms have been applied successfully to combinatorial optimization problems. More recently, Parpinelli and colleagues applied ACO to data mining classification problems, where they introduced a classification algorithm called Ant_Miner. In this paper, we present an improvement to Ant_Miner (we call it Ant_Miner3). The proposed version was tested on two standard problems and performed better than the original Ant_Miner algorithm.**

I. INTRODUCTION

Knowledge discovery in databases (KDD) is the process of extracting models and patterns from large databases. The term *data mining* (DM) is often used as a synonym for the KDD process, although strictly speaking it is just a step within KDD. DM refers to the process of applying the discovery algorithm to the data. In [5], KDD is defined as

"… the process of model abstraction from large databases and searching for valid, novel, and nontrivial patterns and symptoms within the abstracted model".

Rule Discovery is an important data mining task since it generates a set of symbolic rules that describe each class or category in a natural way. The human mind is able to understand rules better than any other data mining model. However, these rules need to be simple and comprehensive; otherwise, a human won't be able to comprehend them. Evolutionary algorithms have been widely used for rule discovery, a well known approach being learning classifier systems.

To our knowledge, Parpinelli, Lopes and Freitas [4] were the first to propose Ant Colony Optimization (ACO) for discovering classification rules, with the system *Ant-Miner*. They argue that an ant-based search is more flexible and robust than traditional approaches. Their method uses a heuristic value based on entropy measure.

[1] Department of Computer Science, JINAN University, Guangzhou, China, 510632. lbxldd@sohu.com

[2] School of Computer Science, University of New South Wales, Australia, ACT 2600. rim@cs.adfa.edu.au

In [9], we presented a modified version of Ant-Miner (i.e. Ant-Miner2), where the core computation heuristic value was based on a simple density estimation heuristic. In this paper, we present a further study and introduce another ant-based algorithm, which uses a different pheromone updating strategy and state transition rule. By comparison with the work of Parpinelli et al, our method can improve the accuracy of rule lists.

The remainder of the paper is organized as follow. In section 1, we present the basic idea of the ant colony systems. In section 2, the Ant_Miner algorithm (Rafael S.Parpinelli et al, 2000) is introduced. In section 3, the density based Ant_miner2 is explained. In section 4, our further improved method (i.e.Ant_Miner3) is shown. Then the computational results are reported in section 5. Finally, we conclude with general remarks on this work and further directions for future research.

II. ANT COLONY SYSTEM (ACS) AND ANT_MINER

Ant Colony Optimization (ACO) [2] is a branch of a newly developed form of artificial intelligence called *swarm intelligenc*e. Swarm intelligence is a field which studies "the emergent collective intelligence of groups of simple agents" [1]. In groups of insects, which live in colonies, such as ants and bees, an individual can only do simple tasks on its own, while the colony's cooperative work is the main reason determining the intelligent behavior it shows. Most real ants are blind. However, each ant while it is walking, deposits a chemical substance on the ground called pheromone [2]. Pheromone encourages the following ants to stay close to previous moves. The pheromone evaporates over time to allow search exploration. In a number of experiments presented in [3], Dorigo and Maniezzo illustrate the complex behavior of ant colonies. For example, a set of ants built a path to some food. An obstacle with two ends was then placed in their way such that one end of the obstacle was more distant than the other. In the beginning, equal numbers of ants spread around the two ends of the obstacle. Since all ants have almost the same speed, the ants going around the nearer end of the obstacle return before the ants going around the farther end (differential path effect). With time, the amount of pheromone the ants deposit increases more rapidly on the shorter path, and so more ants prefer this path. This positive effect is called autocatalysis. The difference between the two paths is called the preferential path effect; it is the result of the differential deposition of pheromone between the two sides of the obstacle, since the ants

following the shorter path will make more visits to the source than those following the longer path. Because of pheromone evaporation, pheromone on the longer path vanishes with time.

The goal of Ant-Miner is to extract classification rules from data (Parpinelli et al., 2002). The algorithm is presented in Figure 1.

```
Training set = all training cases;
WHILE (No. of cases in the Training set > max_uncovered_cases)
        i=0;
        REPEAT
                i=i+1;
                Ant_i incrementally constructs a classification rule;
                Prune the just constructed rule;
                Update the pheromone of the trail followed by Ant_i;
        UNTIL (i ≥ No_of_Ants) or (Ant_i constructed the same rule as the previous No_Rules_Converg-1 Ants)
        Select the best rule among all constructed rules;
        Remove the cases correctly covered by the selected rule from the training set;
END WHILE
```

Figure 1. Overview of Ant-Miner (Parepinelli et al., 2002)

A. Pheromone Initialization

All cells in the pheromone table are initialized equally to the following value:

$$\tau_{ij}(t=0) = \frac{1}{\sum_{i=1}^{a} b_i} \tag{1}$$

where a is the total number of attributes, bi is the number of values in the domain of attribute i.

B. Rule Construction

Each rule in Ant-Miner contains a condition part as the antecedent and a predicted class. The condition part is a conjunction of attribute-operator-value tuples. The operator used in all experiments is "=" since in Ant-Miner2, just as in Ant-Miner, all attributes are assumed to be categorical. Let us assume a rule condition such as $term_{ij} \approx A_i=V_{ij}$, where A_i is the i^{th} attribute and V_{ij} is the j^{th} value in the domain of A_i. The probability, that this condition is added to the current partial rule that the ant is constructing, is given by the following Equation:

$$P_{ij}(t) = \frac{\tau_{ij}(t)\cdot\eta_{ij}}{\sum_{i}^{a}\sum_{j}^{b_i}\tau_{ij}(t)\cdot\eta_{ij}, \forall i \in I} \tag{2}$$

where η_{ij} is a problem-dependent heuristic value for term-ij, τ_{ij} is the amount of pheromone currently available (at time t) on the connection between attribute i and value I is the set of attributes that are not yet used by the ant.

C. Heuristic Value

In traditional ACO, a heuristic value is usually used in conjunction with the pheromone value to decide on the transitions to be made. In Ant-Miner, the heuristic value is taken to be an information theoretic measure for the quality of the term to be added to the rule. The quality here is measured in terms of the entropy for preferring this term to the others, and is given by the following equations:

$$\eta_{ij} = \frac{\log_2(k) - InfoT_{ij}}{\sum_{i}^{a}\sum_{j}^{b_i}\log_2(k) - InfoT_{ij}} \tag{3}$$

$$InfoT_{ij} = -\sum_{w=1}^{k}\left[\frac{freqT_{ij}^{w}}{|T_{ij}|}\right] * \log_2\left[\frac{freqT_{ij}^{w}}{|T_{ij}|}\right] \tag{4}$$

wherek is the number of classes, |Tij| is the total number of cases in partition Tij (partition containing the cases where attribute Ai has value Vij), freqTwij is the number of cases in partition Tij with class w, a is the total number of attributes, and bi is the number of values in the domain of attribute i

The higher the value of $infoT_{ij}$, the less likely that the ant will choose $term_{ij}$ to add to its partial rule.

D. Rule Pruning

Immediately after the ant completes the construction of a rule, rule pruning is undertaken to increase the comprehensibility and accuracy of the rule. After the pruning step, the rule may be assigned a different predicted class based on the majority class in the cases covered by the rule antecedent. The rule pruning procedure iteratively removes the term whose removal will cause a maximum increase in the quality of the rule. The quality of a rule is measured using the following equation:

$$Q = \left(\frac{TruePos}{TruePos + FalseNeg}\right) \times \left(\frac{TrueNeg}{FalsePos + TrueNeg}\right) \tag{5}$$

where TruePos is the number of cases covered by the rule and having the same class as that predicted by the rule, FalsePos is the number of cases covered by the rule and having a different

class from that predicted by the rule, FalseNeg is the number of cases that are not covered by the rule, while having the class predicted by the rule, TrueNeg is the number of cases that are not covered by the rule which have a different class from the class predicted by the rule.

E. Pheromone Update Rule

After each ant completes the construction of its rule, pheromone updating is carried out as follows:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t).Q \ , \ \forall \ \text{term}_{ij} \in \text{the rule} \quad (6)$$

To simulate the phenomenon of pheromone evaporation in real ant colony systems, the amount of pheromone associated with each term_{ij} which does not occur in the constructed rule must be decreased,. The reduction of pheromone of an unused term is performed by dividing the value of each τ_{ij} by the summation of all τ_{ij}.

III. Density based Ant_Miner2

In [9], we proposed an easily computable density estimation equation (7) instead of equation (3). It derived from the view that the ACO algorithm does not need accurate information in its heuristic value, since the pheromone should compensate for small potential errors in the heuristic values. In other words, a simpler heuristic value may do the job as well as a complex one. This simple heuristic produced equivalent results to the entropy function.

$$\eta_{ij} = \frac{majority\_classT_{ij}}{|T_{ij}|} \quad (7)$$

where: $\text{majority\_class}T_{ij}$ is the majority class in partition T_{ij}

IV. Further Proposed System (Ant_Miner3)

It is reasonable that ants select terms according to equation (1); in other words, determined by pheromone amount and heuristic function η which measures the predictive power of a term. But in the above methods, the pheromone of each term is changed after an ant constructs a rule, while η is always the same, so that the next ant tends to choose terms used in the previous rule, whose pheromone is increased, and is unlikely choose unused terms, whose pheromone is decreased. Consequently, the ants converge to a single constructed rule too quickly. This leads to a failure to produce alternative potential rules.

In [9], we showed that Ant-Miner2 is computationally less expensive than the original Ant-Miner1, since in its innermost loop, it uses simple division instead of the logarithm as in Ant-Miner. To be more precise, each heuristic value in Ant-Miner1 requires 2 divisions, 1 multiplication and 1 calculation of the logarithm, whereas Ant-Miner2 requires a single division. This saving in computational time did not change the accuracy of the method and did not require additional iterations.

In the following, we propose a new pheromone updating method and a new state transition rule to increase the accuracy of classification by ACO.

A. Our Pheromone Update Method

After an ant constructs a rule, the amount of pheromone associated with each term that occurs in the constructed rule is updated by equation (8), and the pheromone of unused terms is updated by normalization.

Note that Q varies in the range [0, 1]. The higher Q is, the larger the amount of pheromone associated with each used term. On the other hand, if Q is very small (close to zero), the pheromone level associated with each used term will decrease.

$$\tau_{ij}(t) = (1-\rho)\cdot\tau_{ij}(t-1) + (1-\frac{1}{1+Q})\cdot\tau_{ij}(t-1) \quad (8)$$

where ρ is the pheromone evaporation rate, Q is quality of the constructed rule, ρ is the pheromone evaporation rate, which controls how fast the old path evaporates. This parameter controls the influence of the history on the current pheromone trail [6]. In our method, a large value of ρ indicates a fast evaporation and vice versa. We fix it at 0.1 in our experiments.

B. Choice of Transition

Ants can be regarded as cooperative agents in an ant colony system. These intelligent agents inhabit an environment without global knowledge, but they could benefit from the update of pheromones [7]. Pheromones placed on the edges in ACS play the role of a distributed long-term memory. This memory is not stored within the individual ants, but is distributed on the edges of the route, which allows an indirect form of communication. This benefits exploitation of prior knowledge. But it increases the probability of choosing terms belonging to the previously discovered rules according to equation (2), thus inhibiting the ants from exhibiting a bias toward exploration. In order to enhance the role of exploration, we apply the transition rule shown in figure 3 in choosing

```
If q1≤φ
  Loop
    If q2≤ Σ_{j∈J_i} P_ij
    Then choose term_ij
  Endloop
Else
    Choose term_ij with max P_ij
```

$term_{ij}$,

Figure 3. The proposed State Transition Rule

Where q1 and q2 are random numbers, φis a parameter in [0, 1], J_i is the number of i-th attribute values, P_{ij} is possibility calculated using equation (2).

Therefore, the result depends not only on the heuristic functions η_{ij} and pheromone τ_{ij}, but also on a random number, which increases the likelihood of choosing terms not nused in previously constructed rules. More precisely, q1≥φ corresponds to an exploitation of the knowledge available about the problem, whereas q1≤φ favors more exploration. φ is tunable for controlling exploration. In our experiments, it is set to 0.4.

C. Diversity comparison

The total number of terms provided for constructing a rule is $\sum_{i}^{a} b_i$, where b_i is the number of values in the domain of attribute i, a is the total number of attributes of a training set.

In Ant_Miner, the expected value of the chosen paths is :

$$E1=\sum_{i}^{a} b_i \max_{i,j}\{p_{ij}\};$$

In Ant_Miner3, the expected value of the chosen paths is :

$$E2=1-\varphi\sum_{i}^{a} b_i \max_{i,j}\{p_{ij}\}+\varphi\sum_{i}^{a} b_i \sum_{j}^{b_i} P_{ij}$$

that is,

$$E2=\sum_{i}^{a} b_i \max_{i,j}\{p_{ij}\}+\varphi\sum_{i}^{a} b_i \sum_{j}^{b_i} P_{ij} - \sum_{i}^{a} b_i \max_{i,j}\{p_{ij}\}.$$

Obviously, E2>E1.

V. Experimental Results

Our experiments used two data sets from the UCI data set repository[8]: the Wisconsin breast cancer database, which contains 699 instances, 9 integer-valued attributes and 2 classes (malignant and benign); and the Tic_tac_toe endgame database, which contains 958 instances, 9 numerate-valued attributes and 2 classes (won and lost). We evaluate comparative performance of the proposed method and Ant_Miner1 using ten-fold cross-validation. Each Database is divided into ten partitions, and each method is run ten times, using a different partition as test set each time, with the other nine as training set.

We use the rule list produced by a training set to predict the class of each case in the test set, the accuracy rate being calculated according to equation (9) (where the meanings of TruePos, Trueneg, FalseNeg, FalsePos are as in equation (5)). Every rule list includes a default rule, which has no condition and takes as its class the majority class in the set of training cases, so that we can apply the default rule if none of the rules in the list covers test case.

Table 1 shows accuracy rates for the rule sets produced by Ant_miner1 and Ant_Miner3 for ten runs on the two datasets. The mean accuracy rate and mean number of rule sets produced are reported in Table 2. It can be seen that Ant_Miner3 discovers somewhat more rules than Ant_Miner1, but the mean accuracy of the rule sets discovered by Ant_Miner3 is higher than Ant_Miner1. We conducted student's one-tail t-tests on the differences between the means of the two datasets. The significance level is 0.04 for the Breast Cancer Database and 0.004 for Tic-tac-toe: the differences are statistically significant. This shows that if ants explore a greater variety of different paths, then there is a higher probability that one of them will find an improved solution compared with the case in which they all converge to the same tour.

Table 1. Test Set Accuracy Rate (%)

	Breast Cancer		Tic_tac_toe	
Run Number	Ant_ Miner1	Ant_ Miner3	Ant_ Miner1	Ant_ Miner3
1	92.05	94.32	71.28	82.97
2	93.15	93.15	73.40	72.34
3	91.67	91.67	67.37	78.94
4	95.59	97.06	71.58	80.00
5	88.41	92.75	68.42	72.63
6	94.20	95.65	75.79	80.00
7	90.77	93.84	74.74	81.05
8	96.55	96.55	65.26	74.74
9	91.04	92.54	73.68	75.79
10	92.86	95.71	68.42	67.37

$$\text{Accuracy}=\frac{TruePos+TrueNeg}{TruePos+FalseNeg+FalsePos+TrueNeg} \quad (9)$$

Table 2. Mean accuracy rate and mean number of rule lists

	Breast Cancer		Tic_tac_toe	
Valuation item	Ant_ Miner1	Ant_ Miner3	Ant_ Miner1	Ant_ Miner3
Accuracy rate(%)	92.63	94.32	70.99	76.58
#_rules	10.1	13.2	16.5	18.58

Although Ant_Miner3 requires marginally more ants to find a solution, the density-based-heuristic computational method compensates for ant searching time. In practice, Ant_miner1 and Ant_miner3 required almost identical running time.

VI. Conclusion

Decision tree induction is perhaps the best known method of finding rules in databases. In [4], it is demonstrated that Ant-Miner1 produces a higher accuracy rate and fewer rules than decision-tree induction (C4.5). In this paper, a new method based on a variant of Ant_Miner1 is proposed. We compare the results of both methods and found that our method

features a higher accuracy rate than Ant_Miner1. The main contributions of the work are the following:

1. Our method incorporates a tunable stochastic element when constructing a rule, and so provides a balance between exploitation and exploration in its operation. This more accurately models the behavior of real ants, but also, because it leads to a greater diversity of path choices, assists in finding an optimal rule.

2. A different strategy for controlling the influence of pheromone values was studied. We proposed a pheromone update rule which can cause future ants to make better decisions, i.e. improving the quality of a rule and the accuracy of rule sets.

The application of ACO in data mining is still in its early stages. In future work, we aim to further improve time efficiency.

Ant_Miner3 has a number of system parameters. Detailed experimentation is needed to determine the effects of these parameters, and develop an understanding of methods to set the parameters appropriately for particular learning problems. Such an understanding is required if ACO methods are to scale up to real-world large scale databases.

REFERENCES

[1] Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). Swarm Intelligence: From Natural to Artificial Systems. New York: Oxford University Press.

[2] Dorigo, M., & Caro, G. D. (1999). Ant Algorithms for Discrete Optimization. Artificial Life, 5(3), 137-172.

[3] Dorigo, M., & Maniezzo, V. (1996). The ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, 26(1), 1-13.

[4] Parepinelli, R. S., Lopes, H. S., & Freitas, A. (2002). An Ant Colony Algorithm for Classification Rule Discovery. In H. A. a. R. S. a. C. Newton (Ed.), Data Mining: Heuristic Approach: Idea Group Publishing.

[5] Sarker, R., Abbass, H., & Newton, C. (2002). Introducing data mining and knowledge discovery. In R. sarker & H. Abbass & C. Newton (Eds.), Heuristics and Optimisation for Knowledge Discovery (pp. 1-23): Idea Group Publishing.

[6] Luk Schoofs, Bart Naudts, Ant Colonies are Good at Solving Constraint Satisfaction Problems, Proceedings of the 2000 Congress on Evolutionary Computation, Volume:2, page 1190-1195.

[7] Ruoying Sun, Shoji Tatsumi, Gang Zhao, Multiagent Reinforcement Learning Method with An Improved Ant Colony System, Proceedings of the 2001 IEEE International Conference on Systems, Man and Cybernetics, Volume:3, 2001, page 1612-1617.

RELATED CONFERENCES, CALL FOR PAPERS, AND CAREER OPPORTUNITIES

TCCI Sponsored Conferences

WI 2004
The 2004 IEEE/WIC/ACM International Conference on Web Intelligence
Beijing, China
September 20-24, 2004
http://www.maebashi-it.org/WI04/
http://www.comp.hkbu.edu.hk/WI04/
Submission Deadline: April 4, 2004

Web Intelligence (WI) has been recognized as a new direction for scientific research and development to explore the fundamental roles as well as practical impacts of Artificial Intelligence (AI) (e.g., knowledge representation, planning, knowledge discovery and data mining, intelligent agents, and social network intelligence) and advanced Information Technology (IT) (e.g., wireless networks, ubiquitous devices, social networks, wisdom Web, and data/knowledge grids) on the next generation of Web-empowered products, systems, services, and activities. It is one of the most important as well as promising IT research fields in the era of Web and agent intelligence.

The 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI'04) will be jointly held with the 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04). The IEEE/WIC/ACM 2004 joint conferences are sponsored and organized by IEEE Computer Society Technical Committee on Computational Intelligence (TCCI), Web Intelligence Consortium (WIC), and ACM-SIGART.

Following the great successes of WI'01 held in Maebashi City, Japan and WI'03 held in Halifax, Canada, WI 2004 provides a leading international forum for researchers and practitioners (1) to present the state-of-the-arts of WI technologies; (2) to examine performance characteristics of various a pproaches in Web-based intelligent information technology; and (3) to cross-fertilize ideas on the development of Web-based intelligent information systems among different domains. By idea-sharing and discussions on the underlying foundations and the enabling technologies of Web intelligence, WI 2004 will capture current important developments of new models, new methodologies and new tools for building a variety of embodiments of Web-based intelligent information systems.

IAT 2004
The 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology
Beijing, China
September 20-24, 2004
http://www.maebashi-it.org/IAT04/
http://www.comp.hkbu.edu.hk/IAT04/
Submission Deadline: April 4, 2004

The 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04) will be jointly held with the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI'04). The IEEE/WIC/ACM 2004 joint conferences are sponsored and organized by IEEE Computer Society Technical Committee on Computational Intelligence (TCCI), Web Intelligence Consortium (WIC), and ACM-SIGART. The upcoming meeting in this conference series follows the great success of IAT-99 held in Hong Kong in 1999, IAT-01 held in Maebashi City, Japan in 2001, IAT-03 held in Halifax, Canada.

IAT 2004 provides a leading international forum to bring together researchers and practitioners from diverse fields, such as computer science, information technology, business, education, human factors, systems engineering, and robotics, to (1) examine the design principles and performance characteristics of various approaches in intelligent agent technology, and (2) increase the cross-fertilization of ideas on the development of autonomous agents and multi-agent systems among different domains. By encouraging idea-sharing and discussions on the underlying logical, cognitive, physical, and sociological foundations as well as the enabling technologies of intelligent agents, IAT 2004 will foster the development of novel paradigms and advanced solutions in agent-based computing.

ICDM'04
The Fourth IEEE International Conference on Data Mining
Brighton, UK
November 1-4, 2004
http://icdm04.cs.uni-dortmund.de/
Submission Deadline: June 1, 2004

The IEEE International Conference on Data Mining (ICDM) provides a leading international forum for the sharing of original research results and practical development experiences among researchers and application developers from different data mining related areas such as machine learning, automated scientific discovery, statistics, pattern recognition, knowledge acquisition, soft computing, databases and data warehousing, data visualization, and knowledge-based systems. The conference seeks solutions to challenging problems facing the development of data mining systems, and shapes future directions of research by promoting high quality, novel and daring research findings. As an important part of the conference, the workshops program will focus on new research challenges and initiatives, and the tutorial program will cover emerging data mining technologies and the state-of-the-art of data mining developments.

In addition to business oriented data mining, ICDM has an equal emphasis on engineering, scientific, and medical data for which domain knowledge plays a significant role in knowledge discovery and refinement.

ICDM is held annually, in different regions of the world.

Topics related to the design, analysis and implementation of data mining theory, systems and applications are of interest. See the conference Web site for more information.

KGGI'04
The Second International Workshop on Knowledge Grid and Grid Intelligence
Beijing, China
September 20, 2004
http://kg.ict.ac.cn/kggi04/default.htm
Submission Deadline: June 10, 2004

The purpose of this workshop is to bring researchers and practitioners to identify and explore the issues, opportunities, and solutions for Knowledge Grid and Grid Intelligence. It will provide a forum for free exchange of ideas and will be featured by invited talks and refereed paper presentations. Authors are invited to submit regular papers, reports on work in progress, and position papers. Appropriate topics include, but are not limited to, the following: Knowledge Grid, Semantic Grid and Semantic Web, Grid, Web and Self-organized Intelligence, Data/Information/Knowledge/Service, Integration, Mediation and Middleware , Ontology, Knowledge Discovery, Distributed Knowledge Management, Cooperative Teamwork and Agent-based Workflow, Web/Grid-based Decision Making, Self-organizing Systems and Emergent Organization,, Computational Economy on Grid, Web and Grid Dynamics, and Applications in e-Science, e-Business, and e-Government.

Other Computational Intelligence Conferences

AAMAS'04
The Third International Joint Conference on Autonomous Agents and Multi-Agent Systems
New York City, USA
July 19-21, 2004
http://satchmo.cs.columbia.edu/aamas04/

Agents are one of the most prominent and attractive technologies in computer science at the beginning of the new milennium. The technologies, methods, and theories of agents and multiagent systems are currently contributing to many diverse domains such as information retrieval, user interfaces, electronic commerce, robotics, computer mediated collaboration, computer games, education and training, ubiquitous computing, and social simulation. They are not only a very promising technology, but they are also emerging as a new way of thinking: a conceptual paradigm for analyzing problems and for designing systems, for dealing with complexity, distribution, and interactivity while providing a new perspective on computing and intelligence. The AAMAS conferences aim to bring together the world's researchers active in this important, vibrant, and rapidly growing field.

The AAMAS conference series was initiated in 2002 as a merger of three highly respected individual conferences: AGENTS (International Conference on Autonomous Agents) ICMAS (International Conference on Multi-Agent Systems), and ATAL (International Workshop on Agent Theories, Architectures, and Languages) The aim of the joint conference is to provide a single, high-profile, internationally renowned forum for research in the theory and practice of autonomous agents and multiagent systems. The first two AAMAS conferences (AAMAS-2002, Bologna, Italy and AAMAS-2003, Melbourne, Australia) are significant events in the academic history of agent systems. We expect AAMAS-04 to build on these successes and stand out as a key date on the international computing research calendar.

EEE'04
The 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service
Taipei, Taiwan
March 28-31, 2004
http://bikmrdc.lm.fju.edu.tw/eee04/

The 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE-04) aims to bring together researchers and developers from diverse areas of computing, developers and practitioners to explore and address the challenging research issues on e-technology in order to develop a common research agenda and vision for e-commerce and e-business. The main focus of the conference is on the enabling technologies to facilitate next generation, intelligent e-Business, e-Commerce and e-Government. The conference solicits research papers as well as proposals for tutorials on all aspects of e-Commerce, e-Business, and e-Service.

SDM'04
The Fourth SIAM International Conferenceon Data Mining
Orlando, Florida, USA
April 22-24, 2004
http//www.siam.org/meetings/sdm04/

The Fourth SIAM International Conference on Data Mining will provide a forum for the presentation of peer-reviewed recent results in all aspects of data mining. The conference program will also include keynote talks and tutorials by leading experts, as well as several workshops on topics of current interest on the final day of the conference.

The Fourth SIAM International Conference on Data Mining will provide several workshops on topics of current interest in data mining on April 22, 2004.

Workshop on clustering high dimensional data and its application
http://www.cs.utexas.edu/users/inderjit/sdm04.html
Workshop on mining scientific and engineering datasets
http://www.cs.colorado.edu/∼mburl/MSD04/
Workshop on high performance and distributed data mining
http://www.cis.ohio-state.edu/∼agrawal/hpdm04.html
Workshop on bioinformatics
http://www.ist.temple.edu/sbw04/
Workshop on data mining in resource constrained environments
http://ic.arc.nasa.gov/siam/
Workshop on link analysis, counter-terrorism and privacy preserving data mining
http://www.cs.umn.edu/ aleks/sdm04w/

Proceedings of the conference will be available both online at the SIAM Web site and in hard copy form. The online proceedings of the last year's SIAM data mining conference, which was held in San Francisco, CA, is available at: http://www.siam.org/meetings/sdm03.

ISWC2004
The Third International Semantic Web Conference
Hiroshima, Japan
7-11 November, 2004
http://iswc2004.semanticweb.org/

The vision of the Semantic Web is to make the contents of the Web unambiguously computer interpretable, enabling automation of a diversity of tasks currently performed by human beings. The goal of providing semantics and automated reasoning capabilities to the Web draws upon research in a broad range of areas including Artificial Intelligence, Databases, Software Engineering, Distributed Computing and Information Systems. Contributions to date have included languages for semantic annotation of Web documents, automated reasoning capabilities for Web languages, ontologies, query and view languages, semantic translation of Web contents, semantic integration middleware, technologies and principles for building multi-agent and Grid systems, semantic interoperation of programs and devices, technologies and principles for describing, searching and composing Web Services, and more.

The 3rd International Semantic Web Conference (ISWC2004) follows on the success of previous conferences and workshops in Sanibel Island, USA (2003), Sardinia, Italy (2002), and Stanford, USA (2001).

The organizing committee solicits research submissions for the main research track of the conference, as well as for the accompanying industrial and posters track: Research Track, Industrial Track and Posters Track.

Call For Papers

The IEEE Computational Intelligence Bulletin

Special Issue on Detection of Malicious Attacks

Aim

This special issue will be devoted to the detection of malicious attacks using AI techniques. A malicious attack could be done by various means: malicious code injection, malicious software, unauthorized access, etc.

Software detection mechanisms currently deployed are mostly based on signature techniques. They are ineffective for unknown malicious code and many other forms of attacks requiring constant maintenance to remain effective. Detection mechanisms based on role descriptions can deal with unknown attacks but require manual intervention. Artificial intelligence techniques and algorithms can offer higher level solutions avoiding these shortcomings.

Scope

Contributions should discuss the application of artificial intelligence techniques to the detection of malicious attacks of various forms (e.g. malicious code, unauthorized access). The detection could be network based, host based, or across network systems. The paper should contain experimental data and/or a clearly presented algorithmic approach.

Original papers are solicited in, but not limited to, the following AI techniques applied to malicious attacks detection:

- Expert systems
- Multiagent systems
- Neural networks
- Genetic algorithms
- Fuzzy logic
- Knowledge representation
- Learning algorithm

Important Dates:

Submission deadline:	May 15th, 2004
Notification of acceptance:	Aug 15th, 2004
Final papers:	Sep 30th, 2004
Publication of special issue:	Dec 2004

Submission of Manuscripts:

Paper submission should be sent, via email, to the Guest Editor. The accepted formats are MS Word and LaTeX according to the submission guidelines described at:

`http://www.comp.hkbu.edu.hk/~cib/submission.html`

We would appreciate an intent of submission, in the form of an abstract via email, as early as possible to plan the reviewing process.

For further information please contact the Guest Editor or Feature Article Editor.

Guest Editor:

Dr. Mario Latendresse
Science and Technology Advancement Team
Fleet Numerical Meteorology and Oceanography Center
U.S. Navy
7 Grace Hopper Ave. Stop 1
Monterey, CA 93943-5501 USA
Tel: (831) 656-4826
Fax: (831) 656-4363
Email: mario.latendresse.ca@metnet.navy.mil

Feature Article Editor:

Dr. Michel Desmarais
Computer Engineering,
Ecole Polytechnique de Montreal
P.O. Box 6079, Station Centre-Ville
Montreal (Quebec) Canada
Tel: (514) 340-4711 ext:3914
Email: michel.desmarais@polymtl.ca

Post-Doctoral Research Fellows in Multi-Agent Systems and Grid Computing

Centre for E-Transformation Research (CETR) at Hong Kong Baptist University (URL: http://www.comp.hkbu.edu.hk/~wic-hk) is seeking Post-Doctoral Research Fellows to join in a government-funded research project focusing on advanced research (e.g., novel algorithms and computational architectures) in the areas of multi-agent systems, data mining, and grid computing, with scalable applications to e-business and e-learning.

The positions will be funded for three years. Other major strategic collaborative partners include The Hong Kong University of Science and Technology and E-business Technology Institute, The University of Hong Kong. Close scientific collaborations with other overseas and mainland research institutions will be needed.

With a Ph.D. degree in a relevant area, applicants for the positions must have a demonstrated research record and experiences in web intelligence, multi-agent systems, and/or data mining. Experiences in other areas such as web services, semantic grid, and/or mobile/pervasive computing are also desirable.

Salary will be commensurate with qualifications and experiences.

The positions are immediately available. An initial 9-month contract will be offered with the possibility of being renewed for up to three years.

Interested applicants are invited to send a CV and two letters of recommendation to:

Prof. Jiming Liu
Director, Centre for E-Transformation Research
Head, Department of Computer Science
Hong Kong Baptist University
Kowloon Tong
Hong Kong
Email: jiming@comp.hkbu.edu.hk
URL: http://www.comp.hkbu.edu.hk/~jiming

IEEE Computer Society
1730 Massachusetts Ave, NW
Washington, D.C. 20036-1903

Non-profit Org.
U.S. Postage
PAID
Silver Spring, MD
Permit 1398