THE IEEE
Intelligent Informatics
BULLETIN

IEEE Computer Society
Technical Committee
on Intelligent Informatics

December 2004 Vol. 4 No. 2 (ISSN 1727-5997)

Profile

Conference Reports

Feature Articles

Announcements

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

IEEE Computer Society Technical Committee on Intelligent Informatics (TCII)

The Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society deals with tools and systems using biologically and linguistically motivated computational paradigms such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality.

If you are a member of the IEEE Computer Society, you may join the TCII without cost. Just fill out the form at http://computer.org/tcsignup/.

The IEEE Intelligent Informatics Bulletin

Aims and Scope

The IEEE Intelligent Informatics Bulletin is the official publication of the Technical Committee on Intelligent Informatics (TCII) 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 TCII 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 (TCII sponsored or important/related activities)

Materials suitable for publication at the IEEE Intelligent Informatics 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.

Editorial Board

Publisher: *The IEEE Computer Society Technical Committee on Intelligent Informatics*
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).*

Exploiting the Immune System for Computation

ARTIFICIAL IMMUNE SYSTEMS AT THE UNIVERSITY OF KENT

I. INTRODUCTION

Over the past five years, within The University of Kent's Computing Laboratory, a group has emerged that is investigating the natural immune system as inspiration for computation. The group has established itself as one of the world leaders in the field of Artificial Immune Systems (AIS), through their work on extracting immunological metaphors for applications in machine learning, optimisation, software testing and fault tolerance. The approach adopted is interdisciplinary, with group members that are experts in various areas of computer science, mathematics and immunology. The group at Kent has been instrumental in the establishment of the International Conference on Artificial Immune Systems (ICARIS) and a UK based academic network for AIS known as ARTIST.

The AIS group is part of the larger Applied and Interdisciplinary Informatics Group (AII) at Kent, both of which are headed by Dr. Jon Timmis. The AIS group collaborates with many industrial partners such as Sun Microsystems, Edward Jenner Institute for Vaccine Design, NCR PLC and BAE SYSTEMS, attempting to apply the AIS approach in an industrial setting.

Within the group, Jon Timmis and his team are investigating a number of avenues of research, ranging from theoretical aspects of AIS, abstraction of biologically plausible algorithms, applications of AIS technology and interactions of the immune system with neural and hormonal systems.

The immune system (IS) is a complex biological system essential for survival, which involves a variety of interacting cellular and molecular elements that control either micro- or macro-system dynamics. The strategies of the immune system are based on task distribution to obtain distributed solutions to problems with different cells able to carry out complementary tasks. Thus, cellular interactions can be envisaged as parallel and distributed processes among cells with different dynamical behavior and the resulting immune responses appear to be emergent properties of self-organising processes.

AIS can be defined as adaptive systems inspired by theoretical immunology, observed immune functions, principles and models, which are applied to problem solving. The development of AIS as a field of research has been progressing steadily over recent years. Much work has gone into the development of new algorithms inspired by the immune system for a variety of tasks, ranging from machine learning, data mining, to fault tolerance, and network intrusion detection and so on. Therefore, using immunology as a foundation, a new and exciting research field has evolved that has led to the creation of innovative applications of immune metaphors.

II. RESEARCH AT KENT

At Kent, there are a large number of people involved in AIS research. In this section we outline just some of the current research projects being undertaken within the group. This is not an exhaustive review of all themes of research currently undertaken by the group; a full list can be found at the URL at the end of this article.

A. Immunising Software

Peter May, a PhD student supervised by Prof. Keith Mander and Dr. Timmis, is developing a novel AIS based approach to software testing, in particular mutation testing. Mutation testing is an effective fault-based testing approach that uses large numbers of slightly varying versions of the program-under-test to quantify the test data's adequacy. However, the large number of "mutant" programs means this approach is computationally expensive. Peter's system aims to reduce the number of mutant programs required to ones that represent the most common errors made. Simultaneously his system will evolve high-quality test data. This co-evolutionary approach, grounded on the evolution of antibodies in the immune system, effectively gives a programming-environment specific form of mutation testing, which will hopefully reduce the computational expense associated with mutation testing.

B. Mining the Web for Interesting Pages

PhD student Andrew Secker is investigating the way immune metaphors may be employed to mine content from the web (Andrew is co supervised by Dr. Alex Freitas and Dr. Jon Timmis). Andrew's initial investigations concerned the classification of uninteresting email. At the time, this system, called AISEC, was the first email classifier to recognise that, like natural pathogens, junk email will change over time to evade standard filtering techniques. AISEC used the immune principles of constant adaptation and memory to learn the type of junk email each individual user receives and prevent that reaching his or her inbox. AISEC was shown to be very effective and has since been developed into a user application and is just undergoing readiness for deployment as free software.

Andrew's current investigations are concerned with the mining of interesting information from websites. This is the discovery of relevant pages (like traditional search engines) where these pages must also offer surprising or novel information. He employs an AIS to intelligently follow links to find pages on the assumption that good pages may link to other good pages, and assess a measure of "interestingness" for each one. He adopts the metaphor of interesting webpages to be pathogens and the internet as tissue. His system then allows immune cells into that tissue to find the pathogens (interesting web pages). The immune cells make decisions about where to go to next by choosing hyperlinks on pages based on their affinity (similarity) with the text surrounding each hyperlink. The cells may clone based on how interesting the page is likely to be and may mutate to absorb features of that interesting page. Figure 1, shows the example output from the system.

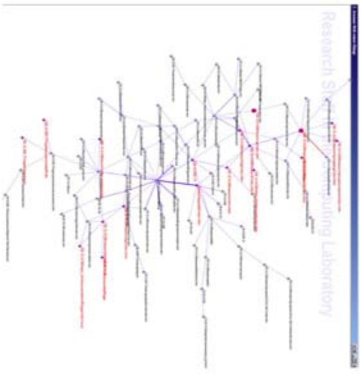


Figure 1 – Output from the Immune Web Miner

C. Immune Memory and Learning

How the immune system learns about and remembers invading pathogens is of great interest. Researchers at Kent have developed a number of immune inspired algorithms, both for supervised and unsupervised machine learning, that capitalise on the memory cells in the immune system.

Andrew Watkins, a PhD student in the group, developed the Artificial Immune Recognition System (AIRS) algorithm, one of the first immune-inspired supervised learning algorithms. AIRS is based on the clonal selection principle, which describes how invading pathogens are defeated by the cloning and mutation of new antibodies. AIRS evolves a set of memory detectors capable of classifying unseen items, and is a supervised learning system. AIRS has recently been extended to a parallel and distributed learning system.

Among the oft-cited reasons for exploring mammalian immune systems as a source of inspiration for computational problem solving include the observations that the immune system is inherently parallel and distributed, with many diverse components working simultaneously and in cooperation to provide all of the services that the immune system provides. Very little has been done in the realm of parallel AIS--that is, applying methods to parallelize existing AIS algorithms in the hopes of efficiency (or other) gains. Two new versions of AIRS have been created, one parallel and the other distributed. Both maintain the accuracy observed in the serial version, but exhibit a significant reduction in computation time.

One theory regarding memory in immunology is known as the immune network theory. Here, B-cells interact with each other to form a meta-stable memory structure though complex interactions of stimulation and suppression. Inspired by this idea, in collaboration with Dr. Nick Ryan of the Computing Laboratory, PhD student Philip Mohr is exploiting the meta-stable properties of immune networks to create a memory structure for use in context aware systems. Philip is developing a system that will identify common behaviors of users based on contextual information such as time, location, day of the month etc. The immune network approach allows for a drastic reduction in the amount of data storage required, making it very attractive for eventual deployment on a hand-held device.

D. Theoretical Investigations

In addition to developing novel solutions to problems, it is essential to have some theoretical understanding of the algorithms. Working with Dr. Andrew Hone of the Institute of Mathematics, Statistics and Actuarial Science at Kent, Dr. Timmis and Dr. Hone are working towards developing a theoretical foundation for AIS based on non-linear dynamics. It is hoped that by employing mathematical techniques widely used for the study of biological systems, it will be possible to analyse the dynamics of AIS algorithms which will give insight into their performance and ultimate usefulness.

E. Integration of Immune, Neural and Endocrine Systems

Finally, in collaboration with Dr. Mark Neal from the University of Wales, Aberystwyth the group is examining the interactions between the immune, neural and endocrine systems and how they lead to homeostasis within an organism. This ability to achieve some kind of steady internal state is both impressive and very useful when one considers the demands of long term autonomy in robotic systems that operate in dynamic environments. Preliminary work has been undertaken to examine the regulatory role of the endocrine system on neural systems for robotic control to enable effective operation in a given domain despite high perturbations in the input space. Work is now progressing on integrating an immune system component to the robot controller to regulate growth within the system.

III. Summary

There is not enough space to explore all the exciting research currently on-going in the University of Kent, but hopefully this has given an insight into some of the activities. There is a great deal of interesting work to be done, and we have only just scratched the surface.

Contact Information
Dr. Jonathan Timmis
Computing Laboratory, University of Kent.
Canterbury. UK
Phone: +44 0 1227 823636
Email : J.Timmis@kent.ac.uk
Websites: www.cs.kent.ac.uk/~jt6
www.artificial-immune-systems.org/artist.htm

The CP-04 Workshop on CSP Techniques with Immediate Application (CSPIA)

BY ROMAN BARTÁK, CSPIA CO-ORGANIZER

The CP-03 Workshop on Immediate Applications of Constraint Programming (Cork, Ireland) started a new tradition of application oriented meetings organized together with the conference on Principles and Practice of Constraint Programming (CP). These meetings are intended as a forum on sharing and exchanging information on applications of Constraint Programming and on techniques improving applicability of constraint satisfaction in solving real-life problems. In 2004, the CP-04 Workshop on CSP Techniques with Immediate Application (CSPIA) was held on August 27, 2004 in Toronto, Canada.

The all day CSPIA-04 workshop started with an invited talk by Mark Wallace (Monash University, Australia), continued by three technical sessions, and concluded by a panel discussion. Mark Wallace's invited talk entitled Three Research Collaborations with the Transportation Industry covered Mark's experience with developing real-life applications in IC-Parc. In particular, Mark talked about logistics with depots, patrol dispatcher, and flight schedule retimer. We include just one conclusion from Mark's talk – "applications are more than algorithms" meaning that technology must meet the requirements, no arbitrary simplifications.

The first technical session consisted of two papers. The first paper by Marius C. Silaghi, Markus Zanker, and Roman Barták proposed a new framework for modeling Distributed CSP with privacy of preferences and showed how this framework helps in solving desk-mates placing problems where the students have secret preferences among their classmates. The second paper by Mats Carlsson and Nicolas Beldiceanu introduced a multiplex dispensation order generation problem, a real-life combinatorial problem in the context of analyzing of large numbers of short to medium length DNA sequences. The authors proposed a constraint model for this optimization problem.

The second technical session included two papers on search techniques and one application paper. The paper by Barry O'Sullivan, Alex Ferguson, and Eugene C. Freuder described an approach that uses knowledge about known solutions to a problem to improve search. In particular, the authors proposed to use decision tree learning to capture a structure of the solution set. This decision tree is built from a small number of known solutions and it is used to give variable ordering as well as a source of additional constraints refining further the search phase. This research was motivated by solving configuration problems. The second paper by Venkata Praveen Guddeti and Berthe Y. Choueiry proposed an improved restart strategy for randomized backtrack search applied to course assignment problems. Their technique dynamically adapts the cutoff limit to the results of the search process. The third paper by Marco Cadoli and colleagues proposed a constraint-based approach to checking finiteness of UML class diagrams.

The last technical session was devoted to interactive configuration and two papers were presented there. The first paper by Sathiamoorthy Subbarayan and his colleagues compared two approaches to complete and backtrack-free interactive product configuration. The authors experimentally showed that the approach based on a symbolic representation using Binary Decision Diagrams outperforms the natural CSP encoding where all the solutions are pre-computed in advance. The second paper by Erik van der Meer and Henrik Reif Anderson proposed a modular language for modeling interactive configuration problems. The authors presented semantics of this language and showed how it can be compiled into an executable form.

The workshop has been concluded by a panel discussion on the market for applications with CSP chaired by Jean Charles Régin. One of the conclusions of this discussion was that the reason why CP is not as widespread as predicted a couple of years ago could be that the technology is becoming too complex to provide solutions for non-expert users. The gap between academic research and applications in CP seems to grow so the goal of next meetings could be bringing these areas back to be closer again.

Further information on the workshop including the proceedings is available on-line from the workshop web pages www.ifi.uni-klu.ac.at/Conferences/cp04cspia.

Dr. Roman Barták is an assistant professor and a researcher at Charles University, Prague (Czech Republic). His main research interests include constraint satisfaction and its application to planning and scheduling.
E-mail: bartak@kti.mff.cuni.cz
Phone: +420 221 914 242
Fax: +420 221 914 323

Web-Based Semantic Pervasive Computing Services

Yugyung Lee, *Member, IEEE,* Soon Ae Chun, *Member, IEEE,* and James Geller

***Abstract*— Pervasive Computing refers to a seamless and invisible computing environment which provides dynamic, proactive and context-aware services to the user by acquiring context knowledge from the environment and composing available services. In this paper, we demonstrate how heterogeneous Web services can be made interoperable and used to support Pervasive Computing. We present an architecture how a service flow can be automatically composed using syntactic, semantic and pragmatic knowledge. Thus, this paper addresses three problems: (1) How do heterogeneous Pervasive Computing services interoperate in a Pervasive Computing service flow, composed by using syntactic, semantic and pragmatic knowledge; (2) How do we define, distinguish between, and justify the need for these three different kinds of knowledge to be used in service descriptions; and (3) How can we perform ontology integration to enable the automatic composition of Web services into a service flow. A Pervasive Computing prototype system, based on this architecture, has been implemented as a proof-of-concept.**

***Index Terms*— Ontology, Semantic Web Services, Service Discovery and Composition, Pragmatic Knowledge, Pervasive Computing**

I. INTRODUCTION

There are reasons to believe that Pervasive Computing may be the next frontier of computing after the Internet revolution. Pervasive Computing aims to revolutionize the current paradigm of human-computer interaction. Computers have been used in various aspects of human life, but in most cases human beings have had to adapt their behavior to existing systems. Pervasive Computing, as envisioned by Weiser [49], is a computing environment in which computing systems weave themselves into the fabric of everyday life and become invisible. Invisibility is the most important aspect of Pervasive Computing. The user is exposed to a few sets of services available to him/her and is oblivious to the complex system implementing those services [38]. This takes the human-computer interaction into a whole different dimension, where the user is surrounded by a complete smart environment with devices/sensors communicating with each other and aggregating their functionalities to provide a set of consolidated services.

In order to build a Pervasive Computing environment, existing methodologies use smart devices, which have some processing power and are specialized to perform a set of specific tasks. Usually the user needs to carry these devices with her/him as s/he moves either within or across Pervasive Computing environments. However, we present an alternate approach and use Semantic Web technologies for Pervasive Computing environments. This allows context information to be stored on the Web, Pervasive Computing services to be dynamically composed based on Web Services, and then shared across Pervasive Computing environments via the Web to provide Pervasive Computing services.

There are several challenges that we are facing in Pervasive Computing. First, it requires acquiring context from the environment and dynamically building computing models dependent on context. Context-awareness is a pivotal aspect of Pervasive Computing. Dey and Abowd [10] defined the concept of context as a piece of information that can be used to characterize the situation of a participant in an interaction. Brown [3] defined the context as location, environment and/or identity of people and time. By sensing context information, context enabled applications can present context relevant information to users, or modify their behavior according to changes in the environment. Context is however very subjective, in the sense that it can include any factors that may affect a users interaction with the system. This makes the modeling of context extremely difficult especially because we have to capture abstract and subjective factors.

In the past few years, the WWW has changed from being nothing more than an indexed repository of documents towards being a repository of interconnected services and documents. Web users are now routinely checking the Web for services such as currency converters, mortgage calculators, shortest driving distance with directions generators, etc. Unfortunately, not every required service is available on the Web, and if it is, it might be hidden at position 1921 of 2000 search engine hits. Therefore Web research has turned to the time-honored approach of its parent discipline and attempts to provide complex services by, in effect, combining simple services in the way of a workflow of services, what we call a *service flow*. However, the problem of creating a service flow for a given specification is difficult, and it is a part of the vision of the Semantic Web [2] to let roaming agents perform this difficult task. For that purpose, (simple) services need to be described in an agent-readable form.

The automatic composition of services requires more than descriptions of service capabilities and input/output parameters. Rather, a service should also indicate in what situations and in what ways it should be used. This is comparable to the manual of an electronic device that provides a service. For example, a cell phone manual describes “use cases” of the services that the cell phone offers: Making phone calls, playing games, maintaining a calendar, etc. In case of an emergency, most cell phones allow a 911 call without the payment of a fee. While it is obvious that this kind of knowledge needs to be provided and bundled with the device itself, it is only recently becoming clear that Web services need to have the same kind of knowledge attached to them.

We call this additional level of description of Web services *pragmatic* or contextual knowledge. A service should be described by a pragmatic annotation that represents this

Y. Lee is with the University of Missouri - Kansas City.

pragmatic knowledge, in addition to the semantic and syntactic knowledge that describes the necessary parameters and functionalities of the service. We propose an ontology as a model for representing knowledge to describe services. Specifically, we use ontologies to represent syntactic, semantic and pragmatic knowledge about services.

Clearly, the service composition faces an immediate problem when every service is described using terms from its own underlying domain. The pragmatic and semantic knowledge ontology may contain a collection of these terms [2]. Therefore, the discovery of correct component Web services will often require additional preliminary steps to integrate the ontologies used to describe these Web services. In many cases it will be necessary to integrate the ontology of an agent, searching for a service, with an ontology describing a service. This will have to be done on the fly and at great speed to decide whether a specific service is a possible candidate for the desired service flow.

In this paper, we demonstrate how heterogeneous Web services can be made interoperable and used to support Pervasive Computing. We present an architecture how a service flow can be automatically composed using syntactic, semantic and pragmatic knowledge. Thus, this paper addresses three problems: (1) How do heterogeneous Pervasive Computing services interoperate in a Pervasive Computing service flow, composed by using syntactic, semantic and pragmatic knowledge; (2) How do we define, distinguish between, and justify the need for these three different kinds of knowledge to be used in service descriptions; and (3) How can we perform ontology integration to enable the automatic composition of Web services into a service flow.

The three different types of compositional knowledge are expressed by compositional rules that a software agent can use for the automatic generation of a service flow. We present an ontology for these compositional rules, applying them to the description of Web services. OWL-S and Jena rule (HP Jena [19]) are used as formats for compositional knowledge [9]. The paper also illustrates one approach how to integrate terms from several ontologies in an efficient manner, using the framework of Terminological Knowledge Bases [13]. These are two-level ontologies where the semantics of concepts at the lower levels are constrained by assigning concepts to semantic types at the upper level.

The paper is organized as follows. In Section II, we introduce our motivating application. In Section III, we discuss different types of compositional knowledge, followed in Section IV by a semantic methodology for heterogenous service composition. In Section V, we present an overall system architecture and implementation for semantic Pervasive Computing services. Relevant work and conclusions are presented in Section VI and Section VII, respectively.

II. Motivation

Existing methodologies for implementing a Pervasive Computing environment use smart devices, which have some processing power and are specialized in performing a set of specific tasks. Usually the user needs to carry these devices with her/him as s/he moves either within or across Pervasive Computing environments. These devices are not readily available and are often difficult to build. The issues that limit fabrication of such personal devices are limitations like battery power, shape and weight, making practical use of such devices extremely difficult. The advantage of using smart devices is their ability to communicate with each other by building and storing contextual information, which may be used by the Pervasive Computing environment to offer services based on the stored information. In addition, current devices are costly, and thus it is difficult to replace all current devices with smart devices to implement Pervasive Computing environments. Finally, smart devices need to have functionality beyond what they are expected to do, because they are integral to the environments.

Our solution reduces the need for smart devices by using the Semantic Web to build dynamic service composition knowledge (context) models as a user moves from one environment to another. We can achieve dynamic building of contexts by sharing knowledge and context information between local Pervasive Computing environments through the Semantic Web. Furthermore, Pervasive Computing services can be dynamically composed by considering the contexts determined by the Pervasive Computing framework. In this approach we can utilize currently available resources (data, information, services, devices, etc), letting the devices do their basic tasks without saddling them with any pre-requisites to participate in Pervasive environments. Also we believe that this approach will help us quickly implement Pervasive Computing, since we can use currently available resources and do not need specialized devices.

III. Definitions: Syntactic, Semantic, Pragmatic Knowledge

In a previous publication we have introduced the use of syntactic, semantic and pragmatic knowledge for services and workflows [42]. Of these, syntactic and semantic knowledge are well known in computer science, but this is less so for pragmatic knowledge. Pragmatic knowledge has been an issue mostly in philosophy of language and some branches of linguistics, such as discourse understanding [25]. Giving a general definition of pragmatic knowledge and distinguishing it from semantic knowledge is difficult. However, by limiting ourselves to the fairly well defined environment of services, the distinction becomes easier.

Instead of jumping directly into a set of definitions, we will clarify our distinctions between syntactic knowledge, semantic knowledge and pragmatic knowledge by the example of a cellular phone. Our basic approach is to observe the different mistakes that users of a cell-phone may make. Every user that does NOT make those mistakes appears to have some knowledge on how to correctly use a cell phone.

This approach is metaphorically related to Cognitive Science methods that study the working brain using data from aphasia patients. By linking observable damage to certain areas of the brain with observable performance failures, it becomes possible to hypothesize which part of the brain is responsible for which cognitive activity. Instead of looking for physical damage linked to performance failures we are looking for presumed missing knowledge items that would lead to performance failures.

The mistakes that a cell phone user can make vary widely, and therefore different kinds of mistakes give rise to observations of different kinds of knowledge.

We will now turn to syntactic errors. If a person types in a 6 digit phone number in the continental United States, this should (after some waiting period) result in a voice saying “your call cannot be completed as dialed.” Attempting to dial a six digit phone number is a syntactic error. In programming languages, syntactic errors can (usually) be detected by a compiler, and similarly, dialing too few digits can be detected by the phone SW. Thus, the essence of a syntactic error is that it violates simple (“knowledge free”) rules that can be checked mechanically without reference to any additional knowledge. Thus, syntactic knowledge is knowledge which can be expressed in rules that do not refer to any outside database. As usual in computer science, syntactic knowledge is easier to understand than semantic knowledge.

We next turn to semantic errors and semantic knowledge. A person attempting to call a friend, who has memorized her phone number incorrectly, is making a semantic error. He will end up dialing a different person, or possibly a non-existent phone number. To verify that a number is non-existent, it is necessary, at least in theory, to have a knowledge base of all existing phone numbers. Thus, this cannot be checked by a set of self-contained rules and requires a substantial data base. To detect that a user is calling a wrong phone number, the phone would need to “know” who the user is trying to call and also need to know the phone number of that person. Again, this requires a database and cannot be done with self-contained rules alone.

In some cases, a database is not sufficient for semantic knowledge. The process of reasoning is essential to meaningful knowledge processing. For example, a person might have the phone number of a friend, but without the area code. Making a phone call without area code would result in a call to the given phone number in the area of the caller. If the caller and the callee live in the same area code, this would be a successful call, but otherwise it would result in a failure by calling a different person than the one intended.

If the caller lives in New Jersey and knows that his friend lives in Manhattan, he can still place a successful call. By knowing the area code for manhattan, he will be able to reason out and construct a complete and correct phone number. Thus, besides simple rules and a database we need to assume a knowledge base with some reasoning abilities for semantic knowledge.

However, semantic knowledge clearly does not avoid all errors. One should not call a friend at 1:00 AM. Doing it would make the friend upset and would be a pragmatic error. On the other hand, if the house of the friend is on fire, one should call him at any time, even at 1:00 AM. If one has an emergency, he should call 911. If one has an emergency and is in Austria, he should call 112.

If one has a phone that does not work, he should call 611 (presumably using another phone that works). If one does not know the phone number of a friend, one should call 411 to get it. If one has no money to call, he should only call 888 and 800 numbers or attempt to make a collect call. All these rules describe pragmatic knowledge that links situations with the actions that should be taken. They do not just require knowledge as it is stored in a database or reasoning that involves a knowledge base, they require situational awareness.

In the simplest cases, situational awareness deals with time (Is this a reasonable time to phone?) and space (In which country am I? In which area code am I?) Sometimes complex combinations of time and space need to be reasoned about. If your friend just left your house and lives an hour’s drive away, it makes no sense to call him at his home phone number after 5 minutes. Determining what constitutes an emergency that would allow one to call at all hours, or to call 911 requires even more complex knowledge of ownership and values of objects. Pragmatic knowledge also includes social relationships and authorities of people. Thus, the essence of pragmatic knowledge for services is that it incorporates some kind of knowledge of the context (or situation) in which a service should be used. A subset of this kind of knowledge may be expressed in terms of time and space, which themselves are already (for time) or in the foreseeable future (GPS systems for space) integrated into all computer systems. This is the point where Pervasive Computing becomes important.

Social situations may be incorporated into services, in organizations with well defined hierarchies such as in the military. A service may provide more information for privileged users (”super-users”).

Thus, the border line between semantic knowledge and pragmatic knowledge in our approach is that semantic knowledge relies on the retrieval or the reasoning with knowledge from a knowledge base, while pragmatic knowledge requires retrieval of situational information from an outside source. Thus, we have formulated a border line between semantic knowledge and pragmatic knowledge, limited to our services domain.

We will now carry over these general remarks to actual services. If we view a service as a procedure or function that takes certain inputs and produces certain outputs, then we can require the same syntactic constraints as on functions:

For functions, the number, order, directionality (in/out), data type and optionality (mandatory/optional) of inputs and outputs need to be correct. Otherwise there is a syntactic mismatch. Our view of syntactic knowledge is guided by this idea.

Syntactic knowledge for a service consists of self-contained rules that can automatically determine whether the input parameters received by the service are correct in number, order, directionality, data type and optionality.

Semantic knowledge for a service consists of rules that describe how to correctly use the service. These rules may access an outside database to retrieve information and/or an outside knowledge base to reason with information. We call them semantic rules.

Pragmatic knowledge for a service consists of rules that describe in what situations to use the service. These rules may access the same information as the rules of semantic knowledge. In addition, these rules may access information about the current situation, such as time, location of the service requester and the service provider, hierarchical ("privilege") status of the service requester and the service provider, etc. We call them pragmatic rules.

IV. A Methodology for Heterogenous Service Composition

In this section we address the issue of the heterogeneity of Web services that were developed independently, using terms from different ontologies, and present the methodology to match those concepts from different ontologies, called OnInt (Ontology Integration).

A. Ontology Integration

Every realistic service model consist necessarily of two kinds of elements. On one hand there are elements that are specific to OWL-S itself. These elements correspond to what would be called "reserved words" in a programming language. The number of types of these elements is strictly limited, but the elements are composable, in the same way in which FOR loops in a programming language may be composed by nesting. On the other hand there are elements that are specific to the service domain itself. These elements would correspond to the variable names, constant names, function names, type names and module names of a program. Every good program closely mirrors its domain by the choice of meaningful function and variable names. Therefore, there are as many different (sets of) function names as there are domains. The same applies to services. Thus, a good service description will need to use terms from its underlying domain, and the number of terms available will be as unlimited as the domains themselves.

When an agent is looking for a service, it will carry with it a description of the kind of service that it is looking for, in terms of its underlying domain. It will encounter service descriptions using the same or different terms from the domain of the service provider. Unfortunately, even if the agent domain and the service provider domain are the same, that does not mean that the agent and the provider can smoothly interact, because there is no global shared ontology of domain terms. The situation is comparable to an Italian tourist in America that tries to order a meal from a Chinese waiter, and both know only subsets of English food language. The waiter and the tourist cannot start talking with each other directly. They need to establish a common language first, by discovering shared terms and finding mappings (hard!) between differing terms.

In ontology research this kind of process is described as a form of ontology integration. The heart of this process is to find mappings between differing terms for the same concept. This integration process has to be performed quickly, as one agent may be visiting many services with service ontologies in its attempt to construct a service flow. For any pair of sizable ontologies it is out of the question to perform a brute force attempt of matching every term in one ontology with every term in the other ontology. To overcome this problem we have developed an extensive method of semantic specification and semantic integration, using two-level ontologies, which is used as a precursor to a the actual integration algorithm [13], [16]. This semantic integration algorithm greatly limits the number of matching attempts involved in every integration task. Details of this matching and integration method would go well beyond the scope of this paper, but we summarize the basic ideas and some of the formalism in this section.

B. Two Level Ontologies for Integration

Definition: Terminological Knowledge Base. We call any structure that consists of (1) a semantic network of semantic types; (2) a thesaurus of concepts; and (3) assignments of every concept to at least one semantic type a *Terminological Knowledge Base* (TKB).

$$TKB =< \hat{\mathcal{C}}, \hat{\mathcal{S}}, \mu > \tag{1}$$

in which $\hat{\mathcal{C}}$ is a set of concepts, $\hat{\mathcal{S}}$ is a set of semantic types (i.e., high-level categories), and μ is a set of assignments of concepts to semantic types. Every concept must be assigned to at least one semantic type. The opposite condition does not hold. We will use capital letters to represent semantic types and small letters to represent concepts.[1]

$$\hat{\mathcal{S}} = \{W, X, Y, ...\}; \quad \hat{\mathcal{C}} = \{a, b, c, d, e, ...\} \tag{2}$$

Finally, μ consists of pairs (c, S) such that the concept c is assigned to the semantic type S.

$$\mu \subset \{(c, S) |\, c \in \hat{\mathcal{C}} \,\&\, S \in \hat{\mathcal{S}}\} \tag{3}$$

We define that two concepts c, d are similar, $c \simeq d$, if they are assigned to exactly the same set of semantic types of a TKB.

$$c \simeq d :\; \forall S \in \hat{\mathcal{S}}\, [(c, S) \in \mu \;\Leftrightarrow\; (d, S) \in \mu] \tag{4}$$

If two concepts c and d are assigned to the same semantic type X, then these two concepts have similar semantics. On the other hand, if a concept a is assigned to X and a concept b is assigned to Y, then a and b will have semantics that are not similar in the formal sense defined above. The case of concepts with sets of several assigned semantic types that may have a non-empty intersection is discussed in [30]. With our notion of similarity, we need to decide how strict we want to be with respect to accepting partial matches of concepts.

There is a spectrum of what requirements one could impose to accept two concepts as matching. On one extreme, one might insist that there be only perfect matches. The other extreme is to insist that all (or almost all) concepts of the smaller ontology are matched against concepts in the larger ontology, as long as there is at least some structural similarity. This extreme could be based on the assumption that both ontology designers did a reasonable job to cover the domain, and thus all fundamental concepts simply have to appear in both ontologies, no matter what each one is called, and no matter how exactly it is structured. Our solution is closer to the

[1] Both roman and italic fonts

second extreme. We are optimistic that with the development of the Semantic Web many subdomains of the world will be described by ontologies which cover their domain to a reasonably complete degree. Thus, one would expect that most concepts of one such ontology exist in the other ontologies for the same domain. We note that for people the lack of exact matches does not normally make communication impossible. Indeed, philosophers would point out that we can never be sure of how another person is thinking about a concept, a fact denoted as “solipsism.”[2]

Now we will show how the two-level structure limits the required number of matching attempts. By our construction of the Terminological Knowledge Bases, two concepts, q from TKB$'$ and r from TKB$'_2$, can only match if they are both assigned to the same semantic type. There are three cases:

(1) Assume a semantic type S exists in TKB$'$ that has assigned concepts $x, y, z, ...$ Further assume that S does not exist in TKB$'_2$ or, there are no concepts assigned to S in TKB$'_2$. Then, by the similarity definitions given above, no concepts corresponding to $x, y, z, ...$ exist anywhere in TKB$'_2$. Thus, these concepts do not need to be matched at all.

(2) The above observation applies in reverse also. If a semantic type S exists in TKB$'_2$ that does not exist in TKB$'$, then the concepts $x, y, z, ...$ assigned to S will not have corresponding concepts anywhere in TKB$'$. Thus, these concepts do not need to be matched at all.

(3) Concepts assigned to the semantic type S in both TKB$'$ and TKB$'_2$ are potentially similar ($\simeq$) and need to be matched. As mentioned above, we allow partial matches between concepts that have been determined to be similar. The exact cut-off is decided by a threshold value.

C. Scoring Concept Similarities

Now we describe details of how scores for concept similarities are computed. We use three aspects to determine whether a match exists between similar concepts. Initially we rank pairs of concepts according to their terms (or synonyms) and then according to attribute similarity. After establishing some initial matches in this way, we use relationships that point from one concept to another concept to iteratively recompute the similarity value between two concepts.

1) Ranking Concepts by their Terms: If two concepts have similar names (defined below, based on bigrams) then they are possibly matches. The existence of synonyms and homonyms causes problems for concept matching. We include the use of synonyms during the concept matching step itself. If no match is found for a concept, then it is attempted to use its synonyms for matching.

The bigram approach [23] is known to be a very effective, simply programmed means of determining a similarity measure between strings. The bigram approach consists of three steps. (1) Sequences of two consecutive letters within a string are extracted (e.g., the word “calendar” contains ’ca’, ’al’, ’le’, . . . ’ar’); (2) Two sequences of bigrams are compared, and a raw similarity score is computed; (3) A matched score is computed from the raw score, i.e., the number of the common bigrams is divided by the average number of bigrams in the two strings.

2) Ranking Candidates by Attributes: Assume that we are given a pair of concepts from two different ontologies. These concepts have different terms, therefore, a priori there is no reason for a computer to assume that they are in fact describing the same concepts. In order to establish whether they are indeed the same concept, we need to compare attributes.

We assign to every pair of concepts a score as follows.

- Two concepts, that have the same number of attributes, are considered perfectly matched, with a score of 1, only if for every attribute in one concept there is an attribute in the other concept of the same name and same data type,
- If two attributes (of two concepts from two ontologies) have the same name but are of different data types, we assign them a score of k ($k < 1$, $k \gg 0$).
- Then we compute the ratio of matched attribute scores divided by the number of attributes of the concept that has more attributes.
- The final decision about similarity is made, based on a minimum threshold for the computed combined score.

3) Ranking Candidates by Relationships using Propagation: In the previous step we have established matches between concepts from two different ontologies, based on pairs of terms and attributes. However, two concepts that point to exactly the same concepts with the same relationships are presumably very similar to each other. We view the relationship targets as data types, and two concepts that point to all the same data types are likely to be quite similar. However, we would have a chicken and egg problem here, if we start with considering relationships from the beginning. That is the case because the relationships targets cannot be used for matching if they themselves have not been matched up.

This is why we start by matching up a few concepts using terms and attributes alone. By this step, we create an initialization for matching up additional concepts by using relationships. Thus, two concepts with different names that point to several target concepts that all have been matched up between two ontologies are presumably themselves a match. We can use a similar ratio criterion as for attributes, however, now the targets carry more semantics than the undifferentiated data types of attributes. Thus, we are willing to assign a pair of relationships a high score if the targets are the same OR if the relationship names are the same. Let us assume now that a set of concept pairs has been established such that the concepts in each pair match and are from two different ontologies. Then any pair of concepts that point to these matched concepts would also be considered highly ranked for being matches. Thus, after establishing initial matches, we continue ranking concepts by similarity using a process similar to a Waltz filtering [48].

The process of finding matches needs to be recomputed until a score change of one concept pair does not result in a score change of any concepts pointing to that pair anymore. This state of equilibrium can be achieved, as we are using a threshold. If there are only changes that do not cross the threshold, the update process would terminate.

[2] IEP

4) Combining Matching Scores: Two concepts are considered matched if their terms, their attributes and their relationships are (on average) similar. A weight is assigned to each similarity aspect of a concept (term similarity, average attribute similarity, average relationship similarity). Considering these three criteria, we now compute the degree of the similarity of concepts from two distinct ontologies. For this purpose, we use a Multiple Attribute Decision Making (MADM) approach, a simple additive weight-based solution [20]. This approach determines a combined score of concept matches between ontologies. Let C_i = $\{C_{i1}, C_{i2}, \ldots C_{im}\}$ and C_j = $\{C_{j1}, C_{j2}, \ldots C_{jn}\}$ be sets of concepts for given ontologies, and let F = $\{F_1, F_2, \ldots F_p\}$ be a set of p features (in this paper p = 3) that characterize the degree of similarity. The weight vector W reflects the importance of each attribute $W = \{W_1, W_2, \ldots, W_p\},\ where \sum W_i$ = 1. We compute the scores for each of the p features for each of l matching cases ($l \ll n$ or m) in a decision matrix $D = d_{ij}$.

The method is comprised of three steps: first, scale the scores into a range [0, 1], with the best score represented by 1, using

$$r_{ij} = (d_{ij} - d_{jmin})/(d_{jmax} - d_{jmin}) \tag{5}$$

Second, apply weights and third, sum up the values for each of the alternatives, using

$$S_i = \frac{\sum W_j r_{ij}}{l} \tag{6}$$

After a combined score has been computed, we compare the weighted sum with a given threshold α. Some matches may be lacking attributes or relationships. In this case, a weight of zero will be assigned to these aspects of a concept. All combined similarity values greater than α are stored in a matrix G_T. In this matrix, rows correspond to concepts from one ontology. Columns represent concepts from the other ontology. At each row/column intersection the similarity value of two terms is stored.

Subsequently, concept pairs with similarity values above the threshold are constructed, starting with the maximal similarity value. If there are several equal maximal similarity values, they are processed in random order. Whenever the next largest similarity value has been identified between two concepts c and d, then the complete row of c and the complete column of d in the similarity matrix G_T are set to 0. This is because c and d are not available for matching anymore. Details of this algorithm are given in [30].

Our approach to ontology integration simplifies the matching task by identifying sets of semantically similar concepts before starting with the actual matching steps. Terms from two ontologies only need to be compared for integration if they are already classified as semantically similar. Therefore, our methodology reduces the computational cost of the matching operations. Fewer pairs of terms have to be matched against each other. For more details on the Terminological Knowledge Base Framework see [13], [16].

V. Pervasive Computing Services

There are several challenges that we are facing in Pervasive Computing. The first is, how to acquire context models from the environment and dynamically build computing models dependent on context. By sensing context information, context enabled applications can present context information to users, or modify their behavior according to changes in the environment. Secondly, the environment should be flexible enough to provide composite services by incorporating existing services of the Pervasive Computing environment at run time. Here we show how the proposed approach, service composition, helps in dealing with these challenges in Pervasive Computing.

A. Context Ontologies

The context ontologies are two-part ontologies in OWL format: The upper ontology provides a description of various concepts that together characterize a particular situation. The lower ontologies describe each of these concepts in more detail. For instance, the upper ontology contains Location as one of the concepts while the lower ontology contains the description of the current location i.e. location of rooms, floors etc.

The context ontologies consist of concepts from our own User Profiling ontology and several extensions of the CONON (COntext ONtology) [50], the SOUPA (the Standard Ontology for Ubiquitous and Pervasive Computing[3]), which is the outcome of a collaborative effort between researchers in the field of Pervasive Computing and Semantic Web and the *Content Selection for Device Independence* (DISelect[4]). The upper ontology is based on the CONON ontology for context. CONON provides all the basic concepts needed to model context. This ontology however needs lower ontologies that are extensions to reflect the current domain. For instance the Location concept can be extended to model a building or a city. The descriptions of Location and Time as context elements was obtained from SOUPA. The following context elements are of specific interest.

- Individual: Representing the person whose context is represented. Our Pervasive Computing framework, called SeMEther [40], contains a user profile ontology, which was mapped to the individual concept in the CONON ontology.
- Time: The temporal features associated with the current situation. They were designed based on the specifications of the SOUPA ontology.
- Location: The spatial location features of the person involved. They were designed based on the specifications of the SOUPA ontology.
- Computing Entity used: The Computing Entity used by the person in the current context. The SeMEther in its current implementation does not have extensive device modeling. The only two existing concepts describing the devices are MobileDevice and StaticDevice which indicate whether a device has a fixed location or whether its location can change. We are currently incorporating the *Content Selection for Device Independence* (DISelect[5]) and *Foundation for Intelligent Physical Agents* (FIPA[6])

[3] http://pervasive.semanticweb.org/soupa-2004-06.html

[4] http://www.w3.org/TR/cselection/

[5] http://www.w3.org/TR/cselection/

[6] http://www.fipa.org/

Ontology Type	SeMEther Ontologies	
Upper Ontology	Context Ontology (CONON)	
Lower Ontologies	Concept	Specific Ontologies
	User context	SeMEther User Profile Ontology
	Location	Standard Ontology for Ubiquitous and Pervasive Computing (SOUPA)
	Time	Standard Ontology for Ubiquitous and Pervasive Computing (SOUPA)
	Computing Entity	W3C DISelect, FIPA Device ontology
	Activity	AIAI Activity ontology

TABLE I
SEMETHER ONTOLOGIES

Ontologies that provide a formal description of devices and device capabilities.

- Activity Performed: The activity being performed by the person. This activity can be deduced (e.g. from his schedule) or explicitly specified.

B. Rules for Pervasive Services Composition

The service composition knowledge (context) model can be used to infer new knowledge about a user's situation. For instance, consider a messaging service that sends messages to the users in their current locations. The messages are sent depending on the devices that the user currently uses. A user having a cell phone may receive SMS messages while a user having a laptop may get an e-mail. Depending on the framework events these messages are sent to the devices, however context reasoning may affect this service.

Consider user A whose schedule indicates that he will be in a meeting from 11:00 AM to 1:00 PM in Room 101. At 12:00 PM a buddy_in_environment event arrives that informs the user that his buddy has entered the environment. The Context Reasoner however reasons that the user's scheduled activity indicates he is in a meeting and the user must be busy at this moment. It then asserts the fact in the knowledge base that the user is currently busy. This suppresses the message service from sending messages to the user. Rather, the message will be forwarded to his/her secretary or converted to an email message depending upon his/her profile.

SeMEther makes extensive use of such reasoning. The domain-specific inferencing requires explicit rules. RULEML[7] is an XML-based rule language and the current Semantic Web efforts[8] include building an RDF-based RuleAxiomLogic layer over the current OWL-based Ontology layer in the Semantic Web stack. Thus, we adopted Jena rules (an RDF based language). Jena Rules, written in the Jena rule syntax similar to the one described above, can be used to direct the services provided to the user. An example rule shown below indicates that if a user is in room 101 (conference room) during the meeting time then his status must be set to busy. Such composition rules can be added to guide the situation.

For pursuing advanced context reasoning, the location context manager, which is a specialized reasoner, was developed to manage the location context. As the user's location changes in an environment, this component tracks his/her spatial position. Events are generated if the user changes floors (floor change event) or rooms (room change event) etc. It reasons with the current spatial position of the user and the spatial model stored in the knowledge base in the form of a lower context ontology described above. The events generated by the location manager are used to trigger location-based services. For instance a music service makes use of the room change event to continue playing the user's music preference, switching it from his old room to the new room.

```
[Rule1:
   (?L
     http://SeMEther/ontology/locationcontext#UserInLocation
     http://SeMEther/ontology/location/floor1#room101)
   (?T_s
     http://SeMEther/ontology/timecontext#EventStartTime
     http://SeMEther/ontology/time/startTime#1100)
   (?T_e
     http://SeMEther/ontology/timecontext#EventEndTime
     http://SeMEther/ontology/time/endTime#1300)
       ->
  (?U
     http://SeMEther/user/usercontext#status
     http://SeMEther/user/usercontext#busy )]

[Rule2:
   (http://SeMEther/ontology/timecontext#Time
    http://SeMEther/ontology/timecontext#CurrentTime
    ?val),
    greaterThan(?val,
     http://SeMEther/ontology/timecontext#EventStartTime)
    lessThan(?val,
     http://SeMEther/ontology/timecontext#EventEndTime)
       ->
   (http://SeMEther/ontology#EventManager
   http://SeMEther/ontology#sendMessage
   http://SeMEther/PatientHabits#CallForwarding)]
```

TABLE II
SEMETHER SERVICE COMPOSITION RULES

To illustrate the working of the location context manager consider the floor change event. To generate this event the location context manager performs the following query in RDQL [36], which is a query language for RDF in Jena models, to find out whether the user's old location ”oldlocationuri” is within the same region as the user's new location ”newlocationuri”:

The location manager can verify whether the region is within the same floor or not. It then compares to see if the two rooms are on the same floor or not and generates the floor change event accordingly.

[7] http://www.ruleml.org/

[8] http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

```
[Query 1
  Select ?a where
  (<oldlocationuri / newlocationuri>,
    <http://a.com/ontology#inRegion>,
    ?a)]
```

TABLE III

SeMEther Context Query Example

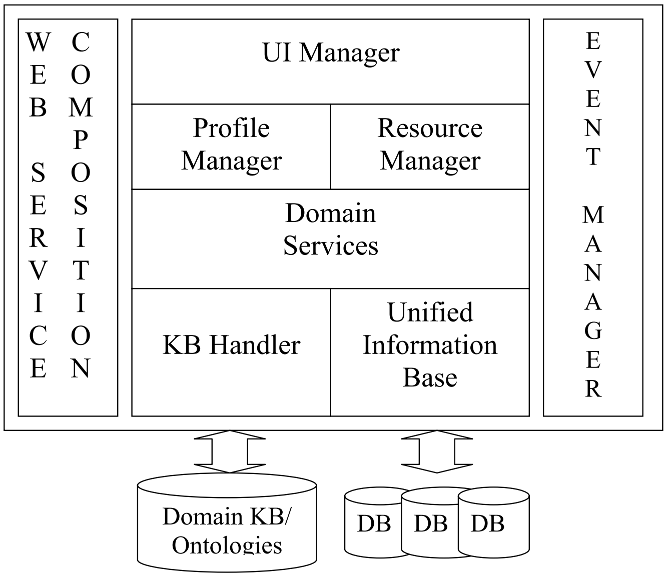


Fig. 1. The SeMEther Framework Architecture

C. SeMEther Framework: Service Composition Subsystem Architecture

As a proof of concept, the Web Service Composition Subsystem (WSCS) has been implemented on top of a Pervasive Computing framework called SeMEther [40]. The SeMEther framework provides an efficient infrastructure for the WSCS design and execution. We are in the process of developing a set of applications, tailored to meet the needs for developing Semantic Pervasive Computing Services. In this paper, we mainly highlight the WSCS. Before we demonstrate the WSCS, we briefly introduce the architecture of the SeMEther framework (Figure 1).

The Event Manager manages event-based communication between components of the framework. Components in the SeMEther framework communicate with each other by throwing and listening to events at the Event Manager. The Event Manager ensures that all services registered for an event receive that event and don't receive any duplicate events. The KB Handler maintains a Knowledge Model which reflects the current context. This context is acquired by listening to events and converting them to knowledge facts which are added or removed from the Knowledge Base (KB). The Resource Manager manages the resources available in the environment. It maintains the status of each resource and also takes care of scheduling resources for different activities. In SeMEther, a resource is anything that can be scheduled, including human actors and devices. Each resource has a semantically annotated schedule, which describes when a particular resource is available. When there is a request for a certain resource, the system will look for its availability and then schedule that resource.

The Profile Manager is responsible for fetching user profile information from the Web. We assume a centralized server, where one can request user profiles or parts of them. The SeMEther communicates with this server to get the user profile or extract specific information about the user from his/her profile.

The framework is designed to provide a dynamic user interface (UI), that is, the interface can be changed depending on the user role and the requirement of the service as well as the device used to communicate with the user. For example, the system generates different UI screens, corresponding to a desktop or a PDA, or sends an SMS over a cell phone, depending on what device the user is currently on. Thus, the framework provides pervasiveness in the sense that it uses an appropriate device to communicate with the user, depending on the context. The Domain Services provide functionalities specific to a given domain. These services use the framework to communicate with each other, and the external environment, and also to access the knowledge base of the system.

The Unified Information Base (UIB) integrates information from disparate data sources present in the environment and presents it at a more conceptual level by linking it to a local domain ontology. For example, a database field 'BP' in a hospital setting can be linked to the concept "BloodPressure" of a standard medical ontology like the UMLS. This creates an abstraction of a single homogeneous data source for other services, which need to refer to the data in terms of domain concepts. The UIB thus allows linkage to a data element and fetch or update of the same. The idea of a unified information base is an extension to our previous work [8].

The proposed system architecture for the Web Service Composition Subsystem (WSCS) is shown in Figure 2.

- **Service Editor:** A platform to model and create Semantic Pervasive Computing Services over existing legacy applications (Figure 3). The editor allows mapping of service parameters (input, output, preconditions, effects) to concepts in predefined ontologies. New ontologies can be loaded into the editor. The editor has an ontology search component which performs keyword based search in the ontologies. The user can map resultant concepts to service parameters. To facilitate faster development and ease of use, we concentrated on development of atomic services, hence making the model simpler and easier to implement. The editor parses the WSDL (Web service Definition Language) documents discovered by the Service Crawler and creates the service grounding descriptions. One interesting feature is the possibility of plugging in of new context ontologies (required for mapping service parameters). Once stored in the composition rule KB, these services are used by the Service Matcher.
- **Service Crawler:** The Service Crawler is crawling the Web for services (Web form, WSDL, or OWL-S). A multi-threaded Service Crawler crawls multiple URLs in parallel. For efficiency, the Crawler crawls the Web

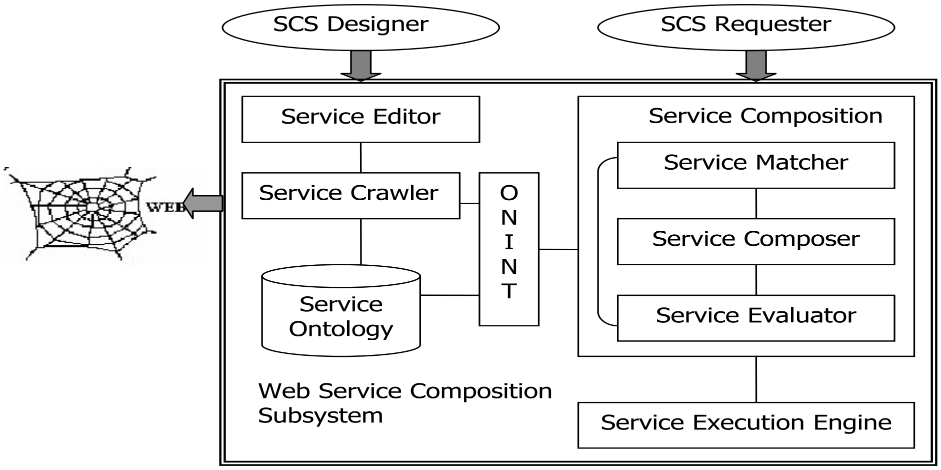


Fig. 2. The Architecture of the Web Service Composition Subsystem

using URLs taken from DMOZ[9], classifies any pages containing such service descriptions into that particular category/domain and stores them in a temporary database.

- **Service Ontology:** The Service Ontology is a repository of services either developed or discovered from the Web. Each service in the Service Ontology is semantically annotated (OWL-S) according to its respective category/domain. Thus, the services discovered by the Service Crawler are transformed into Semantic Web Services (OWL-S).
- **Service Matcher:** The Service Matcher matches a service request to existing services available in the Service Ontology. For this service matching, we could apply the Ontology Integration (OnInt) methodology to each OWL-S profile. In the profile, each service is represented by a type of service and an IOPE (Input, Output, Precondition, Effect) tuple. For two given service descriptions which are the service parameters (IOPE), the Sevice Matcher tries to match them.
- **Service Composer:** Using the Ontology integration (OnInt) methodology as described in Section IV, this module performs semantic matching of concepts. For two given concepts which are service parameters, the component tries to establish a match between them. The service composition required an iterative approach: matching of concepts followed by pragmatic evaluation.
- **Service Execution Engine:** Once the services have been discovered and composed to satisfy the goal, this module executes the services. We used the Taverna service execution tool.[10] This tool mandates the process specification in a specific format. In our case, the process specifications are generated as the result of refinement of the composition process.
- **Service Evaluator:** This component performs evaluation of a service, based on the pragmatics defined for selection of a particular service. We perform evaluation based on some simple evaluation metrics, which are similar to match algorithm described in [34]. In order to select appropriate services, we need to evaluate whether they satisfy the syntactic, semantic and pragmatic requirements of the desired composite service.

D. Implementation

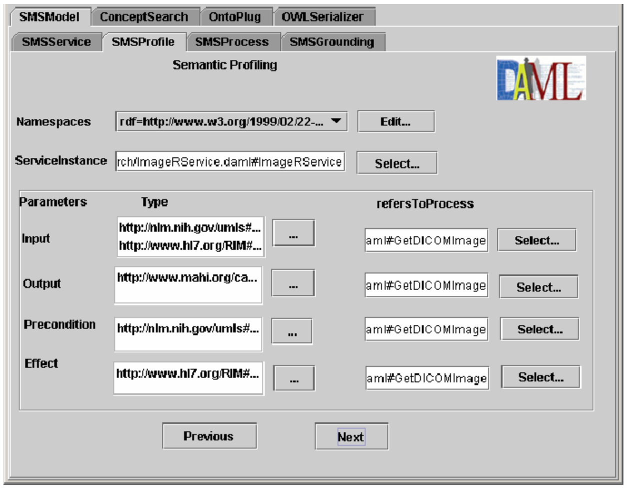
We have implemented a prototype of SeMEther that demonstrates the intended goals and shows the feasibility of the proposed approach. Integrating some common computing devices such as PDAs, cell phones and personal computers, we show how the system actually functions and interacts seamlessly with the user. To bring out the effectiveness of the framework, we have implemented the Pervasive Service Compositions for several scenarios. One simple, yet powerful, service that we have implemented is the “Buddy” service. This service detects buddies of a given user, determines if they are in the vicinity, and in that case contacts them by the best possible means available. The user is detected by a Bluetooth enabled device such as a PDA or Cell Phone. Messages are delivered based on the type of device he is carrying, like a dialog box for the PDA or an SMS for the cell phone. Several other services can be run concurrently on this framework. All these services are pervasive in the sense that a user doesn’t depend on any specific device to get that service. The environment proactively detects the user, and based on his/her preferences, adapts these services and provides him/her via the best available service.

Through the implementation we have verified the viability of: (1) Generating service flow specifications in OWL-S to model context-aware services; (2) Achieving dynamic service matching and binding to the service flow; (3) Performing dynamic service composition using the compositional knowledge we specified in Section 3.

We implemented an OWL-S Editor to assist the service flow designer in modeling OWL-S specifications for the service flow, which in addition provides a graphical interface to model the abstract concepts. The Service Matching Agent (SMA) handles the task of matching the Web Services for

[9] http://www.dmoz.org
[10] http://taverna.sourceforge.net/

given specifications and the service matching rules. The task execution engine was constructed as Java implementation (HP Jena Toolkit [19]) and reads the process specifications in OWL-S to execute the appropriate Web Service from a service pool associated with each task.



Semantic Profiling

Namespaces: rdf=http://www.w3.org/1999/02/22-...

ServiceInstance: rch/ImageRService.daml#ImageRService

Parameters	Type	refersToProcess
Input	http://nlm.nih.gov/umls#... http://www.hl7.org/RIM#...	aml#GetDICOMImage
Output	http://www.mahi.org/ca...	aml#GetDICOMImage
Precondition	http://nlm.nih.gov/umls#...	aml#GetDICOMImage
Effect	http://www.hl7.org/RIM#...	aml#GetDICOMImage

Fig. 3. The WSCS Editor

The context visualization interface provides a Touchgraph interface to visualize changes in context. As changes in the knowledge base occur due to changes in context, the Touchgraph morphs to reflect the changes. The graphl library[11] was used for the visualization. A Java based client has been developed for the visualization. The client makes an HTTP connection to the SeMEther Autonomous System (AS) head to download the latest context model. This model is in the form of an RDF document that is generated every time the service composition knowledge model changes in the knowledge base. The client can be configured to poll the server for new models at specific intervals of time. A screen shot of the context visualization tool is given in Figure 4.

VI. Related Work

Current Web services support a certain level of interoperability in using and accessing them. The next level of interoperability cannot be achieved by just making services available, but requires providing automatic mechanisms so that the services can be linked in appropriate and meaningful ways [14]. Semantic interoperability is essential for automated discovery, matching and composition of services. This enhancement depends on the existence of ontologies for the terms used by Web services. The Semantic Web research work, following the DARPA Agent Markup Language (DAML), includes DAML+OIL [17] for the creation of arbitrary domain ontologies and DAML+OIL/RDF(S) [14] for the semantic mediation between services and workflow logic. Some research has focused on the composition of services using workflow management. Automatic composition of Web services [28] has been achieved through automated mapping, composition and interoperation of services, service verification, and execution monitoring. Process modeling languages such as PIF [29], PSL [41], and frame based models of services [12] were designed to support process management. There are other emerging relevant approaches such as indexing services based on process models [26] and reasoning and matching over service descriptions for choosing computational resources [35].

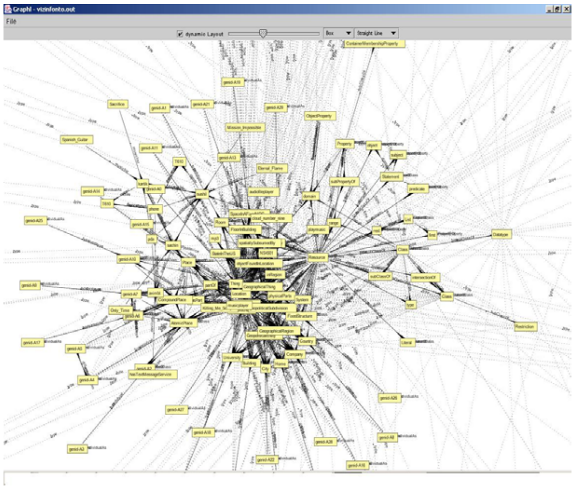


Fig. 4. Visualization of the Pragmatic Knowledge in SeMEther

Several applications require that multiple ontologies are combined into a single coherent ontology [33]. Many lines of research have addressed ontology matching in the context of ontology construction and integration [5] and effective methodologies for automated mappings [31]. Similarity measure studies were introduced for effective ontology integration. Tversky's feature-based approach [47] is one of the most powerful similarity models, but depends on the structure of ontology features. Resnik [37] considered the extent of shared information between concepts. Lin [27] proposed an information-theoretic notion of similarity based on the joint distribution of properties. Jiang and Conrath's similarity measurement [24] is based on the conditional probability of encountering a child synonym set given a parent synonym set.

Ontologies are used for constraining the parameters of dynamic service configurations. Reasoning to ensure the semantic validity of compositions is used for automated workflows. Scientific workflow [4] is supposed to support interoperation through semantics. It may have the potential to support Web service descriptions for service discovery, invocation, activation and execution of an identified service by an agent or other service [28]. Unlike these efforts, our approach emphasizes the importance of different kinds of knowledge, especially pragmatic knowledge, and the ontological methodology for heterogeneous semantics for the automatic composition of service flows.

There have been efforts in representing business contracts for service evaluation and negotiations [15] but how to use such pragmatic knowledge for service matching remains still unresolved. We show the semantic and pragmatic represen-

[11] http://home.subnet.at/flo/mv/graphl/

tations for Pervasive Computing service flows and how the Pervasive Computing community can reap the benefits of using semantic and pragmatic rules over the Semantic Web. In other work [4], workflows in Pervasive Computing settings have been studied, but their efforts are more geared towards QoS (Quality of Service) and workflow execution aspects. We address the need to consider a broad set of pragmatic rules (including QoS) to compose a service flow of Pervasive Computing services for the Semantic Web.

Mennie et al. [32] provide an insight how to achieve dynamic service modification and up-gradation. Specifically, they describe switching or updating services without bringing down the system. A new service can be incorporated into the system by dynamic service composition. Tripathi et al. [46] define access policies for “collaboration spaces” which can also be referred to as pervasive domains. They also describe creating ubiquitous and context-aware applications from high level specifications coupled with a policy driven middleware. Their idea of a user’s ’View’ of the system, with static or dynamic binding to actual resources, governed by access policies, is highly relevant to our approach.

Amann et al. [1] focused on knowledge management in distributed environments, adaptive decision support and assistance with dissemination of relevant information and knowledge among geographically dispersed user groups. The key technical contribution is the integration of the extended tuple space concept to adapt, co-ordinate and control a set of ordered events as well as applications and devices in mobile settings. Henricksen et al. [18] introduced appropriate context modeling issues for Pervasive Computing, such as wide variations in information quality, the existence of complex relationships amongst context information, and temporal aspects of context. They provide a very good understanding and a solid model for modeling context; however they utilize a traditional database to store context information and relationships, while we think an ontology is a better structure to model context.

In Strang et al. [45] Con-text Ontology Language (CoOL) is derived from the model, which may be used to enable context-awareness and contextual interoperability during service discovery and execution in a proposed distributed system architecture. Specifically, Indulska et al. [21] present a location management system able to gather process and manage location information from a variety of physical and virtual location sensors. Their approach scales to the complexity of context-aware applications, to a variety of types and a large number of location sensors and clients, and to a geographical size of the environment. The objective of CoBrA [7] is to provide a centralized model of context that can be shared by all devices, services, and agents in the space. They acquire contextual information from sources that are unreachable by the resource-limited devices. They also reason about contextual information that cannot be directly acquired from the sensors (e.g., intentions, roles, temporal and spatial relations). The main idea in [6] is that of a central context broker which manages the context knowledge base using a context reasoning engine. This is analogous to our idea of a KBHandler. As the centralized KB approach discussed here becomes a bottleneck, this paper proposes a distributed knowledge base. For instance, the building agent maintains knowledge about a building and these agents then exchange/share knowledge. Our approach differs from CoBrA in that we propose a completely service-based architecture, in contrast to their agent-based one.

VII. Conclusions and Future Work

In this paper, we have laid out an architecture of the knowledge processing that is necessary for composing services automatically, using different kinds of knowledge. We showed heterogeneous Pervasive Computing services interoperate in a Pervasive Computing service flow, composed by using syntactic, semantic and pragmatic knowledge. We defined, distinguished between, and justified the need for these three different kinds of knowledge to be used in service descriptions. Finally, we demonstrated principles of ontology integration to enable the automatic composition of Web services into a service flow. We have developed a prototype of a Pervasive Computing service flow as a proof-of-concept. This prototype allows the routing of information to a user with the most appropriate device for a given context.

Future work includes the extension of compositional knowledge to include negotiation rules. When certain services in the process of service selection do not exactly meet the conditions of a rule, then there should be a possibility to relax the conditions to continue with the selection and integration process. This is best modeled as a form of inter-agent negotiation. We are also planning to work on a user service request model and representation that support a wide range of different services.

References

[1] P. Amann, D. Bright, G. Quirchmayr, B. Thomas: Supporting Knowledge Management in Context-Aware and Pervasive Environments Using Event-Based Co-ordination. DEXA Workshops 2003: 929-935

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. Scientific American, 284(5), pp. 34-43, May 2001

[3] Brown, P.J. The Stick-e Document: a Framework for Creating Context-Aware Applications. Electronic Publishing 96 (1996) 259-27.

[4] J. Cardoso and A. Sheth, Semantic e-Workflow Composition, Journal of Intelligent Information Systems, 2003. (In press)

[5] H. Chalupsky. Ontomorph: A translation system for symbolic knowledge. In A.G. Cohn, F. Giunchiglia, and B. Selman, editors, Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning. Morgan Kaufman, San Francisco, CA, 2000.

[6] H. Chen et al., ”An Ontology for Context-Aware Pervasive Computing Environments”, Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, November 2003

[7] H. Chen, T. Finin and A. Joshi, Semantic Web in the Context Broker Architecture, IEEE Conference on Pervasive Computing and Communications (PerCom), Orlando, March, 2004

[8] Q. Chong, A. Marwadi, K. Supekar and Y. Lee, Ontology Based Metadata Management in Medical Domains, Journal of Research and Practice in Information Technology (JRPIT), 2003, 35(2), pp. 139 - 154.

[9] S. A. Chun, V. Atluri and N. R. Adam, Domain Knowledge-based Automatic Workflow Generation, in the proceedings of the 13th International Conference on Database and Expert Systems Applications (DEXA 2002), September 2-6, 2002, Aix en Provence, France.

[10] Dey, A.K. and Abowd, G.D.: Towards a better understanding of Context and Context-Awareness. GVU Technical Report GITGVU-99-22, College of Computing, Georgia Institute of Technology. 2 (1999) 2 14

[11] D.F. D’Souza and A.C. Wills, Objects, Components, and Frameworks with UML: the Catalysis Approach, Addison Wesley, 1999.

[12] M. G. Fugini and S. Faustle. Retrieval of reusable components in a development information system. In Proceedings of Second International Workshop on Software Reusability, IEEE, 1993.

[13] J. Geller, H. Gu, Y. Perl and M. Halper, Semantic refinement and error correction in large terminological knowledge bases, Data & Knowledge Engineering, 45(1), 2003, pp. 1-32.

[14] C. A. Goble. Supporting Web-based biology with ontologies. In The Third IEEE ITAB00, pages 384-390, Arlington, VA, November 2001.

[15] B. N. Grosof, T. C. Poon, SweetDeal: Representing Agent Contracts with Exceptions using XML Rules, Ontologies, and Process Descriptions, In Proc of WWW 2003, May 20-24, 2003, Budapest, Hungary.

[16] H. Gu, Y. Perl, J. Geller, M. Halper, L. Liu and J.J. Cimino, ”Representing the UMLS as an OODB: Modeling Issues and Advantages.” Journal of the American Medical Informatics Association, 7(1), January 2000, pp.66-80.

[17] J. Hendler and D. L. McGuinness. DARPA Agent Markup Language. IEEE Intelligent Systems, 15(6):72-73, 2001.

[18] K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling context information in pervasive computing systems. In Proceedings of the First International Conference on Pervasive Computing, volume 2414 of Lecture Notes in Computer Science, pages 167-180, Zurich, August 2002. Springer-Verlag.

[19] HP Labs Jena 2 Toolkit, http://www.hpl.hp.com/semweb/index.html

[20] C.L. Hwang and K. Yoon, Multiple Attribute Decision Making, Springer-Verlag, Berlin, 1981.

[21] J. Indulska, T. McFadden, M. Kind, and K. Henricksen. Scalable location management for context-aware systems. In Proceedings of the 4th International Conference on Distributed Applications and Interoperable Systems, DAIS 2003, volume 2893 of Lecture Notes in Computer Science, pages 224-235, Paris, France, November 19-21 2003.

[22] Internet Encyclopedia of Philosophy Website
http://www.iep.utm.edu/s/solipsis.htm

[23] F. Jelinek. 1990. Self-organized Language Modeling for Speech Recognition. In Readings in Speech Recognition. Edited by Waibel and Lee. Morgan Kaufmann Publishers.

[24] J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of International Conference on Research in Computational Linguistics, 1997.

[25] A. K. Joshi, B. J. Weber, and I. A. Sag. Elements of Discourse Understanding. Cambridge University Press, Cambridge, 1981.

[26] M. Klein and A. Bernstein. Searching for services on the semantic Web using process ontologies. In Proceedings of the International Semantic Web Working Symposium (SWWS), July 2001.

[27] D. Lin. An information-theoretic definition of similarity. In Proceedings of the 15th International Conference on Machine Learning, 1998.

[28] S. McIlraith, T. Son, and H. Zeng. Semantic Web services. IEEE Intelligent Systems (Special Issue on the Semantic Web), 16(2):46-53, 2001.

[29] J. Lee, G. Yost and the PIF Working Group, The PIF Process Interchange Format and Framework, 1994, http://ccs.mit.edu/pifmain.html

[30] Y. Lee, C. Patel, S. Chun, J. Geller, Towards Intelligent Web Services for Automating Medical Services Composition, in Proceedings of 2004 IEEE International Conference on Web Services (ICWS 2004) July 6-9, 2004, San Diego, California. pp. 384 391.

[31] Maedche. A machine learning perspective for the Semantic Web. In Proceedings of Semantic Web Working Symposium (SWWS), 2001.

[32] David Mennie, Bernard Pagurek, An Architecture to Support Dynamic Composition of Service Components, Proceeding of the WCOP2000, 2000.

[33] N. Noy and M. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In Proceedings of the National Conference on Artificial Intelligence (AAAI), 2000.

[34] Massimo Paolucci, Katia Sycara, and Takahiro Kawamura, ”Delivering Semantic Web Services.” In Proceedings of the Twelves World Wide Web Conference (WWW2003), Budapest, Hungary, May 2003, pp 111- 118 .

[35] R. Raman, M. Livny, and M. Solomon. Matchmaking: An extensible framework for distributed resource management. Cluster Computing: The Journal of Networks, Software Tools and Applications, 2:129-138, 1999.

[36] Query language for RDF Website:
http://www.hpl.hp.com/semweb/doc/tutorial/RDQL/index.html

[37] P. Resnik. Selection and Information: A Class based Approach to Lexical Relationships. PhD thesis, University of Pennsylvania, 1993.

[38] M. Satyanarayanan, ”Pervasive computing: Vision and challenges,” IEEE Personal Communications, vol. 8, pp. 10–17, Aug. 2001

[39] D. Saha, A. Mukherjee, ”Pervasive Computing: A Paradigm for the 21st Century”, IEEE Computer, vol. 36, pp. 25-31, March 2003

[40] S. Singh, S. Puradkar, Y. Lee, Ubiquitous Computing: Connecting Pervasive Computing through Semantic Web, ISEB Journal (Accepted)

[41] Schlenoff et al. The essence of the process specification language. Transactions of the Society for Computer Simulation, 16(4):204-216, 1999. http://ats.nist.gov/psl/

[42] S. A. Chun, Y. Lee, J. Geller , Ontological and Pragmatic Knowledge Management for Web service Composition, 9th International Conference on Database Systems for Advanced Applications (DASFAA 2004) Lecture Notes in Computer Science 2973 Springer 2004, pp. 365-373.

[43] K. Supekar, The OnInt (Ontology Integration) Implementation Report, UMKC, Technical Report, 2002.

[44] W. W. Stead, R. A. Miller, M. A. Musen, and W. R. Hersh, Integration and Beyond: Linking Information from Disparate Sources and into Workflow, Am Med Inform Assoc 2000;7(2):135-145

[45] T. Strang, C. Linnhoff-Popien, and K. Frank. CoOL: A Context Ontology Language to enable Contextual Interoperability. LNCS 2893: Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003), pages 236-247, 2003.

[46] A. Tripathi, T. Ahmed, D. Kulkarni, R. Kumar, and K. Kashiramka, Context-Based Secure Resource Access in Pervasive Computing Environments In 1st IEEE International Workshop on Pervasive Computing and Communications Security(IEEE PerSec’04), To Appear.

[47] Tversky. Features of similarity. Psychological Review, 84:327-352, 1977.

[48] D. Waltz, Understanding Line Drawings with Shadows, P. Winston, The Psychology of Computer Vision, McGraw Hill, New York, 1975, pp. 19-91.

[49] M. Weiser, The Computer for the 21st Century, Scientific American., 1991, pp 94-104; reprinted in IEEE Pervasive Computing, Jan-Mar 2003, pp. 19-25

[50] X. Wang, D. Zhang, T. Gu, H. Fung, Ontology Based Context Modeling and Reasoning using OWL, Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004

Mining Local Data Sources For Learning Global Cluster Models Via Local Model Exchange

Xiao-Feng Zhang, Chak-Man Lam, William K. Cheung, *Member, IEEE*

***Abstract*— Distributed data mining has recently caught a lot of attention as there are many cases where pooling distributed data for mining is probibited, due to either huge data volume or data privacy. In this paper, we addressed the issue of learning a global cluster model, known as the latent class model, by mining distributed data sources. Most of the existing model learning algorithms (e.g., EM) require access to all the available training data. Instead, we studied a methodology based on periodic model exchange and merge, and applied it to Web structure modeling. In addition, we have tested a number of variations of the basic idea, including confining the exchange to some privacy friendly parameters and varying the number of distributed sources. Experimental results show that the proposed distributed learning scheme is effective with accuracy close to the case with all the data physically shared for the learning. Also, our results show empirically that sharing less model parameters as a further mechanism for privacy control does not result in significant performance degradation for our application.**

***Index Terms*— Distributed data mining, model-based learning, latent class model, privacy preservation**

I. Introduction

Most of the machine learning and data mining algorithms work with a rather basic assumption that all the training data can be pooled together in a centralized data repository. Recently, there exist a growing number of cases that the data have to be physically distributed due to some constraints. Examples include the data privacy concern in commercial enterprises where customers' private information are supposed not to be disclosed to other parties without their consent. Another example is mining individuals' incoming e-mails for some global patterns of junk mails, and sharing personal emails with others is a scenario which is almost impossible. Additional relevant examples including distributed medical data analysis, intrusion detection, data fusion in sensor networks, etc.[9] This calls for a lot of recent research interest on distributed machine learning and data mining [7].

A common methodology for distributed machine learning and data mining is of two-stage type — first performing local data analysis and then combining the local results forming the global one. For example, in [10], a meta-learning process was proposed as an additional learning process for combining a set of locally learned classifiers (decision trees in particular) for a global classifier. A related implementation has been realized under a Grid platform known as the Knowledge Grid [11]. In [9], Kargupta *et al.* proposed what they called collective data mining and the distributed data are assumed to possess different sets of features, each being considered as an orthogonal basis. The orthogonal bases are then combined to give the overall result. They have applied it to learning Bayesian Networks for Web log analysis [12], [8].

Regarding incorporation of local data privacy control in distributed data mining, Clifton *et al.* [13], [14], [15] and Du *et al.* [16], [17], [18] have proposed solutions to distributed association rules mining with privacy preserving capability. Under the premise that parties prefer to share the local data mining results instead of the original local data, each party site learns and disclose only their local patterns, which will eventually be aggregated together to form some global patterns. Other than taking associated rule mining, Merugu *et al.* [19], [20], [21], [22], [23] works on mining global clusters (in the form of Gaussian mixture model) of high dimension feature vectors which are distributed in different sites. Their proposed method starts with creating local cluster models and then resampling from the combined models "virtual" global samples for training the global model. A quantitative data privacy measure was proposed and they pointed out that some trade-off between the global model accuracy and local data privacy has to be made.

All the aforementoned methods adopt the two-stage methodology for distributed data mining. The instrinsic limitation is that patterns which emerge only when the local data are aggregated cannot be discovered at all. In this paper, instead of taking the two-stage methodology, we propose to allow the local data mining stage and the result combining stage to interleave. In particular, we choose the latent class model as an example, where the iterative expectation and minimization algorithm is typically used for estimating the model parameters based on some training data. We learn local latent class models based on the local data but allow the immediately learned model parameters to be exchanged. For merging the exchange models which are supposed to be heterogeneous, relative entropy is used as the measure for aligning, and thus merging, of the local latent classes. The main rationale of the proposed methodology lies on the conjecture that periodic sharing of intermediate local analysis results can reduce the biases due to the local data and thus help learn a more accurate global model. For performance evaluation, experiments on applying the proposed methodology to Web cluster analysis using both Web contents and links have been conducted where the WebKB dataset is used for benchmarking. A few variations of the proposed methodology have also been proposed by considering the situation that a higher level of privacy is required as well as that the degree of data distribution is different. We found that the proposed periodic model exchange methodoloy can achieve an global model accuracy higher than the case using the two-stage methodology, and sometimes can even

[1] William K. Cheung is with Hong Kong Baptist University, Hong Kong.

outperform the situation with all the data physically pooled together for the model learning. While the gain is due to the additional communication effort, we also provide the computational complexity and the communication cost analysis for comparing different model exchange settings.

The remaining of the paper is organized as follow. Section 2 describes a particular latent class model for modeling hyperlinked Web pages. Section 3 explains how the proposed periodic model-exchange methodology can be applied to the distributed model learning. Also, the computational complexity as well as the communication overhead involved are analyzed. Details about the experimental setup for evaluating the different variations of the basic idea as well as the corresponding results can be found in Section 4. Section 5 concludes the paper and proposes some possible future research directions.

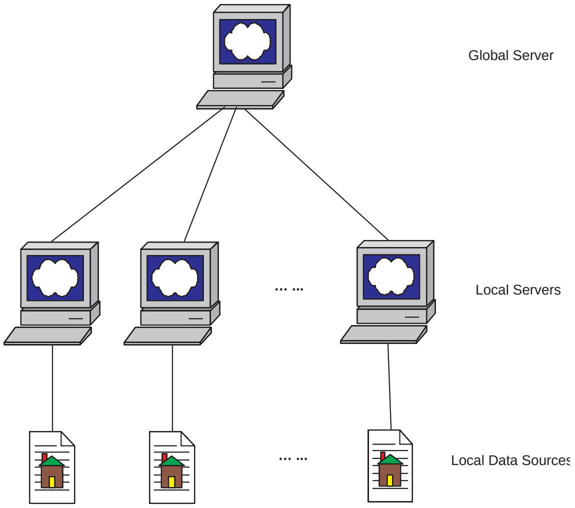


Fig. 1. A senario with a single global server mediating multiple physically distributed local servers.

II. Latent class models and Web structure analysis

The latent class model (LCM) is a statistical model under the family of mixture models. It has been adopted for modeling the co-occurence of multiple random variables with applications to a number of areas. A particular latent class model for analyzing Web contents and Web links was proposed in [2], which can be considered as a joint model of two related latent class models called PLSA [5] (for Web contents) and PHITS [1] (for Web links).

Let t_i denote the i^{th} term, d_j the j^{th} document, c_l the document being cited (or linked), N_{ij} the observed frequency that t_i exists in d_j, A_{lj} the observed frequency that c_l is being linked by d_j.

By assuming that given an underlying latent factor z_k, t_i and c_l are independent of d_j and are independent of each other, the log likelihood $\mathcal{L}$ of the observed data (Web pages) can be given as

$$\begin{aligned}\mathcal{L} &= \sum_j \Bigg[\alpha \sum_i N_{ij} \log \sum_k P(t_i|z_k)P(z_k|d_j) \\ &\quad +(1-\alpha)\sum_l A_{lj} \log \sum_k P(c_l|z_k)P(z_k|d_j)\Bigg] \end{aligned} \tag{1}$$

where α determines the relative importance between observed terms (used in PLSA) and observed links (used in PHITS). Data normalization is adopted as in [2] to reduce the bias due to different document sizes. Model parameters $\{P(t_i|z_k), P(c_l|z_k), P(z_k|d$ are estimated using the tempered Expectation and Maximization (EM) algorithm [2] so as to avoid the local minimum problem of the standard EM algorithm.

III. Model exchange methodology for LCM learning

As mentioned in Section 1, the main focus of this paper is to explore how well physically separately datasets can be used to learn a global cluster model (LCM in our case) through periodic model exchange. The traditional methodolody of distributed learning is to do it in a two-stage manner — finishing local analysis and then merging the local results. For LCM learning, it corresponds to learning the local LCMs $\{LCM^{lm}\}$ first based on terms and hyperlinks information observed at each distributed site, and then performing the model merging subsequently to form the global model LCM^{gm}. In this paper, we view this methodology as an *one-shot* model exchange scheme. Based on this scheme, only the standard LCM learning process is needed at each site and the accuracy of the global estimate is determined only by how well the local models are merged.

Instead of only exchanging models at the final stage, we here propose a *multiple* model exchange scheme, where the two stages of learning interleave to perform some *cross learning*. Other than accessing its local set of data, each local data source will, now, receive from time to time models of the other data sources to help the model estimation task. The EM step implementation needed at each local site for LCM learning will be affected as parameters of local and non-local models are needed to be merged for each exchange before the sequent EM steps can be proceeded. After all the models in the distributed sites converge, the finally merged LCM is denoted as LCM^{gm}.

In the following, details of the one-shot and multiple model exchange schemes are explained. Also, the computational complexity as well as the communication overhead of the proposed schemes will be discussed as both are important for serious applications.

A. One-shot model exchange scheme

In this model exchange scheme, we perform only two main steps, namely *local model learning* and *model merging*. Figure 2 shows the overview of the one-shot model exchange scheme.

1) Local model learning: The local model learning step first estimates the parameters of LCM_p^{lm} using the local term-document matrix N_{ij}^p and link-document matrix A_{lj}^p observed at the p^th site.[1] One can follow the computation as described in Section II to estimate the model parameters' values. For setting the value of α, it is believed that different sites, possessing different data, may require a different value of for optimal performance. In this paper, we learn multiple LCM^{lm}s within a site by varying α from zero to one, with lower and upper extremes corresponding to PHITS and PLSA, as explained in [2]. To find the optimal one, we first use a factored nearest neighbor approach for measuring the factoring accuracy. In particular, for a learned LCM corresponding to a given value of α, a Web page d_j is considered to be correctly factored by that LCM if it belongs to the same class[2] of its neighbors. To define the neighborhood, we compute the cosine value of the Web pages' projections on the factor space, given as

$$sim(\vec{P}(z|d_i), \vec{P}(z|d_j)) = \frac{\vec{P}(z|d_i) \cdot \vec{P}(z|d_j)}{\|\vec{P}(z|d_i)\| \cdot \|\vec{P}(z|d_j)\|}. \quad (2)$$

The model associated to an α which gives the highest overall accuracy will be chosen for the subsequent merging.

2) Model merging: It is common that distributed data sources are heterogeneous. For example, in our case, the data at different Web sites are best described by different parameter sets, involving different terms, links as well as different latent classes (hidden patterns) captured by z. In order to combine different local models $\{LCM_p^{lm}\}$ to form a global one, we first need to assume that the unique identity of each data item can be identified to the extent that repeated appearance of them in different sites can be found. Thus, those repeated data items, after merging, can be re-indexed to aggregate their effect in the learning process. After reindexing, the latent parts of the local models whose identities can never be pre-defined have to be aligned before they can be merged.

Re-indexing: For each local model, we first enlarge and re-index the set of model parameters $\{P(z|d), P(t|z), P(c|z)\}$ by noting the difference between the local model and the other non-local models received from the other data sources. The parameters of the unseen variables are first initialized to zero.

Latent variables matching: As the latent part of each local LCM is induced from their corresponding training datasets, it is hard to have a pre-agreed way to know how they should be matched. Here, we propose to use the relative entropy between the probability distributions of the latent variables for a pair of local LCMs to align their latent variables.

For our application domain, two cases are to be considered: a) Web pages in different sites are non-overlapping, and b) some Web pages are shared in different sites. For the former case, we merely need to consider $P(t_i|z_k)$ and the relative entropy of a pair of latent variables z_k and $z_{k'}$ corresponding to two local models LCM_p^{lm} and $LCM_{p'}^{lm}$ is given as

$$H1_{p,p'}(z_k, z_{k'}) = \sum_i P_p(t_i|z_k) \log \frac{P_p(t_i|z_k)}{P_{p'}(t_i|z_{k'})}. \quad (3)$$

For the latter case, we use $P(t_i, c_l|z_k)$ for computing the relative entropy, given as

$$H2_{p,p'}(z_k, z_{k'}) = \sum_i \sum_l P_p(t_i, c_l|z_k) \log \frac{P_p(t_i, c_l|z_k)}{P_{p'}(t_i, c_l|z_{k'})}. \quad (4)$$

Two latent classes are considered to be closely matched if the value of their relative entropy is close to zero. The best one-to-one matching between the two sets of latent class models are computed based on the matrix $\{H1_{p,p'}(z_k, z_{k'})\}$ or $\{H2_{p,p'}(z_k, z_{k'})\}$. In this paper, we only consider the case where the LCMs have identical numbers of latent variables and assume that their latent variables possess the one-to-one correspondence property. In general, these assumptions should be relaxed.

Parameter merging: After the latent variables are matched, we can readily combine the local and non-local model parameters. For simplicity, we use simple averaging for the merge. A weighted sum based on some accuracy or uncertainty measures of the local models may worth further research effort.

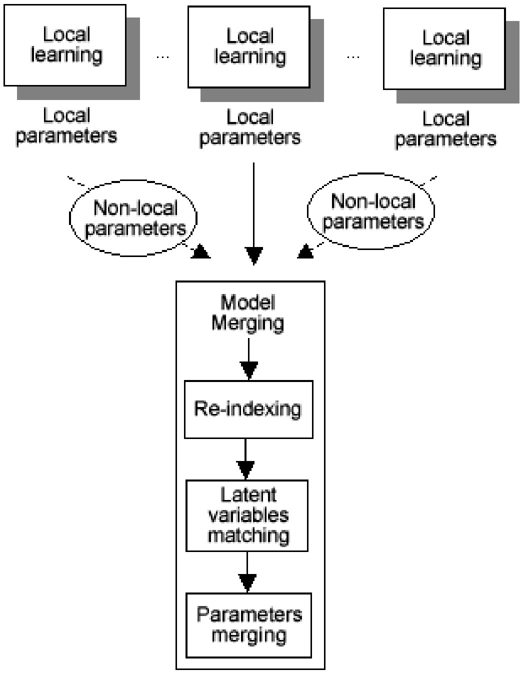


Fig. 2. Overview of one-shot model exchange scheme.

B. Multiple model exchange scheme

Under the multiple model exchange scheme, the local learning and model merging steps for one-shot model exchange

[1] Note that cross-site links are not considered in this pilot study, which however is an important part to be included in our future work.

[2] The class labels are available in the training set.

interleave *during* the learning process, which we call it *cross learning*. Cross learning is here defined as learning a local model with the use of non-local information *during* the learning process. Local model parameterss are exchanged at the intermediate stages, instead of the final stage. Similar to the one-shot model exchange scheme, such a cross learning process involves four steps, namely re-indexing, latent variables matching, parameter merging and local model parameter estimation. Most of them are identical to those for the one-shot model exchange, except for some minor implementation details. However, as the model exchange happens multiple times, the exchanging and merging steps could have much more influence on the overall performance. The main rationale is that periodic sharing of intermediate local analysis results can reduce the biases due to the local data and thus help learn a more accurate overall global model. Figure 3 shown the overview of the periodic model exchange scheme.

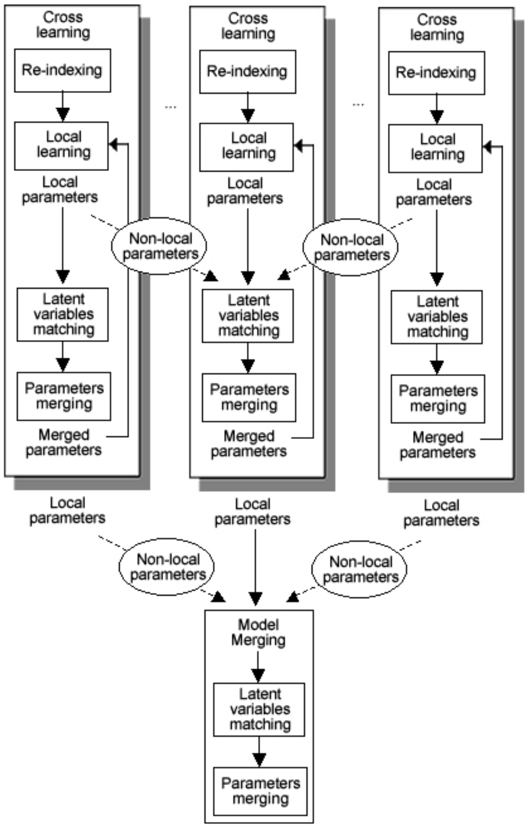


Fig. 3. Overview of multiple model exchange scheme.

C. Communication overhead and computational complexity

In this section, the asymptotic communication overhead and computational complexity of the two model exchange schemes are discussed in detail. Table I shows the notations used. Here, the communication overhead (CO) per model exchange includes parameters transmission. Related overheads for the two schemes are basically the same, given as

TABLE I
NOTATIONS

Notation	Definition
M/M_g	Number of local/global Web pages
N/N_g	Number of local/global hyperlinks
P/P_g	Number of local /global terms
Q	Number of latent variables
R	Number of distributed sources
I_{ter}	Number of EM iterations
I_{ex}	Number of non-local parameters exchanges
CO	Overhead of parameters transmission per model exchange

$$\begin{aligned} CO &= O(Q(M+N+P)/bandwidth) \\ &= O(rQ(M+N+P)) \end{aligned}$$

For the computational complexity, we compare the performance of the two exchange schemes ($O_{one}, O_{multiple}$) as well as the case with a single centralized server hosting all the data ($O_{central}$). They are given as

$$\begin{aligned} O_{central} &= O(I_{ter} M_g Q(N_g + P_g)) \\ O_{one} &= O(I_{ter} MQ(N+P)) \\ O_{multiple} &= O(I_{ter} MQ(N+P) \end{aligned}$$

For the overall complexity ($O^{overall}$), we add up the communication overheads and the computational ones, given as

$$\begin{aligned} O^{all}_{central} &= O(I_{ter} M_g Q(N_g + P_g)) \\ O^{all}_{one} &= O(I_{ter} Q((N+P)(M+r) + rM)) \\ O^{all}_{multiple} &= O(I_{ter} Q((N+P)(M+rI_{ex}) + rM)) \\ &= O^{all}_{one} + O(rI_{ter} I_{ex} Q(N+P)). \end{aligned}$$

Thus, it is noted that the communication overhead (CO) becomes insignificant when the size of the dataset (of the order $M(N+P)$) is much larger than that of the models (of the order $(M+N+P)$). The overall computational complexity will still be dominated by the local learning processes. Furthermore, M is much smaller than M_g in general as R increases (that is the data are more distributed). Therefore, the parallellism gained by the independent learning of the local models $\{LCM^{lm}\}$ should result in a shorter overall learning time when compared with that of the global model LCM^{gm}.

D. Model exchange scheme with additional privacy preserving cabability

One of the important motivations for sharing models instead of data is related to data privacy. For the aforementioned application on Web structure analysis, sharing local LCM models assumes the knowledge of a set of unique identifiers for all the Web pages at different data sources, no matter they are for public access or within the intranets. While each site can relabel all the identifiers based on the URLs of the Web pages, sharing those identifiers may still cause privacy and security concern of internal users of different sites. The situation will become even more obvious if we replace each Web page by a customer and each hyperlink by a product that a customer has purchased [24]. There will not be a company willing to share with others their transaction records. One effective way to alleviate the aforementioned privacy issue is to share only aggregated information. In our case, the model parameters belonging to this category (which we refer as the most *privacy-friendly* parameters as illustrated in Figure 4) is $P(t|z)$.

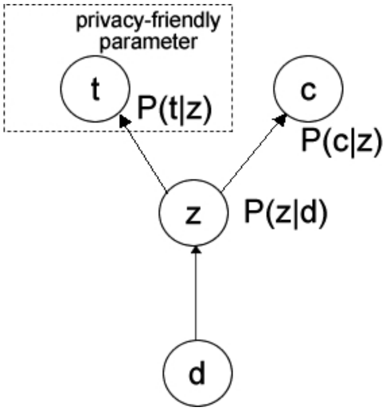


Fig. 4. Privacy-friendly parameters of LCM.

By sharing only $P(t|z)$, we gain additional advantage due to the reduced requirement of communication cost as well as computational complexity. Note that as the value of N and M increases, the increase in P will soon be saturated if the vocabulary under a particular domain is exhausted. The corresponding overall complexity can be reduced to:

$$O^{all}_{multiple} = O^{all}_{one} + O(rQPI_{ter}I_{ex}).$$

IV. Performance Evaluation

We have applied the proposed model exchange approach to the WebKB dataset [6]. As this study is novel and there are no directly related works in the literature, we conduct our experiments to compare the commonly adopted two-stage approach and the proposed multiple model exchange approach. While our experiment focuses only on learning LCM in a distributed manner, we believe that the approach should also apply to distributed learning of other statistical models.

For the training and testing dataset, a total of 546 web pages, which are pre-classified into 3 categories: *course*, *department* and *student* in WebKB, have been used and each class contains 182 pages. In the following, we describe the data pre-processing steps adopted and how the experiments were designed and conducted.

A. Web page preprocessing

As mentioned in Section II, the term-document matrix N_{ij} and hyperlink-document matrix A_{lj} are required for the LCM learning. Hyperlinks between Web pages can easily be identified based on the anchor tags for computing A_{lj}. For Web page contents, we removed all the html tags as well as the contents between the `<SCRIPT>` tags. Also, stopwords removal and stemming [4] were applied subsequently. The remaining terms were all changed to be of lower case. We then extracted only terms with their document frequencies bigger than a threshold value [3]. We have tested the threshold of 5, 10, and 20 (denoted as DF05, DF10, DF20), resulting in datasets with their numbers of distinct words equal 1629, 957 and 550 respectively. The factored nearest neighbor approach (1-nn and 3-nn), as described in Section III-A.1, is used for comparing their accuracy and the corresponding results are shown in Table II. We found that DF10 and DF20 outperform DF5 and the performance of DF10 and DF20 are comparable. As a smaller number of terms implies lower computational complexity as shown in the previous section, DF20 was used in the subsequent experiments.

TABLE II
CLASSIFICATION ACCURACY (%) FOR D05, D10 AND D20.

	1-nn (%)	3-nn (%)
D05	81.46	82.71
D10	86.30	85.20
D20	86.41	86.81

B. Experiment setups for different model exchange schemes

We performed a number of experiments for learning latent class models using the one-shot and multiple model exchange schemes with 1) different parameter exchange periods to indicate different degrees of non-local data availability 2) different numbers of distributed data sources to indicate different degrees of data distribution. In particular, we have tried different exchange periods, 2, 5, 10, 15, 20 and ∞ (which degenerates to one-shot model exchange case) and performed the experiments with 2 to 6 distributed data sources. For preparing the distributed data sources, we partitioned the WebKB dataset so that part of Web pages in one partition also appear in some others. Classification accuracy (as described in Section III-A.1) and the training time are the performance measures we adopted. As the local models learned at the distributed sites have to synchronize at each model exchange stage, in our experiment, we recorded the maximum computational time among those needed by the distributed servers. To contrast the additional privacy concern mentioned in Section III-D, we deliberately learned an LCM by exchanging all three sets of model parameters, i.e., $P(t|z)$, $P(c|z)$ and $P(z|d)$, and another only the privacy-friendly parameters, i.e., $P(t|z)$ for performance comparison. Lastly, as the EM algorithm only gives sub-optimal solutions, for each LCM training, we have tried ten different random initializations and reported the average performance of the ten cases.

C. Experimental Results

1) Performance comparison for exchange different sets of model parameters: The classification accuracy and the training time associated to the distributed LCM learning with all parameters exchanged and with only the privacy-friendly parameters exchanged for two distributed sets are tabulated in Table III and IV, respectively.

TABLE III
CLASSIFICATION ACCURACY (%) AND TRAINING TIME (SEC) BASED ON EXCHANGING THE FULL SET OF PARAMETERS.

Exchange period	**1-nn (%)**	**3-nn (%)**	**Time (mm:ss)**
∞	81.85	80.90	0:54
20	77.93	78.64	1:47
15	78.04	77.91	1:41
10	78.59	79.38	2:11
5	80.73	82.11	2:17
2	**83.26**	**84.71**	2:29

TABLE IV
CLASSIFICATION ACCURACY (%) AND TRAINING TIME (SEC) BASED ON EXCHANGING ONLY THE PRIVACY-FRIENDLY PARAMETERS.

	1-nn (%)	**3-nn (%)**	**Time (mm:ss)**
∞	83.11	82.75	0:56
20	84.45	81.45	0:55
15	82.03	81.47	0:57
10	80.93	81.32	1:03
5	83.11	83.10	1:06
2	**87.51**	**87.55**	1:43

According to Table III and IV, it is observed that the performance of exchanging only the privacy-friendly parameters is always better than that of exchanging the full set of model parameters. In addition, as expected, the computational time for exchanging only the privacy-friendly parameters is significantly less than that for exchanging all model parameters because of the reduced communication overhead. This effect is especially obvious for cases with higher exchange frequencies. Therefore, in the following experiments, we only adopt the scheme of exchanging $\{P(t|z)\}$.

TABLE V
CLASSIFICATION ACCURACY (%) EVALUATED BY 1-NN BASED ON DIFFERENT DEGREES OF DATA DISTRIBUTION AND DIFFERENT MODEL EXCHANGE PERIODS.

	∞	**20**	**15**	**10**	**5**	**2**
2 sets	83.11	81.45	82.03	80.93	83.11	**87.51**
3 sets	**87.23**	86.74	86.52	86.43	85.55	85.82
4 sets	84.93	85.53	**87.57**	87.14	79.43	83.86
5 sets	77.01	79.38	79.93	**84.40**	79.74	83.22
6 sets	76.85	79.62	82.05	83.46	81.52	**83.55**

TABLE VI
CLASSIFICATION ACCURACY (%) EVALUATED BY 3-NN BASED ON DIFFERENT DEGREES OF DATA DISTRIBUTION AND DIFFERENT MODEL EXCHANGE PERIODS.

	∞	**20**	**15**	**10**	**5**	**2**
2 sets	82.75	81.45	81.47	81.32	83.10	**87.55**
3 sets	87.45	87.40	**87.73**	87.42	85.55	86.65
4 sets	84.30	85.27	87.29	**87.78**	85.95	85.79
5 sets	78.11	79.41	80.29	84.07	84.45	**85.11**
6 sets	75.66	78.74	82.05	83.04	84.18	**85.55**

TABLE VII
TRAINING TIME (SEC) BASED ON DIFFERENT DEGREES OF DATA DISTRIBUTION.

	∞	**20**	**15**	**10**	**5**	**2**
2 sets	0:56	0:55	0:57	1:03	1:06	1:43
3 sets	0:50	0:52	0:55	0:50	1:37	1:34
4 sets	0:40	0:41	0:41	0:41	1:33	1:30
5 sets	0:37	0:33	0:41	0:42	1:29	1:28
6 sets	0:30	0:33	0:33	0:35	1:24	1:24

2) Performance sensitivity on different degrees of data distribution and different model exchange periods: In Table V to VII, the classification accuracy evaluated based on the 1-nn and 3-nn factoring approaches as well as the training time for learning the LCM based on different experiment settings are reported. According to Table V and VI, the accuracy decreases monotonically as the number of distributed sources increases. It is possibly due to the fact that when data are distributed to different sites, the amount of available information for each source decreases. Therefore, the overall performance is reduced. The gain, as shown in Table VII, is that the training time is reduced (due to the parallellism). By allowing model exchange, as observed in Table V and VI, we found that the accuracy can be significantly increases, the trend is not especially clear though. In general, allowing parameters exchange more frequently can result in better overall performance of the global model.

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented a methodology for learning a global latent model from distributed data sources by multiple model exchanges with promising results. With the option to exchange only the privacy-friendly parameters, our empirical results show that the overall model gives acceptable and sometimes even better accuracy. In addition, we observed that while the increase in the number of distributed sites can lower the overall accuracy of the global model, the interpolating effect caused by the model exchange can improve the accuracy to some extent.

While this work provides us some interesting and encouraging results for exploring distributed data mining through model (or generally speaking knowledge) exchange, there still exist a number of areas worth further research effort. What we have proposed in this paper is a model-specific methodology for distributed data mining. Ways for generalizing the proposed methodology so as to be applied to different types of models is one of the worth-pursuing research directions.

In addition, the way we exchange all local model parameters or privacy-friendly parameters is based on some fixed time periods. one can go one step further to derive adaptive on-demand model exchange strategies for minimizing the communication cost while still maintaining the desired accuracy. Furthermore, there still exist no guarantee for the convergence of the global model. We are currently investigating different discounting strategies for addressing the model convergence issue.

References

[1] D.Cohn, H.Chang, “Learning to probabilistically identify authoritative documents,” *Proceedings of the 17th International Conference on Machine Learning*, 2000.

[2] D.Cohn, T.Hofmann, “The missing link - A probabilistic model of document content and hypertext connectivity,” *Advances in Neural Information Processing Systems*, 2001.

[3] E. Bingham, J. Kuusisto and K. Lagus, “ICA and SOM in text document analysis,” *Proceedings of SIGIR’02*, 361 - 362 2002

[4] G.Salton and M.J.McGill, *Introduction to modern information retrieval*, McGraw-Hill, New York, 1983.

[5] T. Hofmann, “Probabilistic latent semantic analysis,” *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[6] Web-KB. Available electronically at *http://www.cs.cmu.edu/˜WebKB/*.

[7] Distributed Data Mining Bibliography. Available at *URL: http://www.csee.umbc.edu/˜hillol/DDMBIB/ ddmbib_html/index.html*.

[8] R. Chen and S. Krishnamoorthy, “A new algorithm for learning parameters of a Bayesian network from distributed data,” Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), pages 585-588, Maebashi City, Japan, December 2002.

[9] H. Kargupta, B. Park, D. Hershberger, and E. Johnson, “Collective data mining: A new perspective towards distributed data mining,” *Advances in Distributed and Parallel Knowledge Discovery*, pages 133-184. MIT/AAAI Press, 2000.

[10] A. Prodromidis and P. Chan, “Meta-learning in distributed data mining systems: Issues and approaches,” *Advances of Distributed Data Mining*, MIT/AAAI Press, 2000.

[11] M. Cannataro and D. Talia, “The Knowledge Grid,” *Communications of the ACM*, 46(1):89-93, January 2003.

[12] R. Chen, S. Krishnamoorthy, and H. Kargupta, “Distributed Web mining using Bayesian networks from multiple data streams,” *Proceedings of the IEEE International Conference on Data Mining*, pages 281-288, IEEE Press, November 2001.

[13] J. Vaidya and C. Clifton, “Privacy preserving association rule mining in vertically partitioned data,” *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 639-644, July 2002.

[14] J. Vaidya and C. Clifton, “Privacy preserving K-means clustering over vertically partitioned data,” *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 206-215, August 2003.

[15] C. Clifton, M. Kantarcioglu, J. Vaidya, X.D. Lin and M.Y. Zhu, “Tools for privacy preserving distributed data mining,” *ACM SIGKDD Explorations Newsletter*, 4(2):28 - 34, 2002.

[16] W.L. Du, Yunghsiang S. Han and S. Chen, “Privacy-preserving multivariate statistical analysis: Linear regression and classification,” *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 222-233, 2004.

[17] H. Polat and W.L. Du, “Privacy-preserving collaborative filtering using randomized perturbation techniques,” *Proceedings of The Third IEEE International Conference on Data Mining*, pages 625-628, 2003.

[18] W.L. Du and Z.J. Zhan, “Using randomized response techniques for privacy-preserving data mining,” *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 505-510, 2003.

[19] S. Merugu and J. Ghosh, “A probabilistic approach to privacy-sensitive distributed data mining,” *Proceedings of the Sixth International Conference on Information Technology (CIT)*, 2003.

[20] S. Merugu, A. Banerjee, I. Dhillon and J. Ghosh, “Clustering with Bregman Divergences,” *Proceedings of the Fourth IEEE International Conference on Data Mining*, 2004.

[21] S. Merugu and J. Ghosh, “Distributed data mining with limited knowledge sharing,” *Proceedings of the Fifth International Conference on Advances in Pattern Recognition (ICAPR)*, 2003.

[22] S. Merugu and J. Ghosh., “Privacy-preserving distributed clustering using generative models,” *Proceedings of the Third IEEE International Conference on Data Mining*, 2003.

[23] S. Merugu, J. Ghosh and A. Strehl, “A Consensus Framework for Integrating Distributed Clusterings under Limited Knowledge Sharing,” *Proceedings of the National Science Foundation (NSF) Workshop on Next Generation Data Mining*, 2002.

[24] K. Cheung, K.C. Tsui and J. Liu, “Extended Latent Class Model for Collaborative Recommendation,” IEEE Transactions on Systems, Man and Cybernetics - Part A, Vol.34, No.1, pp. 143-147, January, 2004.

Data Mining: An AI Perspective

Xindong Wu[1], *Senior Member, IEEE*

Abstract--**Data mining, or knowledge discovery in databases (KDD), is an interdisciplinary area that integrates techniques from several fields including machine learning, statistics, and database systems, for the analysis of large volumes of data. This paper reviews the topics of interest from the IEEE International Conference on Data Mining (ICDM) from an AI perspective. We discuss common topics in data mining and AI, including key AI ideas that have been used in both data mining and machine learning.**

Index Terms—**Data Mining, Artificial Intelligence, Machine Learning.**

I. The IEEE International Conference on Data Mining

DATA mining is a fast-growing area. The first Knowledge Discovery in Databases Workshop was held in August 1989, in conjunction with the 1989 International Joint Conference on Artificial Intelligence, and this workshop series became the International Conference on Knowledge Discovery and Data Mining in 1995. In 2003, there were a total of 15 data mining conferences, most of which are listed at http://www.kdnuggets.com/meetings/meetings-2003-past.html:

- Data Warehousing and Data Mining in Drug Development (January 13-14, 2003, Philadelphia, PA, USA)
- First Annual Forum on Data Mining Technology for Military and Government Applications (February 25-26, 2003, Washington DC, USA)
- SPIE Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology V (21-22 April 2003, http://www.spie.org/Conferences/Programs/03/or/conferences/index.cfm?fuseaction=5098)
- PAKDD-03: 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining (April 30 - May 2, 2003, Seoul, Korea)
- SDM 03: 3rd SIAM International Conference on Data Mining (May 1-3, 2003, San Francisco, CA, USA)
- MLDM 2003: Machine Learning and Data Mining (July 5-7, 2003, Leipzig, Germany)
- KDD-2003, 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (August 24-27, 2003, Washington DC, USA)
- IDA-2003, 5th International Symposium on Intelligent Data Analysis (August 28-30, 2003, Berlin, Germany)
- DaWaK 2003: 5th International Conference on Data Warehousing and Knowledge Discovery (September 3-5, 2003, Prague, Czech Repblic)
- PKDD-2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (September 22-26, 2003, Cavtat-Dubrovnik, Croatia)
- SAS M2003: 6th Annual Data Mining Technology Conference (October 13-14, 2003, Las Vegas, NV, USA)
- Data Warehousing & Data Mining for Energy Companies (October 16-17, 2003, Houston, TX, USA)
- CAMDA 2003: Critical Assessment of Microarray Data Analysis (November 12-14, 2003, Durham, NC, USA)
- ICDM-2003: 3rd IEEE International Conference on Data Mining (November 19 - 22, 2003, Melbourne, FL, USA)
- The Australasian Data Mining Workshop (December 8, 2003, Canberra, Australia, http://datamining.csiro.au/adm03/)

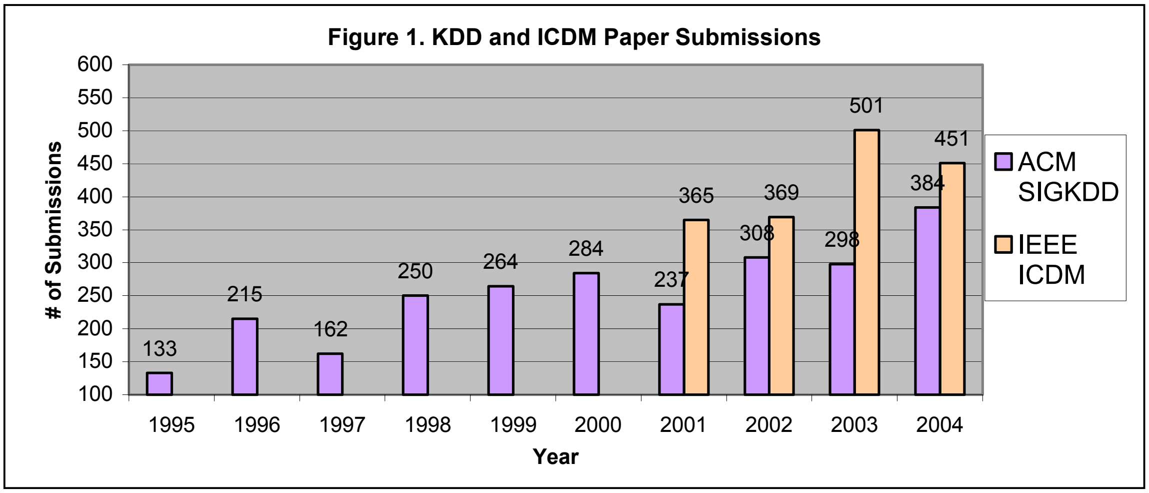
These 15 conferences do not include various artificial intelligence (AI), statistics and database conferences (and their workshops) that also solicited and accepted data mining related papers, such as IJCAI, ICML, ICTAI, COMPSTAT, AI & Statistics, SIGMOD, VLDB, ICDE, and CIKM.

Among various data mining conferences, KDD and ICDM are arguably (or unarguably) the two premier ones in the field. ICDM was established in 2000, sponsored by the IEEE Computer Society, and had its first annual meeting in 2001. Figure 1 shows the number of paper submissions to each KDD and ICDM conference.

Topics of interest from the ICDM 2003 call for papers [http://www.cs.uvm.edu/~xwu/icdm-03.shtml] are listed here:

1. Foundations of data mining
2. Data mining and machine learning algorithms and methods in traditional areas (such as classification, regression, clustering, probabilistic modeling, and association analysis), and in new areas
3. Mining text and semi-structured data, and mining temporal, spatial and multimedia data
4. Data and knowledge representation for data mining
5. Complexity, efficiency, and scalability issues in data mining

[1] Xindong Wu is with the Department of Computer Science, University of Vermont Burlington, VT 05405, USA (e-mail: xwu@cs.uvm.edu).



6. Data pre-processing, data reduction, feature selection and feature transformation
7. Post-processing of data mining results
8. Statistics and probability in large-scale data mining
9. Soft computing (including neural networks, fuzzy logic, evolutionary computation, and rough sets) and uncertainty management for data mining
10. Integration of data warehousing, OLAP and data mining
11. Human-machine interaction and visualization in data mining, and visual data mining
12. High performance and distributed data mining
13. Pattern recognition and scientific discovery
14. Quality assessment and interestingness metrics of data mining results
15. Process-centric data mining and models of data mining process
16. Security, privacy and social impact of data mining
17. Data mining applications in electronic commerce, bioinformatics, computer security, Web intelligence, intelligent learning database systems, finance, marketing, healthcare, telecommunications, and other fields

Clearly, some of the above topics are of interest from the database and statistics perspectives [Chen, Han and Yu 1996; Elder and Pregibon 1996; Zhou 2003]. Since the database perspective [Chen, Han and Yu 1996] and statistical perspective [Elder and Pregibon 1996] have been discussed and reviewed in detail in the literature, this paper concentrates on an AI perspective. We list the best papers selected from ICDM '01, '02, and '03 in Section 2, and discuss common topics in data mining and AI in Section 3.

II. Best Papers Selected from ICDM 2001, 2002, and 2003

Below are the best papers selected from ICDM 2001, 2002 and 2003, which have been expanded and revised for publication in Knowledge and Information Systems (http://www.cs.uvm.edu/~kais/), a peer-reviewed archival journal published by Springer-Verlag. The reference number before each paper, such as S336, M557 and R281, is the original submission number to each year's ICDM conference. We will see in Section III.A that these papers are all relevant to machine learning topics in AI.

ICDM 2001:

1. [S336] Discovering Similar Patterns for Characterising Time Series in a Medical Domain, by Fernando Alonso, Juan P. Caraça-Valente, Loïc Martínez, and Cesar Montes
2. [S409] Preprocessing Opportunities in Optimal Numerical Range Partitioning, by Tapio Elomaa and Juho Rousu
3. [S430] Using Artitificial Anomalies to Detect Known and Unknown Network Intrusions, by Wei Fan, Matthew Miller, Salvatore J. Stolfo, and Wenke Lee
4. [S457] Meta-Patterns: Revealing Hidden Periodic Patterns, by Wei Wang, Jiong Yang, and Philip Yu
5. [S516] Closing the Loop: an Agenda- and Justification-Based Framework for Selecting the Next Discovery Task to Perform, by Gary R. Livingston, John M. Rosenberg, and Bruce G. Buchanan

ICDM 2002:

1. [M557] Convex Hull Ensemble Machine, by Yongdai Kim
2. [M572] Phrase-based Document Similarity Based on an Index Graph Model, by Khaled Hammouda and Mohamed Kamel
3. [M632] High Performance Data Mining Using the Nearest Neighbor Join, by Christian Bohm and Florian Krebs
4. [M741] Efficient Discovery of Common Substructures in Macromolecules, by Srinivasan Parthasarathy and Matt Coatney
5. [M782] On the Mining of Substitution Rules for Statistically Dependent Items, by Wei-Guang Teng, Ming-Jyh Hsieh, and Ming-Syan Chen

ICDM 2003:

1. [R281] Clustering of Streaming Time Series is Meaningless: Implications for Previous and Future Research, by Jessica Lin, Eamonn Keogh, and Wagner Truppel
2. [R405] A High-Performance Distributed Algorithm for Mining Association Rules, by Ran Wolff, Assaf Schuster, and Dan Trock
3. [R493] TSP: Mining Top-K Closed Sequential Patterns, by Petre Tzvetkov, Xifeng Yan, and Jiawei Han
4. [R528] ExAMiner: Optimized Level-wise Frequent Pattern Mining with Monotone Constraints, by Francesco Bonchi, Fosca Giannotti, Alessio Mazzanti, and Dino Pedreschi
5. [R565] Reliable Detection of Episodes in Event Sequences, by Robert Gwadera, Mikhail Atallah, and Wojciech Szpankowski
6. [R620] On the Privacy Preserving Properties of Random Data Perturbation Techniques, by Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar

III. Common Topics in Data Mining and AI

A. Data Mining Papers on Machine Learning Topics

Machine learning in AI is the most relevant area to data mining, from the AI perspective. ICML 2003 [http://www.hpl.hp.com/conferences/icml03/] especially invited paper submissions on the following topics:

1. Applications of machine learning, particularly:
   a. exploratory research that describes novel learning tasks;
   b. applications that require non-standard techniques or shed light on limitations of existing learning techniques; and
   c. work that investigates the effect of the developers' decisions about problem formulation, representation or data quality on the learning process.
2. Analysis of learning algorithms that demonstrate generalization ability and also lead to better understanding of the computational complexity of learning.
3. The role of learning in spatial reasoning, motor control, and more generally in the performance of intelligent autonomous agents.
4. The discovery of scientific laws and taxonomies, and the induction of structured models from data.
5. Computational models of human learning.
6. Novel formulations of and insights into data clustering.
7. Learning from non-static data sources: incremental induction, on-line learning and learning from data streams.

Apart from Topic 5, all other topics above are relevant in significant ways to the topics of the 2003 IEEE International Conference on Data Mining listed in Section 1. Topic 2 is relevant to topics 2 and 5 in Section 1, Topic 3 overlaps with topics 3 and 1 in Section 1, and Topic 1 above and topic 17 in Section 1 both deal with applications. In practice, it is rather difficult to clearly distinguish a data mining application from a machine learning application, as long as an induction/learning task in involved. In fact, data mining and machine learning share the emphases on efficiency, effectiveness, and validity [Zhou 2003].

Meanwhile, every best paper from ICDM 2001, 2002 and 2003 in Section 2 can fit in the above ICML 2003 topics. With the exception of data pre-processing and post-processing, which might not involve any particular mining task, a data mining paper can generally find its relevance to a machine learning conference.

B. Three Fundamental AI Techniques in Data Mining

AI is a broader area than machine learning. AI systems are knowledge processing systems. Knowledge representation, knowledge acquisition, and inference including search and control, are three fundamental techniques in AI.

- ❖ **Knowledge representation**. Data mining seeks to discover interesting patterns from large volumes of data. These patterns can take various forms, such as association rules, classification rules, and decision trees, and therefore, knowledge representation (Topic 4 of ICDM 2003 in Section 1) becomes an issue of interest in data mining.
- ❖ **Knowledge acquisition**. The discovery process shares various algorithms and methods (Topics 2 and 6) with machine learning for the same purpose of knowledge acquisition from data [Wu 1995] or learning from examples.
- ❖ **Knowledge inference**. The patterns discovered from data need to be verified in various applications (Topics 7 and 17) and so deduction of mining results is an essential technique in data mining applications.

Therefore, knowledge representation, knowledge acquisition and knowledge inference, the three fundamental techniques in AI are all relevant to data mining.

Meanwhile, data mining was explicitly listed in the IJCAI 2003 call for papers [http://www.ijcai-03.org/1024/index.html] as an area keyword.

C. Key Methods Shared in AI and Data Mining

AI research is concerned with the principles and design of rational agents [Russell and Norvig 2003], and data mining systems can be good examples of such rational agents. Most AI research areas (such as reasoning, planning, natural language processing, game playing and robotics) have concentrated on the development of symbolic and heuristic methods to solve complex problems efficiently. These methods have also found extensive use in data mining.

- **Symbolic computation**. Many data mining algorithms deal with symbolic values. As a matter of fact, since a large number of data mining algorithms were developed to primarily deal with symbolic values, discretization of continuous attributes has been a popular and important topic in data mining for many years, so that those algorithms can be extended to handle both symbolic and real-valued attributes.
- **Heuristic search**. As in AI, many data mining problems are NP-hard, such as constructing the best decision tree from a given data set, and clustering a given number of data objects into an optimal number of groups. Therefore, heuristic search, divide and conquer, and knowledge acquisition from multiple sources [Zhang, Zhang and Wu 2004] have been common techniques in both data mining and machine learning.

For example, Ross Quinlan's information gain and gain ratio methods for decision tree construction, which uses a greedy search with divide and conquer, is introduced in both [Russell and Norvig 2003] and [Han and Kamber 2000], which are probably the most popular textbooks in AI and data mining respectively. Decision tree construction can make use of both symbolic and real-valued attributes.

Neural networks and evolutional algorithms (including genetic algorithms) are also covered in various AI and data mining references.

IV. Conclusion

Knowledge discovery from large volumes of data is a research frontier for both data mining and AI, and has seen sustained research in recent years. From the analysis of their common topics, this sustained research also acts as a link between the two fields, thus offering a dual benefit. First, because data mining is finding wide application in many fields, AI research obviously stands to gain from this greater exposure. Second, AI techniques can further augment the ability of existing data mining systems to represent, acquire, and process various types of knowledge and patterns that can be integrated into many large, advanced applications, such as computational biology, Web mining, and fraud detection.

Acknowledgements

The author would like to express his appreciation for the anonymous feedback from the Editorial Board on an earlier version of this paper, which has helped improve the paper.

References

[1] Ming-Syan Chen, Jiawei Han, and Philip Yu, Data Mining: An Overview from a Database Perspective, *IEEE Transactions on Knowledge and Data Engineering*, **8** (1996), 6: 866-883.

[2] John Elder IV and Daryl Pregibon, A Statistical Perspective on Knowledge Discovery in Databases, in *Advances in Knowledge Discovery and Data Mining*, Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy (Eds.), AAAI Press, 1996, 83-113.

[3] Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2000.

[4] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach, Second Edition*, Prentice-Hall, 2003.

[5] X. Wu, *Knowledge Acquisition from Databases*, Ablex Publishing Corp., U.S.A., 1995.

[6] S Zhang, C Zhang, and X Wu, *Knowledge Discovery in Multiple Databases*, Springer-Verlag, 2004.

[7] Zhi-Hua Zhou, Three Perspectives of Data Mining, *Artificial Intelligence*, **143**(2003), 1: 139-146.

RELATED CONFERENCES, CALL FOR PAPERS, AND CAREER OPPORTUNITIES

TCII Sponsored Conferences

WI 2005
The 2005 IEEE/WIC/ACM International Conference on Web Intelligence

Compiègne, France
September 19-21, 2005
http://www.comp.hkbu.edu.hk/WI05/
Submission Deadline: April 3, 2005

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 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05) will be jointly held with The 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05). The IEEE/WIC/ACM 2005 joint conferences are sponsored and organized by IEEE Computer Society Technical Committee on Intelligent Informatics (TCII), Web Intelligence Consortium (WIC), and ACM-SIGART.

Following the great successes of WI'01 held in Maebashi City, Japan , WI'03 held in Halifax, Canada, and WI'04 held in Beijing, China. WI 2005 provides a leading international forum for researchers and practitioners (1) to present the state-of-the-art of 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 2005 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 2005
The 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology

Compiègne, France
September 19-21, 2005
http://www.comp.hkbu.edu.hk/IAT05/
Submission Deadline: April 3, 2005

The 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05) will be jointly held with The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05). The IEEE/WIC/ACM 2005 joint conferences are sponsored and organized by IEEE Computer Society Technical Committee on Intelligent Informatics (TCII), 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, and IAT-04 held in Beijing, China.

IAT 2005 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 2005 will foster the development of novel paradigms and advanced solutions in agent-based computing.

————————-

ICDM'05
The Fifth IEEE International Conference on Data Mining

New Orleans, Louisiana, USA
November 26-30, 2005
http://www.cacs.louisiana.edu/∼icdm05/
Submission Deadline: June 15, 2005

The 2005 IEEE International Conference on Data Mining (IEEE ICDM '05) provides a premier forum for the dissemination of innovative, practical development experiences as well as original research results in data mining, spanning applications, algorithms, software and systems. The conference draws researchers and application developers from a wide range of data mining related areas such as statistics, machine learning, pattern recognition, databases and data warehousing, data visualization, knowledge-based systems and high performance computing. By promoting high quality and novel research findings, and innovative solutions to challenging data mining problems, the conference seeks to continuously advance the state of the art in data mining. As an important part of the conference, the workshops program will focus on new research challenges and initiatives, and the tutorials program will cover emerging data mining technologies and the latest developments in data mining. technologies and the state-of-the-art of data mining developments.

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.

————————-

AWIC'05
The Third Atlantic Web Intelligence Conference

Lodz, Poland
June 6-9, 2005
http://wic.ics.p.lodz.pl/awic/
Submission Deadline: December 20, 2004

The 3rd Atlantic Web Intelligence Conference (Madrid - 2003, Cancun - 2004) brings together scientists, engineers, computer users, and students to exchange a nd share their experiences, new ideas, and research results about all aspects (theory, applications and tools) of intelligent methods applied to Web based systems, and to discuss the practical challenges encountered and the solutions adopted.

The conference will cover a broad set of intelligent methods, with particular emphasis on soft computing. Methods such as (but not restricted to):

Neural Networks, Fuzzy Logic, Multivalued Logic, Rough Sets, Ontologies, Evolutionary Programming, Intelligent CBR, Genetic Algorithms, Semantic Networks, Intelligent Agents, Reinforcement Learning, Knowledge Management, etc.

must be related to applications on the Web like:

Web Design, Information Retrieval, Electronic Commerce, Conversational Systems, Recommender Systems, Browsing and Exploration, Adaptive Web, User Profiling/Clustering, E-mail/SMS filtering, Negotiation Systems, Security, Privacy, and Trust, Web-log Mining, etc.

Related Conferences

AAMAS'05
The Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems
Utrecht, The Netherlands
July 25-29, 2005
http://www.aamas2005.nl/

AAMAS-05 encourages the submission of theoretical, experimental, methodological, and applications papers. Theory papers should make clear the significance and relevance of their results to the AAMAS community. Similarly, applied papers should make clear both their scientific and technical contributions, and are expected to demonstrate a thorough evaluation of their strengths and weaknesses in practice. Papers that address isolated agent capabilities (for example, planning or learning) are discouraged unless they are placed in the overall context of autonomous agent architectures or multiagent system organization and performance. A thorough evaluation is considered an essential component of any submission. Authors are also requested to make clear the implications of any theoretical and empirical results, as well as how their work relates to the state of the art in autonomous agents and multi-agent systems research as evidenced in, for example, previous AAMAS conferences. All submissions will be rigorously peer reviewed and evaluated on the basis of the quality of their technical contribution, originality, soundness, significance, presentation, understanding of the state of the art, and overall quality.

In addition to conventional conference papers, AAMAS-05 also welcomes the submission of papers that focus on implemented systems, software, or robot prototypes. These papers require a demonstration of the prototype at the conference and should include a detailed project or system description specifying the hardware and software features and requirements.

———————————-

IJCAI'05
The Nineteenth International Joint Conferenceon on Artificial Intelligence
Edinburgh, Scotland
July 30 - August 5, 2005
http://ijcai05.csd.abdn.ac.uk/
Submission Deadline: January 21, 2005

The IJCAI-05 Program Committee invites submissions of full technical papers for IJCAI-05, to be held in Edinburgh, Scotland, 30 July - 5 August, 2005. Submissions are invited on substantial, original, and previously unpublished research on all aspects of artificial intelligence.

———————————-

EEE'05
The 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service
Hong Kong, China
March 29 - April 1, 2005
http://www.comp.hkbu.edu.hk/∼eee05/

The 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE-05) 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 focus of this year is two-fold: 1) emerging enabling technologies to facilitate next generation e-transformation, and 2) their application and deployment experience in different e-themes, including e-Business, e-Learning, e-Government, e-Finance, etc. The conference solicits research papers as well as proposals for tutorials and workshops on related e-topics. The conference is organized around topics (including, but not limited to):

Emerging E-Technology Track
- Web/Grid Service Oriented Computing
- Ontology, Semantic Web and Ontology
- WI, Agents and Personalization
- Pervasive, Mobile and P2P Computing
- Context-Aware, Autonomous Computing
- Trust and Reputation for e/m-Services
- Payment Technologies for e/m-Services
- Middleware for e/m-Services

E-Commerce, E-Service and Experience Track
- Business Processes Interoperation
- Supply Chain Integration & Management
- Business Intelligence and e-CRM
- Electronic Contracting and Commitment
- Computational Markets and Economy
- Quality Metrics for Web Content/Services
- Applications to other e-themes

———————————-

SDM'05
The 2005 SIAM International Conference on Data Mining
Newport Beach, CA, USA
April 21-23, 2004
http://www.siam.org/meetings/sdm05/

Advances in information technology and data collection methods have led to the availability of large data sets in commercial enterprises and in a wide variety of scientific and engineering disciplines. We have an unprecedented opportunity to analyze this data and extract intelligent and useful information from it. The field of data mining draws upon extensive work in areas such as statistics, machine learning, pattern recognition, databases, and high performance computing to discover interesting and previously unknown information in data.

This conference will provide a forum for the presentation of recent results in data mining, including applications, algorithms, software, and systems. There will be peer reviewed, contributed papers as well as invited talks and tutorials. Best paper awards will be given in different categories. Proceedings of the conference will be available both online at the SIAM Web site and in hard copy form. In addition, several workshops on topics of current interest will be held on the final day of the conference, including workshops on 1) data mining in sensor networks, 2) link analysis, counterterrorism and security, 3) high performance and distributed mining 4) feature selection for data mining - interfacing machine learning and statistics 5) clustering high dimensional data and its applications, and 6) mining scientific and engineering datasets

———————————-

ISWC2004
The Fourth International Semantic Web Conference
Galway, Ireland
6-10 November, 2005
http://iswc2005.semanticweb.org/

ISWC is a major international forum at which research on all aspects of the Semantic Web is presented. ISWC2005 follows the 1st International Semantic Web Conference (ISWC2002 which was held in Sardinia, Italy, 9-12 June 2002), 2nd International Semantic Web Conference (ISWC2003 which was held in Florida, USA, 20 - 23 October 2003) and 3rd International Semantic Web Conference (ISWC2004 which was held in Hiroshima, Japan, 7-11 November 2004).

Career Opportunities

Associate / Assistant Professors / Lecturer/ Instructors in Computer Science at Hong Kong Baptist University

http://www.comp.hkbu.edu.hk/en/news/?year=2004&id=091104
Apply by March 31, 2005

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