WSS 2003

WI/IAT 2003 Workshop on Applications, Products and Services of Web-based Support Systems

Halifax, Canada, October 13, 2003

Edited by

J. T. Yao
University of Regina

P. Lingras
Saint Mary's University

ISBN 0-9734039-1-8

WSS 2003

WI/IAT 2003 Workshop on
Applications, Products and Services of Web-based Support Systems

Halifax, Canada, October 13, 2003

Edited by

J. T. Yao
University of Regina

P. Lingras
Saint Mary's University

Published by
Department of Mathematics and Computing Science

Saint Mary's University
Halifax, Nova Scotia, Canada

Technical Report Number: 2003-03 October, 2003
ISBN: 0-9734039-1-8

Program Chairs

Dr. JingTao Yao
University of Regina
Canada

Dr. Pawan Lingras
Saint Mary's University
Canada

Program Committee Members

Additional Reviewers

Table of Contents

Preface

The workshop on Applications, Products and Services of Web-based Support Systems is held on October 13, 2003 at Halifax, Canada. It aims to a particular field of Web Intelligence by providing a forum for the discussion and exchange of ideas and information by researchers, students, and professionals on the issues and challenges brought on by the Web technology for various support systems. One of our goals is to find out how applications and adaptations of existing methodologies on the Web platform benefit our decision-makings and various activities.

We are quite pleased with the quality and diversity of the accepted papers. Although this is the first workshop on this topic, we can see the acceptance of the theme by the public through the submissions. The first CFP was sent out on July 12, 2003. Within less than two months time, we received more than 40 submissions. The distribution of authors spans a large geographic area. Here are the names of more than a dozens countries and regions: Australia, Brazil, Canada, China, Czech Republic, Denmark, France, Germany, Hong Kong, Italy, Lithuania, Netherlands, Taiwan, Thailand, UK, and USA.

Finally, we would like to express our thanks to program committee members for their efforts in reviewing submissions in a short time. Many people helped make the workshop possible in one way or another. Most importantly, we would like to thank all the contributing authors for their submissions and participation in the workshop.

Enjoy the workshop, and have fun in Halifax.

Workshop chairs
JingTao Yao and Pawan Lingras

Web-based Support Systems

J.T. Yao Y. Y. Yao
Department of Computer Science, University of Regina
Regina, S4S 0A2, Canada
E-mail: {jtyao, yyao}@cs.uregina.ca

Abstract

Web-based support systems (WSS) concern multidisciplinary investigations which combine computer technologies and domain specific studies. Domain specific studies focus on the investigation of activities in a particular domain. Computer technologies are used to build systems that support these activities. Fundamental issues of WSS are examined, a framework of WSS is presented, and research on WSS is discussed. It is expected that WSS will be accepted as a new research area.

1. Introduction

The advances in computer technologies have affected everyone in the use of computerized support in various activities. Traditional decision support systems focus on computerized support for making decision with respect to managerial problems [11]. There is an emerging and fast growing interest in computerized support systems in many other domains such as information retrieval support systems [12, 14], research support systems [14], teaching and learning support systems, computerized medical support systems [9], knowledge management support systems[1, 5], and many more. The recent development of the Web generates further momentum to the design and implementation of support systems.

This paper investigates the emerging field of computerized support systems in general and Web-based support systems (WSS) in specific. WSS are viewed as a multidisciplinary research involving the integration of domain specific studies and other disciplines such as computer science, information systems, and the Web technology, to only name a few. There is a sufficient evidence showing a strong trend for studies of computerized support systems in addition to decision support systems. Investigations of WSS in a wide context may result in many new research topics and more effective systems.

In the rest of the paper, we focus on the following specific objectives:

- to provide a precise characterization of computerized support systems, and to identify and examine the needs, rationalities, as well as trends of such systems (Section 2.1);
- to understand, study and analyze the feasibility and advantages of transferring support systems to the Web platform (Section 2.2);
- to identify the scope of WSS (Section 2.3);
- to establish a general framework for Web-based support systems (Section 3);
- to address some basic research issues related to WSS (Section 4).

2. Issues of Web-based Support Systems

2.1. Computerized Support systems

It is a dream of every computer scientist to develop a fully automated computer system which has the same or even a higher level of intelligence as human beings. However, the technologies we mastered can only design and develop systems that have some abilities to assist, support, and aid us for various activities. In fact, one of the popular definitions of artificial intelligence (AI) is "*the study of how to make computers do things at which, at the moment, people are better*" [7]. AI is one of the important and popular research topics in computer science. The research proves that it is almost impossible to replace human intelligence with computer systems, at least within the foreseeable future. With this restriction, we have to lower the expectation of our dreams. Decision support systems (DSS), computer aided software engineering (CASE), and computer aided design (CAD) systems are some examples of such systems to fulfill more practical goals.

As a field of study, computerized support systems is an interdisciplinary research area. A particular support system with specific domain knowledge provides support to a specific field. The most popular and successful example is the decision support systems (DSS). DSS was defined as *"computer-based information systems that combine models and data in an attempt to solve nonstructured problems with extensive user involvement through a friendly user interface"* [11]. It can be viewed as a hybrid product of two domains of studies. DSS are derived from management science and computer science. The same principle applies to other types of support systems. For instance, a medical support system or a medical expert system is the product of the marriage between medical science and computer science. Research support systems are the combination of research methodology and computer science [14]. In general, a specific support system aims to support activities and operations of the specific domain.

Various support systems have been studied for a long time. Schematically, suppose $\mathcal{A}$ is a specific domain, a support system for domain $\mathcal{A}$ can be termed as an $\mathcal{A}$ support system. Following this, we used one of the most popular search engines Google [3] for our background studies. Table 1 shows the search results we obtained in August 2003. The first column 'Search Phrase' is the phrase we used for exact phrase search. The second column '# of Hits' is the number of links returned by Google with the search phrase. It can be seen that people have done numerous research on various support systems. Decision support system(s), business support system(s), negotiation support system(s) and medical support system(s) are amongst the highest returned hits. An interesting observation from Table 1 is that the majority of support systems with high hit rates are business and management oriented. Technical oriented support systems had not been paid attention by researchers. Therefore, we should investigate more on technical oriented support systems such support as for data mining, research, and learning. Further more, there are also emerging needs for moving support systems to the Web platform.

2.2. Support systems in the Web age

The Web provides a new medium for storing, presenting, gathering, sharing, processing and using information. The impacts of the Web can be felt in almost all aspects of life. We aim to study the issues and challenges brought on by the Web technology for various support systems. One of the goals is to find out how applications and adaptations of existing methodologies on the Web platform benefit our decision-makings and various activities. A list of benefits of the Web technology is given bellow.

1. The Web provides a distributed infrastructure for information processing.
2. The Web is used as a channel to discuss one of the most popular support systems, DSS [4].
3. The Web can deliver timely, secure information and tools with user friendly interface such as Internet Explorer and Netscape.
4. The Web has no time or geographic restrictions. Users can access the system at any time, any place.
5. Users can control and retrieve results remotely and instantly.

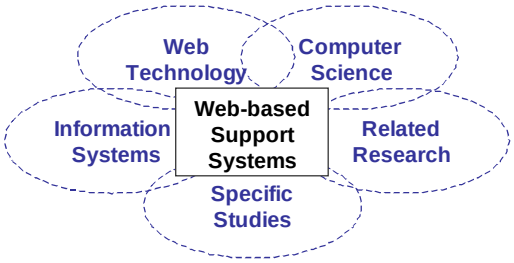


Figure 1. WSS: A multidisciplinary research

Although the advantages of applying the Web technology to support systems are obvious, the concept of Web-based support systems has not been paid enough attention by researchers. It is clear to see from the search results obtained in Table 1 that the number of hits for each type of Web-based support systems is dramatically lower than its computerized support system counterpart. For instance, the hits of the search of "Medical support system" and "Medical support systems" both reached 1,000. However, there was none when we change the phrase to "Web-based medical support system" or "Web-based medical support systems". The majority of returns from "Web-based medical support" were not related to computerized systems. Although the hits were 33, Google returned only 18 links with similar sites omitted according to its criteria. In fact, 13 out of 18 links pointed to a single research paper entitled "Intranet Health Clinic: Web-based medical support services employing XML" [8]. Web-based decision support systems [6] is one of the pioneer research areas of WSS. The returns of "Web-based decision support system(s)" were also higher than others.

2.3. Scope of Web-based support systems

WSS is a multidisciplinary research area as depicted in Figure 1. It involves many research domains. We classify the scope of WSS in four categories: WSS for specific domains, Web-based applications, techniques related to WSS,

Search Phrase	# of Hits
Decision support system	212,000
Decision support systems	332,000
Web-based decision support system	891
Web-based decision support systems	583
Web-based decision support	3,460
Business support system	4,180
Business support systems	11,400
Web-based business support system	3
Web-based business support systems	27
Web-based business support	87
Negotiation support system	1,270
Negotiation support systems	1,680
Web-based negotiation support system	96
Web-based negotiation support systems	294
Web-based negotiation support	408
Information retrieval support system	39
Information retrieval support systems	98
Web-based information retrieval support system	0
Web-based information retrieval support systems	33
Web-based information retrieval support	33
Research support system	750
Research support systems	48
Web-based research support system	2
Web-based research support systems	25
Web-based research support	33
Teaching support system	231
Teaching support systems	118
Web-based teaching support system	1
Web-based teaching support systems	2
Web-based teaching support	108
Medical support system	1,180
Medical support systems	1,010
Web-based medical support system	0
Web-based medical support systems	0
Web-based medical support	33
Knowledge management support system	433
Knowledge management support systems	90
Web-based knowledge management support system	340
Web-based knowledge management support systems	1
Web-based knowledge management support	414
Data mining support system	7
Data mining support systems	2
Web-based data mining support system	0
Web-based data mining support systems	0
Web-based data mining support	0

Table 1. Search results with Google

and design and development of WSS. Some suggested topics are listed below:

- Web-based support systems for specific domains:
  - Web-based decision support systems
  - Enterprise-wide decision support systems
  - Web-based group decision support systems
  - Web-based executive support systems
  - Web-based business support systems
  - Web-based negotiation support systems
  - Web-based medical support systems
  - Web-based research support systems
  - Web-based information retrieval support systems
  - Web-based education support systems
  - Web-based learning support systems
  - Web-based teaching support systems
- Web-based applications
  - Web-based knowledge management systems
  - Web-based groupware systems
  - Web-based financial and economic systems
  - Internet banking systems
  - Web-based multimedia systems
- Techniques related to WSS:
  - XML and data management on the Web
  - Web information management
  - Web information retrieval
  - Web data mining and farming
  - Web search engines
- Design and development of WSS:
  - Web-based systems development
  - CASE tools and software for developing Web-based applications
  - Systems analysis and design methods for Web-based applications
  - User-interface design issues for Web-based applications
  - Visualizations of Web-based systems
  - Security issues related to Web-based applications

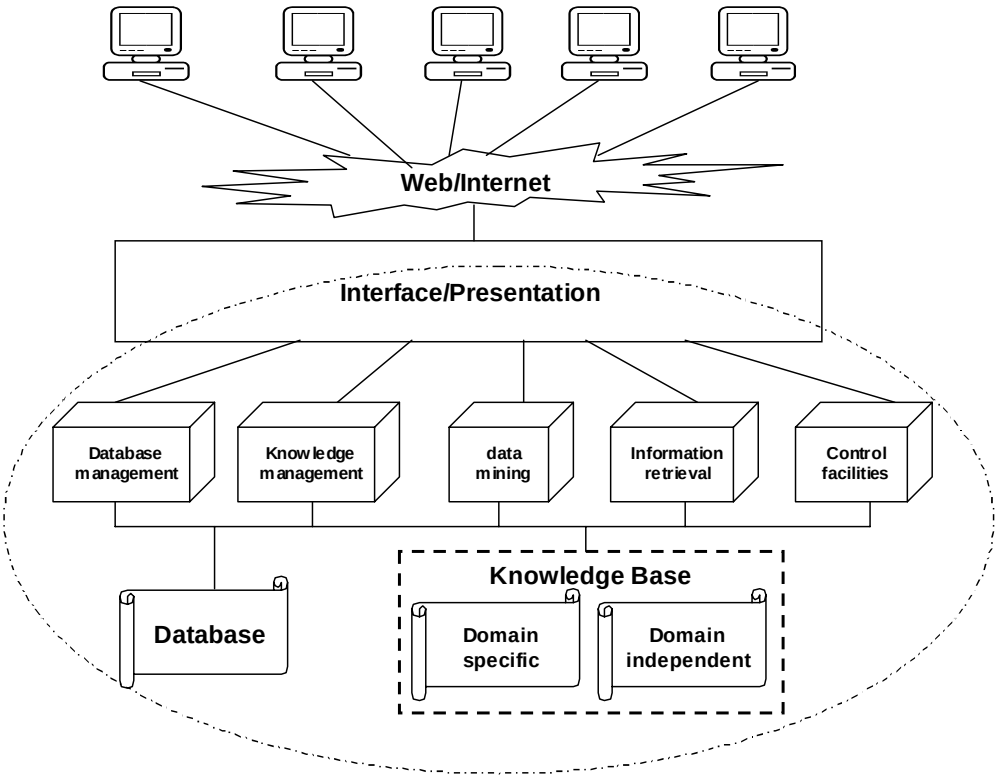


Figure 2. An Architecture of Web-based Support Systems

3. A Framework of Web-based Support Systems

Interface, functionality, and databases are some of the components which need to be considered when we design a system. We can view the architecture of WSS as a (thin) clint/server structure [2] as shown in Figure 2. The users, including decision makers and information seekers, are clients on the top layer. They access the system with browsers via the Web and the Internet. The interface that is designed on the server side will be presented on the client's side by browsers. The lower layers and components encapsulated by the oval dotted line are, in fact, very similar to conventional computerized support systems. In other words, a Web-based support system is a support system with the Web and Internet as the interface.

There are two components on the data layer. Database is a basic component in any modern system. WSS is not an exception. Another major component is the knowledge base. It stores all rules, principles and guidelines used in supporting activities. We intend to divide the knowledge base into two parts: domain specific knowledge base and domain independent knowledge base. The former is the knowledge specific to the domain. The latter involves general knowledge for all support systems.

Knowledge management, data management, information retrieval, data mining and other control facilities form the management layer. They serve as the middleware of the three-tier client/server architecture. They are the intermediaries between interface and data layers. Reasoning, inference and agent technologies will play important roles on this layer. The split of data and user results in a secure and standardized system. To take advantage of the Web technology, these processes are distributed over the Internet to form a virtual server. In fact, databases and knowledge bases on the lower tier are also distributed.

Web-based support systems can be classified into three levels. The first level is support for personal activities. An example of such support is research support for individuals [14]. Personal research activities such as search, retrieval, reading and writing are supported. The second level is the organizational support, such as research support on an institute level [10]. The top level is the network level. The collaborations between organizations or decision making by a group of people like in group decision support systems fall in this level. The group decision support room may be a vir-

tual room on the Web.

4. Research on Web-based Support Systems

The research on Web-based support systems can be classified into a few categories. The first class is the study of a specific support system and related technology as indicated in Section 2.3. There are four types of existing research, namely, WSS for specific domains, Web-based applications, techniques related to WSS and design, and development of WSS, that can be classified as WSS research.

On a more general level of research on WSS, we may include the study of WSS operations and support facilities. The study of WSS operations aims to understand the needs of supporting domains such as business logic and management concerns. The study of support facilities focuses on potential support functionalities that computer science and Web technology can provide. There are two types of operations, i.e, domain independent operations and domain specific operations. Domain independent support facilities and domain specific support facilities are two types of support facilities.

The study of operations will help us to gain a deeper understanding of WSS. Domain independent operations may include operational controls such as report generating and graphical multimedia presentation, managerial control such as negotiation and evaluation, strategic planning such as technology adoption and quality assurance. These domain specific operations may include class schedules for teaching support and images processing for medical support.

With the understanding of operations, various support facilities can be studied. They may include techniques such as data mining, information retrieval, optimization, simulation heuristics, and inference. The support facilities could also be classified into levels. For instance, a Web-based research support may provide two levels of support: managing support for management staff and activities support for individual researchers [10, 14].

5. Conclusion

The research of Web-based support systems is a natural evolution of the existing research. The first step is the extension of decision support systems to computerized support systems. With the emergence of Web technology and Web intelligence, the need to study Web-based support systems are obvious. We identify the domain and scope of Web-based support systems. A framework with the viewing angle from a client/server facility is presented. We also discuss the issues of research on WSS. It is expected that WSS, as a new identified research area, will attract more research.

References

[1] M. Ginsburg, A. Kambil, Annotate: A Web-based Knowledge Management Support System for Document Collections, *Procedeeings of HICSS-32*, 1999.

[2] J. Goldman, P. Rawles, J. Mariga, *Client/server information systems: a business-oriented approach*, John Wiley & Sons, 1999.

[3] Google: http://www.google.com

[4] ISWorld DSS research page: http://www.isworld.org/dss/index.htm.

[5] R. Otondo, J. Simon, A Model for the Study of Knowledge Management Support Systems, *Proceedings of the 6th Americas Conference for Information Systems*, Long Beach, 2000.

[6] D.J. Power, S. Kaparthi, Building Web-based decision support systems, *Studies in Informatics and Control*, **11**, 291-302, 2002.

[7] E. Rich, K. Knight, *Artificial Intelligence*, McGraw-Hill, 1991.

[8] G. Stalidis, A. Prentza, I.N. Vlachos, G. Anogianakis, S. Maglavera, D. Koutsouris, Intranet Health Clinic: Web-based medical support services employing XML, *Proceedings of the Medical Informatics Europe*, pp1112-1116, 2000.

[9] G. Stalidis, A. Prentza, I.N. Vlachos, S. Maglavera, D. Koutsouris, Medical support system for continuation of care based on XML Web technology, *International Journal of Medical Informatics*, **64**, 385-400, 2001.

[10] H. Tang, Y. Wu, J.T. Yao, G.Y. Wang, Y. Y. Yao, CUP-TRSS: a Web-based Research Support System, *Proceedings WSS'03*, 2003.

[11] E. Turban, J.E. Aronson, *Decision Support Systems and Intelligent System*, Prentice Hall, New Jersey, 2001.

[12] J.T. Yao, Y.Y. Yao, Web-based information retrieval support systems: building research tools for scientists in the new information age, *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, 2003.

[13] Y.Y. Yao, Information retrieval support systems, *Proceedings of FUZZ-IEEE'02*, 773-778, 2002

[14] Y.Y. Yao, A framework for Web-based research support systems, proceedings of COMPSAC'2003, Dallas, USA, Nov 2003 (to appear).

Framework and Implementation of A Web-based Multi-objective Decision Support System: WMODSS

Jie Lu, Guangquan Zhang and Chenggen Shi
Faculty of Information Technology, University of Technology Sydney, Australia
E-mail: jielu@it.uts.edu.au

Abstract

This paper presents a web-based general decision support system for solving multi-objective decision problems. This is referred to WMODSS. The system provides three decision-making methods in its method base. They deal with a wide range of linear multi-objective decision-making problems and different user requirements for the solution process. In particular, the WMODSS provides an intelligent guide to help users select a most suitable method from the method base by evaluating the relevant problems and user requirements. A satisfactory solution can thus be obtained in an interactive and flexible manner.
Key words: Decision support systems, Web-based systems, Multi-objective decision-making, Multi-objective decision support systems

1. Introduction

Various decision support systems (DSS) have, over the years, been successfully implemented in many organizations at different organizational levels. DSS always concern with information and how to make effective use of it in decision-making scenarios [8]. To deal with multi-objective decision-making (MODM) problems by using DSS, one or more MODM methods must be embedded into the DSS to form multi-objective decision support systems (MODSS). MODSS have been considered as a 'specific' type of systems within the broad family of DSS.

Traditionally, DSS had to be installed in a specified location, such as a decision room, supported by a specified operating system. Web technology allows organizations to make decisions in a distributed environment that supports remote data access and communication. Since the advance of web technology, which allows users fast and inexpensive access to an unprecedented amount of information provided by websites, digital libraries and other sources, web-based DSS have been proposed to extend the applications of traditional DSS to a global environment with a unified web platform. The computer facilities for utilizing DSS have been moving towards a more widespread use of the Internet with its graphical user interface, the web [14].

More recently, both E-business and E-government are increasing their demands for more online data analysis and decision support. The web platform, which is also a platform for E-business and E-government development, lends itself to widespread use and adoption of DSS in organizations [6, 2]. From the web platform, managers who have not used DSS will find the new decision support tools powerful and convenient. New managers who have not been exposed to client-server tools or other DSS tools in the 1980s and 1990s find that web-based DSS are what they needed: easy to use and available from their office, home and client locations. The organizations where decision makers are distributed in different locations can use web-based DSS to assist them in making organizational strategic decisions by way of the Internet [13].

This paper presents a web-based multi-objective decision support system, called WMODSS, and describes its framework, design and implementation.

2. Multi-objective decision support and web technology

In a MODM problem the objectives usually conflict with each other and any improvement in one objective can be achieved only at the expense of another. With this observation, decisions on optimality are not determined uniquely. A final satisficing solution must be selected from among a set of possible close to optimal solutions. Consequently, the main task in solving MODM problems is to derive a satisficing solution based on subjective judgments and preferences for alternatives [9]. A number of MODM methods have been reported to provide solutions and lead to better decision outcomes for MODM problems [16].

As a specific type of the DSS family, MODSS have special characteristics that distinguish them from other DSS. The main characteristics are that they allow analysis of multiple objectives; they use a variety of

MODM methods to compute efficient solutions; and they incorporate user input in the various phases of modeling and solving a problem. While it has been customary to consider algorithms as the focal point of decision support, emphasis is shifting to database and modeling activities [7].

Traditionally, users needed proper training to learn how to use a DSS. Because the Internet is easily accessible, web-based DSS are automatically available to large number of decision makers [10]. Web-based DSS don not require any special software on a user's computer. This means that the operating system used, or compilers available, are not important to users. Further, the web enables a convenient and graphical user interface with visualization possibilities. The main issue for web-based DSS users is that the only requirement for using the DSS is a connection to the Internet, and a web browser.

Web-based DSS have reduced technological barriers and made it easier and less costly to make decision-relevant information and model-driven DSS available to decision makers in geographically distributed organizations [1, 12]. Because of the Internet infrastructure, enterprise-wide DSS can now be implemented in geographically dispersed companies at a relatively low cost. Using web-based DSS, organizations can provide DSS capability to managers over a proprietary Intranet, to customers and suppliers over an Extranet, or to any stakeholder over the global Internet. The web has increased access to information and thus should increase the use of DSS in organizations. Several web-based DSS have been developed in the last few years. These systems mainly focus on criteria-driven decision support, except for WWW-NIMBUS. WWW-NIMBUS is a web-based multi-objective DSS, in which NIMBUS is a MODM method [10]. As some MODM methods are more suitable than others for particular users and particular decision problems, our study proposes the WMODSS which contains three typical linear MODM methods. Also, in order to help users to choose the most suitable methods for their decision problems, the WMODSS provides an intelligent guide as a front-end. These methods have different characteristics when dealing with multi-objective decision problems. Therefore, WMODSS has more advantages than existing systems as it can be used for a more wide range of decision-making problems and decision makers.

3. Framework of WMODSS

This research proposes a conceptual framework of web-based DSS for solving multi-objective decision problems using interactive modules. This is shown in Figure 1. Key issues involved in this framework include system integrity, communication, shared information space, method management and data management. The framework integrates a database, a method base and intelligent user interface. In the framework, data used for decision making is mainly obtained from internal database and user input, but can also be obtained from web resources. All MODM methods in the method base can be accessed and executed locally or remotely by users. The method base has an appropriate linkage with the database. Users can be distributed in different locations using the system through web browsers.

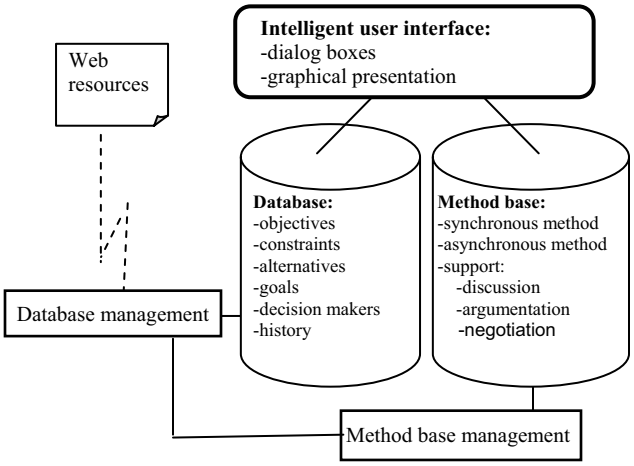


Figure 1. A conceptual framework of WMODSS

4. Implementation of WMODSS

4.1 Web sever

A web server manages all the web pages of the WMODSS, traces user information and provides simultaneously services to multiple users through sessions, applications and coking facilities. All the web pages displayed dynamically to users are created on the fly by the web server. Using the server side application program, the web server can manage and implement client tasks. For example, when a user wants to access WMODSS using existing problem data, the process is described as follows. (1) The user submits a task to the web server. (2) The web sever asks the database sever to confirm the user, and fetches a stored MODM decision problem. (3) The web then sends the decision problems to the user. (4) The system guide will help the user select a suitable MODM method. (5) After a decision problem is chosen (or given) and a MODM method is selected, the web server gives this information to the method base management component. (6) When the web server gets a solution by running the MODM method, it will let the database sever store the solution together with the user's

information. (7) Finally, the web server constructs a solution page displaying the solution to the user.

4.2 Method base

It has been found that there are many differences between different MODM methods in their applications. Some MODM methods are easy to use but require pre-data, such as goals of objective functions. Some methods encourage decision makers to explore other alternative solutions in an interactive fashion with the aim of finding a satisficing solution but users have to have such ability to do relaxation. The ultimate success of MODSS lies in its ability to help decision makers produce the most satisfactory solution through providing pre-data or directly interacting with analytical models. This is why the WMODSS builds a MODM method base which contains different kinds of methods. Three popular MODM methods are selected in the method base of WMODSS: the Efficient Solution via Goal Programming (ESGP) method [3], Linear Goal Programming (LGP) method [4], and STEUER method [15]. These methods are developed as independent executables to facilitate the flexibility required of the system. Figure 2 shows a set of alternative solutions for a product planning problem using STEUER method. Figure 3 shows the algorithm of STEUER method.

4.3 Database

The WMODSS database is designed to share problem data and solution data among users. Relational data mode technology is used in the database design. There are three main entities in the WMODSS database: user, problem and solution data.

4.4 Method selection guide

The system builds a method selection guide for users selecting a suitable MODM method from the method base. Based on research results of Poh [11] and Lu & Quaddus [9], the various characteristics of MODM methods are classified into four classes: user-related, method-related, problem-related and solution-related. Each class has a group of special characteristics. For example, the user-related characteristics concern user preference for selecting a method. Such characteristics include users' desire to interact with the system, and users' ability to provide pre-data for a specific MODM method. For example, LGP needs pre-data of weights, goals and priorities for its objectives, but STEUER and ESGP do not.

When a user chooses the guide, a set of questions are shown firstly. The system uses a response-characteristic-method match algorithm to get a recommendation of a suitable method to the user based on their responses. Figure 4 shows the question page of the guide.

4.5 Web page connection

Web pages of the WMODSS are designed and implemented to support users accessing the system. Figure 5 shows possible web pages involved when using STEUER method and their connections. Basically there are two ways to get a decision problem. One is to set a new problem by inputting its objectives and constraints. Another is to use an existing decision problem which has been set before and solutions may have been obtained by other users. Figure 5 shows two groups of pages to handle the two ways respectively.

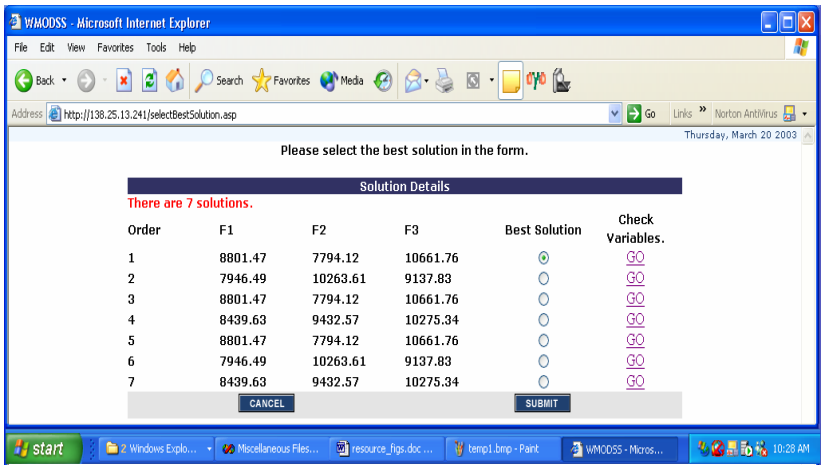


Figure 2. A number of alternative solutions

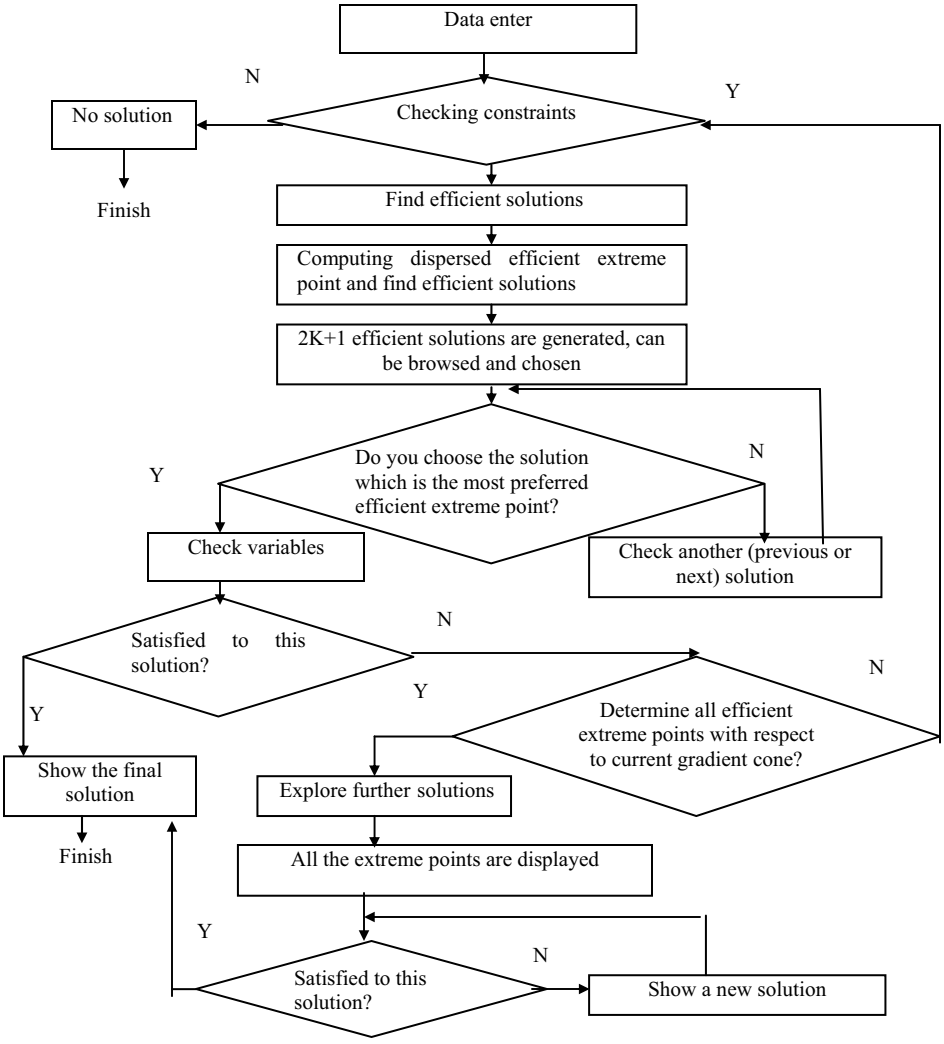


Figure 3. Algorithm and execution of STEUER method

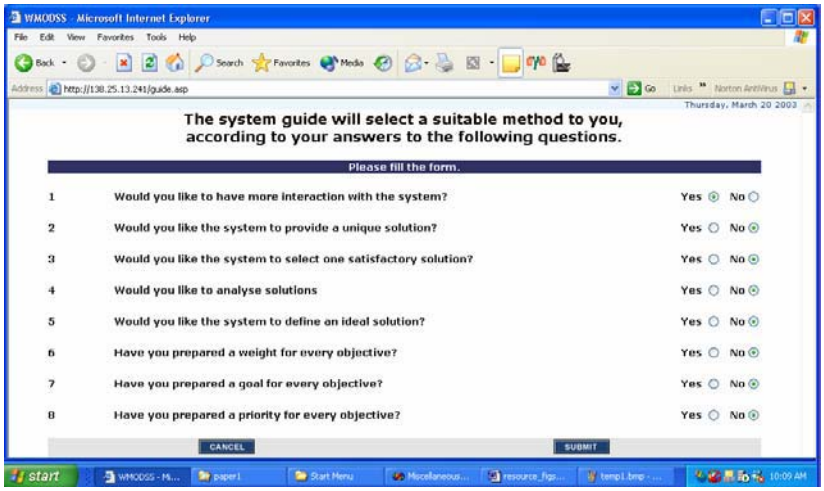


Figure 4. Question page of the guide in WMODSS

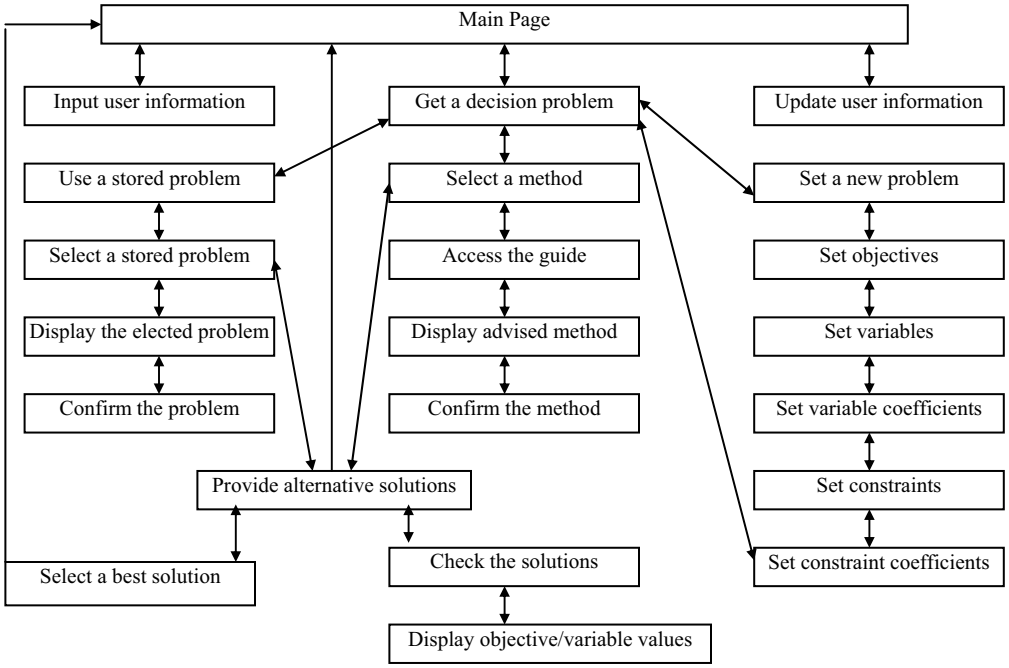


Figure 5. Page connections when using STEUER method

5. Future study

The WMODSS is interactive, flexible and easy to use for various linear multi-objective decision problems. It is expected to be applied to practical multi-objective decision problems such as product planning, resource management, research project funding, and the determination of optimal price changes. We hope to get feedback from our users and to make the system widely available. We are planning to enhance the systems to support group decision-making involving multi-objective decision problems.

References

[1] S. Ba, R. Kalakota and A.B. Whinston, "Using client--broker--server architecture for Intranet decision support", *Decision Support Systems,* Vol.19, No. 3 (1997), pp. 171-192.

[2] H.K. Bhargava, R. Krishnan and R. Muller, "Decision support on demanded: emerging electronic markets for decision technologies", *Decision Support Systems,* Vol. 19, No. 3(1997), pp.193-214.

[3] C.L. Hwang and A.S.M. Masud, *Multiple Objective Decision-making - Methods and Applications: A State of the Art Survey,* Springer, New York, 1979.

[4] J. P. Ignizio, *Goal Programming and Extensions,* Massachusetts, 1976.

[5] J.P. Ignizio, "The determination of a subset of efficient solutions via goal programming", *Computer and Operations Research,* Vol. 8 (1981), pp. 9-16.

[6] G.E. Kersten and S.J. Noronha, "WWW-based negotiation support: design, implementation and use", *Decision Support Systems,* Vol. 25, No. 2 (1999), pp. 135-154.

[7] P. Korhonen and J. Wallenius, "A Multiple Objective Linear Programming Decision Support System", *Decision Support Systems,* Vol. 6, No. 4 (1990), pp. 243-251.

[8] R.L. Lang and , A.S. Whinston, "A design of a DSS intermediary for electronic markets", *Decision Support Systems,* Vol. 25, No. 3 (1999), pp. 181-197.

[9] J. Lu and M.A. Quaddus, "Integrating knowledge based guidance system with multi-objective decision-making", *The New Zealand Journal of Applied Computer and Information Technology*, Vol. 5, No.1 (2001), pp. 53-59.

[10] K. Miettinen and M.M. Mäkelä, "Interactive multi-objective optimization system WWW-NIMBUS on the Internet", *Computers and Operations Research,* Vol. 27, No. 7-8 (2000), pp. 709-723.

[11] K. Poh, "Knowledge-based guidance system for multi-attribute decision making", *Artificial Intelligence in Engineering,* Vol. 12, No. 3 (1998), pp. 315-326

[12] D.J. Power, "Decision support systems glossary. DSS resources, World Wide Web", http://www.DSSResources.COM/glossary/ 1999.

[13] J.P. Shim, M. Warkentin, J.F. Courtney, D.J. Power, R. Sharda and C. Carlsson, "Past, present, and future of decision support technology", *Decision Support Systems,* Vol. 33, No. 2 (2002), pp.111-126.

[14] S. Sridhar, "Decision support using the Intranet", *Decision Support Systems,* Vol. 23, No.1 (1998), pp. 19-28.

[15] R.E. Steuer, "An interactive multiple objective linear programming procedure", *TIMS Studies in the Management Sciences,* Vol. 6 (1977), pp. 225-239.

[16] R.E. Steuer, L.R. Gardiner and J. Gray, "A bibliographic survey of the activities and international nature of multiple criteria decision-making", *Journal of Multi-criteria Decision Analysis,* No. 5 (1996), pp.195-217.

Web-Based Decision Support for Software Release Planning

Jingzhou Li
University of Calgary
2500 University Dr NW
Calgary, AB, Canada, T2N 1N4
jingli@ucalgary.ca

Guenther Ruhe
University of Calgary
2500 University Dr NW
Calgary, AB, Canada, T2N 1N4
ruhe@ucalgary.ca

Abstract

Web technology and Web Services represent a great opportunity for improving knowledge and experience exchange. In this paper, the focus is on intelligent decision support for software release planning. We first characterize the problem of software release planning, and then review Web technology and Web-based Decision Support Systems. By deriving some major requirements on Web-based decision support for release planning, we then put forward a suggestion for an architectural design. We discuss the first steps of its realization and future directions of its real-world application.

1. Motivation

Software development is a multi-disciplinary, knowledge-intensive, and human-centric process in which decisions about business, technology, market, software quality and process, distribution of resources, and so on, must be effectively and efficiently made in order to achieve the software product goals and eventually the business goals. Software Engineering Decision Support (SEDS) [1] considers the software planning, development and evolution process as a continuous problem solving and decision making activity. The expected result is a better understanding, management and control of software process, products, resources, tools, and technologies. SEDS will employ possible theories and technologies to fulfill its goals.

Web technology is now applied in various applications as the Internet provides oceans of information and knowledge, and more and more effective information publishing and retrieval techniques. In addition to information, there are services available on the Web, i.e. Web services [2][3]. With the availability of both information and services, Web technology will have a more profound impact on our society.

Web-based Decision Support Systems (DSS)[4][5] try to take the advantage of the Web technology to facilitate the support for decision-making. Some pilot development and application of Web-based DSS has been proved to be effective in many application domains [6].

In this paper, we will use the body of SEDS as a basis to explore the role of Web technology as a support mechanism for software engineering decision-making. Particularly, we will take the software release planning process as an example to illustrate this idea. We first introduce software release planning in Section 2, then the new development of Web technology in Section 3. After deriving some major demands of software release planning on Web technology in Section 4, we put forward an architectural design for Web-based decision support system for software release planning. Finally the conclusions and future work are presented.

2. Software release planning

Incremental software development replaces monolithic-type development by offering a series of increments with additive functionality. The process is a central part of planning in incremental software development. Typically each increment is a complete system that is of value to the client. This means that each new increment can be evaluated by the client. The results feed back to the developers, who then take that information into account when implementing subsequent phases. This feedback may introduce changes to requirements or new requirements, priorities, and constraints.

Software release planning determines which customer gets what features and quality at what point in time. It is a generalization of the “requirements triage” [21] defined as the process of determining which requirements a product should satisfy given the time and resources available. Simultaneously, it is one of the most brilliant examples of decision-making as part of requirements engineering activities. Without good release planning ‘critical’ features are jammed into the release late in the cycle

without removing features or adjusting dates [22]. This might result in unsatisfied customers, time and budget overruns, and a loss in market share.

Incremental development has many advantages over the traditional waterfall approach. First, prioritization of requirements ensures that the most important requirements are delivered first. This implies that benefits of the new system are realized earlier. Consequently, less important requirements are left until later and so, if the schedule or budget is not sufficient, the least important requirements are the ones more likely to be omitted. Second, customers receive part of the system early on and so are more likely to support the system and to provide feedback on it. Thirdly, the schedule/cost for each delivery stage is easier to estimate due to smaller system size. Fourth, user feedback can be obtained at each stage and plans adjusted accordingly. Fifth and most importantly, an incremental approach is sensitive to changes or additions to requirements.

The process of requirements engineering of software systems is a complex problem solving activity involving many stakeholders and many decisions. As a key activity in requirements engineering, software release planning is a complex, human-centric process in which intensive knowledge and intelligent support are heavily needed. Release planning for incremental software development assigns requirements to releases such that all technical, resource and budget constraints are met. Achieving an optimal balance of conflicting stakeholder opinions requires a prioritization of these opinions. For example, it has to be decided what sort of requirements will go into products, which releases the requirements are put into, and when they will be implemented. [7][8][9].

To summarize main characteristics of software release planning:

- **Requirements are not well specified and understood**: There is usually no formal way to describe the requirements. Non-standard format of requirement specification often leads to incomplete descriptions and makes it harder for stakeholders to properly understand and evaluate the requirements.
- **Stakeholder involvement**: Stakeholder examples are user (novice, advanced, expert), shareholder, project manager, and developer. In most cases, stakeholders are not sufficiently involved into the planning process. This is especially true for the final users of the system. Often, stakeholders are unsure why certain plans were suggested. In the case of conflicting priorities, knowing the details of compromises and why they were taken would be useful. All these issues add to the complexity of the problem at hand and if not handled properly, they create a huge possibility for project failures. On the other hand, how the stakeholders can easily participate the planning process is also a big problem. It is impossibe to have all the stakeholders gathered together to have a discussion. Wired or wireless network support is necessary for stakeholders to be involved easily and frequently in the release planning process.
- **Change of requirements and other problem parameters**: Requirements always change as the project progresses. If a large number of requirements increase the complexity of the project, their dynamic nature can pose another challenge. Other parameters such as the number of stakeholders, and their priorities, etc., also change with time - adding to the overall complexity.
- **Size and complexity of the problem**: Size and complexity are major problems for project managers when choosing release plans - some projects may have hundreds or even thousands of requirements. The size and complexity of the problem, and the tendency for not involving all of the contributing factors, makes the problem prohibitively difficult to solve by individual judgment or trial and error type methods.
- **Uncertainty of data**: Meaningful data for release planning are hard to gather and/or uncertain. Specifically, estimates of the available effort, dependencies of requirements, and definition of preferences from the perspective of involved stakeholders are difficult to gauge.
- **Availability of data**: Different types of information is necessary for actually conducting release planning. Some of the required data are available from other information sources within or external to the organization. Ideally, release planning is incorporated into existing Enterprise Resource Planning or other organizational information systems.
- **Constraints**: A project manager has to consider various constraints while allocating the requirements to various releases. Most frequently, these constraints are related to resources, schedule, budget or effort. Some of the constraints are hard constraints, others are soft ones.
- **Unclear objectives**: “Good” release plans are hard to define at the beginning. There are competing objectives such as cost and benefit, time and quality, and it is unclear which target level should be achieved.

- **Efficiency and effectiveness of release planning**: Release plans have to be updated frequently due to changing project and organizational parameters. Ad hoc methods help determine solutions but are far behind objective demands.
- **Tool support**: Currently, only general-purpose tools for requirements management are available. None of them focuses on the characteristics of release planning.

Some tasks to be done and questions to be answered for software release planning:

- **Ensure the release plans satisfy the precedence constraints**: technical (e.g. software quality especially reliability and maintainability, system architecture), logical (e.g. business process, functional relations, market policy and competition), and resource.
- **Determine the requirements priorities**: by different groups of stakeholder such as users, developers, managers, and shareholders, depending on their different opinions of requirements.
- **Effort estimation**: when new or additive requirements added, explore the ripple effects: change of data structure, system architecture, etc., which may affect the effort.
- **Resources allocation**: person, hardware, software, COTS, and technology when making release plan.

Some of these characteristics and tasks have been partially addressed in some literature. In [7], a method based on genetic algorithm called EVELOVE is used to generate the releases incrementally with the inputs of effort, stakeholder priorities, precedence and coupling. However, determining the value of these inputs still requires a great deal of information and effort. In [8], release planning is discussed from a perspective of market driven requirements engineering processes. Requirements dependency is described as a crucial task of release planning. Unfortunately, no practical methods were proposed. Paper [9] mainly discussed the requirements interdependencies in release planning, without considering other factors.

From the main characteristics and the above preliminary literature review, we conclude the relevance of support for the procurement of information, knowledge, and services (such as remote collaboration and negotiation over the Web either wired or wireless) are also crucial in decision-making support for release planning. Web technologies offer a great opportunity to achieve these goals.

3. Web Technology

Web (or Internet/Intranet/Extranet) technology with its rapid development has been providing great benefit for the whole world. Known for its ability to organize and traverse oceans of information as a "web", Web technology now has a new member—Web services which are a new breed of Web application. Web services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes.

Now that we have both information and services available on the Web, it really becomes true that "network is computer." Therefore, the concept of Internet Operating System (IOS) or Grid [10] has emerged as a way to generalize the view of the Web. Taking IOS as a platform, we are now able to assembly an application from the Web using its contents (or information) and services. The contents on the Web are scattered across the Web sites over the world. They are multi-sources, multi-forms, and different in management mechanisms, to name a few. To use the information on the Web effectively, the Virtual Database (VDB) technology [11] has come into being. By combining the Semantic Web technology and Web services [2][12], discovery of Web services can be performed at a high level in accordance with business requirements [13].

Compared to traditional network applications and the early-age applications of the Internet, the Web-based applications will benefit from the following aspects which are also the challenges the new Web technology faces:

- Grid computing and peer-to-peer systems applications
- Distributed and replicated file system management
- Content caching, replication, and distribution; content delivery networks
- Secure and reliable messaging based on HTTP, SOAP, and XML and also the security and confidentiality of Web-based system
- Directories of distributed and replicated content and services
- Ubiquitous user interfaces aggregating Web applications
- Management of distributed processes and orchestration of services
- Monitoring and management of applications based on Web contents and services
- The orchestration of services and their related data

Because of DSS's characteristics, Web-based DSS [14][15] [16] seems promising in that Web technology provides much support that the traditional technologies could not, e.g. the availability of both internal and external information, and simple yet powerful user interface. In [17], guidance for building Web-based DSS is provided.

Although there are web services already available, their practical use still does not exploit the potential benefit.

4. Demands of Web-based decision support for software release planning

There is some obvious evidence about the potential benefits of web-based DSS, but what are the demands in more detail, especially when focusing on web-based DSS for software release planning?

- **(R1) Different decision support functionality:** inclusion of data driven, document driven, communication driven, model driven, and knowledge driven; or the combination of the above functionality; other intelligent support components, such as reasoning and analysis methods, interface component, and explanation component.

- **(R2) Interaction** among and **integration** of various DSS components to allow information exchange and sharing in order to fulfill their functionality individually and as a whole.

- **(R3) Multi-sources and multi-levels abstractness** of information and knowledge: internal/external, domain-dependent/domain-independent, structured/unstructured, business/technical, for decision-makers and decision support components. For example, decision-makers may ask how a result data comes and where it comes from for explanation.

- **(R4) Negotiation among stakeholders**: One example in software release planning is that stakeholders need to negotiate the priority of requirements to be released according to their opinions of the requirements.

- **(R5) Collaboration and coordination** among stakeholders.

- **(R6) Integration** with existing information systems and resources. The decision support process will share data with existing information systems and some decision support functions may need to interact with some components of existing information systems.

- **(R7) Friendly user interface**: users should be able to access all the information on the web from as many devices as possible at any time, for example, wireless equipment.

In dependence of the problem, different aspects of these requirements may become more important than others.

5. Web-based decision support for release planning

From the above discussion we know that the Web essentially offers to the user services and contents through a unified interface tool – the web browser. This is illustrated by Figure 1.

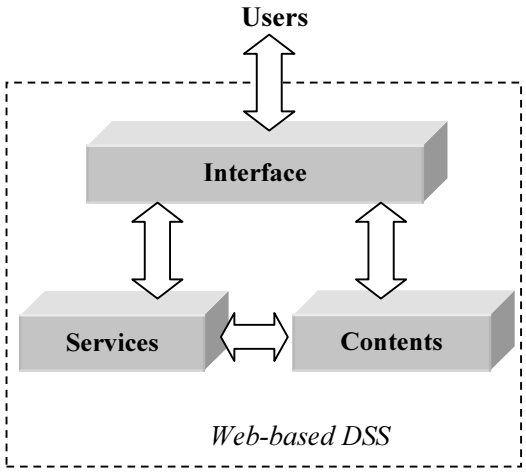


Figure 1. Conceptual Components of Web-based DSS

Here the services may be functions of local and legacy information systems, web-based or not, and Web services on both Intranet and Extranet. The functionality of the whole decision support system is thus divided into the aforementioned services. What are the advantages of using the concept of providing services?

- Services can be (self-) described, published and discovered on the web so that they can be invoked as needed;

- Services may be implemented and invoked platform-independently;

- Services can be published by providers, discovered and invoked by consumers. The two sides communicate with each other using protocols, such

as HTTP and SOAP for web services: thus the services are loosely coupled by the Web, which increases the independency of system components;

- Systems with service architecture that are loosely coupled by the Web can be easily integrated;
- With the rapid development of Web service technology, there will be an increasing variety of services available on the Web, which makes the development of new applications increasingly easier than before.

The contents components in Figure 1 may be any kind of data or knowledge existing on the Web, both Intranet and Extranet, ranging from structured data stored in databases to text documents and multimedia files. So far, the most distinctive advantage of Web is that it may contain multi-sources, multi-forms, and distributed information.

Combining the services and contents on the Web, or consuming the Web contents by the services on the Web, Web-based applications will be easily constructed, and easily operated by users through the unified Web browser interface.

Corresponding to the characteristics of the release planning problem and the demands of Web-based decision support for release planning, the following aspects of decision support for release planning are further discussed based on the notion illustrated by Figure 1. The requirements in Section 4, R1 to R7, covered by the aspects are indicated in the parenthesis.

- **DSS components (R1, R2)**: can be organized and implemented as services. For example, if general models and analysis methods for release planning in a model centered component are implemented as services and probably provided by some producers on the web, then decision alternatives can be obtained from different services, evaluated, and chosen. The construction of this component becomes relatively easy. Take EVOLVE [7] for release planning as another example. The method may be realized wholly or partially as Web services and put on the Web so that it can be implemented and invoked easily. Meanwhile, the interaction between components becomes simple since the services are loosely coupled by the Web. So are similar situations for the other components.
- **Web contents (R3)**: the problem of multi-resources and multi-forms of data has been tackled by Web contents that encompass any kind of data, documents, or even knowledge [18] that is distributed on the Web.
- **Collaboration/negotiation (R4, R5)**: over the Web [19], easy communication with both wired, such as Netmeeting, and wireless equipment, such as Smartphone and PDA, especially when using Web services, so that stakeholders may communicate and negotiate over the Web. Likewise, the developers may collaborate over the Web.
- **System integration (R6)**: decision support system can be easily integrated with the existing information systems under the service architecture: information sharing, document formalizing/editing/searching/management based on XML technology.
- **User interface (R7)**: Easy-to-use and easy access from the Web browser on thin clients. Wired or wireless devices can also be used from anywhere at anytime.

We propose a more detailed composition of Web-based decision support components as shown in Figure 2.

There are vertical components in Figure 2 (a), which are specific functions, and horizontal components in Figure 2 (b), which are commonly used by all the vertical components.

Horizontal components include, for example, negotiation, collaboration/coordination support over the Web as a common means for all vertical components. The explanation component is used for tracing back the solutions the vertical components provided to help the decision makers to understand the problem and problem-solving process.

Vertical components include, for example, COTS DSS [20] for COTS-based software development; communication-driven support mainly for negotiation, collaboration, and coordination among stakeholders; document-driven support for managing a variety of e-document formats, keeping track of textually represented knowledge for decision-making; model-driven support for modeling and simulation; knowledge-driven support for management of “expertise”, “experience”, and “skill” for specialized domain or problems. The functionality of these components is normally implemented and used separately under traditional DSS architecture. However they can be integrated under the Web-based and service centered architecture.

The contents, on the other hand, can be put under the management of Virtual DB, Web knowledge management,

or simply left as normal Web contents for browsing using the Web browser.

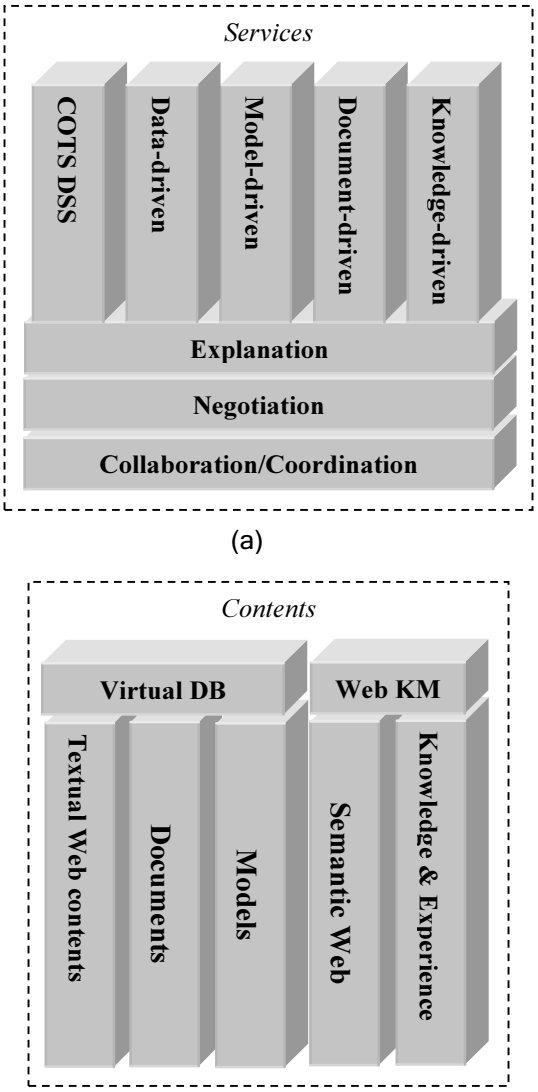


Figure 2. Detailed components of Services and Contents

Putting all the considerations discussed above together, we propose an architectural design for Web-based decision support for software release planning, as illustrated by Figure 3.

6. Conclusions and future work

In this paper, we have presented an architectural design for web-based decision support for release planning based on the characteristics of release planning and the demands on its Web-based decision support Although it has not been implemented and applied in the real world, it provides an insight view on the possibility and benefit of employing Web-based technology for software release planning. In fact, the architectural design can also be used to support other kinds of release planning such as project or product planning.

The Web-based and service centered architecture has some advantages over the traditional ones:

- Multi-sources and multi-formats of information: This is the most advantageous aspect for decision support where a great amount and variety of information is needed. For XML-based Web contents, it will be easier to retrieve both information and knowledge.

- Loosely coupled service architecture: Services can be provided, discovered, and invoked over the Web. They are loosely coupled by the Web via standard languages, such as XML, WSDL, OWL, and protocols, such as HTTP, or SOAP [2][3]. This makes it easier for interaction between components, and the integration with the legacy information systems.

- Easy collaboration and involvement over the Web: Stakeholders can be easily brought together on the Web for collaboration and negotiation purpose, either synchronously, e.g. through Netmeeting, or asynchronously, e.g. e-mail.

- Easy access through Web interface: Users can easily access the Web contents through the Web browser, especially non-computer professionals who have been accustomed and always willing to use the simple yet powerful Web browser. On the other hand, users may access the Web at anytime from anywhere through wired or wireless, simple or complex devices.

Our future work will focus on the following aspects:

- Develop a prototype to evaluate the proposed notions and architecture in cooperation with industry in order to make it more practical.

- Find ways to facilitate the integration of legacy information systems with the Web.

- Explore the application of Web services in order to get the first-hand experience of creating and using services on the Web.

- Develop a high-level description of Web services and more general concept of services in order to map the high-level business

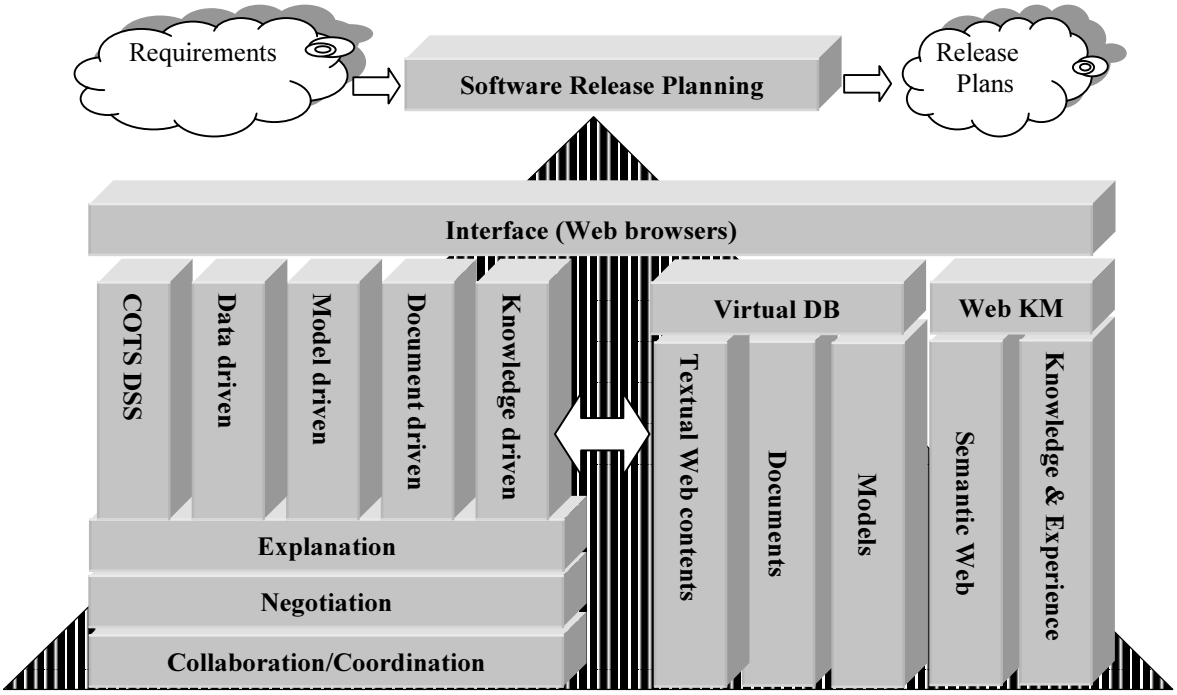


Figure 3. Architectural design for Web-based DSS for release planning

requirements to the low-level implementation of services such as the Web services at present.

- Web-based knowledge management: knowledge representation, discovery, and storage on the Web.
- Put more semantics into both the contents and services on the Web for "meaningful" discovery using semantic Web and intelligent Web technologies.
- Incorporate available and effective Web related technology into decision support for release planning first, then the other software engineering activities.
- Build a prototype of Web-based DSS supporting wireless and mobile devices that can be used by stakeholders to participate in the planning process through simple Web browser.

Acknowledgements

The authors would like to thank the Alberta Informatics Circle of Research Excellence (iCORE) for its financial support of this research.

References

[1] Guenther Ruhe, "Software Engineering Decision Support—Methodology and Applications", *chapter 5 in Innovations in Decision Support Systems*, Graziella Tonfoni, Lakhmi Jain (Eds.), International Series on Advanced Intelligence, Volume 3, 2003, pp. 143-174.

[2] Michael P. Papazoglou, "The World of e-Business: Web-Services, Workflows, and Business Transactions", *Web Services, e-Business, and the Semantic Web*, Bussler et al. (Eds.), CAiSE 2002 International Workshop, WES 2002, Toronto, Canada, May 2002.

[3] Paul Cowles, "Web Services and the Semantic Web", *Web Services Journal*, Volume 02, Issue 12, 2002.

[4] J.P. Shim, et al., "Past, present, and future of decision support technology", *Decision Support Systems*, Issue 33, 2002, pp. 111-126.

[5] C. Carlsson, et al., "DSS: directions for the next decade", *Decision Support Systems*, Issue 33, 2002, pp. 105-110.

[6] Allan Leck Jensen, Iver Thysen, Peter S. Boll and B.K. Pathak, "Pl@nteInfo: A World Wide Web based Decision Support System for Crop Production Management in

Denmark", *Agricultural Information Technology in Asia and Oceania 1998*, pp.125-125.

[7] D. Greer, Guenther Ruhe, "Software Release Planning: An Evolutionary and Iterative Approach", *Information and Software Technology* (Accepted).

[8] P. Carlshamre, B. Regnell, B., "Requirements lifecycle management and release planning in market-driven requirements engineering processes", *Proceedings of 11th International Workshop on Database and Expert Systems Applications*, Sept. 2000, pp. 961-965.

[9] Carlshamre P., et al., "An industrial survey of requirements interdependencies in software product release planning", *Proceedings of Fifth IEEE International Symposium on Requirements Engineering*, August 2001, pp. 84-91.

[10] Dmitri Tcherevik, "Internet Operating System", *Web Services Journal*, Volume 02, Issue 09, 2002.

[11] Anand Rajaraman, Peter Norvig, "Virtual Database Technology: Transforming The Internet Into A Database", *IEEE Internet Computing*, Vol. 2, Issue 4, July/August 1998, pp. 55-58.

[12] Joram Borenstein, Joshua Fox, "Semantic Discovery for Web Services", *Web Services Journal*, Volume 03, Issue 04, 2003.

[13] Andrew Bibby, Cesar Brea, "Business Reengineering Through Web Services", *Web Services Journal*, Volume 02, Issue 09, 2002.

[14] Benjamin Khoo, Guisseppe Forgionne, "Web-Based Decision Making Support Systems", *Hawaii International Conference on Business, June, 2003*, Hawaii, USA.

[15] Helen S. Du, Xiaohua Jia, "A Framework for Web-based Intelligent Decision Support Enterprise", *IEEE Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW03)*, 2003.

[16] Suresh Sridhar, "Decision support using the Intranet", *Decision Support Systems*, 23 1998, pp. 19-28.

[17] Daniel Power, Shashidhar Kaparthi, "Building Web-based Decision support Systems", *Studies in Informatics and Control*, Vol. 11, No. 4, December 2002, pp. 291-302.

[18] Sergio A. Alvarez, et al., Introduction to Part IV: Data Mining and Web Knowledge Management, *Web Knowledge Management and Decision Support*, Oskar Bartenstein, et al. (Eds.), 14th International Conference on Applications of Prolog, INAP 2001 Tokyo, Japan, October 2001. LNAI 2543, Springer, 2003.

[19] Gregory E. Kersten, Sunil J. Noronha, "WWW-based negotiation support: design, implementation, and use", *Decision Support Systems*, 25 1999, pp. 135-154.

[20] Guenther Ruhe, "Intelligent Support for Selection of COTS Products", *Proceedings of the Net.ObjectDays 2002*, Erfurt, Springer 2003, pp. 34-45.

[21] Davis, A.M., The Art of Requirements Triage, IEEE Computer 36(3), Mar 2003, pp 42- 49.

[22] Penny, D., An Estimation-Based Management Framework for Enhancive Maintenance in Commercial Software Products, Proc. International Conference on Software Maintenance, 2002.

CUPTRSS: A Web-based Research Support System

Hong Tang[†] Yu Wu[‡] J.T. Yao[§] Gouyin Wang[‡] Y. Y. Yao[§]

[†]Office of Science and Technology, [‡]Institute of Computer Science and Technology
Chongqing University of Posts and Telecommunications, Chongqing 400065, P. R. China]
Email: {tanghong,wuyu, wanggy}@cqupt.edu.cn
[§]Department of Computer Science, University of Regina, Regina, Canada S4S 0A2
Email: {jtyao, yyao}@cs.uregina.ca

Abstract

Research Support Systems (RSS) provide research organizations and scientists with information and facility for improving their research capacity, quality and productivity. Application of Web Intelligence technologies in developing the Web-based Research Support System (WRSS) will make such systems more effective and convenient. This paper analyzes research support services and proposes a Web-based intelligent system to provide integrated research support. A prototype system, CUPTRSS, is discussed, which will be implemented at Chongqing University of Posts and Telecommunications (CUPT).

1. Introduction

With the never ending growth of the Internet and ever expanding of information on the Web, we have arrived at a new information age. The Web provides a new media for gathering, storing, processing, presenting, sharing, and using information. The impacts of the Web can be felt in almost all aspects of life. To meet the challenges and take advantages of the opportunities offered by the Web, a subfield of computer science, called Web Intelligence (WI), has been emerged recently. There is fast growing interest in Web intelligence research [6, 16].

According to Yao *et al.* [15],

> "*Web Intelligence (WI) exploits Artificial Intelligence (AI) and advanced Information Technology (IT) on the Web and Internet*."

The goals of WI research is to study, based on artificial intelligence and information technology, theories, methodologies, technologies, and algorithms for the design and implementation of Intelligent Web Information Systems (IWIS). Many types of IWIS are needed to serve different groups of users and purposes.

In this paper, we investigate and examine a special type of IWIS called Web-based Research Support Systems (WRSS) [10, 14]. WRSS could be viewed as a concrete research area of Web Intelligence. The objective is to build new and effective tools for research institutions, researchers and scientists in order to support their research activities. Such tools will assist researchers to improve their research quality and productivity.

The design and the implementation of viable Research Support Systems (RSS) depend on a clear understanding of research activities and process. Research activities can be broadly classified into two levels, the institutional level and the individual level. The institutional level activities deal with the management of research and research projects in an institution. The individual level is the the actual research precess of a scientist. A good support system at institutional level will maximize the efficiency of research activity and guarantee conditions where research staff can concentrate on research. At the individual level, the system assists researchers at every phase in the research process [14]. The support at institutional level is closely related to Decision Support Systems (DSS), and the support at individual level is concerned with the integration of existing software systems and tools [14].

The study of Web-based research support systems on its own will lead to new theories, technologies and tools for supporting scientific research in the Web age. To illustrate the basic ideas, we discuss in detail a prototype system, CUPTRSS, which will be implemented at Chongqing University of Posts and Telecommunications (CUPT). The initial feedback from the research community in CUPT on the proposed system is positive and encouraging.

2. Motivations and Related Studies

In the management context, decision support systems (DSS) have been studied extensively to support and improve

decision making for managers [11]. To a large extent, DSS apply existing computer technologies to build systems that support management decision making in an organization. The development of computer science affects the design and implementation of DSS, as new computer technologies have been constantly added to DSS. For example, the development of the Web leads to Web-based decision support systems [8]. By applying the basic ideas of DSS to the context of research management and research process, one may consider the notion of research support systems (RSS). Tang considered RSS at the institutional level by focused on research management and administration tasks [10]. Yao proposed a framework of RSS at the individual level by focusing on the actual research activities of a researcher [14]. The combination of the two levels of support produces a complete model of RSS. Furthermore, the development of RSS on the Web results in Web-based research support systems (WRSS).

The notion of research support systems has been used by many organizations and research institutions. There is typically a research office at a university or a research institution that provides supports to research activities. At the information age, every community runs a Web site to provide important information and support in different scales. Some systems are called Research Support Systems and some are called Research Management Systems (RMS). By querying with the search engine Google with exact phrase "research support system" in February 2003, we obtained 515 hits. It returned 48 hits with exact phrase "research and development system" at the same time.

From the Web search results, it is found that many organizations and research institutions adopt some kind of research support systems. Some of examples are presented below:

- The Research and Development Support System at the Defense Threat Reduction Agency's (DTRA) Center for Monitoring Research (CMR) (http://www.cmr.gov/) provides a broad range of support to the nuclear explosion monitoring R&D community.
- The Chinese Natural Language Processing (CNLP) platform (http://www.nlp.org.cn/), developed and maintained by Software Division of Institute of Computing Technology, Chinese Academy of Sciences, is providing sharing resources for the research on natural language processing.
- The Research Support Libraries Programme (RSLP) in United Kingdom (http://www.rslp.ac.uk/) aims to facilitate the best possible arrangements for research support in UK libraries. "The programme has been 'managed', and has attempted to take a holistic view of library and archive activity throughout the UK."
- National Institute of Health (NIH, United States) has planned 3 NIH-wide information system projects (http://www.nih.gov/), they are CRIS (Clinical Research Information System), eRA (Electronic Research Administration) and NBRSS (Business and Research Support System).
- DCU Genius (http://rss.dcu.ie/GeniusSearch/genius.asp) is a research support system of Dublin City University in Ireland that contains comprehensive information on research staff at the university, including details of consultancies and other research outputs, publications etc.
- InforShare (http://infoshare.mednet.ucla.edu/infosys.htm) is a Research Support System in University of California at Los Angeles (UCLA). It integrates resources researchers need to conduct research such as a database of research equipment, a database of UCLA research databases, a database of grant funding information, faculty research interests, and grant application and financial systems.

Although research support systems have been widely used, there is still a lack of systematic study of such systems. Furthermore, most of systems focus on the management and administration of research activities at the institutional level.

Existing studies on individual level support focus on some specific tasks of research activities. For example, search/retrieval, reading and writing may be considered as three basic activities. Support systems for such activities have been studied by many authors. As an illustration, we searched the Google in August 2003 and found the following results:

Exact Search Phrase	Number of Hits
search support system	114
search support systems	21
retrieval support system	141
retrieval support systems	121
reading support system	120
reading support systems	8
writing support system	92
writing support systems	18

A careful examination of the returned Web pages reveals several useful observations. Different types of support systems have long been considered in many contexts. The recent advances in digital library and Web show the necessity of search/retrieval, and reading support. Many studies concentrate on the support of a particular research activity relatively independent to each other. For example, Web-based information retrieval support systems (WIRSS) assist retrieval related activities, such as browsing, searching, organization, and utilization of information [12, 13] on the

Web platform. WIRSS provide models, languages, utilities, and tools to assist a scientist or researcher in exploring, searching, analyzing, understanding, and organizing a document collection and search results. These tools allow the user to explore both semantic and structural information of each individual document, as well as the entire collection. There is clearly a need to integrate and study those systems in a unified model. A recently proposed framework of Web-based research support systems (WIRSS) attempts to address this problem. A brief description of this framework will be provided in the next section.

3. A Model of Research Support Systems

Based on the results of related studies reviewed in the last section, we propose a model of research support systems by considering the two levels of support. For simplicity, we used university in the discussion. A university research support system services two group of people, university research management staff and individual researchers. It in fact represents two types of research support services. The former provides administrative support and guidance on all matters related to the research effort. The latter provides support to the full range of research activities in order to meet the needs of research.

3.1. Research Support for Management Staff

Research management provides administrative support and guidance on all matters related to the research effort. According to a survey conducted by Association of Commonwealth Universities in 2001, there are four research management models [9]:

Model A: One Central Office or One Stop Shop One office administers grants and deals with industrial liaison and commercialization. Some offices are more comprehensive than others even dealing with financial management of awards.

Model B: Multiple Central Offices Two or more nonfinancial offices are involved in the main functions of research management. The most commonly involved offices are industrial liaison, technology transfer that make contact with businesses and deals with commercialization.

Model C: No research office The research management functions are carried out in the office of senior management or in another administrative office.

Model D: Partial research office The central research office carries out only part of the process, most commonly grant administration.

Model A and B count for 78% from the respondents. With a half of the remaining universities fall into Model C and D category are from Africa and Asia. The survey shows that over the past 10 years there has been considerable movement towards centralized structures for research management [9]. Even though there are different research management strategies in different research communities, the research management services are concentrated in 4 areas, as shown in table 1.

We may conclude from the above discussion that there are two types of services in research management: (1) To provide various information and guidelines related to research activities; (2) To deal with administrative transaction and collaborate with both of researchers and grand sponsors. To support research managers to cope with the research management, an integrated, web-based management system, which combines Office Automation (OA) with the function of a Management Information System (MIS), is needed. Although individual researcher has his/her way of doing research, we can also roughly divide the procedure into three stages from management supporting of views:

Survey and proposal investigating the interested academic field, assessing its state and finding the issues to be solved, making choice on topics to research and preparing the proposal

Research and development planning and designing how to do in research project, collecting and analyzing relative information and data, developing related technologies, making simulation and testing the results;

Summarizing and evaluation making conclusion of the project, evaluating the results, presenting publications and figuring out future research issues.

It is obvious that research information, research environment and research collaboration are three critical factors leading to successful research. Research outcomes such as publication, commercialization, etc are also important to both researchers and funding agencies. Figure 1 shows the research activities in three stages from management perspective. A Research Support System should provide both managers and researchers with services, including information services, sharing resources and collaborative work support. In addition, it should be easily accessible.

In general, a WRSS should fulfill the needs of a research office such as,

- To help researchers effectively and efficiently identify funding opportunities, prepare grants proposals and contracts;
- To provides researchers with information retrieval support that help them find their interested information efficiently;

	Responsibilities	Provision
Comprehensive administration	Gather and disseminate research related information Help researchers to build and sustain partnerships with industry, government, universities, public and private agencies Maintain archives Fulfill research plan,policies and strategy Provide research training Monitor and promote ethical practices in research	Policies and strategies structure Planning report Training courses Research ethics Scientific and technological archives
Project management	Provide information on funding opportunities Organize proposal and application for projects Negotiate contracts Supervise progress of projects Organize result evaluation	Knowledge of funding opportunities Advice on proposal Annual management report
Result management	R&D statistics Intellectual property protection and management Technology transfer Publication/Dissemination Seminar sponsor	Marketing intelligence Identification and exploitation intellectual property Annual R&D statistic report Publication
Financial Management	Administration of research funds and grants Producing certified financial acquittance for individual grants and contracts Outlay control	Project outlay report Financial final accounts report

Table 1. Research Management Responsibilities and Provisions

- To provide public resources sharing, such as data, computing capacity, programming and testing environment, experiment condition, etc.
- To help research managers effectively deal with administrative affairs

Therefore a WRSS could be designed as an integrated system which consists of 4 subsystems as shown in Figure 2. Research management subsystem provides research management support services for research managers. Information Retrieval Support subsystem provides information support services. Resource Sharing Support subsystem provides public research resources information and environment. Collaborative Research Support subsystem provides a public platform to support collaborative work.

The system consists of three layers, namely infrastructure, application system and user access control. The WRSS is established on campus network and interconnected with Internet. It is a distributed system that connects to the Research Support Office, the Finance Department, the Library, the Laboratory & Equipment Management Office, the Network Center and Academic Departments. Each unit operates and maintains related database. And it deploys the "one-stop shop" model, namely all users access to the system and are served through a unique portal.

In fact, many universities have constructed various computer management information systems, such as Office Automation system, Research Management System, Financial Management System, Asset Management System, Digital Library, Human Resource Management System, etc. These systems are usually separated in different departments or offices and are valuable in establishing the integrated Research Support Systems.

3.2. Research Support for Individual Researchers

For the individual level support, we briefly summarize the framework recently proposed by Yao [14].

Research is a highly complex and subtle human activity, which may be difficult to formulate formally. By adopting Graziano and Raulin's model [4], we can model research procedures into seven phases, namely, Idea generating, Problem definition, Observation/experimentation, Data analysis, Results interpretation phase and Communication phases.

Idea-generating phase. The objective is to identify a topic

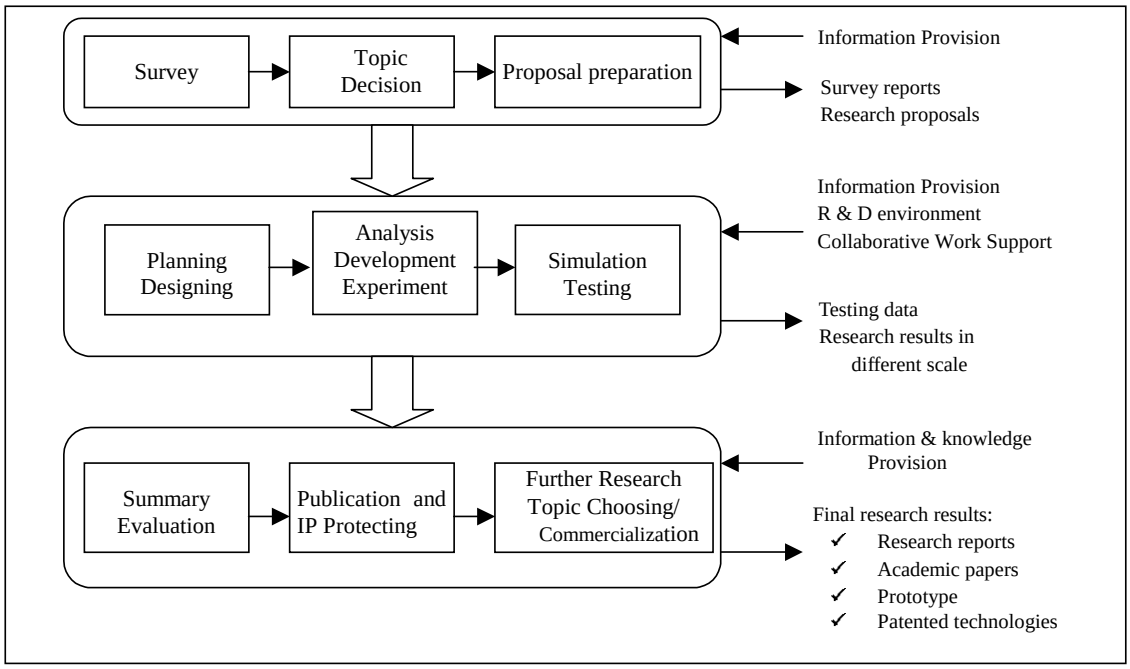


Figure 1. Management Support to Research Activities

of interest to study. Initial ideas can emerge from vague thoughts and in very non-scientific ways. Literature search and reading also play an important role in this phase.

Problem-definition phase. The objective of this phase is to precisely and clearly define and formulate vague and general ideas generated in the previous phase. Problem definition involves careful conceptualization and abstraction. The success in problem definition increases the probability of a successful research project.

Procedure-design/planning phase. The objective is to make a workable research plan by considering all issues involved, such as expected findings and results, available tools and methodologies, experiments, system implementation, time and resource constraints, and so on. This phase deals with planning and organizing research at strategic level.

Observation/experimentation phase. The objective is to observe real world phenomenon, collect data, and carry out experiments. Depending on the nature of the research disciplines, various tools and equipment, as well as different methods, can be used.

Data-analysis phase. The objective is to make sense out of the data collected. One extracts potentially useful information, abstraction, findings, and knowledge from data.

Results-interpretation phase. The objective is to build rational models and theories that explain the results from the data-analysis phase. The connections to other concepts and existing studies may also be established.

Communication phase. The objective is to present the research results to the research community. Communication can be done in either a formal or an informal manner. Books and scientific journals are the traditional communication media. Web publication is a new means of communication. Oral presentation at a conference, or discussion with colleagues, is an interactive means of communication.

To assist researchers in each of the above phases, one needs to consider the following specific supporting functionalities:

- **Exploring support**. In the early stage of research, a scientist may have a vague idea and may not be aware of the works of fellow researchers. Exploration thus plays an important role. There are many means of exploration, such as browsing databases, libraries, and the Web. If the Web is used for browsing, the historical data can be tracked. The collected data can be analyzed using machine learning and data mining tools to provide a scientist useful information and hints. Currently, Web browsers are a useful exploration tool. Their functions need to be expanded for providing support to research.

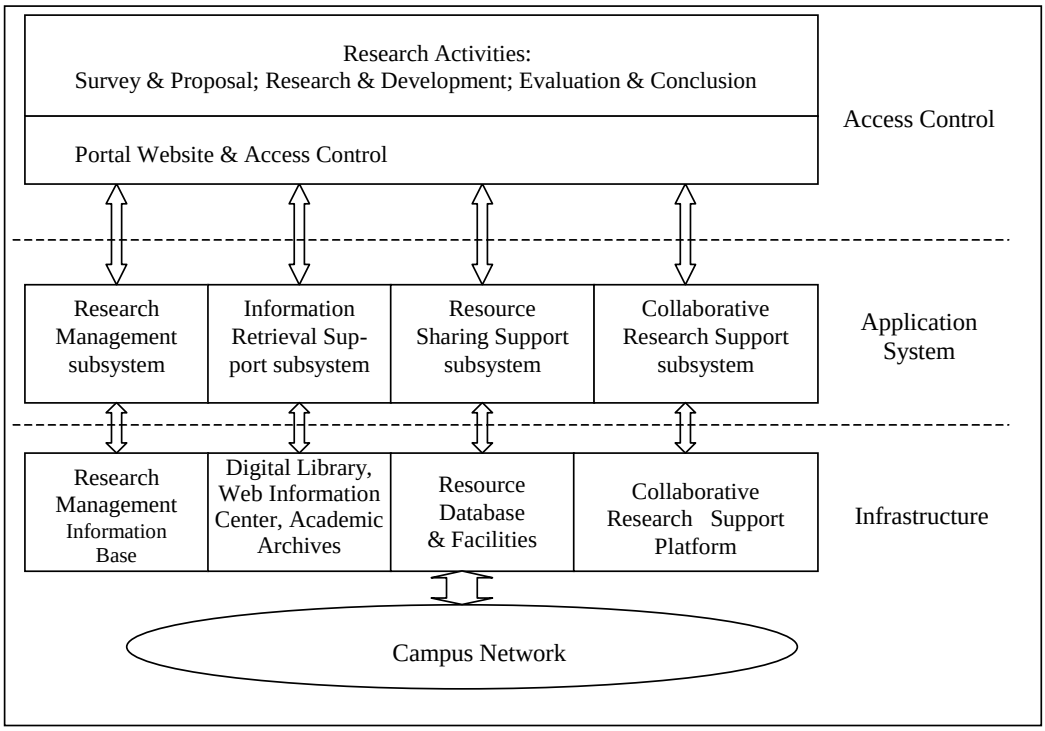


Figure 2. An Architecture of Web-based Research Support Systems

- **Retrieval support**. Once a scientist forms relatively solid ideas, it is necessary to search literature to find relevant information. Retrieval support assists retrieval related activities, such as browsing, searching, organization, and utilization of information [13, 14, 12]. A more detailed discussion on retrieval support will be given in the next section.
- **Reading support**. Reading critically and extensively is important, especially in the preparation stage [2, 5]. The advances in digital libraries and electronic publications make the reading support a necessity [3]. Software packages exist so that a reader can add book marks, make notes, link different parts of an article, and make logical connections of different articles. A reading support system needs to assist a reader in actively finding relevant materials, as well as constructing cognitive maps of the materials read. Reading support systems can be combined with exploring and retrieval support systems. Machine learning and text mining methods can be used to assist a reader by learning from the reading history. Agent technology can be used to actively look for useful information and periodically inform scientists with new information. On-line dictionaries may also be useful in reading support.
- **Analyzing support**. Successful analyzing support depends on tool management. It is necessary to help a scientist find the right tool for a particular problem in analyzing data. In addition, the system should also assist a scientist in using a tool. An explanation feature may be needed, which answers the question why a particular tool is used. If the functions of tools are described as plain text, information retrieval systems can be used to find the right tool. Computer graphics and visualization may be useful in analyzing support.
- **Writing support**. There are many writing support software tools, such as word-processor and typesetting software. Many packages come with additional functions, such as spelling-checking, grammar-checking, and various other agents. A writing support system should also contain some functions mentioned in the retrieval support systems. For example, a writing support system can find relevant articles based on the text written by a scientist and suggest possible references.

A research support system for individual research consists of many sub-systems to support different activities. The sub-systems share common data and knowledge bases. As one can not have a clear classification of research activities, it is difficult to have a clear classification of different types of support sub-systems.

4 A Prototype System: CUPTRSS

To support research and its activities at Chongqing University of Posts and Telecommunications (CUPT), China,

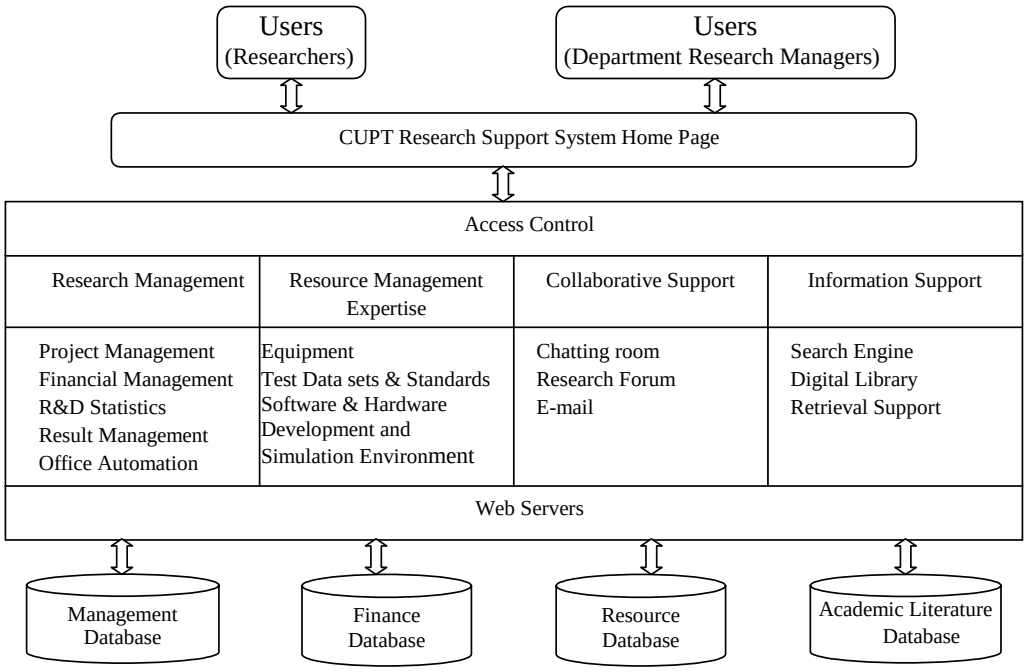


Figure 3. CUPT Research support System

we design and implement a prototype system for web-based research support. The system is maintained by the Office of Science and Technology of CUPT. It aims at providing researchers with accurate and timely information, improving research management and integrating public research resources at the university. It is also aimed to implement the frame we introduced in previous section. The current design of the system is mainly for management support. The system is also acts as test bed or platform for further research in WI related technologies. The structure of the system is shown in Figure 3. We adopt multi-layered architecture. The top layer is different users of this systems. The homepage represents layers of presentation and business logic. Under these two layers are access control layer and databases. The users access to the system through the unique portal, namely CUPTRSS homepage. There is an authentication module to manage the access control. Users, databases as well as web servers are distributed in different offices and departments.

The main function of the system includes:

Research management It is a combination of an Office Automation (OA) system and a Management Information System (MIS) for supporting research managers in terms of dealing with administrative transactions and providing relative services. It includes project management, contract management, financial management, result management, intellectual property, policy and strategy, research archives management and research and development statistics.

Research resource management This is an management subsystem that mainly focuses on research resources including personnel, data, software and hardware. It provides an online platform for instruments, simulation hardware and software. Acquisition and supply, human resource and expertise database are also included. All resources can be shared by different departments.

Information Support It includes collection and dissemination of information about funding opportunities, project guidelines and procedures, science & technologies news and marketing information. It provides information retrieval support including a Chinese Bibliography search system. A web-based intelligent search engine employing fuzzy, rough etc soft computing technologies is also the function of this module.

Collaborative work support It includes some subsystems such as a research forum for posting research papers in order to get comments and feedback from peers, a research chatting room for exchanging research ideas and discussion with other researchers, and many others.

The CUPTRSS is an ongoing project and used while research and development are going on. An open platform is adopted for future improvement and adding new functionalities as the new needs appeared as time going.

5 Conclusion and Future Work

Developing Research Support Systems is a very important research topic in the domain of Web Intelligence. RSS especially WRSS support research activities for researchers and research offices in universities. They also aim to promote research and related activities in an institution. Web-based technologies make the WRSS easy to use and access. We present a model for Web-based research support systems. WRSS provide two level of support, namely institute level for management staff and individual level for researchers. We also present a prototyped WRSS implemented at CUPT. Its main function is to server the needs in the research office of CUPT. It may also support research activities for some researchers. There are some other issues remain unsolved such as how to merge the goals of researchers and management staff. Future work include fine tune the system and enhance it functionality.

Acknowledgement

Tang, Wu and Wang would like to thank the partial support to this project from the National Science Foundation of P. R. China (No.69803014), the National Climb Program of P. R. China, the Foundation for University Key Teacher by the State Education Ministry of P. R. China (No.GG-520-10617-1001), the Application Science Foundation of Chongqing, the Science and Technology Research Program of the Municipal Education Committee of Chongqing, and the Research Foundation of Chongqing University of Posts and Telecommunications.

References

[1] G.D. Anderson, T. Snider, B. Robinson, J. Toporek, An Integrated Research Support System for Interpackage Communication and Handling Large Volume Output from Statistical Database Analysis Operations, *Proceedings of the 2nd International Workshop on Statistical Database Management*, USA, 1983, pp104-110.

[2] W.I.B. Beveridge, *The Art of Scientific Investigation*, Vintage Books, New York, 1957.

[3] G. Crane, Cultural heritage digital libraries: Needs and components, *The 6th European Conference on Digital Libraries*, Rome, Italy, 2002, pp626-637.

[4] A.M. Graziano, M.L. Raulin, *Research Methods: A Process of Inquiry*, 4th edition, Allyn and Bacon, Boston, 2000.

[5] G.W. Ladd, *Imagination in Research: An Economist's View*, Iowa State University Press, Ames, Iowa, 1987.

[6] J. Liu, N. Zhong, Y.Y Yao, Z.W. Ras, The Wisdom Web: new challenges for Web Intelligence (WI), Special issue guest editors' introduction, *Journal of Intelligence Information Systems*, **20**, 5-9, 2003.

[7] T. Ozono, S. Goto, N. Fujimaki and T. Shintani, P2P Based Knowledge Source Discovery on Research Support System Papits, *Proceedings of the 1st International Joint Conference on Autonomous Agents & Multiagent Systems*, 2002, Italy, pp49-50.

[8] D.J. Power, S. Kaparthi, Building Web-based decision support systems, *Studies in Informatics and Control*, **11**, 291-302, 2002.

[9] J. Stackhouse, J. Kubler, "Survey of Research Management in Commonwealth Universities", *Research Opportunities*, No. 4, 12-15, 2002.

[10] H. Tang, Web-based research support systems, Manuscript, University of Regina, 2003.

[11] E. Turban, J. E. Aronson, *Decision Support Systems and Intelligent System*, Prentice Hall, New Jersey, 2001.

[12] J.T. Yao, Y.Y. Yao, Web-based information retrieval support systems: building research tools for scientists in the new information age, *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, 2003.

[13] Y.Y. Yao, Information retrieval support systems, *Proceedings of FUZZ-IEEE'02*, 2002, pp773-778.

[14] Y.Y. Yao, A framework for web-based research support systems, proceedings of COMPSAC'2003, Dallas, USA, Nov 2003 (to appear).

[15] Y.Y. Yao, N. Zhong, J. Liu, S. Ohsuga,"Web Intelligence (WI): Research Challenges and Trends in the New Information Age", in N. Zhong, Y.Y. Yao, J. Liu, and S. Ohsuga, (eds.) Web Intelligence: Research and Development, LNAI 2198, Springer-Verlag 2001, pp1-17.

[16] N. Zhong, J. Liu, and Y.Y. Yao (Eds.), *Web Intelligence*, Springer, Berlin, 2003.

A Web-based Collaboratory for Supporting Environmental Science Research

Xiaorong Xiang Yingping Huang Gregory Madey
Department of Computer Science & Engineering
University of Notre Dame
Notre Dame, IN 46556
{xxiang1, yhuang3, gmadey}@nd.edu

Steve Cabaniss
Department of Chemistry
University of New Mexico
Albuquerque, NM 67131
cabaniss@unm.edu

Abstract

A scientific collaboratory for supporting research in the field of environmental science is presented in this paper. The purpose for building this Web-based research support system is to promote collaboration among a geographically separated group of NSF sponsored scientists from different research areas and allow them to share their data and information across distributed sites. An XML-based Markup Language, NOML, is provided to build the XML-based Web components and facilitate Web services development in the future. This collaboratory also consists of a set of on-line simulators with an intelligent interface for guiding simulation configuration and various electronic communication tools. The development of this collaboratory takes advantage of the J2EE and RDBMS technologies to provide scientists a robust and flexible environment to do science on the Web.

1. Introduction

Prior research into Web-based research support systems has had two orientations: 1) system that support the individual researcher [25], and 2) collaboratories that support group of researchers.

A collaboratory is defined as "an open meta-laboratory that spans multiple geographical areas with collaborators interacting via electronic means" [11]. It is a merger of the words "collaboration" and "laboratory" first coined by Wulf (1996) [12] who was elected as President of the National Academy of Engineering in 1997. Collaboratories are intended to promote collaborations among scientists in various research areas across geographic boundaries. They allow scientists to share expensive research instruments as well as data and information stored at distributed sites, to exchange personal experiences, and to accelerate the development and dissemination of knowledge. Several collaboratories have been developed for various scientific purposes, including the Diesel Collaboratory [20] in Combustion Science, BioCoRE [2] in Biology Science, and the EMSL Collaboratory [9][16] in Environmental Science. Sonnenwald et al. (2003) [22] did an evaluation of scientific collaboratories and investigated their capabilities and disadvantages. They concluded that "there is positive potential for the development of scientific collaboratory systems."

Creating a collaboratory involves integrating existing software and hardware tools to build a seamless environment where scientists can work together virtually. A number of commercial and research tools have been developed to allow Web-based collaboration. These tools can be separated into two categories, synchronous tools and asynchronous tools. Electronic mail, discussion boards, and electronic notebooks are typical asynchronous tools, while audio/video conferencing, chat boxes, and white boards are synchronous tools. However, some collaboration tools, such as audio/video conferencing, rely on high performance hardware support. Audio/video conferencing allows scientists to see and hear each other and to communicate with each other, as if face-to-face. Although audio/video conferencing greatly supports scientific interaction, it is adopted in very few collaboratories due to the cost, hardware restrictions, low network bandwidth, and poor quality. Therefore, determining which tools are suitable for building into a collaboratory depends on whether the tools add value to the scientific interaction process or not. It also heavily depends on the requirements of end-users, the collaborators.

Besides these generic communication tools that can be integrated into the collaboratory, the software packages that scientists use to do science should be developed separately and integrated in such a way that they correspond to different scientific topics. The design and implementation of collaboratories are important to create a flexible collaboratory.

The NOM project involves a group of NSF sponsored researchers from different research areas, including chemists, ecologists, biologists, and computer scientists. As is often the case, these researchers are also geographically sepa-

rated. In order to enable scientists from different geographical areas and research areas to work together virtually, a Web-based collaboratory for supporting scientific collaborations in the NOM (Natural Organic Matter) research community is built based on the Java 2 Enterprise Edition (J2EE) [8] platform from Sun Microsystem. The NOM collaboratory provides capabilities for scientists to share computational resources (including large-scale databases and a high performance simulation cluster), analysis tools, simulation results, and data and information. It also allows scientists to communicate with each other through a wide range of tools. The NOM collaboratory also provides a XML-based Markup Language, NOML, which is used to manage molecule information and simulation configurations. Additionally, NOML is used to build the XML-based Web components to support the collaboration. Using the universal XML-based data representations enhances component reuse, enables data sharing, and facilitates Web services development. The NOM collaboratory provides several online simulation models to allow scientists to do various experiments via the Web. This collaboratory has been built, evaluated, and used by several environment research groups [1][24][4][14][13].

In this paper, an overview of the NOM collaboratory is introduced in section 2. NOML, a XML-based format for describing the molecule structure and the simulation configuration, is presented in section 3. The implementation of several collaboration components is described in section 4, and conclusions are drawn and future work are described in section 5.

2. The NOM collaboratory overview

The NOM collaboratory provides the following functions:

- Distributed computational resource utilization: Users can configure and invoke their simulations through a Web interface, and computational resources on remote sites are allocated transparently by a job manager.
- Data analysis: Users can view their simulation data, generated by the NOM simulator, from a Web-based interface. These data and information are represented in various types of graphs (bar charts, pie charts, line charts) and statistical reports by employing data query and data mining technologies.
- Information sharing: Users can share the results of their simulation, the molecule definitions, and the simulation configurations through web interfaces and a search engine.
- Data repository: Oracle databases are used to store the internal data that are generated from the NOM simulator. Additionally, external data, including publications, technical reports, and other forms of dissemination, which are uploaded by scientists, are also stored in the database.
- Secure access: Users do not have the same level of access privileges to all the tools in the NOM collaboratory. Some particular tools, such as the “Molecule validator” and the “NOM simulator”, can only be accessed by authorized persons. Users have access to their own simulations, and other users can not access data which have not been authorized for public usage.
- Communication tools integration: A discussion board and a chat room are integrated into the collaboratory in order to facilitate communications between users.

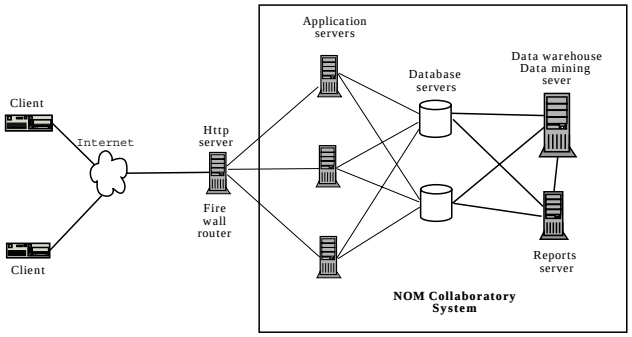


Figure 1. The architecture of NOM collaboratory

The NOM collaboratory is built upon the J2EE architecture running on a distributed cluster. This cluster has multiple dual processor PCs running Redhat Linux 8.0 and Windows 2000 operating systems. These machines in the cluster include a HTTP server, application servers, database servers, a reports server, and a data mining server. The architecture is showed as Figure 1.

The number of servers can be scaled to meet our requirements for better performance and reliability.

The implementation of the NOM collaboratory is supported by the J2EE architecture and a relational database management system (RDBMS). The Oracle RDBMS [18] was chosen for this purpose. OC4J (Oracle9iAS container for J2EE) was selected for the application servers. In order to overcome the disadvantages of traditional alternatives, such as Common Gateway Interface (CGI), for the dynamice creation of Web pages, Java Server Pages (JSPs), Servlets, JavaBeans, and Java Database Connection (JDBC) technology are applied. In order to easily maintain the programs, add new components, and reuse existing components, the design of the collaboratory follows the Model-View-Controller (MVC) design diagram [10]. Figure 2 illustrates the Web-based interfaces and components in the NOM collaboratory. Users can access these tools and ser-

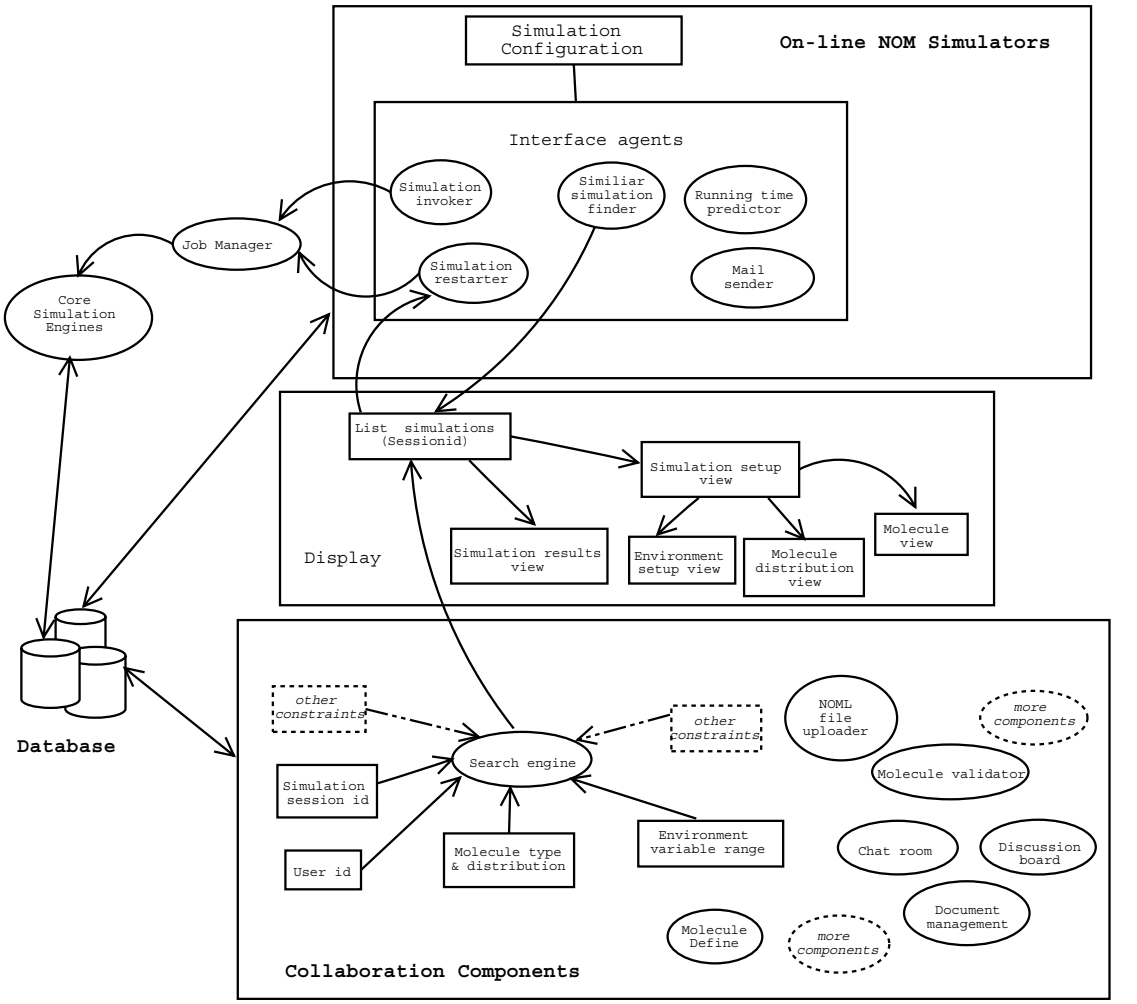


Figure 2. Components of the NOM collaboratory environment for supporting cooperation among remotely located scientists.

vices in the collaboratory environment through multiple Web-based interfaces from a standard Web browser.

The NOM collaboratory includes the following components:

- Web-based NOM simulators: NOM simulators are core parts of this collaboratory. Users can access these on-line NOM simulators through an intelligent Web interface using a standard Web browser. Several simulation models can be chosen to do their experiments. They can configure their simulations with multiple HTML and JSP pages and invoke their simulations on the remote computer cluster. The intelligent web interface provides a set of intelligent agents to find similar simulations, stop and resume simulations, predict running time of simulations, and send email to users after simulations are finished. These agents guide users to use the computational resources more efficiently. A user can submit one or more simulations and several users can submit their simulations simultaneously. A simple job manager assigns tasks on several simulation servers to achieve load balancing.

- Search engine: An ad-hoc query-based search engine provides users maximum flexibilities to search any information in the data repositories.

- NOML data format and file uploader: This Web service is built to support uploading of NOML format files. These files are parsed on the fly and the parsed data pieces are stored in the database.

- Molecule editor: Users can access the Molecule editor to define new molecule structures through a Web-based interface or upload a NOML format file.

- Molecule validator: Authorized users can access this tool from the Web interface to validate the newly cre-

ated molecule types or add new molecule types for public usage.

- Chat room and discussion board: These two collaboration tools are integrated into the collaboratory in order to facilitate the communication between users.

- Document management: This tool is integrated to exchange and disseminate information.

3. XML-based NOM Markup Language (NOML)

The XML-based model ensures uniformity across components and helps to abstract the component structure and implementation from the component interface. McLaughlin (2001)[15] and Ray (2001) [21] provide more details about XML.

The NOM collaboratory provides a set of standardized XML DTD definitions that form an ontology for the simulation configuration and the molecular structure definition. This standard format can reduce efforts of information interchange between users. Users can create XML documents, conforming to the standardized NOML DTD definitions, to describe simulation configurations and molecular structures. Once users have stored the information in NOML format files, they can attach these files into email or documents and share the data with others. Users can also upload files using the NOML uploader service that is embedded in the NOM collaboratory. Without using the Web-based configuration interface, a simulation can also be invoked after a NOML-based configuration file is uploaded. Three DTDs are defined in NOML to describe various data information.

(1) *environment.dtd* describes the format of the environment information. Users can define a set of environment parameters using tags that are defined in the DTD and used to indicate the owner as well as the accessing privilege.

(2) *molecules.dtd* describes the format of the molecular information. Within a XML file, users can define more than one new molecule structure and specify the accessing privilege.

(3) *setup.dtd* describes the format of the simulation configuration. Users can define one set of parameters, including environment parameters, molecule types, and molecule distributions, to configure a simulation.

Users can view the DTDs and download the NOML examples from the Web. The simple tree structure of the sample documents in NOML format are shown in Figure 3.

NOML is used to accomplish data exchange between users and applications. It can be extended by defining more DTDs to support various Web services.

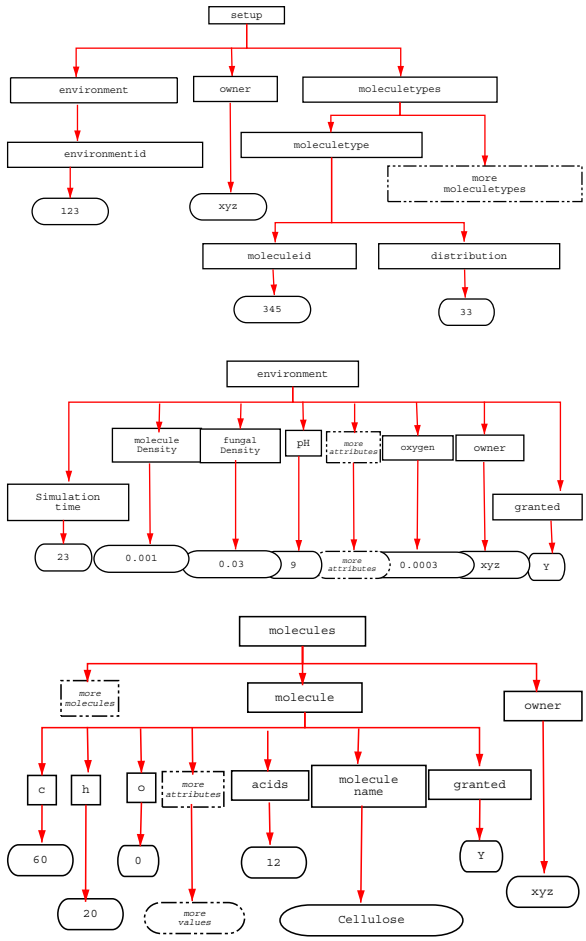


Figure 3. Tree structured view of three types of documents in NOML format. Note: not all the elements are shown in the graph.

4. Collaboratory components

All components of the NOM collaboratory are implemented within a thin-client, three-tier architecture by following the MVC design pattern. Components can be easily added, removed, and modified without changing any other features.

A client tier interacts with end users and displays information from the server to end users. In general, HTML and Java applets in a client container implement this tier. For the design of our Web interface, we use standard HTML, XML/XSL style sheets, and JSPs instead of Java applets for the performance concern.

A Web tier accepts users' responses from the client tier and generates the presentation logic. We used JSP pages for presentation logic and Servlets for session management.

An application tier handles the core scientific logic of the application. EJB components in an EJB container and Jav-

aBeans are used for implementing the application tier. In the NOM system, we used JavaBeans.

An Oracle database has been designed as the backend to store all the data input and output.

4.1. Web-based NOM Simulators

Natural organic matter (NOM) is a mixture of molecular compounds with different types of structures, compositions, functional group concentrations, molecular weights, and different degree of reactivity. NOM comes from animal and plant material in the natural environment. It exists everywhere in the world, from terrestrial ecosystems to aquatic environments. NOM plays a crucial role in ecological and bio-geochemical processes such as the evolution of soils, the transport of pollutants, and the global biochemical and geochemical cycling of elements [3]. The evolution of NOM over time from precursor molecules to mineralization is an important research area in a wide range of disciplines, including biology, geochemistry, ecology, soil science, and water resources. NOM, a prevalent constituent of natural waters, is highly reactive with mineral surfaces [3]. While NOM is transported through soil pores by water, it can be adsorbed onto or desorbed from mineral surfaces. Sorption of NOM is an important consideration in the treatment of drinking water.

NOM is present in all surface and soil waters. The amount and composition of NOM differ with climate and physical location, as well as a number of other environmental factors. NOM represents a significant fraction of the solute present in fresh water and influences all chemical and biological processes in the aquatic environment [17].

NOM, micro-organisms, and their environment form a complex system. The global phenomenon of a complex system can often be observed by simulating the dynamic behavior of individual components and their interactions in the system. Currently, in order to meet different requirments from different users, two different NOM simulators have been built into the collaboratory. Each of these NOM simulators is an agent-based stochastic model with an intelligent Web interface that can model NOM, mineral surfaces, and microbial interactions near the surface of the soil. More details on the modeling and the implementation of core simulation engines are described in Xiang(2003) [23] and Huang(2003) [7]. By simulating the behaviors of individual molecules in the system, the distributions of physical, chemical, and biological properties of NOM can be predicted. Additionally, such simulations can provide scientists in biology, geochemistry, and ecology valuable information for their studies.

The purpose of these NOM simulators is to provide scientists a testbed for their theoretical analysis and experimental results for NOM study. As the simulation system is refined, we expect that the system will help many scientists better understand the NOM complex system by providing them with information for predicting the properties of the NOM system over time.

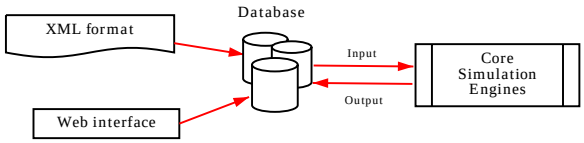


Figure 4. Simulation Input and Output

4.1.1. NOM simulation model overview The NOM simulation model includes three parts: an intelligent Web interface that assists users in configuring simulation parameters, core simulation engines that do the computations and simulate the complex behavior of a large number molecules, and a data analysis package that allows users to view their simulation results through a standard Web browser. Users can configure a simulation by inputing the simulation parameters either from the Web interface or from a NOML file. These inputs are stored in the backend Oracle database and a unique identification number is assigned to this particular simulation. The output data is stored into the database while the simulation is running. Figure 4 shows this process.

4.1.2. Intelligent agent components The Web interface provides remote users with a configuration tool. Users can specify the simulation parameters and invoke the simulation from a standard Web browser using the HTTP protocol. The intelligent configuration interface can also guide users in setting up their simulations step-by-step. Intelligent agents in the Web interface can not only provide a novice user with guidance during the configuration process, but can also provide the experienced user more information on how to better use the simulator. These intelligent agents are components based on JavaBean technology, and every component is a separate module. A summary of these intelligent agents is described as follows:

- The sending email agent

  The agent automatically sends an email to the user after the simulation is done. This email message gives the user information about the status of the simulation, either it was successfully completed or it was terminated due to any exception that was caused by hardware malfunction or software problem.

- The running time prediction agent

  A user can ask this agent to statically predict how long it will take to run the simulation according to the user's configuration inputs before the user invokes the simulation. According to the time returned by the

agent, the user can decide if the simulation configuration should be adjusted.

This agent also provides the capability for dynamically predicting the running time while the simulation is running. Users can submit a request that asks how much time remains before the finish of their simulation by simply clicking a button.

- The similar simulation finder

  Since executing a simulation is a time-consuming task, especially when the problem size is large and the running time is long, we provide users the option to take advantage of simulation results that were generated earlier by the user or by other users. A similar-simulation-finder-agent helps users find finished simulations that have some similarities with the current setup in the database. Users can either retrieve the simulation results from the database without running their simulations or submit the simulation for execution according to their own preference.

  By employing the "Euclidean distance," the distance from the current input data set to other data sets retrieved from the database can be computed. The data sets that have the smallest distance have the highest degree of similarity with the current setup.

- Automatic restarting agent

  The agent helps users extend their simulations for longer time steps by restarting their simulation from the point where it stopped. While one simulation is running, the data information about reactions is stored in the database every time step for further data analysis. In addition to this, the state of the simulation system is stored in the database at different checkpoints. The insertion and updating of the state at each checkpoint are treated as a data transaction process to guarantee data integrity.

4.2. Data analysis and visualization

One of the primary goals of this collaboratory is to provide an explanation of the simulation results, to help users better understand their data, and to predict the global properties of NOM over time. Large-scale scientific simulations take days to run and produce massive amounts of data. A user typically needs to run several variations and compare many sets of results with their experimental data. It is very difficult for users to do this with only raw data. Gray (2002) [5] shows how important new data mining algorithms are in helping scientists to access their on-line data more quickly. The Oracle9i tools are used to build the data warehousing and data mining and to incorporate them into the simulation model in order to organize, visualize, and analyze multitudinous data [6]. We enabled the display of simulation

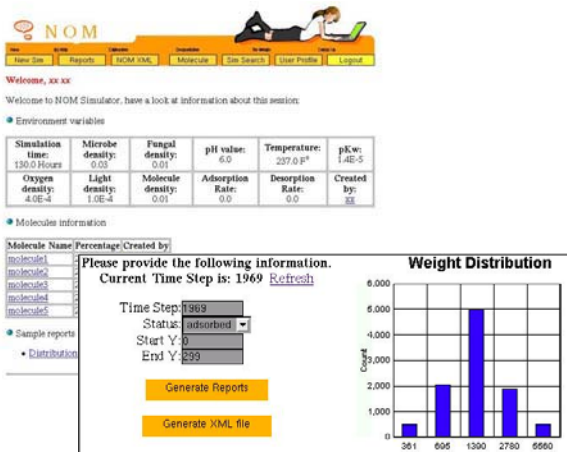


Figure 5. Two screenshots of the configuration and sample output of one simulation

results independently from the processing. Users can examine the results and download the data in XML format from the Web site during and after the simulation completes running. Figure 5 shows example of both input information and of sample output from a simulation executed by a scientist.

4.3. Search engine

Collaborative research demands sharing information and experiments to reduce the time it takes to produce new results. Executing a large-scale scientific simulation is a time-consuming task. The NOM collaboratory offers the capability of allowing users to share the configurations and results of their simulations by providing a search engine. Users can take advantage of the simulation's stored results from previous executions instead of doing the same simulation over and over again. By achieving this capability, the instruments and computational resources can be used more effectively.

Users can access the search engine through a Web interface by providing several search conditions. Users can either leave all the fields blank or make reasonable combinations and submit the request form to the server. The remote server processes the request and returns a list of simulations associated with the simulation configurations that meet with users' requests.

Users can view data information corresponding to each simulation in more depth by clicking on links. These information include the simulation configuration and result for a particular simulation. If the simulation results are valuable for users, they can extend the simulation by entering extra simulation time and resuming the simulation from the Web interface.

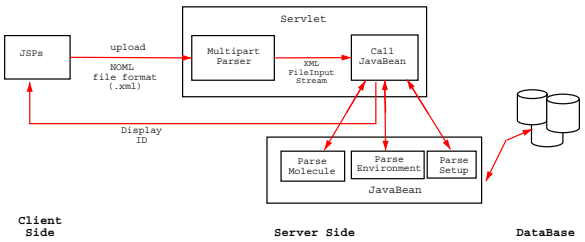


Figure 6. The design model for NOML file uploader

4.4. NOML file uploader

NOML file uploader provides an interface for users to upload a set of NOML format files. Users issue a Multipart Request by submitting a HTML form from the client side. On the server side, instead of saving files in the disk on the server, a Servlet reads incoming files and parameters and parses the incoming file's information to a XML FileInputStream. The FileInputStream is fed into a corresponding JavaBean according to the content of the incoming files. JavaBeans are implemented to validate and parse the NOML documents, establish a connection to the Oracle database, and write the data in the database. The Servlet returns corresponding ID to users for further reference. The design model is shown in Figure 6.

Users can access the NOML file uploader by providing a file name. The data validation is done by the server side JavaBeans. No information is stored to the database if the data is invalide and corresponding error messages are returned to users.

4.5. Molecule editor

Several simple rules are encoded to automatically validate the newly defined molecule in order to prevent the specification of chemically impossible molecule structures. Besides providing a Web interface for the molecule definition, users can also define new molecule types by uploading a NOML format file to the server. The uploaded file is parsed on the fly and the data is stored in the database through the NOML file uploader.

Users can also search the corresponding molecules according to the molecule name and owner of the molecules.

4.6. Molecule validator

Before one molecule type definition is ready to be shared by the NOM community, this molecule definition must be validated. However, the molecular validation is not only an objective task that can be accomplished by a piece of software, but also a subjective task that needs human involvement.

General rules that can be encoded in software are not sufficient for validating a newly created molecule. Therefore, an administrator role is provided to an authorized person who can validate newly defined molecules manually. The authorized user, a chemist in the general case, can access the validator. Only validated molecules can be shared by every user who participates in the collaboratory. Non-validated molecules can only be used by their owners.

An authorized user can also define the molecule type from the separate molecule editor or upload a NOML format file to the server. This user can remove a particular molecule from the NOM system.

4.7. Chat room and discussion board

In order to facilitate the communications between researchers, a chat room (a synchronous collaboration tool) and a threaded discussion board (an asynchronous tool) are integrated into the collaboratory. Users can enter the chat room through a Web interface by providing a user ID and password. The threaded discussion board helps users to easily trace discussion contents about one topic.

4.8. Document management

Collaborative research often demands exchange of large data files and documentation for faster dissemination of the knowledge. Users can upload research papers and data sets through a Web interface into a shared place residing on the remote servers. These files are accessible to other researchers after they are uploaded.

5. Conclusions and Future Work

In this paper, a description of a Web-based collaboratory environment that supports new ways for conducting scientific research on NOM is presented. The NOM collaboratory deploys collaborative technologies to NOM researchers geographically separated or in different disciplines. This system integrates a number of collaboration tools, including NOM simulators, a molecular validator, a molecule editor, search engine, document management, a discussion board, and a chat room. The architectural design of the NOM collaboratory makes it easy to add, modify, and remove components in the collaboratory according to the requirements of scientists.

In the future, some powerful third-party tools (such as video/audio conferencing) which rely on a fast and efficient high bandwidth network technology and advanced hardware (e.g., Internet2) may be integrated. Oracle Collaboration Suite [19] offers integrated email, voice mail, phone,

fax, scheduling, calendaring, meeting management, and file management. It could also be considered for integration into the NOM collaboratory.

An XML-based NOM Markup Language, NOML, which provides a standard definition for the molecule and simulation configuration is described. The NOML not only supports data communication between users but also give us flexibility in extending the on-line NOM simulation. It also supports building new XML-based Web services in the near future. A set of new simulation models for the NOM study will be built and integrated into the collaboratory. The communication between these models can be achieved by the extension of the NOML. The NOM collaboratory allows users from different places to collaborate by sharing their simulation data and configurations as well as by hosting discussions among the NOM community.

6. Acknowledgments

This research was supported in part by a NSF ITR Grant No. 0112820 and by the Center for Environmental Science & Technology at the University of Notre Dame. We acknowledge the contributions of Patricia Maurice and Leilani Arthurs of the Department of Civil Engineering & Geological Science at the Univeristy of Notre Dame for their respective discussions on the Web interface, input, and testing of the NOM collaboratory.

References

[1] L. Arthurs, P. Maurice, X. Xiang, G. Madey, and Y. Huang. Agent-based simulation of biocomplexity: Effects of adsorption on natural organic mobility through soils. In *American Chemical Society National Meeting*, 2003.

[2] M. Bhandarkar, G. Budescu, W. F. Humphrey, J. A. Izaguirre, S. Izrailev, L. V. Kalé, D. Kosztin, F. Molnar, J. C. Phillips, and K. Schulten. Biocore: A collaboratory for structural biology. In *Proceedings of the SCS International Conference on Web-Based Modeling and Simulation*, pages 242–251, 1999. Also available at: http://www.ks.uiuc.edu/Research/biocore.

[3] S. Cabaniss. Modeling and stochastic simulation of NOM reactions, working paper. http://www.nd.edu/ñom/papers/papers.html, July 2002.

[4] S. Cabaniss, G. Madey, P. Maurice, L. Leff, Y. Huang, X. Xiang, E. Chanowich, and O. Olapade. Stochastic synthesis model for the evolution of natural organic matter. In *225th American Chemical Society National Meeting*, 2003.

[5] J. Gray and A. Szalay. The world-wide telescope. *Communication of the ACM*, 45(11):51–55, November 2002.

[6] Y. Huang. Infrastructure, query optimization, data warehousing and datamining for scientific simulation. Master's thesis, University of Notre Dame, 2002.

[7] Y. Huang, G. Madey, X. Xaing, and E. Chanowich. Web-based molecular modeling using JAVA/SWARM, J2EE, and RDBMS Technologies. In *Seventh Annual Swarm Researchers Meeting (Swarm2003)*, 2003.

[8] Java 2 platform, enterprise edition. http://java.sun.com/j2ee/.

[9] G. C. Jr., J. Myers, and D. Hoyt. Social networks in the virtual science laboratory. *Communications of the ACM*, 45(8):87–92, August 2002. Also available at: http://collaboratory.emsl.pnl.gov/.

[10] N. Kassem and the Enterprise team. *Designing enterprise applications with the Java 2 Platform, Enterprise Edition*. Addison-Wesley, 2000.

[11] S. H. Koslow and M. F. Huerta, editors. *Electronic collaboration in science*. Lawrence Erlbaum Associates, 2000.

[12] R. T. Kouzes, J. D. Myers, and W. A. Wulf. Collaboratories: Doing science on the Internet. *IEEE Computer*, 1996.

[13] G. Madey, Y. Huang, X. Xiang, E. Chanowich, S. Cabaniss, and P. Maurice. Complex system simulation: interactions of nom molecules, mineral surfaces, and microorganism in soils. In *Workshop on Modeling Complex Systems, US Geological Survey*, 2002.

[14] G. Madey, Y. Huang, X. Xiang, E. Chanowich, S. Cabaniss, and P. Maurice. Simulation of biocomplexity: Interactions of nom, mineral surfaces, and microorganism in soils. In *ASLO 2003 Aquatic Sciences Meeting*, 2003.

[15] B. McLaughlin. *Java & XML, 2nd Edition*. O'Reilly & Associates, 2001.

[16] J. D. Myers, A. R. Chappell, M. Elder, A. Geist, and J. Schwidder. Re-integrating the research record. *Computing in Science and Engineering*, pages 44–50, May/June 2003. Also available: http://collaboratory.emsl.pnl.gov/.

[17] Natural organic matter in the nordic countries. http://www.kjemi.uio.no/envir/nominic/documents/background.html.

[18] Oracle database. http://www.oracle.com/ip/deploy/database/oracle9i/index.html.

[19] Oracle collaboration suite. http://otn.oracle.com/products/cs/content.html.

[20] C. M. Pancerella, L. A. Rahn, and C. L. Yang. The diesel combustion collaboratory: combustion researchers collaborating over the Internet. In *Proceedings of the 1999 ACM/IEEE conference on Supercomputing*. ACM Press, 1999.

[21] E. T. Ray. *Learning XML*. O'Reilly & Associates, 2001.

[22] D. H. Sonnenwald, M. C. Whitton, and K. L. Maglaughlin. Evaluating a scientific collaboratory: Results of a controlled experiment. *ACM Transactions on Computer-Human Interaction*, 10(2):150–176, June 2003.

[23] X. Xiang. Agent-based scientific applications and collaboration using java. Master's thesis, University of Notre Dame, 2003.

[24] X. Xiang, G. Madey, Y. Huang, and S. Cabaniss. A stochastic simulation of natural organic matter and microbes in the environment. In *2003 World Conference on Natural Resource Modeling*, 2003.

[25] Y. Yao. A framework for web-based research support system. In *Computer Software and Applications Conference (COMPSAC2003)*, Dallas, TX, Nov. 2003.

A RESEARCH SUPPORT SYSTEM FRAMEWORK FOR WEB DATA MINING

Jin Xu *Yingping Huang* *Gregory Madey*

Dept. of Computer Science
University of Notre Dame
Notre Dame, IN 46556
Email: {jxu1, yhuang3, gmadey}@cse.nd.edu

ABSTRACT

Design and implementation of a research support system for web data mining has become a challenge for researchers wishing to utilize useful information on the web. This paper proposes a framework for web data mining support systems. These systems are designed for identifying, extracting, filtering and analyzing data from web resources. They combines web retrieval and data mining techniques together to provide an efficient infrastructure to support web data mining for research.

Keywords: Data mining, web retrieval, clustering, classification, association rules, research support system

1. INTRODUCTION

The evolution of the World Wide Web has brought us enormous and ever growing amounts of data and information. With the abundant data provided by the web, it has become an important resource for research. However, traditional data extraction and mining techniques can not be applied directly to the web due to its semi-structured or even unstructured nature. Web pages are Hypertext documents, which contain both text and hyperlinks to other documents. Furthermore, web data are heterogeneous and dynamic. Thus, design and implementation of a web data mining research support system has become a challenge for researchers in order to utilize useful information from the web.

Usually, web mining is categorized as web content mining and web usage mining. Web content mining studies the search and retreval of information on the web, while web usage mining discovers and analyzes user access pattern [1]. A knowledge discovery tool, WebLogMiner, is discussed in [8] which uses OLAP and data mining techniques for mining web server log files. In [5], a web mining framework which integrated both usage and content attibutes of a site is described. Specific techniques based on clustering and association rules are proposed. Yao [3, 6, 7] presents a framework and information retrieval techniques to support individual scientists doing research. This paper proposes a framework for designing web data mining research support systems. These systems are designed for identifying, extracting, filtering and analyzing data from web resources. They combine web retrieval and data mining techniques together to provide an efficient infrastructure to support web data mining for research. This framework is composed of several stages. Features of each stage are explored and implemention techniques are presented. A case study on mining data from a large software development site is provided as an example of how to use this framework. Our work provides a general solution which researchers can follow to utilize web resources in their research.

The rest of this paper is organized as follows: Section 2 presents a framework for web mining research support systems and a brief overview of its components. Section 3 describes design and implementation of web data retrieval. Section 4 focuses on processing and analyzing web data. Section 5 presents a case study based on this web mining research support system. Conclusions and future work are given in Section 6.

2. FRAMEWORK OVERVIEW

In order to explore web data, we construct a research support system framework for web data mining, as shown in Fig 1, consisting of four phases: source identification, content selection, information retrieval and data mining.

In the first phase, proper web sites should be chosen according to research needs. This includes identifying availability, relevance and importance of web sites. Key words searching by using search engine can be used to find appropriate web sites.

After finding all web sites identified by the first phase, the second phase is to select appropriate contents on those web sites, such as documentation, newsgroups, forums, mailing lists, etc. Usually, a web site contains many web pages, including relevant and irrelevant information. This phase is important because it decides which web information should be extracted. The selection of web pages is based on research purpose and a researcher's experience.

In the information retrieval phase, a crawler is designed

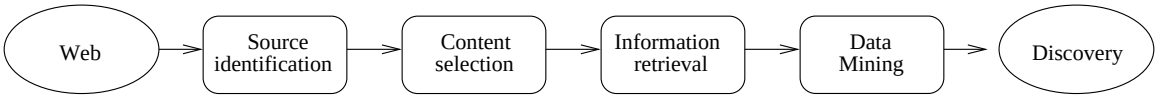


Fig. 1. Research support system framework for web data mining

to automatically extract information selected during the selection phase. Specific tools and techniques are employed to effectively retrieve useful knowledge/information from web sources. Additional effort may be required for dynamic content retrieval and specfic data sources such as newsgroup, forum, etc.

The final phase is to conduct data mining on extracted web data. It includes preparing data for analysis. An extracted web page may contain missing data, extraneous data, wrong format and unnecessary characters. Furthermore, some data should be processed in order to protect privacy. Advanced data mining techniques are employed here to help analyzing data.

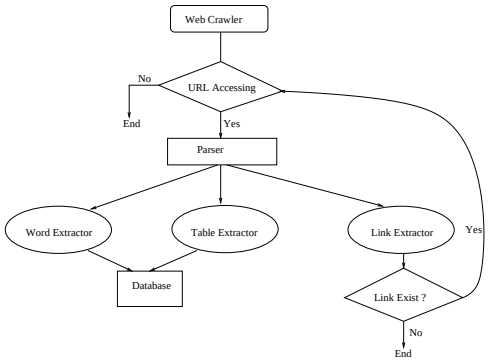


Fig. 2. The web crawler

3. INFORMATION RETRIEVAL

Information retrieval is used to provide access to data on the web. This implies a web mining research support system should be able to search for and retrieve specific contents on the web efficiently and effectively. There are two major categories of searching tools on the Web: directories (Yahoo, Netscape, etc.) and search engines (Lycos, Google, etc.). It is hard to use directories with the increase of web sites. Search engines cannot meet every search requirement

In our system, a web crawler based on advanced tools and techniques is developed to help find useful information from web resources. Web crawlers are also called spiders, robots, worms, etc. A web crawler is a program which automatically traverses web sites, downloads documents and follows links to other pages [4]. It keeps a copy of all visited pages for later uses. Many web search engines use web crawlers to create entries for indexing. They can also be used in other possible applications such as page validation, structural analysis and visualization, update notification, mirroring and personal web assistants/agents etc. [2]. Search engines are not adequate for web mining for a research project. It is necessary to design a web crawler which includes methods to find and gather the research related information from the web. Although different research projects have different web information which leads to different web crawlers, those crawlers still have some common designs, as shown in Figure 2. They can be implemented by Java, Perl, Python, etc.

A web crawler should start with a URL, identify other links on the HTML page, visit those links, extract information from web pages and store information into databases. Thus, a web crawler consists of a URL access method, a web page parser with some extractors, and databases. The access function of a web crawler should prevent repeatedly accessing the same web address and should identify dead links. The parser recognizes start tags, end tags, text and comments. The databases provide storage for extracted web information.

The key component of a web crawler is the parser, which includes a word extractor, a table extractor and a link extractor. The word extractor is used to extract word information. It should provide string checking functions. Tables are used commonly in web pages to align information. A table extractor identifies the location of the data in a table. A link extractor retrieves links contained in a web page. There are two types of links – absolute links and relative links. An absolute link gives the full address of a web page, while a relative link needs to be converted to a full address by adding a prefix.

4. DATA MINING TECHNOLOGY

To facilitate web mining, the following data mining algorithms can be applied to find patterns and trends in the data collected from the web: clustering, classification, association rules. In the following sections, we will introduce each of the algorithms and explain their applications.

4.1. Association Rules

Association rules mining tries to find interesting association or correlation relationship among a large set of data items. A typical example of association rules mining is the market basket analysis. An association rule is something like ”80% of people who buy beer also buy fried chicken”.

Association rules mining can also be applied to predict web access patterns for personalization. For example, we

may discover that 80% of people who access page A and page B also access page C. Page C might not have a direct link from either page A or page B. The information discovered might be used to create a link to page C from page A or page B. One example of this application is amazon.com. We often see something like "customers who buy this book also buy book A". The association rules mining can be applied to web data to explore the behavior of web users and find patterns of their behaviors.

4.2. Classification

The goal of classification is to predict which of several classes a case (or an observation) belongs to. Each case consists of n attributes, one of which is the target attribute, all others are predictor attributes. Each of the target attribute's value is a class to be predicted based on the $n-1$ predictor attributes.

Classification is a two-step process. First, a classification model is built based on training data set. Second, the model is applied to new data for classification. In the middle of the two steps, some other steps might be taken, such as lift computation. Lift computation is a way of verifying whether a classification model is valuable. A value larger than 1 is normally good.

Classification models can be applied on the web to make business decisions. Applications include classifying email messages as junk mails, detecting credit card fraud, network intrusion detection, etc.

4.3. Clustering

Clustering is used to find natural groupings of data. These natural groupings are clusters. A cluster is a collection of data that are similar to one another. A good clustering algorithm produces clusters such that inter-cluster similarity is low and intra-cluster similarity is high. Clustering can be used to group customers with similar behavior and to make business decisions in industry.

5. CASE STUDY: OPEN SOURCE SOFTWARE (OSS) DEVELOPMENT

The OSS community has developed a substantial amount of the infrastructure of the Internet, and has several outstanding technical achievements, including Apache, Perl, Linux, etc. These programs were written, developed, and debugged largely by part time contributors, who in most cases were not paid for their work, and without the benefit of any traditional project management techniques. A research study of how the OSS community functions may help IT planners make more informed decisions and develop more effective strategies for using OSS software.

The OSS development community is a global virtual community. Thus, we have the advantage in that their digital interactions are archived and can be data mined. With approximately 70,000 projects, 90,000 developers, and 700,000 registered users, SourceForge.net, sponsored by VA Software is the largest OSS development and collaboration site. This site provides highly detailed information about the projects and the developers, including project characteristics, most active projects, and "top ranked" developers.

5.1. Data Collection

After informing SourceForge of our plans and receiving permission, we gathered data monthly at SourceForge. SourceForge provides project management tools, bug tracking, mail list services, discussion forums, version control software for hosted projects. Data is collected at the community, project, and developer level, characterizing the entire OSS phenomenon, across multiple numbers of projects, investigating behaviors and mechanisms at work at the project and developer levels.

The primary data required for this research are two tables – project statistics and developers. The project statistics table, shown in Figure 1, consists of records with 9 fields: project ID, lifespan, rank, page views, downloads, bugs, support, patches and CVS. The developers table has 2 fields: project ID and developer ID. Because projects can have many developers and developers can be on many projects, neither field is unique primary key. Thus the composite key composed of both attributes serves as a primary key. Each project in SourceForge has a unique ID when registering with SourceForge.

A web crawler, implemented by Perl and CPAN (Comprehensive Perl Archive - the repository of Perl module/libraries) modules, traversed the SourceForge web server to collect the necessary data. All project home pages in SourceForge have a similar top-level design. Many of these pages are dynamically generated from a database.The web crawler uses LWP, the libwww-Perl library, to fetch each project's homepage. CPAN has a generic HTML parser to recognize start tags, end tags, text and comments, etc. Because both statistical and member information are stored in tables, the web crawler uses an existing Perl Module called *HTML::TableExtract* and string comparisons provided by Perl to extract information. Link extractors are used if there are more than one page of members.

5.2. Data Mining

There are several data mining algorithms that can be applied to the web data. Among them are Naive Bayes, Classification and Regression Tree (CART), for classification, A Priori for association rules mining, K-means and Orthogonal-Cluster for clustering. Let's first focus on the classification

Table 1. Project statistics

project ID	lifespan	rank	page views	downloads	bugs	support	patches	all trackers	tasks	cvs
1	1355 days	31	12,163,712	71,478	4,160	46,811	277	52,732	44	0
2	1355 days	226	4,399,238	662,961	0	53	0	62	0	14,979
3	1355 days	301	1,656,020	1,064,236	364	35	15	421	0	12,263
7	1355 days	3322	50,257	20,091	0	0	0	12	0	0
8	1355 days	2849	6,541,480	0	17	1	1	26	0	13,896

algorithms: Naive Bayes and CART.

5.2.1. Classification

The Naive Bayes algorithms makes prediction using Bayes' Theorem. Naive Bayes assumes that each attribute is independent from others. In this case study, that is not the case. For example, the "downloads" feature is closely related to the "cvs" feature, the "rank" feature is closely related to other features, since it is calculated from other features.

CART builds classifi cation and regression tress for predicting continuous dependent variables (regression) and categorical predictor variables (classifi cation). We are only interested in the classifi cation type of problems since the software we are using only handles this type of problems. Oracle's implementation of CART is called Adaptive Bayes Network (ABN). ABN predicts binary and multi-class targets. Thus discretizing the target attribute is desirable.

In this case study, we try to predict downloads from other features. As stated previously, the "downloads" feature is binned into ten equal buckets. We predict the downloads resides which buckets based on the values of other features. As expected, the Naive Bayes algorithms is not suitable for predicting "downloads", since it is related to other features, such as "cvs". The accuracy of Naive Bayes is less than 10%. While Naive Bayes performs badly on predicting "downloads", the ABN algorithms can predict "downloads" quite accurately which is about 63%. At fi rst sight, the accuracy 63% is not attractive, but it is a good prediction since we could only get 10% of correct predictions without classifi cation. The lift computation confi rms that the resulting classifi cation model is quite good, as shown in Figure 3. This fi gures shows that we found all the records whose "downloads" feature is 1 in just the fi rst 20% of records. The rules built by the ABN classifi cation model show that "downloads" is closely related to "cvs".

We conclude that the ABN algorithms is suitable for predicting "downloads" in our case study. The following table compares the two algorithms, namely, ABN and Naive Bayes. From the table, we see that ABN takes much longer time to build a classifi cation model, but the resulting model is much more accurate.

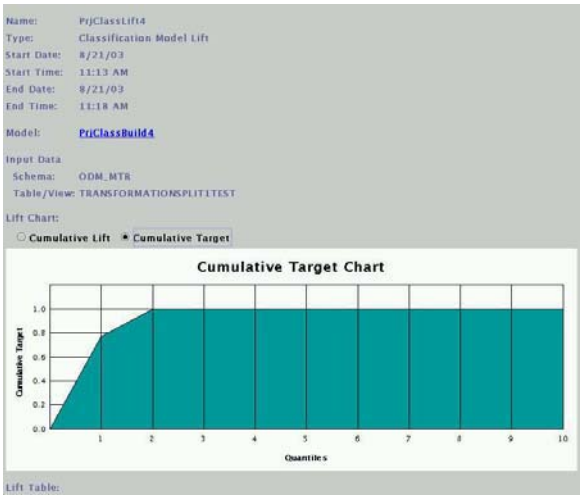


Fig. 3. The lift chart

Table 2. Comparison of ABN and Naive Bayes

Name	Build Time	Accuracy
ABN	0:19:56	63%
Naive Bayes	0:0:30	9%

5.2.2. Association Rules

The association rules mining problem can be decomposed into two subproblems:

- Find all combinations of items, called frequent itemsets, whose support is greater then the minimum support.
- Use the frequent itemsets to generate the association rules. For example, if AB and A are frequent itemsets, then the rule $A \rightarrow B$ holds if the ratio of support(AB) to support(A) is greater then the minimum confi dence.

One of the most famous association rules mining algorithms is called A Priori. Oracle implements this algorithm using SQL. We use the algorithm to try to fi nd correlations between features of projects. The algorithms takes two inputs, namely, the minimum support and the minimum confi dence. We choose 0.01 for minimum support and 0.5 for

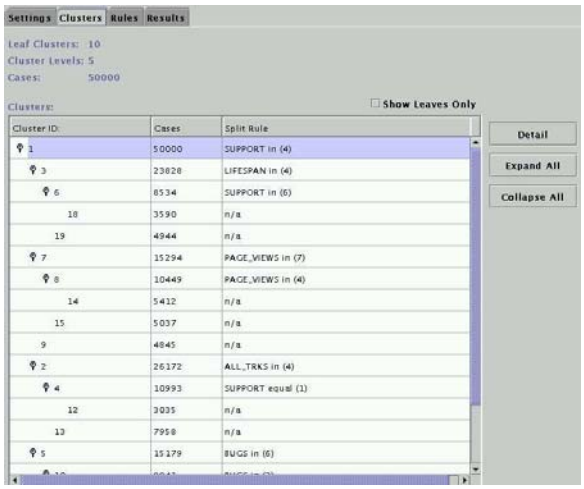


Fig. 4. The clusters

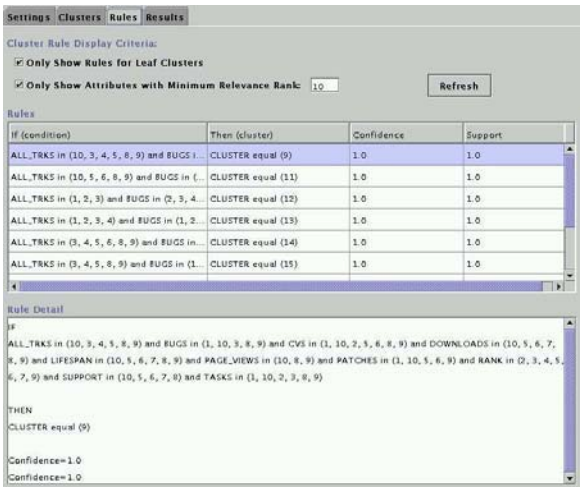


Fig. 5. The rules that define clusters

minimum confidence. we find that the feature "all trks", "cvs" and "downloads" are "associated". More rules can be seen from Figure 5. Not all of the rules discovered are of interest.

5.2.3. Clustering

We are interested in putting the projects with similar features together to form clusters. Two algorithms can be used to accomplish this: k-means and o-cluster. The k-means algorithm is a distance-based clustering algorithm, which partitions the data into predefined number of clusters. The o-cluster algorithm is a hierarchical and grid-based algorithm. The resulting clusters define dense areas in the attribute space. The dimension of the attribute space is the number of attributes involved in the clustering algorithm.

We apply the two clustering algorithms to projects in this case study. Figure 4 and Figure 5 shows the resulting clusters and the rules that define the clusters.

6. CONCLUSION AND FUTURE WORK

This paper discusses a framework for web mining research support system and describes its procedures. It then discusses implementing techniques on web data extraction and analysis. A sourceforge web mining case is presented as an example of how to apply this framework.

This work is an exploratory study of web data retrieval and data mining on web data. We try to evaluate the data extraction process and data mining software which can be used to discover knowledge in the web data. The actual interesting discoveries are still in progress. We are expected to discover interesting patterns from the data.

This research was partially supported by the US National Science Foundation, CISE/IIS-Digital Science and Technology, under Grant No. 0222829.

7. REFERENCES

[1] R. Cooley, B. Mobasher, and J. Srivastava. Web mining: Information and pattern discovery on the world wide web. In *International Conference on Tools with Artificial Intelligence*, pages 558–567, Newpot Beach, 1997.

[2] Francis Crimmins. Web crawler review. http://dev.funnelback.com/crawler-review.html, 2001.

[3] Yao J.T. and Yao Y.Y. Web-based information retrieval support systems: building research tools for scientists in the new information age. In *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, Halifax, Canada, 2003.

[4] M. Koster. The web robots pages. http://info.webcrawler.com/mak/projects/robots/robots.html, 1999.

[5] Bamshad Mobasher, Honghua Dai, Tao Luo, Yuqing Sun, and Jiang Zhu. Integrating web usage and content mining for more effective personalization. In *EC-Web*, pages 165–176, 2000.

[6] Yao Y.Y. Information retrieval support systems. In *FUZZ-IEEE'02 in The 2002 IEEE World Congress on Computational Intelligence*, Honolulu, Hawaii, USA, 2002.

[7] Yao Y.Y. A framework for web-based research support systems. In *Computer Software and Application Conference, (COMPOSAC 2003)*, Dallas, Texas, 2003.

[8] Osmar R. Zaïane, Man Xin, and Jiawei Han. Discovering web access patterns and trends by applying OLAP and data mining technology on web logs. In *Advances in Digital Libraries*, pages 19–29, 1998.

Web-based Learning Support Systems

Lisa Fan and Yiyu Yao,
Department of Computer Science, University of Regina, SK, Canada, S4S 0A2
{fan,yyao}@cs.uregina.ca

Abstract

With the rapid development of the Internet and advances in multimedia technologies, web-based educational systems are becoming popular. Due to different types of learners using these systems, it is necessary to provide them with an individualized learning support system. A framework of web-based learning support system (WLSS) is presented by focusing on learning process and activities, as well as the technology support needed. Based on the learner-centered mode, we demonstrate an online course design and development that supports the students with the flexibility and the adaptability. We present an approach on the use of check point analysis mechanism which guide them to the relevant learning materials in order to achieve their learning goals effectively.

1. Introduction

Web-based educational systems are becoming more and more popular and are used for teaching and learning. However, most systems are still limited to dissemination of teaching materials [1]. The learning process is more complex than navigating between different static pages and reading them. There is a need for mechanisms that modify the navigation alternatives by some sort of adaptation, so that the students can be guided to achieve their learning goals.

Not all students have the same ability and skills to learn a subject. Students may have different background knowledge for a subject, which may affect their learning. Some students need more explanations than others. Other differences among students related to personal features such as age, interests, preferences, etc. may also affect their learning [1]. Moreover, the results of each student's work during the learning session must be taken into account in order to select the next study topics to the student [2].

It is very important to design learning support systems in order to maximize the strength of the WWW and fully utilize the functions that support interactive, personalized and collaborative learning.

In this paper, we first briefly discuss the concept of web-based learning support systems (WLSS) and examine the main functions and characteristics of such systems. Based on the general guideline, we report our experience in the design and implementation of a web-based learning support system for teaching an undergraduate course.

2. Learning and Learning Support Systems

A definition of learning is given by Gagne : [3]

"a process of which man and the animals are capable. It typically involves interaction with the external environment (or with a representation of this interaction, stored in the learner's memory). Learning is inferred when a change or modification in behavior occurs that persists over relatively long periods during the life of the individual."

Learning is an interactive, dynamic and active feedback process with imagination driving action in exploring and interacting with an external environment [4]. There are two main learning styles: group learning and individual learning. Group learning is used in the traditional classroom learning. Teacher and students communicate in real time manner. This is the teacher-centered form of education. Feedback is the two-way communication between teacher and students. It requires high operational cost. Individual learning is student-centered form of education. In this learning style, learners study the material individually, and the teacher acts as a supporter, such as web education. This learning style provides personal flexibility with low cost.

2.1 Web-based Instruction

The major difference between the web-based and conventional instruction system is that students can choose their own paces for learning. They can skip those materials that they have already learned or known. They can replay the course that they were not thoroughly understood. However, most of web-based courses are not as "flexible" as human instructor [5]. Typically, course material is a network of static hypertext pages with some media enhancement. Neither the teacher nor the delivery system can adapt the course presentation to different students. As a result, some students waste their time

learning irrelevant or already known material, and some students fail to understand (or misunderstand) the material and consequently overload a distance teacher with multiple questions and requests for additional information. Therefore, the web-based system needs to overcome the deficiencies of inflexible instruction from conventional face-to-face group learning, and potential inflexibility from not having face-to-face feedback from students to instructors.

2.2 Characteristics of Web-based Learning Support Systems

Building a web-based learning support system is relatively easy from the technical point of view. However, analyzing, designing and implementing the system to achieve better teaching and learning result is a difficult process. The system should consider the following features.

- Complexity of learning support:

Obtaining knowledge means going through a process of learning. The learning process is complex. Many human senses interact and collaborate. Already obtained knowledge and experiences are used to prove and verify the new cognition [5]. Discussions are used for information exchange, and examples help to strengthen and solidify skills. Different learning styles complicate the situation. Without taking these into consideration, the knowledge is often presented in a fixed manner. Neither textbooks nor online texts can actually answer questions. The student is provided with only information.

- Individuality and adaptability support:

Individuality means that a WLSS must adapt itself to the ability and skill level of individual student. Adaptive methods and techniques in learning have been introduced and evaluated since the 1950's in the area of adaptive instruction and the psychology of learning [6]. Adaptive instructional methods adapted the content of the instruction, the sequencing of learning units, the difficulty of units, and other instructional parameters to the students' knowledge. These methods have been empirically evaluated and shown to increase learning speed and to help students gain a better understanding through individualized instruction.

According to Brusilovsky [3], there are several goals that can be achieved with adaptive navigation support techniques, though they are not clearly distinct. Most of the existing adaptive systems use link hiding or link annotation in order to provide adaptive navigation support. Link hiding is currently the most frequently used technique for adaptive navigation support. The idea is to restrict the navigation space by hiding links that do not lead to "relevant" pages, i.e., not related to the user's current goal or not ready to be seen. Users with different goals and knowledge may be interested in different pieces of information and may use different links for navigation. Irrelevant information and links just overload their working memories and screen [3].

De Bra [7] presented a course that uses a system they developed to track student progress and based on that, generate document and link structure adapted to each particular student. Links to nodes that are no longer relevant/necessary or links to information that the student is not yet ready to access are either physically removed or displayed as normal text.

Da Silva et al [8] use typed and weighted links to link concepts to documents and to other concepts The student's knowledge of each concept is used to guide him/her towards the appropriate documents.

- Interaction support

The web-based learning support system must be interactive. Students must be able to communicate with the system. Users should be able to add personalized annotations and notes to the prepared knowledge base. It should allow the students asking questions and automatically retrieving a proper answer. WBIRSS (Web Based Information Retrieval Support System) may be a useful solution to this problem [9]. Discussion group is an important feature of the support system to improve the learning efficiency.

- Activity and assessment support

One of the most difficult challenges of web-based learning mechanism is the assessment of students' learning process. It is hard to judge the behavior of a student since the instructor is separated spatially from the students. Testing and check points are important from the point of view of evaluating and assessing the student progress [10].

Examples and exercises are used to strengthen the students understanding through practice. The system should provide the students the possibility not only to look at the examples, but also be able to modify them, try them out and get feedback on their solutions.

3. Implementation of a WLSS Using WebCT

The design of on-line courses involves different actors. These actors have different requirements. The teacher needs a tool easy to use in order to create the educational

material. The student needs something more than a mere transfer of a book in electronic format. They need some sort of guidance, a high level of interactivity and a tool for accessing the learning process. The site administrator, finally, needs an easy to maintain system for updating the content and the information about the users.

WebCT (Web Course Tools) is a web-based instructional delivery tool, which facilitates the construction of adaptive learning environments for the web. It enables instructors to create and customize their courses for distance education [11]. It provides a set of educational tools to facilitate learning, communication, and collaboration.

The system can facilitates active learning and handles diverse learning styles. It allows the instructor to present his/her course materials using a variety of mediums to make the course dynamic and interactive. By creating a module of selected material and having a quiz at the end, the instructor can selectively release course materials based on the quiz score from the previous module. The student tracking within the Content Module allows the instructor to observe individual student activity. This can be used for online grading based on participation.

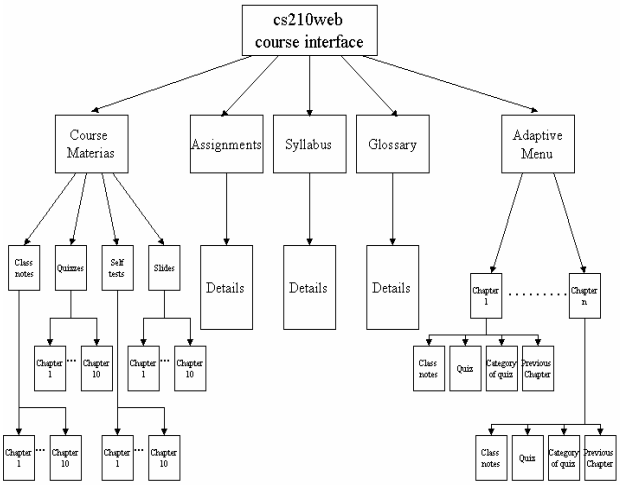


Figure 1: The design structure for the course

The WebCT system uses a client/server architecture. This model is based on the distribution of functions between two types of independent and autonomous processes: client and server. The system consists of WebCT software and a number of hierarchically organized files for each course. The user accesses the data on the server through a web browser. All the WebCT software resides on and runs off a server, which means any changes made to courses are accessible to students immediately after the change is made. The system provides an environment to cover all aspects of a course such as tools for creating the course materials, lectures, assignments, quizzes, and discussion groups.

Using WebCT, a core computer science course "data structure and algorithm analysis" has been designed. In order to provide an in depth understanding of the fundamental data structures and to motivate the students, a web-based adaptive course with analysis tool based on student learning styles has been proposed. The course has been designed with adaptability and it has been taken into considerations for student individual learning styles. The web learning course structure can be better understood through Fig. 1. One of the screen shot of the course material is shown in Fig. 2.

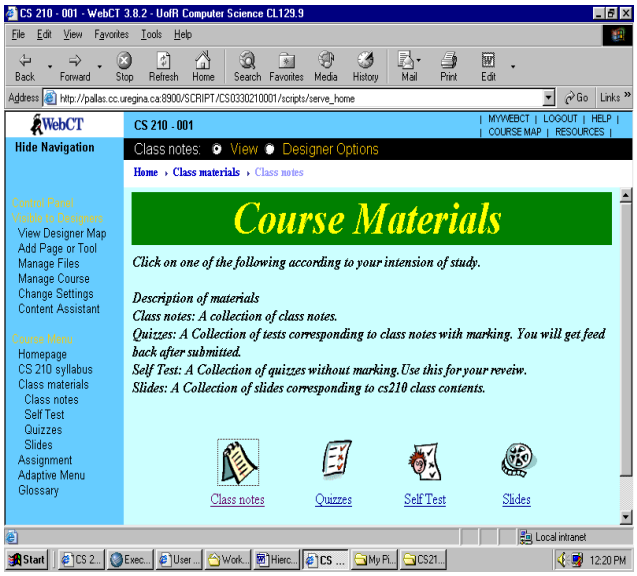


Figure 2: Screen shot of the web page of course materials

In the following sections the structure of the web course and how this structure contributes to our approach to provide adaptation will be presented.

The students' goal of taking the course is to learn all or most of the course materials. However, the learning goal can be differentiated among different students according to both their expectations with respect to the course and their knowledge about the subject being taught, the latter

being the most important user feature in web-based educational system [3].

Our approach utilizes the adaptive navigation support system in WebCT. At the current stage, the structure of the course is organized around chapters that consist of several concepts. The contents of the course is designed and implemented as a collection of modules (under Content Module). Each module has well defined learning objectives. Figure 3 shows the adaptive menu of the web course material.

At the end of each chapter, the web course provides a quiz with automatic feedback. The quiz consists of a set of questions with predefined answers, and the result obtained by the student will have influences over the presentation of the subsequent course contents. The web course system allows the students to test his/her level of knowledge and understanding of the concepts and also permits the dynamical paths among the contents of the course.

Figure 3: Screen shot of the demonstration of the adaptive course material

For example, as showed in Figure 3, if the student complete the Chapter One, he need to take the quiz, which will determine whether or not he acquired the minimum level of knowledge required to moving forward to the next stage of learning. This process will identify the next chapter to be presented. Therefore the following subsequent chapters are set up as conditional.

These chapters are only presented when the student satisfy the basic requirement. In the case the student has failed the quiz, the student will be recommended to go back and study the same content one more time. The questions contained in the quiz are designed to determine whether or not the student has acquired the required level of knowledge for the chapter. The difficulty levels of the individual question associated to the content have been taken into account. Hence each question has different weighting towards the total grade of the quiz.

4. Check Point Analysis

The notion central to our implementation of the web-based support system is check point analysis. We divide the learning process into many stages. The success of a student is checked at many points.

Several software systems have been designed with online testing [12][13]. Most of these systems provide automatic grading for quizzes with multiple choice and true/false questions. WebCT has been designed to perform online testing of answers whose correct syntax can be specified as regular expressions. The online grading system provides limited feedbacks towards the students learning problems with the course materials. As mentioned earlier, if the student has failed the quiz, he has to go back and study the material again. However, these systems do not provide the specifics on which section he needs to spend more time to study and which concepts he has difficulty to understand.

A primary concern for web-based learning support system is the design of an appropriate structure so that a student can easily and naturally find the most relevant information depending on his/her needs.

Table 1. The relationships between questions and concepts for algorithm analysis module

	C_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
q_1	x							
q_2		x	x	x				
q_3		x						
q_4			x					
q_5				x				
q_6					x	x		
q_7					x			
q_8					x			
q_9							x	
q_{10}								x

Table 2. The description of concepts and the related questions

	Concept Description	Associated questions
c_1	Big O	1
c_2	Empirical method	2,3
c_3	Simulation method	2,4
c_4	Analytical method	2,5
c_5	Complicated computer model	6,7,8
c_6	Simplified computer model	7
c_7	Recursive functions	9
c_8	Repeated substitution	10

Table 3. The questions and the related concepts

Questions	Associated Concepts
q_1	c_1
q_2	c_2,c_3,c_4
q_3	c_2
q_4	c_3
q_5	c_4
q_6	c_5,c_6
q_7	c_5
q_8	c_5
q_9	c_7
q_{10}	c_8

In order to aid the students to improve their study effectiveness and efficiency, a check point analysis mechanism is adopted in the online course design. As demonstrated in Tables 1, 2 and 3, the students will be provided with these three tables in each module that clearly indicate the relationships between the questions and the related concepts in the course materials.

For example, Algorithm Analysis module, in Table 1, question q_6 is related to concepts c_5 and c_6. In Table 2, concept 5 is related to "complicated computer model", concept 6 is associated with "simplified computer model"; Table 3 lists all the questions that related to concept 6.

If a student has failed the quiz, and the most marks have been deducted from the question 6, he can look up the tables to analyze his weak point. First he would find out that the question 6 is related to concept 5 and 6, which is related to "complicated computer model" and "simplified computer model". This tells the student that he needs to spend more time on these two concepts related sections in the course contents. If the student has more than one question wrong, the table can show him whether these questions are fall into the same category of concepts or different ones. Therefore students can select the relevant part of the course materials to study.

These tables also provide very important information about the course and student learning to the instructors too. The instructor will be able to use the error rate to redesign the course material and analyze the students learning in order to provide relevant support.

Figure 4 shows the example of Object Oriented Design module with the check point analysis.

Figure 4: Screen shot of Object Oriented Design module

5. Conclusions

The introduction of the web offers new opportunities and challenges for educators. It opens a new door for computer-aided instruction. Web-based learning support systems (WLSS) are designed based on the student-centered philosophy. Such systems assist students in every stage of learning.

The basic features and ideas of WLSS are discussed in this paper. To illustrate those ideas, we implemented a learning support system for helping the students' study of the data structure and algorithm analysis course.

Our primary goal is to provide an adaptive learning support environment that will effectively accommodate a

wide variety of students with different skills, background, and learning styles.

A model for check point analysis has been presented. The system allows the students to analyze their learning progress, and guides them towards the relevant course contents to achieve their learning goals during their learning process. The model also provides important information about the course and student learning to the instructor too. The instructor will able to use the error rate to redesign the course material and analyze the students learning in order to provide relevant support.

Neural network and data mining techniques may be applied in our future work to analyze the students learning patterns, and provide the instructors or designers to organize the online course more effectively.

6. References

[1] Nill, A., "Providing Useable and Useful Information by Adaptability", GMD – German National Research Center for Information Technology, Sankt Augustin, Germany, Http://zeus.gmd.de/~nill/flexht97.html/

[2] Brusilovsky, P., Anderson, J., "An adaptive System for Learning Cognitive Psychology on the Web", WebNet 98 World Conference of the WWW, Internet & Intranet, Orlando, Florida, November 7-12, 1998, pp.92-97.

[3] Gagne, R.M., Driscoll, M. P., "Essentials of Learning for Instruction", New Jersey, Prentice Hall, 1998.

[4] Papandreou, C.A., Adamopoulos, D.X., "Modelling a multimedia communication system for education and training", Computer Communications 21, 1998, pp.584-589.

[5] Brusilovsky, P., "Methods and Techniques of Adaptive Hypermedia", User Modelling and User Adapted Interaction, Vol. 6, N2-3, 1996, pp. 87-129

[6] Tennyson, R.D., Rothen, W., "Pre-task and On-task adaptive design strategies for selecting number of instances in concept acquisition". Journal of Educational Psychology, Volume 69, 1977, pp.586-592.

[7] De Bra, P., "Teaching Through Adaptive Hypertext on the WWW." *International Journal of Educational Telecommunications.* August 1997, pp. 163-179

[8] Da Silva, D.P., Van Durm, R., Duval, E. & Olivi, H., "Concepts and Documents for Adaptive Educational Hypermedia: a Model and a Prototype", Second workshop on Adaptive Hypertext and Hypermedia, Ninth ACM Conference on Hypertext and Hypermedia, Pittsburgh, USA, June 20-24, 1998, pp.35-43.

[9] Yao, J.T., Yao, Y.Y., "Web-based Information Retrieval Support Systems: building research tools for scientists in the new information age", Proceedings of the IEEE/WIC International Conference on Web Intelligence, 2003.

[10] Wade., V. P., Power, C., "Evaluating the Design and Delivery of WWW Based Educational Environments and Courseware", ITICSE'98, Dublin, Ireland, 1998.

[11] Getting Started Tutorial for WebCT. http://www.webct.com/

[12] Mason, D., Woit, D., "Integrating Technology into Computer Science Examinations." Proceeding of 29th SIGCSE., 1998, pp.140-144

[13] Mason, D., Woit, D., "Providing Mark-up and Feedback to Students with Online Marking", Proceeding of 30th SIGCSE, 1999, pp.3-6.

Developing an Intelligent Web-Based Thai Tutor: Some Issues in the Temporal Expert

Rattana Wetprasit
Department of Mathematics and Computer,
Prince of Songkla University, Phuket campus,
Vichitsongkram Rd., Kathu, Phuket 83120 Thailand.
Email: rattana@phuket.psu.ac.th

Abstract

Computer aided language learning system is an attractive application of Artificial Intelligence in Education. This paper presents details of a web-based Thai language tutoring system. The system, called Thai Tutor (TT), assists students who are learning Thai as a second language. Since temporal information constitutes an important part to the meaning of a sentence, we concentrate on a module of the system called the temporal expert. This domain dependent expert is devoted to represent and reason the temporal knowledge by using Allen's interval-based temporal logic. The temporal expert provides information about the order of events so as to convey the intended meaning of the sentence.

Keywords: intelligent language tutoring system, web-based language instruction, temporal reasoning

1. Introduction

Capability in language learning is individual. This may depend on his/her first language, culture, talent, and teaching environment. Effective learning method needs a self-controlled system that allows students to practice and revise the lessons on demand. Classical lecture conveys all subjects' contents as scheduled in the teaching plan. In contrast, tutors, who normally accommodate only a small group of students, can follow the progress of each student through assignments and revision exercises as needed. Specifically, one-to-one human tutoring is widely accepted as a well-understood way of communicating knowledge. It allows learning to be highly individualized and consistently yields better outcomes than other methods of teaching [2]. However, one-to-one tutoring is limited by the number of qualified tutors as well as their availability.

An intelligent tutoring system (ITS) is a program capable of providing students with tutorial guidance in a given subject. The program mimics the way one-to-one tutors conduct their classes. Lessons and practices are dynamically chosen according to the student's language proficiency and progress. A full fledged ITS: a) has specific domain expertise; b) is capable of modeling the student's knowledge in order to discover the reason(s) of his mistakes; and c) is able to make teaching more effective by applying different tutorial strategies [4]. Records and updated histories of individual students are kept in *Student Model*. These details assist in the determination of the next activity for that student. The evaluation process of the student, in order to designate the proper activity, is performed in the *Tutorial Module*. Using the expert system concept, the ITS stores the various teaching methodologies on how to convey subject knowledge to students with diverse background and ability. It allows the overall system to demonstrate or model a correct way of teaching the subject. Both the *Student Model* and *Tutorial Module* will refer to knowledge about ideal student actions in the expert system.

Our preliminary paper proposed in [12] investigates the *Thai Tutor* (TT) system architecture, various learning behavior, and basic Thai lessons structure. Here, we extend the system's ability to train students about tenses by using Allen's interval-based temporal logic.

2. Characteristics of Thai language

2.1 Sentence structure

Thai grammar refers mostly to word order and the use of words like "*dai*" and "*laeo*", so called *function words*. These words have basic meanings related to time and action, which alter tenses or give phrases and sentences different shades of meaning. Verbs in Thai sentences consist only their root form. There is no transformation according to tenses, number, gender of subjects, or subjective mood. The expression of different situation is done by adding to the sentence different wordings with the desired meaning.

Thai words are often interchangeable as parts of speech. Moreover, some of the words in a sentence may be optional. All Thai sentences have a complete form, where words may be omitted depending on the context and the formality of the situation [7]. In general, the more wordings a sentence has, the more formal it sounds. This interchangeability and flexibility of the language increase learning difficulties, especially to learners who are more familiar with languages where tenses are intentional.

2.2 Expression of temporal information in Thai

An important part of a sentence meaning is constituted by temporal information. The information implied by tenses indicates the order of events or situations. An event can be either time interval or instance of time in which the situation described in the sentence takes place and the moment in which the sentence is uttered. The relationship between events described the series of events constituting a scenario. Even though, verbs in Thai sentences are not modified according to tenses as mentioned earlier, using proper functional words or adverbs of time are essential for communicating the right meaning. For example, consider the sentence "I bought some food when I went to the market." The action "I bought some food" happened sometime when I was at the market. In Thai, we can express this two-clause sentence as:

ฉัน*ซื้ออาหาร***ขณะที่**ฉัน*ไป*ตลาด or

ฉัน*ซื้ออาหาร***ตอนที่**ฉัน*ไป*ตลาด

The *italic* words indicate verbs of each clause, while the **bold** words specify the conjunctions having the same meaning "while".

However, consider

ฉัน*ซื้ออาหาร***หลังจากที่**ฉัน*ไป*ตลาด

By changing the conjunction, the meaning is changed to "I bought some food after I went to the market", in other words, the first clause happened after the second. Note that the verbs in each clause for all the sentences remain the same (no modification according to tenses). Each sentence uses the bold wording to specify the tense and order of the clauses.

3. System architecture

3.1 The web-based *Thai Tutor* system

As web technologies for adapting existing educational content converge with increased bandwidth, web-based ITSs with integrated multimedia are increasingly expected to have higher degrees of adaptiveness and interactivity [6]. The diversity of users presents a challenge in system design. It needs to be general enough to suit a variety of competencies, and backgrounds. For example, when analyzing the mistakes from completed assignments, the intelligent language tutor has to do it based on the native language of the student [13] at least. An online system must adopt a more generic scheme in order to accommodate international access and cases where the native language of the user might not be known. However, system performance and efficiency are also an important issue. A visible delay between the browser and the servers defeats the purpose of an interactive system, and can distract the problem solving activity [3]. The *Thai Tutor* we are developing resides on a Linux machine and uses English as the communication tool.

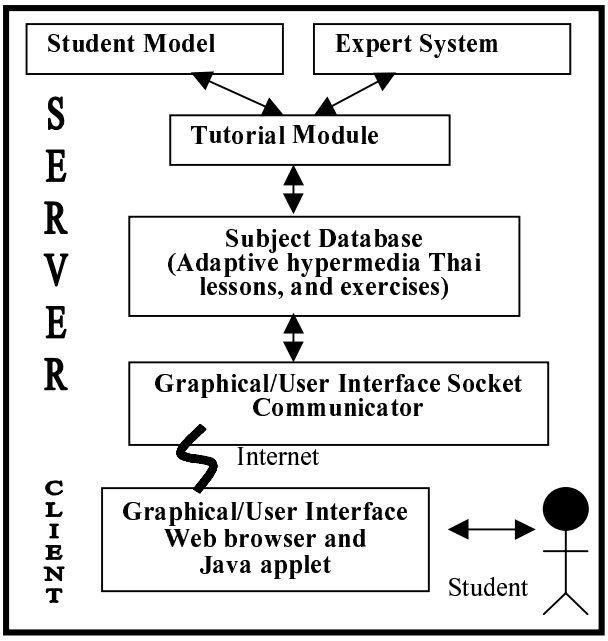


Figure 2. Web-based Thai tutoring system

Displaying Thai characters is a big issue when implementing a digital system. German and French, for example, employ English alphabets and some special symbols in writing, but this is not the case in Thai. Chinese writings, like Thai in some aspects, also have their own characters instead of using English alphabets to describe the languages. However, writing Thai is more complicated as compared to others. With one Thai character, there can be three levels of symbols located above and below it. Therefore, to provide the international access without any modification on the client side, Thai words are stored as image files. In this case, the system performance will be acceptable as the major display will be in English along with other adaptive multimedia files, such as sound and animation. Architecture of the *Thai Tutor* is then shown in Figure 2.

3.2 *Student Model*

In the literature, a number of student modeling techniques have been employed for ITS. The *Student Model* functions as an accumulative and adaptive database for each user. Therefore, a challenging task in implementing multi-user system is to identify users while maintaining user models adaptable to the individual. A number of solutions have been proposed which range from cookies [10], structured URLs [3], and hidden fields to login screens [13]. Deciding on an appropriate alternative depends primarily on the purpose of the application.

Our *Thai Tutor* system stores a database of users, each entry established by an initial login. The student login and password are used to identify a user. This is sufficient since students do not navigate through different HTML pages during learning, but can access a consistent applet. *Student Model* requires the user's identification for two main functions: a) to store scores across a number of error types, or nodes, such as pronunciation, vocabulary, punctuation, etc. Each node is broken down into more fine-grained categories. Dealing with pronunciation learning, the error will be categorized into various word tones of low, medium, and high sound characters. For example, a medium sound consonant when forming a word can have five tones. Thus, the student's error can be finely recognized as a unique tone to the medium sound word. This information is shown to the user at the end of each exercise set; b) to keep and update history of the learners. Depending on student's input, the score for each node will go up or down. These data will be used to adjust the lesson, emphasis of an exercise, etc. These two functions together allow the system to perform a fine-tuned assessment of student competency. Thus a single-error will not drastically change the student's overall assessment.

4. Temporal expert

4.1 Knowledge representation

Our Thai tutor needs an ability to generate and exercise the correct tense for each sentence. Since there is no tense transformation to the verb, TT system concentrates only on how to choose the appropriate function words, adverbs of time, and conjunctions, and also the degree of politeness. Based on Allen's temporal knowledge representation paradigm [1], we consider each primitive event as an interval of time. While a time interval can be an event, an activity or a situation.

The representation of a sentence starts with the sentence structure, which may consist of one or more clauses. The task to be solved is indicated in "Part_to_solve", e.g., finding proper function words, adverbs of time, and conjunctions. Information about each clause is schematically represented by a number of attributes that are:

- *Clause_type*: role of the clause i.e., main, co-ordinate, or subordinate clause;
- *Clause_form*: intention of the clause i.e., narration, question, negation, request, demand;
- *Relation*: a set of sub-attributes (a related interval, possible relation 1, possible relation 2, …, possible relation *n*);
- *Formality*: degree of formality to which situation the clause applies.

A possible relation between two intervals is a disjunction of the thirteen primitive relations proposed by Allen [1]. They are before (*b*), equal (*e*), meet (*m*), overlap (*o*), during (*d*), start (*s*), finish (*f*), and their inverses which indicate by an "*i*" after the relation. These relations can be graphically shown in Table 1. The disjunction of these primitive relations expressively represents ambiguity between two events when the starting and finishing points of the events cannot be clearly stated.

Relation	Symbol	Symbol for Inverse	Pictorial Example
X before Y	b	a	XXX YYY
X equal Y	e	e	XXX YYY
X meets Y	m	mi	XXXYYY
X overlaps Y	o	oi	XXX YYY
X during Y	d	di	XXX YYYYYY
X starts Y	s	si	XXX YYYYY
X finishes Y	f	fi	XXX YYYYY

Table 1. Allen's thirteen interval relations.

4.2 Temporal reasoning

In this subsection, we introduce two examples of temporal reasoning in Thai tutor. The first example shows a simple sentence when the relation between two clauses implies the proper conjunction. The second example introduces the reasoning process when there is ambiguity between three events.

Example 1: Suppose the system proposes an exercise to students in order to find a proper conjunction indicating the order of two events ("I bought some food" and "I went to the market") as follows:

ฉันซื้ออาหาร ... (conjunction) ... ฉันไปตลาด

Here the meaning we intend to train students is "I bought some food when I went to the market". The schematic description of this sentence is the following:

Sentence: S1
  Structure: C1, C2
  Part_to_solve: conjunction
Clause: C1
  Clause_type: main
  Clause_form: narrative
  Relation: (C2, *s, f, d, e*)
  Formality: 2
  Part_to_solve: none
Clause: C2
  Clause_type: subordinate
  Clause_form: narrative
  Relation: (C1, *si, fi, di, e*)
  Formality: 2
  Part_to_solve: none

Typically, a Thai sentence can be applied with more than one function word to make the sentence sound more formal and polite. This depends on the context of the sentence. Here we represent the degree of formality as an integer in order to demonstrate the natural conversation. From the intended meaning of sentence S1, we can imply that

- The starting of the interval C1 ("I bought some food") was during the time period when the clause C2 ("I went to the market") was carrying on;
- The event C1 terminated before the ending of the event C2.

Therefore, the clause "I bought some food" could start, finish, during, or happen at the same time (equal) as the clause "I went to the market". When TT verifies the answer from learners, the temporal expert matches the relation between two clauses with a conjunction table to obtain the proper conjunction.

This exercise can be simply resolved by table look up techniques. However, the significant part of this task is the construction of the table. Each entry of the table contains a Thai word (e.g., conjunctions, function words, adverbs of time, etc.) that matches the corresponding situation (e.g., tense, formality, etc.).

■

In real situation, a conversation may refer to more than two events. In such cases, the temporal relationship between any pair of events may be unknown. In some cases, the system may have complete information about how the events could be related. But when new temporal information is entered, all relations will have to be revised to maintain the consistent knowledge of the overall scenario. To generate or verify an appropriate tense for a given clause or sentence, we need a reasoning process that infers the unknown relations or eliminates the inapplicable relations.

Example 2: Suppose we further know that "A friend came to my house after I left home for the market". This knowledge helps us to infer the relationship between the events "A friend came to my house" and "I went to the market" as the disjunction of relations "after (*a*)", "met by (*mi*)", "overlapped by (*oi*)", "during (*d*)", and "finish (*fi*)". However, the temporal relation between events "I bought some food" and "A friend came to my house" remains unknown. The schematic representation of the additional knowledge can be shown as follows:

Sentence: S2
  Structure: C3, C2
  Part_to_solve: none
Clause: C3
  Clause_type: main
  Clause_form: narrative
  Relation: (C2, *a, mi, oi, d, f*)
  Formality: 2
  Part_to_solve: none
Clause: C2
  Clause_type: subordinate
  Clause_form: narrative
  Relation: (C3, *b, m, o, di, fi*)
  Formality: 2
  Part_to_solve: none

The reasoning task here is to generate the unknown relation between clause C1 and C3. The new fact (the relation between C3 and C2) adds a constraint about how the two events could be related. This may in turn introduce new constraints between other events. In this scenario, there are only three events and the relation between C1 and C3 is not predefined. The consequence of the added knowledge will identify the relation between C1 and C3. To achieve the task, we adopted Allen's temporal reasoning algorithm called *Constraints (R1, R2)*, where *R1* is the disjunctive relation between clauses C1 and C2, and *R2* is the disjunctive relation between clauses C2 and C3. This algorithm was later modified to the so-called Path Consistency algorithm [8].

```
Constraints (R1, R2)
    C ← ξ
    For each r1 in R1
        For each r2 in R2
            C ← C ∪ T (r1, r2)
    Return C
```

U is the mathematical union operation. T(*r1, r2*) is the transitivity function describing the inferred relations from primitive relation *r1* to *r2*. For instance, if interval *i* is during interval *j*, and the fact that interval *j* happens before interval *k* is added, then the transitivity function infers that interval *i* must be before interval *k*.

In our scenario, *R1* is the set {*s, f, d, e*} and *R2* is the set {*b, m, o, di, fi*}. After the reasoning process, the relation between clauses C1 and C3 may be any of the possible thirteen relations. Therefore, we cannot strictly

specify the order of the two events (C1 and C3). In other words, all conjunctions are possible to conjugate the clauses. Until more new knowledge about temporal relation is provided, relation between the intervals can be further restricted.

■

5. Related works

Several ITSs for teaching languages have been proposed. A system for teaching English as a second language presented by Fum, et al., concentrated on generating in a cognitively transparent way, the right tense for the verb(s) appearing in exercises [4]. A related work by Fum, et al., [5] focused on the relationships between naive grammar (knowledge derived from textbooks and school grammars), and formal grammar (developed by theoretical and computational linguists). Another ITS, called *Tutor Assistant*, is designed to be an authoring tool for English language instructors to create their own lessons and exercises [11]. This study also evaluated the degree to which instructors can author good quality content for an *English Tutor* and established benchmarks for development times. ALICE uses Natural Language Processing (NLP) as a basis both for assisting instructors in preparing exercises and for evaluating student responses [9]. Finally, a *German Tutor* was developed by Heift and Nicholson [6]. This attempted to implement generality, interactivity and modularity into the system with an emphasis on efficient and adaptive hypermedia.

As for the Thai learning system, there are several Thai courseware either on a stand-alone machine, or publicly on the Internet. However, they are conventional Computer Assisted Instruction (CAI) setups.

6. Conclusion and ongoing research

Verbs in Thai sentences are not modified according to the chronological order of events. They need additional words (e.g., adverb of time, function words, conjunctions, etc.) to express the tenses. This paper presented a web-based ITS for teaching Thai as a second language, called *Thai Tutor*. The system is able to handle the temporal information when tenses are involved. A module, called *Temporal Expert,* is devoted to represent and reason the temporal knowledge by using Allen's interval-based temporal logic. The temporal expert provides information about the order of events so as to convey the intended meaning of the sentence. The long-term project for a fruitful *Thai Tutor* requires developing the ability to recognize and diagnose learners' pronunciation when speeches are emphasized. Unfortunately, current technology does not yet practically support this requirement.

7. References

[1] Allen, F.J., 1983. Maintaining Knowledge about Temporal Intervals. In Brachman, R., and Levesque, H., (eds). *Readings in Knowledge Representation*. San Mateo: Morgan Kaufman, pp. 510-521.

[2] Bloom, B.S., 1984. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring, *Educational Researcher*, 13, 6, June/July.

[3] Eliot, C., 1997. Implementing Web-Based Intelligent Tutors. In *Proceedings of the Workshop "Adaptive Systems and User Modeling on the World Wide Web", 6th International Conference on User Modeling*, Chia Laguna, Sardinia.

[4] Fum, D., Giangrandi, P. and Tasso, C., 1989. Tense Generation in an Intelligent Tutor for Foreign Language Teaching: Some Issues in the Design of the Verb Expert. In *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics,* Manchester, pp. 124-129.

[5] Fum, D., Pani, B. and Tasso, C., 1992. Naive vs. Formal Grammars: A case for integration in the design of a foreign language tutor. In: M.L. Swartz and M. Yazdani (Eds.) *Intelligent Tutoring Systems for Foreign Language Learning*. Berlin, Springer, pp. 51-64.

[6] Heift, T., and Nicholson, D., 2000. Theoretical and Practical Consideration for Web-based Intelligent Language Tutoring Systems. In *Proceedings of the 5th International Conference, ITS 2000,* Montreal, Canada, pp.354-362.

[7] Higbie, J., and Thinsan, S., 2002. *Thai Reference Grammar: The structure of spoken Thai*. Orchid Press, Bangkok.

[8] Ladkin, P.B., and Maddux, R.D., 1994. On Binary Constraint Problems. *Journal of the Association for Computing Machinery*. Vol. 41, No. 3, pp. 435-469.

[9] Leisibach, T.B., 2001. I-CALL and Second Language Acquisition (SLA). *Applications of Computational Linguistics*. .

[10] Stern, M., 1997. The Difficulties in Web-based Tutoring, and Some Possible Solutions. In *Proceedings of the Workshop "Intelligent Educational Systems on the World Wide Web", 8th Conference of the AIED-Society*, Kobe, Japan.

[11] Toole, J. and Heift, T., 2002. The Tutor Assistant: An Authoring System for a Web-based Intelligent Language Tutor. *Computer Assisted Language Learning*, 15, (4): pp. 373-386.

[12] Wetprasit, R., 2003. An Intelligent Tutoring System for Teaching Thai as a Second Language. In *Proceedings of the third International Symposium on Communications and Information Technologies (ISCIT2003),* Thailand.

[13] Yang, J. and Akahori, K., 1997. Development of Computer Assisted Language Learning System for Japanese Writing Using Natural Language Processing Techniques: A Study on Passive Voice. In *Proceedings of the Workshop "Intelligent Educational Systems on the World Wide Web", 8th Conference of the AIED-Society*, Kobe, Japan.

A Framework for Adaptive Educational Hypermedia System

José M Parente de Oliveira
Clovis Torres Fernandes
Computer Science Division, Technological Institute of Aeronautics
Brazil
{parente, clovis}@comp.ita.br

Abstract

Adaptive Educational Hypermedia Systems (AEHS) have been used to support customized learning. The adaptation mechanisms provided usually try to define the better concept sequence to be presented and to select the materials and activities more appropriate for a given learner. Nevertheless, despite the primary purpose of AEHS in supporting learning, adaptation mechanisms in these systems have no compromise in using an instructional design theory as source of information. This paper addresses a way in which an instructional design or learning theory is used in conjunction with learner's domain knowledge, background, preferences and learning style as source of information for adaptation. To make that viable, a framework for AEHS was defined. The paper describes the framework's features and its potential benefits.

1. Introduction

Adaptive Educational Hypermedia Systems (AEHS) are a kind of Web-Based Educational Systems that tries to provide a customized interaction for learners, in the form of content and navigation adaptations [1, 2]. To provide adaptation, AEHS normally use learner's domain knowledge, background, and preferences as reference. Another source of information for adaptation found in the literature is learner's cognitive or learning styles [3, 4, 5, 6].

Adaptation based on these kinds of information basically tries to define the better concept sequence to be presented and to select the materials and activities more appropriate for a given learner. That is, from a pre-defined curriculum the system defines the better concept sequence and the appropriate materials to be presented.

Despite the primary purpose of AEHS in supporting learning, adaptation in these systems has no compromise in using an instructional design theory. That means it is not common the use of instructional design or learning theories as source of information for adaptation in AEHS. Due to that, it was hypothesized that the use of instructional design or learning theories could be used in conjunction with the above mentioned source of information as driving forces for adaptation mechanisms.

To solve the problem of how to use an instructional design or learning theory in conjunction with learner's domain knowledge, background, preferences and learning style as source of information for adaptation, a framework for AEHS was conceived.

This paper mainly presents a description of the proposed framework and exemplifies how it functions in order to provide adaptation, as well as presents the framework's benefits.

The paper is organized in the following way. Section 2 presents the framework and its components. Section 3 describes how framework's components communication takes place. Section 4 presents an implementation architecture for the framework. Section 5 presents some concluding remarks.

2. Framework for AEHS

To solve the problem of using an instructional design or learning theory as source of information for adaptation, a previous problem should be solved first. This problem was how to explicitly include an instructional design or learning theory in AEHS, given that AEHS did not contemplate an instructional design or learning theory in their architecture. But before this problem there was another one. This other problem was how to define a framework that was capable of facilitating the inclusion of such a theory.

To address these problems, a framework for AEHS [28] was conceived and has been evolving as the result of the analysis of AEHS, Intelligent Tutoring Systems, Web-Based Educational Systems and Adaptive Systems described in the literature [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 35, 36]. Figure 1 shows the framework for AEHS. It should be noted that the framework is a conceptual model that describes what is intended by each component and how the components are related to other components.

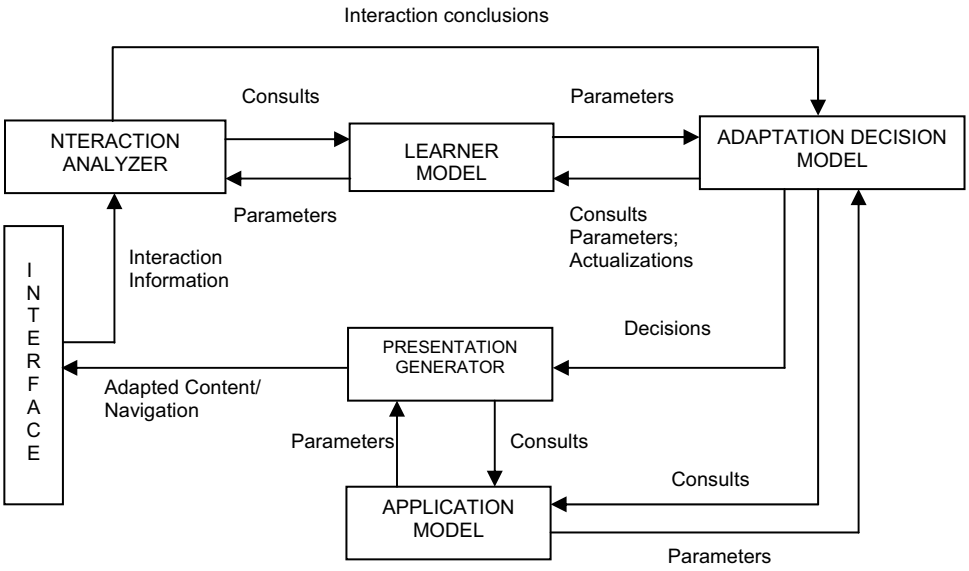


Figure 1. Framework for Adaptive Hypermedia Educational System.

A description of the framework's components is presented next.

2.1 Interaction Analyzer

The Interaction Analyzer is responsible for acquiring information on learner's behavior. It has two main functions [8, 9]:

- To monitor learner-system interaction in order to get information on activated links, selected tools, etc.
- To infer relevant conclusions on the learner's behavior, as for example if the learner entered into a specific learning unit, if he or she has finished a unit, if he or she has taken the initiative to change some parameters in the learner model etc.

2.2 Application Model

The Application Model represents the main features of the application in terms of a model of the domain, the instructional or learning theory used and how the domain concepts are grouped into learning units. The Application Model comprises three sub-models: Domain, Learning and Hyperbase Sub-models.

2.2.1 Domain Sub-model

For sure there are several ways to represent the Domain Sub-model. A simple way to represent it is by a list of topics and subtopics. But a more expressive way of representing it is by means of a concept map [32], where the semantic of concept relationships is specified. Additionally, the concept map can be enriched with some meta-information for the map's concepts. Figure 2 shows a representation of the domain sub-model using Backus-Naur Form [23].

<Domain>:: <SingleDomain> | <CompositeDomain>
<CompositeDomain>:: <SingleDomain> |
    <SingleDomain> <CompositeDomain>
<SingleDomain>:: <Topic> <TopicsRelashionship>
    <Topic>
<Topic>:: TOPICID <TopicKnowledgeType>
    <TopicProperty >
< TopicKnowledgeType>::
    <SingleTopicKnowledgeType > |
    <MultipleTopicKnowledgeType>
< MultipleTopicKnowledgeType >::
    <SingleTopicKnowledgeType > |
    <SingleTopicKnowledgeType >
    <MultipleTopicKnowledgeType >
< SingleTopicKnowledgeType>:: CONCEPTUAL |
    PROCEDURAL | OPERATIONAL
<TopicProperty>:: <TopicImportance>
<TopicDificult>
<TopicImportance>:: LOW | INTERMEDIATE | HIGH
<TopicDifficulty>:: EASY | MODERATE |
DIFFICULT
<TopicsRelashionship>:: <TopicsRelashionshipType>
    <TopicsRelashionshipDirection>
<TopicRelashionshipType>:: PREREQUISITE |
    PART-OF | TYPE-OF

<TopicsRelashionshipDirection>:: <FromTo> |
<Bidirectional> | NULL
<FromTo>:: <SourceTopic> <DestinationTopic>
<SourceTopic>:: TOPICID
<DestinationTopic>:: TOPICID
<Bidirectional>:: TOPICID TOPICID

Figure 2. Representation of the domain sub-model using Backus-Naur Form.

The representation of the domain sub-model using Backus-Naur Form has a twofold purpose. First, it is useful to express the syntactical aspects of the model, offering flexibility to incorporate new elements or to change some of them. Second, it is independent of the form of implementation. It could be implemented as a formal ontology or as a graph.

Figure 2 shows that a domain can consist of a single topic or several topics, and that each topic has an identifier (TOPICID), a single or multiple knowledge type and some other properties, importance and difficulty. Also the topics are connected by semantic relationships, which can be of types prerequisite, part-of, type-of etc.

2.2.2 Learning Sub-model

The Learning Sub-model represents the instructional design theory used in the application [24] and the learning units defined on the domain model. This sub-model represents the main organizing way the learning process is to be carried out.

An example of an instructional theory is Ausubel's Meaningful Learning [25, 26, 32]. Meaningful learning is a process in which new information is related to an existing relevant aspect of an individual's knowledge structure, usually occurring when more specific, less inclusive concepts are linked to more general existing concepts [32]. Thus according to this process, the most general, most inclusive concepts are introduced first and then these concepts are progressively differentiated in terms of detail and specificity [32]. The meaningful learning includes also the inverse of progressive differentiation, the integrative reconciliation. In the integrative reconciliation more general ideas are obtained from more specific ones.

Another important aspect of meaningful learning is the use of as an artifact advance organizers. Advance organizers aim at bridging the learner's present knowledge and new one to be acquired. A common type of advance organizer is a concept map.

On the basis of meaningful learning and learning units, the following instructional strategies can be defined for the Learning Sub-model:

- At the beginning of a course, the system presents a course overview with a short description of the learning units.
- At the beginning of a learning unit, the system presents an advance organizer, a unit overview or the content of the most inclusive topic of the unit.
- In a given learning unit, the learner accesses the topics according to the restrictions imposed by the topic relationships defined in the domain model.
- Having visited every not known topic in a given unit, the learner is provided with an exercise, an integrative reconciliation and a test, respectively in this order.
- Topic contents are presented in accordance with the type of objective of a learning unit, which can be conceptual, procedural or operational.
- When a learner reaches the last topic in a giving unit, the system suggests links to content of the next type.
- If the current topic is conceptual and the subordinated topics have been visited, then present synthesis for conceptual content.
- If the current topic is conceptual and procedural and the subordinated topics have been visited, then present synthesis for procedural content.
- If the current topic is conceptual and operational and the conceptual subordinated topics have been visited, then present operational content.

Using again the Backus-Naur Form, the Learning Sub-model could be represented as in Figure 3.

<Course>:: <LearningUnit> | <SetOfLearningUnits>
< SetOfLearningUnits >:: < LearningUnit > |
<LearningUnit> <SetOfLearningUnits>
<LearningUnit>:: <Objective> <SetOfTopics>
<SetOfTopics>:: TOPICID | TOPICID <SetOfTopics>
<InstructionalStrategies>:: <SetOfConditions>
<SetOfInstructionalActions>
<SetOfConditions>:: <Condition> | <Condition>
<SetOfConditions>
<SetOfInstructionalActions>:: <InstructionalAction> |
<InstructionalAction> <SetOfInstructionalActions>

<Condition>:: COURSEBEGNNING |
UNITBEGNNING |
UNITEND |
CONCEPTUALTOPICSMASTERED |
PROCEDURALTOPICSMASTERED |
TOPICSOFTHEUNITVISITED |
UNITEXAMPLEACESSED |
UNITEXERCISEDONE |
UNITTESTDONE |
EXAMPLE REQUEST |
EXERCISEREQUEST |
TESTREQUEST
<InstrutionalAction>::
PRESENTCOURSEOVERVIEW |
PRESENTLIKSTOUNITS |
PRESENTUNITOVERVIEW |
PRESENTADVANCEORGANIZER |
PRESENTINTEGRATIVERECONCILIATION |
PRESENTCONCEPTUALTOPIC |
PRESENTPROCEDURALTOPIC |
PRESENTOPERACIONALTOPIC |
PRESENTSINTESISOFCONCEPTUALTOPIC |
PRESENTEXAMPLE | PRESENTEXERCISE |
PRESENTTEST

Figure 3. Representation of the Learning Sub-model.

It should be noted that the semantic for each individual instructional strategy is not defined. The idea here is to keep conditions and actions independent in order to allow the strategies be defined in a later moment by the instructional designer. This way conditions and actions can be combined in different ways.

2.2.3 Hyperbase Sub-model

The last sub-model of the Application Model, Hyperbase Sub-model, keeps the metadata library of the learning objects and the learning object properly. The learning objects include the concepts to be learnt, examples, exercises, learning evaluations, and specific contents related to the instructional design theory defined in the Learning Sub-model. For the instructional strategies defined in Figure 3, the following subset of metadata categories and the correspondent elements can be used [33, 34]:

1) General: Identifier of the Learning Object
2) Technical: URL of the Learning Object
3) Educational: Learning Resource Type:
- Problem Statement
- Example
- Exercise
- Topic Content Presentation:
  - Conceptual
  - Procedural
  - Operational
- Test
- Course Overview
- Unit Overview
- Advance Organizer
- Integrative Reconciliation

4) Classification: Domain Topic Correspondent: TOPICID

It is worthwhile noting that the learning objects of type topic content presentation can include the three types of content, as defined in the Domain Sub-model, in the same object. Then each type is presented in accordance with the instructional strategies.

2.3 Learner Model

The Learner Model is the structure that contains the information on the learner's characteristics that allow the AEHS to adapt to these characteristics [27].

For the purpose of an AEHS that uses a concept map for the domain and as instructional theory the meaningful learning, a learner model can take into account the domain knowledge, learning styles [29, 30] and learner's media preferences. The learning styles considered are serialists and holists. The serialist learners prefer to study a limited number of issues in sequence, while holists tend to set a wider focus, opening up more topics in a learning episode and hence working with a more complex organizational scheme [27].

The learning styles have a profound influence on the navigation adaptation [4, 5]. For example, in a domain represented as a concept map in which the topics are progressively differentiated from top to base, a serialist learner would be provided with a depth-first navigation adaptation, with the system suggesting more specific topics, while a holist learner would be provided with a breadth-first navigation adaptation, with the system suggesting topics with the same level of abstraction of the current topic.

The learner's domain knowledge has also influence on navigation adaptation. Once the learner knows a topic the system would not suggest it to be visited. The learner's media preferences have influence on the content presentation. Based on this information, the system tries to present the contents accordingly.

2.4 Adaptation Decision Model

The Adaptation Decision Model is responsible for deciding what the system should do in terms of presentation and navigation adaptation on the basis of the conclusions draw by the Interaction Analyzer, parameters from the Learner Model and information from the Application Model.

The Adaptation Decision Model uses a "four-layer adaptation model". At the highest level is the instructional design governing the main structure of the learning activities. At the second level is the learner's domain knowledge. At the next levels are the learning style and the learner's preferences. With this information, the Adaptation Decision Model decides what should be done in terms of presentation and navigation adaptations.

The Adaptation Decision Model is represented by a set of rules describing the decisions that are sent to the Presentation Generator Model. Some samples of high level rules, expressed as a conjunction of antecedents and a conjunction of consequents, are shown in Figure 4.

(Course beginning && There is partial order within course units && Learner does not know the units && Learning style is holistic) ➔ *(Present a course overview && Present a concept map of the course && Allow access to first unit in the order).*
(Current topic type is conceptual and procedural && Children topics have been visited) ➔ *(Present procedural content).*
(Current topic is conceptual and operational && Procedural children topics have been visited) ➔ *(Present operational content).*
(All unit topics visited && Unit exercise done && Integrative reconciliation presented) ➔ *(Present a test for the unit).*

Figure 4: Sample of High Level Rules of the Adaptation Decision Model.

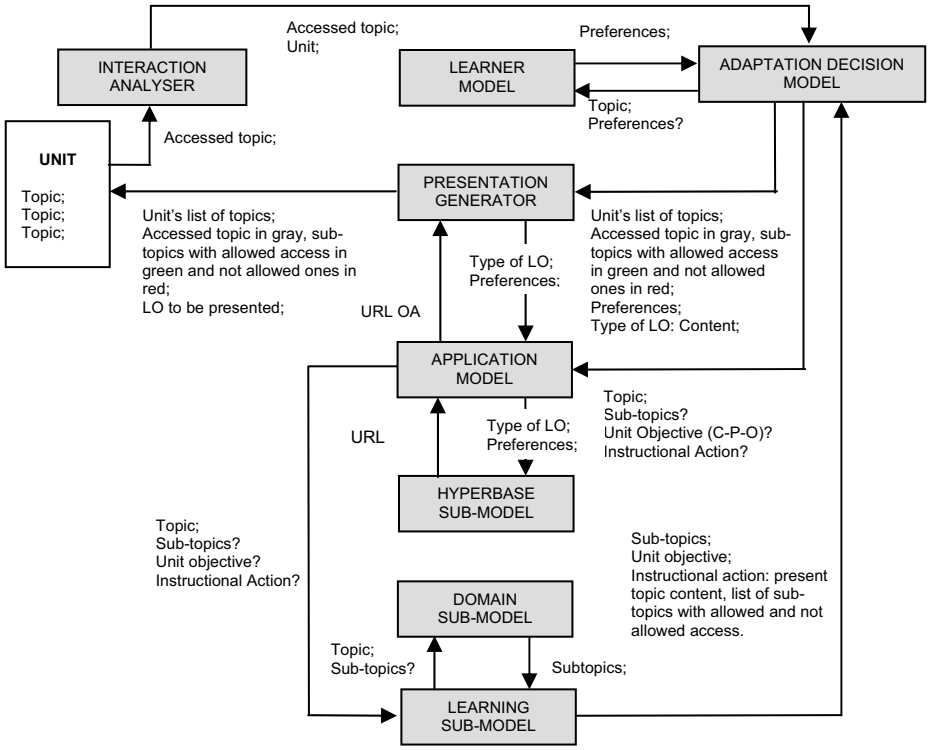


Figure 5: Representation of framework's components communication.

2.5 Presentation Generator

The Presentation Generator is responsible for generating what will be presented to the learner as a result of processing the information received from the Adaptation Decision Model and Application Model. For example, it can present an annotated list of topics and subtopics of the current learning unit, the topics previously accessed and those being suggested, as well as can present the content of a learning object informed by the Application Model.

The Presentation Generator is the final state of the framework's components communication. Figure 5 shows a schematic representation of this communication.

Having the learner clicked on a topic, the Interaction Analyzer identifies the topic and informs it to the Adaptation Decision Model. The Adaptation Decision Model checks the learner preferences on the Learner Model and then informs the just accessed topic to the Application Model, as well as asks this model the sub-topics of the present topic, the unit objective and the instructional action. The Application Model contacts the Learning Sub-model in order to inform the present topic and to ask the sub-topics, the unit objective and the instructional action. The Learning Sub-model asks the Domain Sub-model the sub-topics and returns the sub-topics, unit objective and the instructional action.

On the basis of the information received, the Adaptation Decision Model decides for the kind of adaptation to be carried out. In this case, it decides for a color-annotated list of topics, with the present topic in gray, the allowed topics in green and the ones not allowed in red. The decisions are sent to the Presentation Generator which asks the Application Model the URL of a Learning Object that corresponds to the information provided. Finally, the Presentation Generator presents on the user interface the information received from the Adaptation Decision Model and from the Application Model.

4 Implementation Architecture

This section briefly presents the way the proposed framework is being implemented. The implementation architecture chosen is based on Java technologies to accomplish the idea of framework's component modularity, component modifications and component reuse. Figure 6 presents the implementation architecture for the framework.

As can be seen from Figure 6, every requisition coming from the learner's web browser are interpreted by the servlet Interaction Analyzer that can access the Learner Model and Application Model data bases. After drawing conclusions on the requisition, the Interaction Analyzer contacts the Adaptation Decision Model, which is implemented by means of JavaBeans. The Adaptation Decision Model then accesses the Learner Model and Application Model data bases to get the information required. After getting this information, the Adaptation Decision Model sends its decisions to the Presentation Generator, which is also implemented by means of JavaBeans. After receiving a decision, the Presentation Generator accesses the Application Model data base to get the URL of a learning object to be presented. Finally, the Presentation Generator requests a composition of the

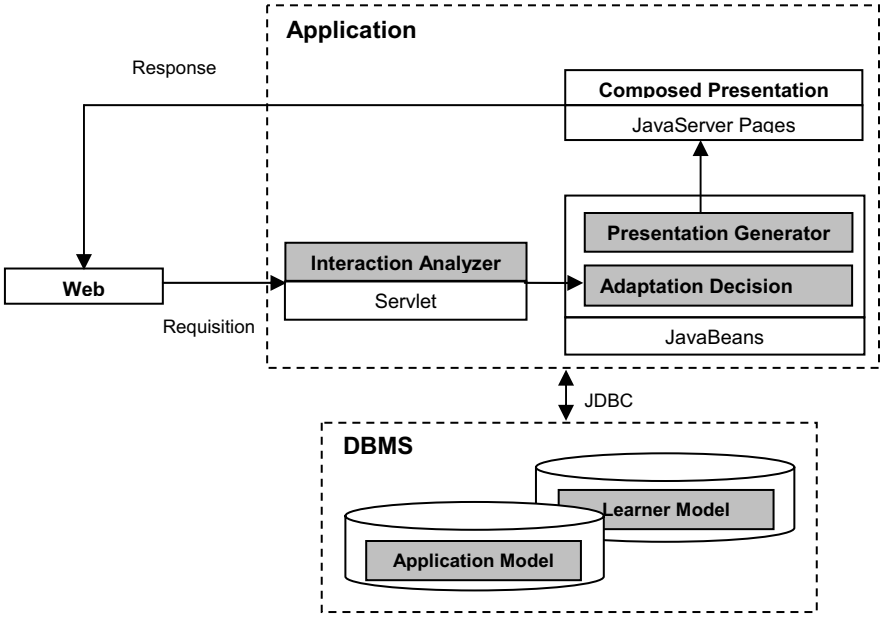


Figure 6. Implementation architecture for the AEHS framework.

presentation to the JavaServer Pages.

As JavaBeans are components of an application in the context of JavaServer Pages or Servlets, they are suitable to implement the Presentation Generator and the Adaptation Decision Model. Additionally, the JavaBeans offer advantages of separating the programming logics from presentation, as for the Presentation Generator and the final page composition in Figure 6.

5 Conclusion

The framework for AEHS has been a solution for the two main problems raised. First, the framework was a solution to the problem of how to use an instructional design or learning theory in conjunction with learner's domain knowledge, background, preferences and learning style as source of information for adaptation in AEHS. Second, it was also a solution to the problem on how to explicitly represent an instructional design theory in an AEHS, instead of having it diluted over other system's components. Additionally, as a core concern in the framework's conception was the modularity of the components, the result was a framework with well defined and separated roles for the components.

With the separation of the components' roles it can be speculated that be possible the reuse of materials from one application to another. For example, the domain model and some learning objects could be reused in an application with a different instructional design theory.

Even being based on a single case of use of some aspects of an instructional design theory, it is possible to envisage the use of other theories in the framework.

The possibility of reuse of material and the use of different instructional design theories raises another possibility. Supposing for example that an application would be defined using two different instructional design theories, then they could be put together to form a single application. With that, besides the presentation and navigation adaptations provided, this new application could also chose the instructional theory more suited to the learner's learning style.

Another speculation that requires further studies is related to the quality of the support provided by the adaptation obtained with the framework. Would the use of an instructional/learning theory in conjunction with knowledge domain, learning styles and learner preferences make adaptation strategies more supportive in terms of learning? It is not an easy question to be answered.

The framework can also serve as a basis for creating authoring tools, as long as several tools would be required for authoring an application. Among the authoring tools required are domain, instructional design and learning object metadata.

By not committing to any specific technology, in contrast to the ongoing implementation, the framework can be adapted in order to be implemented with more intelligent technologies, like formal ontologies and intelligent agents.

References

[1] P. BRUSILOVSKY. Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies. *Proceedings of Workshop "WWW-Based Tutoring" at the 4th International Conference on Intelligent Tutoring Systems* (ITS'98), San Antonio, TX, August 16-19, 1998.

[2] P. BRUSILOVSKY. Adaptive Hypermedia. *User Modeling and User Adapted Interaction*. Kluwer Academic Publishers, v.11, 2001, pp. 87-110.

[3] M. GRAFF. Learning from Hypertext and the Analyst-Intuition Dimension of Cognitive Style. *World Conference on E-Learning in Coporate, Government, Healthcare & Higher Education* – E-Learn 2002, 2002, Montreal, Canadá, AACE, 2002,p p. 361 – 368.

[4] E. TRIANTAFILLOU, A. POMPORTSIS, and E. GEORGIADOU. AES-CS: Adaptive Educational Systems Based on Cognitive Styles. *AH'2002 Workshop on Adaptive Systems for Web-based Education*, Málaga, Spain, 2002.

[5] N. BAJRAKTAREVIC, W. HALL, and P FULLICK. Incorporating Learning Styles in Hypermedia Environment: Empirical Evalutation. *AH'2003 Workshop on Adaptive Hypermedia and Adaptive Web-based Systems*, Budapest, Hugary, 2003.

[6] A. CRISTEA and L. CALVI. The Three Layers of Adaptation Granularity. *User Modeling' 2003*, Pittsburg, US, 2003.

[7] H. WU, G. J. HOUBEN, and P. De BRA. AHAM: A Reference Model to Support Adaptive Hypermedia Authoring. *Conference on Information Science*, Antwerp, 1998, pp. 51-76.

[8] C. KARAGIANNIDIS, D. SAMPSON, and P. BRUSILOVSKY. Layered Evaluation of Adaptive and Personalized Educational Applications and Services. *AI-ED 2001*, *Workshop on Assessment Methods in Web-Based Learning Environments & Adaptive Hypermedia*, May 19, 2001, pp. 21-29.

[9] A. F. NORCIO and J. STANLEY. Adaptive Humman-Computer Interfaces: A Literature Survey and Perspective. *IEEE Transactions on Systems, Man and Cybernetics*, v. 19, n. 2, Mar./Apr 1989, pp. 399-408.

[10] T. A. P. FERNÁNDEZ. *Un Hiperentorno Adaptativo para el Aprendizaje Instructivo/Constructivo*. Universidad del País Vasco, Departamento de

Lenguajes y Sistemas Informático, Memoria para el Grado de Doctor en Informática, 2000.

[11] N. HENZE. *Adaptive Hyperbooks: Adaptation for Project-Based Learning Resources*. University of Hanover, Department of Mathematics and Informatic, Doctoral Dissertation, 2000.

[12] P. De BRA and L. CALVI. AHA! An Open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia*. Taylor Graham Publishers, v. 4, 1998, pp. 115-139.

[13] R. S. G. BARBEIRO. *Characterizing and Modeling of Adaptive Hypermedia Courses*. National Spatial Researching Institute (INPE), Master Degree Dissertation. 2001 (Available in Portuguese).

[14] M. G. B. MARIETTO. *Definição Dinâmica de Estratégias Instrucionais em Sistemas de Tutoria Inteligente: Uma Abordagem Multiagentes na WWW*. Tese (Doutorado em Informática) - Divisão de Ciência da Computação, Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, Brazil, 2000.

[15] WEBER, G.; BRUSILOVSKY, P. ELM-ART: An Adaptive Versatile System for Web-Based Instruction. *International Journal of Artificial Intelligence in Education*, n. 12, 2001.

[16] H. WU. A Reference Architecture for Adaptive Hypermedia Systems. In: *ACM Conference on Hypertext and Hypermedia –Hypertext' 01*, Arhus, Denmark, 2001.

[17] R. MIZOGUCHI and J. BOURDEAU. Theory-Aware Authoring Environment – Ontological Enginnering Approach. In: ICCE Workshop on Concepts and Ontologies in Web-Based Educational Systems, Auckland, New Zealand, 2002. Eindhonven: Eindhoven University of Technology, 2002, p. 51-56.

[18] L. AROYO and D. DICHEVA. Domain and User Knowledge in a Web-based Courseware Engineering Course. In: T. Hruska, M. Hashimoto (Eds.), *Joint Conference Knowledge-Based Software Engineering* 2000. Amsterdam: IOS Press, 2000, pp. 293-300.

[19] A. CRISTEA and L. AROYO. Adaptive Authoring of Adaptive Educational Hypermedia. Adaptive Hypermedia 2002, *Second International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*. LNCS 2347, Springer, 2002, pp. 122-132.

[20] L. AROYO, D. DICHEVA, and A. CRISTEA. An Ontological Support for Web Courseware Authoring. *ITS 2002, Intelligent Tutoring Systems*. LNCS 2363, Springer, 2002, pp. 270-280.

[21] N. HENZE and W. NEJDL. Kowledge Modeling for Open Adaptive Hypermedia. *Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, Málaga, Spain, 2002.

[22] N. HENZE and W. NEJDL. Logically Characterizing Adaptive Educational Hypermedia Systems. *AH'2003 Workshop, World Wide Web Conference, 2003*.

[23] J. HOPCROFT and J. ULLMAN. *Introduction to Automata Theory, Languages and Computation*. Menlo Park, Addison –Wesley, 1979.

[24] C. REIGELUTH. What Is Instructional-Design Theory and How Is IT Changing? In: REIGELUTH, Charles M (Ed.). *Instructional-Design Theories and Models*. Mahwah, Lawrence Erlbaum , 1999, pp. 5-29.

[25] D. P. AUSUBEL, J. D. NOVAK, and H. HANESIAN. *Psicología Educativa*. Ciudad de México, Trilhas, 1989.

[26] J. POZO. *Teorias Cognitivas del Aprendizaje*. Madrid: Ediciones Morata, 1994.

[27] P. HOLT, S. DUBS, M. JONES, and J. GREER. The State of Student Modelling. In: GREER, J.; McCALLA, G. (Eds*.). Student Modelling: The Key to Individualized Knowledge-Based Instruction*. Berlin: Springer-Verlag, 1994.

[28] J. M. P. OLIVEIRA and C. T. FERNANDES. Adaptation Architecture for Adaptive Educational Hypermedia Systems. *World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education -E-Learn 2002*, 2002, Montreal, Canada, AACE, 2002.

[29] G. PASK. Styles and Strategies of Learning. *British Journal of Educational Psychology*, 46, 1976.

[30] C. S. CLAXTON and P. H. MURREL. *Learning Styles – Implications for Improving Educational Practices*. ASHE-ERIC Higher Education Report No. 4. Washington: Association for the Study of Higher Education, 1987.

[31] J. D. NOVAK. *Learning Creating and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations*. New Jersey, Lawrence Erlbaum Associates, 1998.

[32] S. GARLATTI and S. IKSAL. A Semantic Web Approach for Adaptive Hypermedia. *AH'2003, Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems*, Budapest, Hungary, 2003.

[33] IEEE. *Draft Standard for Learning Object Metada*. IEEE P1484.12/D4.0, 2000.

[34] G. WEBER, H. C. KUHL, and S WEIBELZAHL. Developing Adaptive Internet Based Courses with the Authoring System NetCoach. *Twelfth ACM Conference on Hypertext and Hypermedia – Hypertext' 01*. Arhus, Denmark, 2001.

[35] P. BRUSILOVSKY, E. SCHWARZ, and G. GERHARD. ELM-ART: An intelligent tutoring system on World Wide Web. In C. Frasson, G. Gauthier, & A. Lesgold (Eds.), *Third International Conference on Intelligent Tutoring Systems, ITS-96* (LNCS, Vol. 1086). Berlin, Springer-Verlag, 1996, pp. 261-269.

A Web-based Intelligent Case-based Reasoning Legal Aid Retrieval Information System

Kevin Curran, Lee Higgins
University of Ulster, Magee Campus, Londonderry, N. Ireland, BT48 7JL
Kj.curran@ulster.ac.uk

Abstract

The Internet has for some time been used by a wide class of lawyers as both a general business tool and more specifically as a research tool. Indications are that this trend is set to increase and infect all areas of legal practice. It should therefore be cultivated and exploited. However as the demand for legal services increases, two related challenges are presented to the Information Technology and Legal Communities. The first being the provision of easy to use services/applications which are cost-effective to develop and which improve the efficiency of the lawyer's research task and the second challenge; and the second of finding effective means of making such services widely and publicly available across the Internet. This paper demonstrates in the first instance how such services might be created by building on lessons learnt from an investigation into current legal applications. Secondly it examines those technologies that offer an appealing means of realizing the second goal above, with the eventual goal of describing the framework of an application provides the basis for meeting both challenges.

1 Introduction

While the Internet remains a powerful and efficient means of exchanging and communicating information, the standard web-site approach of fetching documents through predetermined hyperlinks or via keyword search through search engines, is not likely to prove overly helpful in satisfying the sophisticated information needs of lawyers. In other words the static nature of the Web seriously limits its usefulness in this context. However the Java language promises to transform the Web into a truly interactive information forum and also provides simple yet powerful means of making services not particularly designed for the web available across the Internet. It could thus help to better meet the aforementioned information needs. At the same time it is crucial (and a central argument in this work) that we recognize that most modern legal information retrieval applications (whether web-based or not) are, for some basic reasons failing to service the requirements of non-specialist lawyers. In light of the above the current situation poses 2 distinct but related challenges for I.T. The first, providing easy to use services which improve the efficiency of the lawyers research tasks and thus effectively meeting the information needs of non-specialist lawyers and secondly, making these services widely and publicly available across the Internet.

Our proposed system aims at supporting the lawyer in his research tasks for a problem situation. Therefore some understanding of the how this task is generally carried out and how we might improve efficiency here is called for. The overriding goal for the lawyer in this context is to get to potentially relevant legal resources (here decided cases and doctrinal writings) that can help him better understand the legal issues he is dealing with and how he might go about tackling the legal problems at hand.

1.2 Improving the Efficiency of Research Task

Making information publicly available i.e. via the Internet, obviously overcomes the problem of obtaining sources which are identified as relevant. However our problem is also one of efficiently identifying which materials could be relevant. It is argued that the efficiency of the lawyer in this context could be greatly improved if an information retrieval system could boast, at the very least, the following functionality

- An interface designed the guidance of a legal expert which 'walks' the lawyer through the various possible issues in the case – this interface (through a series of yes/no questions) could help the lawyer build up a profile of his case at hand.

- Once a basic profile is built up the system should indicate (through a process of basic pattern matching) those important cases in the field which best match the profile of the current problem case. We do not aim here at reasoning through stare decisis. Instead the goal is not to retrieve cases which shall be used in an actual court hearing etc,

but cases which are most likely to discuss the kind of issues the lawyers problem case involves. This functionality would therefore serve as a springboard into more intensive and informed research.

- If the system does indicate which cases best 'match' the input case, it should explain how this match occurs and also indicate how the retrieved cases are distinguishable from the current case.

- Lawyers search by concept (i.e. legal issues) not (potentially) random keywords. Any index of our document repository should allow the lawyer to find, inter alia, the leading case on a given issue, the latest case on a given issue, important cases where a given issue is discussed and also doctrinal articles where a given issue is discussed.

- When looking at a given case the lawyer should be able quickly to identify other cases where this case was distinguished or cases similar to the case or doctrinal writings where the actual case is discussed

1.3 Problems With Ai-Legal Applications

Despite the intensive and laborious research conducted into such machines they have largely failed to attain their goals and very few have made the transition from research ventures to applied systems. This failure, it is submitted is due to fundamental problems both at the philosophical/theoretical level and the practical level. Firstly all such systems involve the creation of a model of the legal domain – referred to as an '*ontology*'. The overriding goal here is one of representing the knowledge in a manner that is at once computer encodeable, and at the same time remains true to the meaning of the original source material. Making this knowledge computer encodeable almost always involves viewing the law as a (fixed) set of rules. It is almost universally accepted that the law is slightly more complex than this. The law is not self-contained and autonomous; instead it's meaning must be interpreted in the light of many implicit and ever-changing assumptions in the political and social context. It is seriously doubted whether current technologies can handle such a complex model. It thus follows that this process of isomorphism has yet to be achieved and representing legal reasoning in a computer encodeable form involves a certain distortion of the subject material.

Given the work involved in building a satisfactory model, it comes as no surprise that such machines are notoriously costly to develop, and given the underlying complexity they are extremely difficult to maintain (ease of maintenance being one of the cornerstones of any applied system) and update [1]. Developing intelligent systems that can easily handle change is no trivial matter and this problem is all the worse if we accept that the law is a notably fickle and changeable creature [2]. Furthermore unlike other areas of AI the complexity involved in automating or providing support for legal reasoning means that no generic commercial shells are available and most systems (capable of covering only one or two legal problem areas) must be built from scratch [3]. In addition such applications (whether EBS, KBS or DSS) fail to recognize the realities of legal practice in the sense that they tend to place too much emphasis on the law as an entity embodied in written texts rather then the product of an oral tradition. Computer technologies should therefore assist with mechanical research/retrieval tasks and not delve into more creative (and inherently uncertain) task of legal reasoning. We might also ask ourselves whether such machines have a large enough target audience to justify the massive effort required in building them. To make sense of the complicated output they produce the user must have already a considerable knowledge of the target area of the law and sophisticated I.T. skills - qualities missing in our target (and most) users.In addition we might note that the complex reasoning strategies and output they produce are likely only to be of use in cases decided in the highest courts in the land (about 1%) [4]. The CBR process of comparing cases based on the notion of factors is, it is argued, relatively easy to replicate. It is also quite useful to (and a common strategy adopted by) lawyers who use it not for any substantive purpose of legal reasoning but to identify cases that could help them better understand the issues involved in their case. Bearing this limited goal in mind our system shall attempt to implement some form of basic pattern-matching mechanism.

2 Legal-Aid Database Implementation

Most modern web applications (or indeed any category of application) to be truly interactive, informative and useful require access to structured data, which is not embedded in the application. Such data is most usually (and usefully) contained in databases. Database management system (DBMS) products, consist of a series of programmes which together offer highly effective means of managing the data. Through the use of powerful data definition languages and data manipulation languages (DDL, DML) such as SQL**,** these products offer an excellent basis for populating, querying and otherwise communicating with databases.

We choose to connect to the online database using the Java Database Connectivity (JDBC). The JDBC API allows for efficient development of multithreaded database applications allowing for almost seamless integration with powerful middleware solutions of the Java family. The API basically defines a number of Java interfaces, which enable developers to use Java as the host language for applications which access data independently of the actual database product. In essence JDBC shields an application from the specifics of individual database applications. Hence interoperability, combined with platform independence is JDBC's major selling point in this context [5].

2.1 The Major Components of the System

The system/application proposed here basically aims at improving the lawyer's research task by providing web-based access to a legal document repository, which resides on the server. To improve the efficiency of information retrieval performed on this document corpus the user can request documents by one of two methods – Legal Database Servlet or Case-Match Servlet. The components operate as follows (Figure 1).

- **Legal database Servlet**

  User at web browser connects to the server and requests the database service inputting data into HTML forms (1), this data is passed to the servlet (2), which uses it to run a query against the database and receive results (3), the servlet formats this data into HTML tables and returns this to the client (4), using the returned results the user makes a request for documents from the server (5), the server retrieves these documents (6) and returns these to the client (7).

- **Case-Match Servlet**

  User at a web browser connects to a server and submits a profile of his problem case (1), this data is passed to the Case-Match servlet (2) which runs a match against stored cases, selects the best matches, formats the results into HTML, and returns these to the user (4). Based on the information returned the user makes a request of the server for documents (5), the server retrieves these (6) and returns them to the user (7).

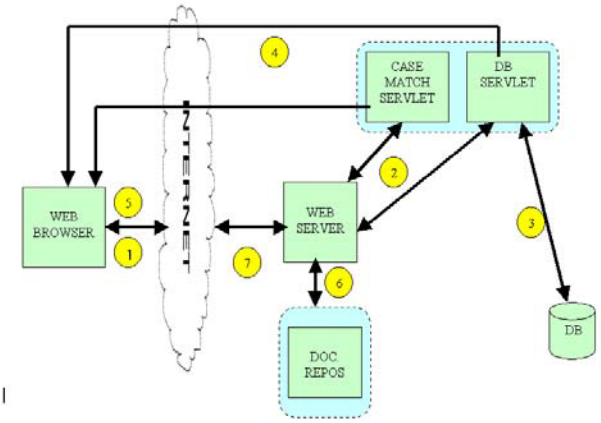


Figure 1 : General System Architecture

The representation of the documents referenced (legal cases and doctrinal writings) aims at providing efficiently for information needs. We represent the documents in the database in accordance with our target area of the law being broken down into a series of *'factors'* symbolising legal issues. These issues/factors (we identified 8 for demonstration purposes) may or may not be present in our case or doctrinal writing. Here these 8 factors are denoted I1→I8. Thus a document can be described in terms of the issues it deals with. Importantly a document can also be described in terms of other cases. The database contains tables made up of rows (and columns), which correspond to our legal documents. Each document is represented as a tuple, having a unique identifier (Cnum/Anum) with the other attributes being used to describe various facets of the document referenced (See Figure 2 and Figure 3). The attribute values are used for query and retrieval purposes.

	CNum	CName	Citation	Date	Link	Verdict	Lead	Main	Dist	Similar
	C1	Re Company A	2AllER(1990)67	30/01/90	<a href=...	P	I1	I2I3I6		C3
	C2	Re Company B	2AllER(1992)55	22/03/92	<a href=...	D	I2	I4I5	C1	C1
	C3	Re Company C	2AllER(1994)11	19/01/94	<a href=...	P	I4	I1I4I8	C1C2	C6
▶	C4	Re Company D	2AllER(1997)24	05/10/97	<a href=...	P	I8	I4I6I7I8	C3	C15

Figure 2 : The legal case table

The column headers include (for cases), the unique identifier, name, law reports citation, date of judgement, the full address of the document on the server and the verdict of the case (either pro-plaintiff, or pro-defendant). These fields are pretty self-explanatory. However certain other fields will represent the opinion of some expert in the legal area –

- **Lead** – Is this case the leading case on an issue? i.e. the most important case. If so we insert the identifier of the appropriate issue.
- **Main** – What are the main issues discussed in this case? Insert the identifier(s) for the most important issue(s) discussed here
- **Distinguished** – Has this case been distinguished in any other case? If so insert the unique identifier(s) for the appropriate case(s).
- **Similar** – Are there any cases which closely resemble this one? If so enter the unique identifier(s) for the appropriate case(s)

Please note that instead of using multi-valued attributes here we use continuous strings. Thus, for example, case C1 above is the leading case on Issue 1, the main issues discussed in the case are issues 2, 3 & 6, it hasn't been distinguished in any case but is similar to case C3.

	AName	Author	Citation	Link	Date	Issue	Case
	UPC1	J.Lowry	[1993] 2BLR 26	<a href=.....	100593	I4I7	C1
	UPC2	J.Dine	[1997] 2BLR 11	<a href=.....	100597	I1I2I3	C2C3
	UPC3	B.Cheffins	[1999] 1BLR 58	<a href=.....	100599	I5I6I8	C1C3
	UPC4	P.Birds	[1999] 3BLR 34	<a href=.....	100799	I2I7I8	C4
*							

Figure 3 : The doctrinal writings table

The 'Article' table is structured along the same lines as the 'Case' table, also having similarly column headers for name, author etc. The 'Issue' field informs us as to what legal issues are dealt with in the doctrinal writing. If any cases are discussed this is stated in the 'Case' field. For example article A1 above deals with issues 4 and 7 and discusses case C1. Overall this structure allows us to move away from the 'false drop' inducing keyword search. The number of possible queries on the database is potentially massive. For this prototype application we have created only a select number of queries that correspond to the most likely information needs. The types of query we specifically cater for here include -

- Find the latest case on a given issue
- Find the leading case on a given issue
- Find those cases where a given issue is discussed in detail & doctrinal writings which deal with a given issue
- Given a case find cases similar to that case & given a case find cases where that case distinguished
- Given a case find articles where that case discussed
- Find the latest case/article on the general area

2.2 The Web-Based Interface

The interface (Figure 4) is presented in the lawyers own terms i.e. the user is invited to find leading cases, main cases etc. according to a certain legal issue, or the user can request cases which are distinguished from the current case he is reading. This enables the lawyer to more easily make specific and meaningful queries. The lawyer basically selects a legal issue within the appropriate form and submits by clicking 'Find'. The 'Find Latest Case/article' forms at the top of the page do not allow the user to specify input parameters – instead here we use 'hidden' fields.

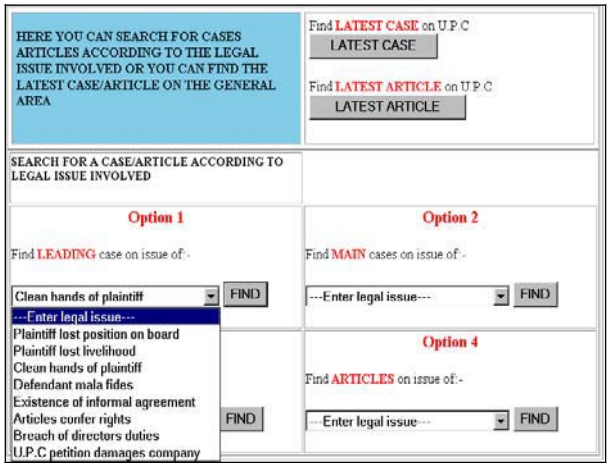


Figure 4 : Legal Aid Web Page

When reading a legal case or article the user may also access the database. In this instance however the user does not specify any input data and the data sent is decided according to the actual page we are on (i.e. the value attached to the name is the case identifier). By clicking the form buttons the user submits a query to the Servlet. A successful request returns a HTML table to the user which contains details of and a link to, documents satisfying the request. The user can then follow the link to obtain more details on the document.

2.3 The Case-Match Servlet

Here we describe how we might use a strategy based on that in the previous to perform a case-match function. To recap, the main goal of this component of the proposed application is not to produce highly complex and detailed reasoning strategies which devices such as HYPO and CATO generate but instead to provide the lawyer with a useful guide as to what important cases could prove useful in their reasoning task. As such a very simple pattern-matching algorithm/method, based on the model is employed. This basically operates on the simple premise that those cases which bear the most

positives similarities to the lawyer's current problem case are most likely to be of use in building an argument which supports the lawyers case. The corollary here is that those cases with more negative factors (i.e. TCS and PDS factors) will generally not be so beneficial.

The value the lawyer would take from such an application would reside in the output of a core set of cases that could help form the basis of the legal reasoning process. Here, no attempt is made to simulate or usurp the reasoning task of the lawyer, instead an effort is made to speed up the initial input (the legal research), which forms the basis of this task. Importantly the core cases returned to the user would be described in terms of comparison to the current problem scenario (i.e. in terms of PPS etc. so that the user could easily determine how the retrieved case might be used in their favour or to their detriment[1]). The case-match Servlet operates with a web browser which the user uses to create a profile of their problem case by answering a set of yes/no questions. The more info column allows the user to access materials that help him to better answer this question. This leads to a series of HTML documents which help clarify what constitutes a Yes or No with regards to each question (as this can be a grey area at times). Data is submitted in the same way as with the previous Servlet . The parameters input by the user are then compared against the pre-defined database data. Comparison would be based on the PPS/OCS/TCS/PDS approach. At the same time, a counter is set up to help determine the best matching cases. The count of each pre-defined case is represented as a cell in an array. What forms a good match (and scores highest) is based on the simple approach described above. For example PPS could score 2pts, OCS 1pt, TCS 0pt and PDS –1pt. Once all comparisons have been run, we search for and identify the highest (possibly 3) scoring cases. Once the best matching cases are found a second round of comparisons is run (between the best matching cases and the user-defined case). The goal here is to classify each retrieved case in terms of PPS/OCS/TCS/PDS factors. We then use the output from this to generate 'HTML-on-the-fly', and send a response back to the user which describes the 'best' matching cases in terms of PPS etc and also attributes such as name, citation, verdict and location.

[1] e.g. if a case exhibits PPS factors then these factors could be used to advance the current case and should be stressed. However if a case exhibits TCS factors then we must explain why these factors are not of crucial importance in this case if we wish to use the case in our favour.

3 Conclusions

The Internet (or more specifically the World Wide Web) has become the forum for information gathering and will surely be an essential tool of all modern lawyers. The Java language can enable us to transform the Web into a truly interactive law library. Although more information is available on the web, the efficient and effective retrieval and management of these web documents are still very challenging research issues. Intelligent information retrieval involves much more than retrieving free text, it involves systems that enable users to create, process, summarise, present, interact with (e.g., query, browse, navigate), and organise information within and across heterogeneous media. When navigating the web, with such a vast collection of linked documents, users can easily get lost in its depths. Information retrieval also poses users problems in finding appropriate resources and extracting information from within documents. Text and relational databases can be searched on content and indexing terms. Most modern legal information is contained (or can be structured as we have described) within databases. The use of JBDC can enable us (remote client) to efficiently access this information from a central location where the ever-changing knowledge base can be controlled and updated as required.

4 References

[1] Poulin et al : Coping with Change (1991): http://www.confpriv.qc.ca/crdp/en/equipes/technologie/textes/ia/bratley91a.html

[2] Bratley et al :The effect of change on legal applications (1991): http://liguria.crdp.umontreal.ca/crdp/en/equipes/technologie/textes/ia/bratley91b.html

[3] Hunter & Zeleznikov, supra

[4] Morrisson & Leith, The Barristers World and The Nature of Law (1992), Open University Press

[5] Ablan, Developing Intranet Applications with Java (1996), Sams.net

Idea Work Style – A Hypothetical Web-Based Approach to Monitoring the Innovative Health of Organizations

John C. Stratton
Aliant Inc.
john.stratton@aliant.ca

Abstract

Increasingly, ideas matter to companies. This paper reports the preliminary findings of a managed innovation pilot project at Aliant, a telecommunications company in Canada. A post-hoc analysis of the data collected suggests that a new measure of behavioural style – Idea Work Style – can be used to monitor the conditions for innovation in organizations.

Idea Work Style, determined from interactions within a web–based idea management tool, is proposed as a measure of individual behavioural style. Hypothetically, the innovative health of organizational environments in which individuals find themselves immersed can be described by comparing IWS to a measure of individual cognitive style.

1. Introduction – Idea Management in the "Knowledge Economy"

Innovation management tools are emerging as implements for improving the organization of "intellectual capital" in corporations. Aliant[1], a Canadian telecommunications company, recently conducted a pilot project in managed innovation using a web-based idea management tool supplied by IdeaPilot AS[2] of Denmark. The literature of knowledge management research provides a context in which this kind of tool can be understood as part of a larger knowledge management toolkit.

Knowledge management helps businesses account for intangible value – "a paradigm where sustainable competitive advantage is tied to individual workers' and organizational knowledge" [1]. Accordingly, intangible assets - knowledge and other forms of intellectual capital - are now seen by many companies as sources of previously hidden value and as strategically important resources. Companies use a variety of methods – among them the Skandia "Navigator" model and Robert Kaplan's and David Norton's "Balanced Scorecard" approach [1] - to measure and manage intangible non-financial assets as well as financial ones.

Tools for managing "intangibles" – like the Balanced Scorecard and Navigator approaches – may provide frameworks within which to quantify the potential value of knowledge and ideas and thereby provide a basis for management. In the words of Kaplan and Norton, "as companies around the world transform themselves for competition that is based on information, their ability to exploit intangible assets has become far more decisive than their ability to invest in and manage physical assets." [2]

Innovation management provides a practical way to deal with intellectual capital within knowledge asset frameworks. According to Gartner, the international research and consulting firm, innovation management is coming into its own as a bona fide business practice, with value as "a process for evaluating innovations rapidly and determining which will provide the best value on investment"[3]:

> Many commercial applications are emerging to provide broad support of innovation management. These are robust tools that are becoming an essential part of innovation in many leading enterprises. These tools have also moved from supporting single users in creative thinking or "brainstorming" to supporting group or team dynamics and enterprisewide processes. [4]

"Intellectual capital" - a term coined by John Kenneth Galbraith in 1969 [1] - is now part of the lexicon of knowledge economics. Implicit in the Balanced Scorecard and other intangible asset management approaches, it is an explicit component of the Skandia

[1] Aliant Inc. (TSX: AIT) is the incumbent telecommunications service provider in Atlantic Canada. More information on the company can be found at www.aliant.ca

[2] IdeaPilot offers a combination of software tools, training and coaching in the area of business creativity. More information on the company can be found at www.ideapilot.com

model. According to this model, intellectual capital is the sum of human and structural capital where:

> Human Capital is defined as the combined knowledge, skill, innovativeness, and ability of the company's individual employees to meet the task at hand. It also includes the company's values, culture, and philosophy. Human capital cannot be owned by the company.
>
> Structural Capital is the hardware, software, databases, organizational structure, patents, trademarks, and everything else of organizational capability that supports those employees' productivity – in other words, everything that gets left behind at the office when employees go home. … Unlike human capital, structural capital can be owned and thereby traded. [1]

In Gartner's view, innovation management tools are emerging in five distinct categories [4]:

- Idea management
- Innovation life cycle management
- Product development
- Environmental innovation management
- "outside-the-box" innovation management

Management approaches such as Balanced Scorecard and Navigator are frameworks for understanding how intellectual capital (and other intangible assets) creates value in the context of a company's overall value system. They are not, per se, methods for assessing the value of knowledge assets. If the value of ideas is to be effectively harnessed, measurable and manageable parameters must be identified. This is the role of innovation management tools.

The managed innovation project at Aliant was restricted to idea management, according to the Gartner categorization, focusing on "capturing ideas from individuals … and making them available to others …in a knowledge sharing sense … or for further evaluation" [4].

A structured idea management approach has the potential to change employees' ideas from intangible to tangible form. In doing so, it may enable companies to capture and exploit their potential value, at least in theory. Seen in the context of the Skandia Navigator model, idea management software tools and the data contained in their data bases are part of the structural capital (owned by, and therefore the tradable property) of a company[3].

Hargadon [5] theorizes that certain innovation networks "link people, ideas, and objects together in ways that form effective and lasting communities and technologies." Within the context of managing knowledge assets, a network-based idea management tool can be understood as a mechanism for transforming tacit, intangible human capital into explicit, tangible structural capital. Intellectual capital in explicit form can be managed, manipulated, exposed to and combined with other tangible and intangible resources and objects – people, funds and other ideas, for example. This is the intended function of idea management, but as we will see, the process of conducting this function in a web-based environment generates residual data which can be used to monitor the organizational environment for innovation.

2. Idea Management Pilot Project

Following a period of customization, Aliant completed a pilot implementation project with 172 participants over the period June 3 – September 30, 2002. Several business units cooperated in sponsoring the project and supporting a project team charged with overseeing the customization work, conducting an employee trial, measuring success and determining the next steps for future implementation. During the employee trial, the project team trained sponsors, facilitators and participants, performed basic administrative and troubleshooting functions and measured, evaluated and made minor adjustments to the tool.

The basic premise of the IdeaPilot approach is that, while employees have plenty of good ideas, they often have difficulty putting them into practice. The web-based idea management tool is designed to help employees innovate by removing two key barriers to innovation in companies:

- No one *knows* everyone – many employees simply don't know the key people to contact and work with to draw out the potential of their ideas and understand if there is a benefit in implementing them.
- No one *sees* everyone – there are significant geographic barriers to innovation. Even when

[3] The issue of the ownership of intellectual capital is interesting and contentious, but beyond the scope of this paper. For an excellent discussion on the conflict between knowledge workers (the controllers of intellectual capital) and investors (the controllers of financial capital) see Martin & Moldoveanu, "Capital Versus Talent".

people know who can help them with their ideas, it can be very difficult to get people together in a timely and cost-effective way.

The tool presented participants with specific business challenges facing the company. Participants were asked to enter new ideas - related to the challenges - into the system, as well as to contribute to the ideas created by others. Through the ensuing on-line discussions between diverse participants, ideas would be turned into fully developed concepts for presentation to the appropriate decision makers, known as "strategic anchors".

The intent was to capture ideas and develop them into concepts in a collaborative, asynchronous on-line environment. Within this process, there are several terms, which, having general meaning in ordinary use, acquired specific meaning in the idea management context:

> An **idea** is the conception of a single person - it often comes from a single perspective and may lack the benefit of others' experience, knowledge and skills.
>
> A **concept**, which includes all appropriate points of view, is completely ready for consideration as a project, in the correct strategic context and with the input of all relevant stakeholders.
>
> A **challenge** is a business problem or opportunity faced by the company.

The idea management tool presented participants with various challenges. Participants entered new ideas related to the challenges. They also contributed comments to ideas (both their own and other people's) already existing in the system. Collectively, the comments constituted a collaborative effort to convert ideas into fully developed concepts, which were subsequently presented to decision makers for evaluation and judgment. This method of transforming an idea into a concept is known as "**idea processing**".

The tool allowed employees to submit ideas within carefully defined strategic challenges. A person enters an idea - usually defined from a single perspective and lacking the benefit of others' experience and skills - into the system. Processing happens, and value is added, when other users comment on the idea. They bring in different, but relevant, points of view, gradually turning the idea into a concept; ready for consideration as a project, in the correct strategic context and with the input of the appropriate stakeholders. The results illustrate that there may be significant differences in the way employees work with ideas. Some employees, the data show, prefer generating new ideas, while others would rather work on refining the ideas of others. Good idea **generators** are not necessarily good idea **processors**.

A company's innovative ability depends both on its employees' ability to create new ideas and on their ability to process other people's ideas. Idea processing lowers the inherent risk in implementing an idea by forcing managers to address the details before committing to action. Effective processing seems to happen when employees are able to work on ideas in a way that suits their individual preferences. Processing an idea requires discussions between several departments to understand how it can – or can't – work within a business organization. A significant amount of value is added (and cost is avoided) in processing ideas, but it is an area that is seldom well understood by managers.

3. Pilot Project Results

The results of the pilot project were evaluated and, while the positive aspects of the program were recognized, the leadership team felt that there were too many other strategic priorities competing for its attention to proceed with a broader implementation, at least for the time being. High-level executive support had been identified by the project team as a critical success factor, so plans for further experimentation with the tool were cancelled in early 2003.

Some key findings of the pilot project were:

- Management involvement, leadership and facilitation were critical factors for success in taking ideas from conception to implementation.
- The company did not have a problem generating and capturing high quality, strategically aligned ideas. (The ability of the IdeaPilot tool to support the alignment of ideas with strategy is consistent with some authors' views of intangible asset management as a link between short-term action and long-term strategy. Kaplan and Norton see Balanced Scorecard as a "Strategic Management System" providing a strategy-action connection that is missing in traditional management systems. [2])
- "Idea networking" appeared to be a real phenomenon – sharing ideas in a common

forum appeared to stimulate the creation of more ideas.

- Idea *processing* seemed to be more difficult for employees than idea *generation* (Many people treated the system as a "suggestion box" rather than as a true forum for collaboration. In the suggestion box model, ideas can be submitted by all employees, but are processed and evaluated by a select group. This model requires less effort on the part of participants, but it has some problems; the process is often not transparent, for example.)
- Individuals in the contributor population can be described by their tendency to act as 'generators' (people whose primary contribution is to come up with new ideas) or as 'processors' (people who prefer to work on ideas created by others). The continuum from generator to processor can be quantified by a new measure, Idea Work Style (IWS).

These findings reflect the opinions of the Gartner analysts [4] on idea management technologies and conditions for their success in workplaces:

> The success of [idea management] initiatives depends strongly on culture, high-quality business processes and executive support. Technology enables the business processes.
>
> These tools simplify the process of idea generation capture and analysis. When they are well-supported by leadership and reward systems, they can be very productive in terms of the number of viable ideas generated.
>
> The challenge is how to take ideas to the next step, how to reconcile them with each other, and how to measure the results of the idea generation program. Enterprises must have a strategic framework for evaluating ideas and an overall process to continually drive good ideas to commercialization or implementation.

Considering the place of executive support in a managed innovation agenda, Aliant's decision to suspend its idea management program is an appropriate one. Other organizations, though, may be able to build on Aliant's experimental results. One outcome warranting further investigation is the concept of Idea Work Style.

4. Idea Work Style

At the conclusion of the employee trial period, the data contained in the idea management tool were examined with a view to discovering underlying constructs.

The data set I examined represents the contributions (including the creation of the ideas themselves) of 66 active participants to 68 ideas across three challenges. The data set covers the period from the project start date, 3 June 2002 up to 20 September 2002.

Data from the trial project sheds some light on the way employees work with ideas. The results suggest a new measure, Idea Work Style, which may help to explain how people work with ideas. For any given individual, IWS Score is defined as:

$$\mathbf{IWS = i - c\ (I/C)}$$

Where:
i = number of original ideas generated by the individual participant;
c = number of comments made by the individual participant to ideas generated by others;
I = total number of ideas in system;
C = total number of comments made to other participants' ideas (excluding "follow-up" comments by one individual to her own idea).

For the Aliant pilot project, IWS scores for the 66 active participants were distributed as shown in Figure 1 (with a normal distribution superimposed).

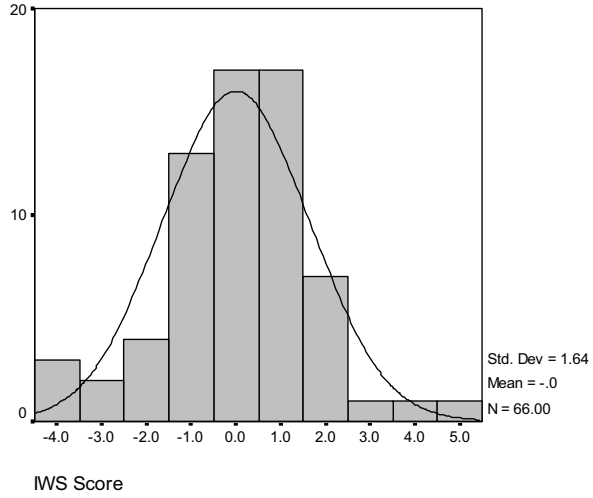


Figure 1. Idea Work Style Scores

A negative IWS score indicates an individual who prefers commenting on existing ideas to creating original new ideas. A positive score indicates one who prefers to generate new ideas and typically does not comment on those generated by other people. A zero score signifies someone whose ratio of ideas to comments (on the ideas of others) is identical to the idea/comment ratio (where 'follow-up' comments to individual's own ideas are excluded) for the entire participant population. Idea Work Style score is a measure of **imbalance** in the approach that employees take to working with ideas.

Fifty percent of pilot project participants scored between –0.7 and 1.0 on the IWS scale. In other words, about half the people on the pilot did not exhibit a preference for either creating or processing; they were comfortable in either role. The other half was fairly evenly split, between those who prefer generating, and those who prefer processing. Generators and processors are therefore not eccentric fringe groups; they are represented in significant numbers.

A hierarchical cluster analysis supports the finding that there are at least two distinct behavioural roles within the active participant population. A principle components analysis of the idea-comment data together with the additional variables 'follow-ups' (comments made by an individual to his or her own idea) and 'challenges' (the number of challenges in which a participant is active) suggests a modified IWS with these variables included could explain as much as 76% of variance in the data. (However, because there is insufficient data to form strong conclusions, the simpler IWS, incorporating only ideas and comments, is presented here). Whether it is IWS or some similar construct, which may include 'follow-ups', 'challenges', or some other as-yet unobserved variables, it seems there are observable variations in the way different people work with ideas.

5. Behavioral Style versus Cognitive Style – IWS and Kirton Adaption-Innovation Inventory

IdeaPilot is a structured tool that allows both synchronous and asynchronous input. Prante et al [6] find that a synchronous (as opposed to turn-taking, which differs from asynchronous) capability and a structured "idea space" were factors which supported the *generation* of ideas in tools for computer supported collaborative work. It seems that these factors also contributed to the success of the IdeaPilot tool as an idea generator in the Aliant project. The project team and the software developers were surprised to find that they did not translate into idea *processing* success. However, I suspect that this is largely due to a combination of participants' lack of processing experience and their familiarity with "suggestion box" methods. Absent these interfering conditions, these factors may also contribute to good idea processing. The appearance of two distinct behavioural roles – generator and processor – highlights the difference between these functions.

The IWS score appears to have potential for classifying individuals by the behaviours they exhibit in performing idea work. IWS is a measure of behavioural *style*, that is, the behaviours which individuals prefer in a collaborative innovation setting. It closely resembles a measure of *cognitive style* - Kirton Adaption-Innovation Inventory (KAI). Kirton's Adaption-Innovation theory (A-I theory) is an attempt to explain "differences in the thinking style of individuals, with particular reference to creativity, problem solving and decision making" [7]. Thinking style differs from problem-solving or creative ability (or for that matter, intelligence) in that it is a measure of *how* a person chooses to go about solving a problem, rather than *how well* the problem is solved. Hypothetically, IWS is a behavioural analog to KAI's measure of cognitive style. It is a measure of *how* an individual behaves in an idea work situation, rather than *how well* he or she works with ideas.

On the KAI scale, populations are distributed normally on a continuum. 'Adaptors' and 'innovators' are situated, like processors and generators, at opposite ends:

> *Adaptors* characteristically produce a sufficiency of ideas… based closely on, but stretching, existing agreed definitions of the problem and likely solutions. They look at these in detail and proceed within the established paradigm (theories policies, mores, practices) that is established in their organisations. Much of their effort in effecting change is in improving and 'doing better'...
>
> *Innovators*, by contrast, are more likely in the pursuit of change to reconstruct the problem, separating it from its enveloping accepted thought, paradigms and customary viewpoints, and emerge with much less expected, and probably less acceptable solutions … They are less concerned with 'doing things better' and more with 'doing things differently'.

Behaviour, Kirton states, is flexible, while cognitive style is unvarying. But behaviour is determined by the environment as well as by characteristics of the individual. Markides asserts, for example, that "the single most important determinant of employee behavior is the underlying context or environment of the organization" [8]. If the individuals in an organization do not change, then behaviour changes must result from differences in the environment.

My hypothesis is that, under conditions which are conducive to innovation, IWS correlates with KAI, with adaptors exhibiting processor behaviour and innovators exhibiting generator behaviour. If the hypothesis is true, a misalignment between individuals' KAI and IWS scores would indicate problems in the organizational environment for innovation, since KAI, part of a deep seated dimension of personality [7], is constant. To test this hypothesis, a measure of organizational environment quality is required.

Amabile's KEYS instrument is a good example of a tool for measuring workplace conditions for innovation. KEYS is a survey-based instrument that addresses the work environment (including Environmental Stimulants to Creativity and Environmental Obstacles to Creativity) and the "work outcomes of creativity and productivity". Amabile explains that the tool helps researchers "understand the social environment in organizations and how it might impact creativity" [9].

KAI, KEYS, and IWS could be combined, in an experimental setting, to develop a better understanding of the dynamics between personality, environment, and behaviour and to validate IWS as an appropriate and direct measure of innovative behaviour. But, any two of the three factors (personality, environment and behaviour) should be sufficient to provide a complete picture of the innovative health of an organization. Why then, since tools such as KEYS and KAI are already established and validated, is a measure of innovative behaviour required?

Many existing tools are survey-based. Personal experience suggests that employees' patience for answering surveys is quite finite, limiting the possibility for frequent measurement. Another problem with surveys is that they can only inquire about *past* behaviour. A combined IWS-KAI method would take a different approach, relying on individuals in an organization as sentinels for the health of their environment. This approach promises some advantages as a continuous and direct measure of the creative process. Because it draws on data collected from an on-line collaborative tool, IWS can be periodically (or even continuously) calculated, giving real-time updates not just on the conditions which will encourage creative behaviour, but on actual creative outputs themselves.

6. Conclusion and Future Work

Various models for measuring and managing knowledge assets, among them, Balanced Scorecard and Skandia Navigator, are emerging. They constitute frameworks for making intangible value explicit as well as for linking strategy and short-term action. Within these frameworks, idea management tools can be seen as mechanisms for transforming the intellectual capital in employees' ideas from a human capital asset to a structural capital asset, in the process shifting its ownership from employees (where it is often not actionable) to the organization (where it can accounted for, managed, optimized and combined with other tangible and intangible assets).

The behavioural preferences of individuals seem to be important factors in innovation management performance. In the Aliant pilot project, I observed significant differences in the way individuals interacted with an idea management system. The Idea Work Style measure developed as a result of these observations may be useful as a monitoring tool for the innovative health of organizations.

The next step will be to design and conduct experiments to test the hypothetical linkage between Idea Work Style and Kirton Adaption-Innovation Inventory. While it is unlikely that Aliant will continue this research, others are invited to adapt and extend these concepts.

7. References

[1] Bontis, Nick, "Assessing Knowledge Assets: A Review of the Models Used to Measure Intellectual Capital", *Framework Paper 00-01, Queen's Management Research Centre for Knowledge-Based Enterprises* (www.business.queensu.ca/kbe), Queen's University, Kingston, ON, April, 2000.

[2] Kaplan, Robert S. and David P. Norton, "Using the Balanced Scorecard as a Strategic Management System", *Harvard Business Review*, January-February 1996 pp75-85.

[3] Caldwell, French and Jackie Fenn, "Managing Innovation: Is it Possible?" *Letter From the Editor*, LE-15-2511, Gartner, Inc., 2002.

[4] Rozwell, C., K. Harris and F. Caldwell, "Survey of Innovation Management Technology" *Research Note* M-15-1388, Gartner, Inc., 2002.

[5] Hargadon, Andrew, "How Breakthroughs Happen: The Surprising Truth about How Companies Innovate", Harvard Business School Press, Boston, 2003.

[6] Prante, Thorsten, Carsten Magerkurth and Norbert Streitz, "Developing CSCW Tools for Idea Finding – Empirical Results and Implications for Design" In: *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2002)*, New Orleans, LA, ACM Press, November 16-20, 2002. pp 106-115.

[7] Kirton, M. J., "Adaptors & Innovators – Why New Initiatives Get Blocked", www.kaicentre.com. Accessed August 21, 2003.

[8] Markides, Constantinos C., "All the Right Moves: A Guide to Crafting Breakthrough Strategy", Harvard Business School Press, Boston, 2000.

[9] Teresa M. Amabile, "Motivating Creativity in Organizations: On Doing What You Love and Loving What You Do", *California Management Review*, Vol. 40, No. 1, Fall 1997. pp 39-58.

Roger L. Martin and Mihnea C. Moldoveanu, "Capital Versus Talent", *Harvard Business Review*, Harvard Business School Publishing, Boston, July 2003, pp…

Requirements of a Data Model for Modelbases

Thadthong Bhrammanee and Vilas Wuwongse
Computer Science & Information Management program,
Asian Institute of Technology
{ttb, vw}@cs.ait.ac.th

Abstract

Although existing model representations attempt to give modelbases—a vital component of model-driven Decision Support Systems (DSS)—a powerful capability for storing decision models, most of them are prone to not supporting modeling life cycles and Web-based modeling incompatibility. More attention is thereby paid to improving the development of data models. Data model requirements are presented in an attempt to assist the development of a new modelbase representation and the improvement of existing ones.

1. Introduction

Modelbases—vital components of model-driven Decision Support Systems (DSS)—immensely benefit from efficient and expressive data models. The term "data model" is used here to refer to a particular language for modeling decision models. The term "decision model" represents a quantitative model used in Management Science and Operations Research (MS/OR). As decision models are a valuable organizational asset, their proper storage certainly promotes reuse and sharing, which thereby economizes efforts and the time spent on the modeling activities.

Recently, Internet and Web technologies have contributed to the design, implementation, and employment of modelbases in at least four ways. Firstly, there is a frog leap advancement of computation via the Web, i.e., server-side and client-side computation. This advancement, in particular server-side computation, makes model execution independent of a client's platform. Secondly, an increasing proficiency of DSS technologies offers a greater variety of DSS tools (such as a mobile device) for storing or exploiting a model. Thirdly, a recent perspective of "Web as computer" publicizes the Web known as a "large repository for decision models". Fourthly, model users are currently demanding a decision model for convenient use just as a service on the Internet [2]. Fortunately, the concept of Web services [22] is just in time to enable that realization of the demand of users. It offers services via the Web and is a new resource for creating value added business services.

Due to these four aspects, the modelbase community must reconsider the development of data models. There exist various efforts—either directly or indirectly related to design issues of data models—which offer certain important requirements which a data model should meet; unfortunately, none of those efforts point out them all, nor are sufficiently concerned with the problem of compatibility with Web environments. For example, Geoffrion [8] and Maturana [18] discuss design issues for mathematical programming, but specifically only for an algebraic modeling language. Bharadwaj et al. [1] point out phases in a modeling life cycle, but they emphasize a model management system. Hence, key requirements/considerations for a data model, applicable within the modern modeling environment, must be well analyzed and understood.

Section 2 reviews major approaches to decision model representation, Section 3 presents data model requirements and Section 4 draws conclusions.

2. Model representation overview

Current approaches to the development of data models can be grouped by their levels of interest in the structure of decision models and their specification techniques. Note that, generally speaking, a single representation framework may use a combination of various approaches and techniques.

Levels of interest in the structure of a decision model are driven by a concern regarding the context of a decision model, ranging from ignoring the algorithmic aspect to high emphasis on the executable system of a decision model. The first—*data-centric*—level employs

a traditional database data model (such as relational and hierarchical structures) as a design principle and does not contain an explicit representation of a computation relationship. It has the advantage that users are familiar with the data model under study [3,5]. The second—*structure-centric*—level presents more information of a decision model, so that users are able to understand the large picture of a model structure in terms of a definitional system as well as are able to specify more details of the problem context. It has the ability to group the sub-model or to make a module which creates a concise representation [7]. The third—*abstraction-centric*—level reduces the complexity and increases the efficiency of a model by hiding all but the relevant data. Some frameworks (e.g., [14] and [15]) fully adopt and utilize the mechanism of an object-oriented (OO) data model so that a wide range of operations can be performed on the model. The fourth—*logic modeling-centric*—level applies a logic-based theory as an underlying principle. Integrity constraints are generally employed to enforce a syntactic structure. An inference mechanism is employed for retrieving implicit derived information from the model specification [13]. However, it requires a user to learn a number of model vocabularies and grammars. The last—*computation-centric*—level represents a decision model close to the algebraic or algorithmic form, so that users can investigate and formulate a mathematical equation as well as easily execute the model. However, most frameworks in this group lack a formal data model; their model specifications are in plain English [6,7,10].

Another perspective to classify the design of a data model is by the form (or specification technique) specifying it. The first—*graphic*—form represents the relations between parts of the model graphically. Diagrams and icons are popular tools for creating a visual perception of a decision model [3,4,7,15,19]. The second—*text*—form is more descriptive than the graphical form since a use of graphics may face either space limitation or unavailability of a proper modeling environment. Some frameworks in this group use human-understanding text to explain certain details of a model construct, e.g., a plain text description in the "interpretation" part of [7]. In contrast, machine-understandable languages are widely used. A decision model may either utilize an existing programming language such as Prolog or issue its own syntax [6,7,10]. The third—*algebraic*—form represents a decision model in a way which is close to algebraic notation. It requires the use of symbolic subscripts. Algebraic representation is commonly used as part of text-based modeling. The last—*schematic*—form restricts the structure and content of a decision model to a certain schema which expresses shared vocabularies and rules. This approach emerges due to the increasing demand for exchanging a decision model in agreement with a common vocabulary. The schema also allows machine validation of document structure [19,11].

3. Data model requirements

A data model typically describes a conceptual schema, constraints and operations. Yet, the modern modeling environment demands convergence between model representation and Web practice, whence a data model also has to integrate the following salient characteristics:

Model creation/formulation support contributes to conversion of a problem description into a particular form in which either a human or a machine can perform an analytical task. Two popular methods for the formulation of a new decision model are, firstly, creation from scratch [4,6] and, secondly, creation from template [14,15,17].

Regardless of the model formulation method, a data model should be sufficiently scalable so that large and small problems as well as various problem types can be formulated. For this purpose, a data model should have a concise and sufficient notation which covers problem types and model elements, be able to hide detail of abstraction and possess a grouping mechanism.

In addition, a data model should provide a supporting tool suitable for different types of model users and assisting decision model conceptualization. However, should those tools migrate toward a Web application, the bandwidth speed is a vital consideration in delivering the graphical content.

Model advertisement/registration support aims to create user awareness of an available decision model by announcing and providing public information of the model. In particular, the "Web as computer" era requires model features to be easily understood by remote users/applications (with a heterogeneous platform). Private and public conventions are generally two modes of model advertisement/registration.

For the *private convention mode*, decision models are advertised on the model owner's model repository. Each repository has a distinct way to index/arrange models. In contrast, the *public convention mode* demands a decision model to register with respect to the universally agreed convention of external register services [21].

In addition, metamodel has a crucial role in model advertisement, whence the framework should have precise definitions of the constructs and rules needed for expressing a decision model. In particular, a metamodel

should explain well detailed information such as required inputs/outputs and the model access method.

Model discovery/selection support is the task of finding and eliciting a suitable decision model from a model repository. The data model should be designed to make it easy to apply a matching operator at various element levels.

At this stage, there are at least three commonly available model selection strategies. The simplest one makes a request from the tree structure of a problem type [9]. The next one selects a model at the input/output level. A relational schema may be employed by introducing into each model "input" and "output" relations. Finally, employment of a frame, another approach to deliver efficient model selection, enables exploration of internal detail of the decision model [17].

Nevertheless, should modelbases be integrated with the Web services environment, this requires that the data model of a decision model description adheres to or is compatible with a common accepted description language such as Web Services Description Language (WSDL) [23]. Such language describes the interface to Web applications in order to enable a proper query result.

Model modification/customization support enables a decision model to either be adjusted to fit an individual's specification or be modified from other existing models.

Firstly, a data model is required to provide *ease of modification,* i.e., a data model should be so simple that users take little time to understand, learn as well as extend the model with minimum effort. However, a tradeoff of the lack of informative internal configuration of the decision model should be considered. Secondly, *integrity constraints* are rules which constrain valid states. A data model pertains to at least two kinds of constraint. The first one is that of the built-into data model itself. For example, specification of the connection of the inheritance class assures that any change in the parent is automatically passed on to all its children. The second constraint is application-dependent and allows storage of user-defined assertion.

Finally, a data model should enable *model transformation across the modeling environment*. The reasoning behind this is simple. Two different organizations sharing the same decision model might be able to afford different kinds of solvers. The employment of XML-base language to represent a decision model would promote an opportunity to transform a model across the modeling environment, since XML is currently a standard way of encoding both text and data across diverse platforms [11].

Model composition/integration support: Even though the terms model composition and model integration are often used interchangeably, Krishnan and Chari [12] point out the differences between these two terms, i.e., that the former is to link independent models without modifying any of them [3], while models in the latter are modified.

It is also essential to emphasize that data models should allow for embedding a conflict resolving mechanism. Three major model integration conflicts [12] are *naming conflict* (use of the same name to refer to individuals), *granularity conflict* (different granularity basis of models, e.g., compute annually or quarterly) and *dimensional/unit of measurement conflict*. Nevertheless, the literature shows that model integration involves extensively human intervention or determination (such as [7]) and it is a challenging research area of the model management community.

Model execution support relates to the process of generating results. A data model should support transformation from the model to the solution space, i.e., it should not burden the creation of the representation/notation which a model solver can understand. Frameworks in a computation-centric group, i.e., algebraic modeling languages, are executable and are understood by the solvers. Unfortunately, they have the poor meta-level of a data model. In contrast, frameworks which employ a high conceptual level are hardly executable. Nevertheless, Web technologies have affected the execution of decision models, whence frameworks should be ready to survive in a Web computing environment.

Web technologies provide flexible computation and have seen advances in both server-side and client-side computing [2]. Exploiting server-side computing is so fertile that it gives an organization, which does not own a model solver, an opportunity to execute a model. Even so, a solver algorithm residing in a remote machine requires a specific data structure, while a model can be represented in many forms such as graphical and textual models. Hence, the issue of executing a model without heavy transformation should be considered.

Support of interoperability is essential in a Web-enable modeling environment so that a data model must support communication among software and hardware on heterogeneous vendors and heterogeneous platforms. In particular, an activity on the Web services platform demands for sending messages plain XML to ensure interoperability, since XML is the current standard for data representation and interchange among various Web applications. Altogether, if a modeling activity should

migrate to a Web services framework, model representation should abide by the exchange standard.

Support of representation of mathematical equations is the ability of a data model to express mathematical notation, including algebraic formulae and symbols. The intent of presentation of a mathematical equation is to allow quantitative inspection and understanding of a decision problem in a quantitative manner. Although the high conceptual level of model representation provides the model user with an easily-understood practice (such as a graphical language), some types of model users—especially MS/OR model expert—find it to be insufficient. Those high conceptual ones are just a front-end to the underlying mathematical expression.

As the modeling environment moves toward the Web, an approach to data modeling may consider employment of XML-based markup language such as MathML to represent mathematical expressions on the Web. It is expected that there will be on the Web more engines which can understand mathematical expressions, especially in MathML format.

Support of indexing is "one of the critical operations a modeling support system must perform", said Lazimy [15]. In any circumstance, a data model is required to support an indexing operation—the ability to characterize a set of data and allow performance of basic operations, i.e., declaration of index set, define index set and application of functions to an index set.

Use of index and subscript are mandatory in most algebraic modeling languages. They are exhaustively discussed in [18] and [8] as an important design issue of a modeling language for mathematical programming. Nevertheless, there are also frameworks which question the use of symbolic subscripts for indexing. Hence, Lin et al. [16] and Lazimy [15] propose a data model which can eliminate the use of subscripts by employment of relational database theory and object-oriented theory, respectively. In essence, these subscript-free modeling languages benefit the user in at least two ways, i.e., ease of model formulation and ease for whose not originally create the model to understand it.

Function on index set is another crucial operation. Common index set functions are "subset of", "union", "intersection" and "complement". Yet, not all existing frameworks support well index representation requirement, i.e., even if they can represent the index set, it is not simple to apply functions to it [3,5,11].

Support of representation of incomplete information—one of the major challenges of model representation—is the ability to represent unknown information. Note that in this context, unknown information differs from a "variable" (such as a decision variable) which is a basic model element. It is the ability to form and allow storing a model even when some parts of the model (such as the objective function) are still needed.

This is essential, because there is a chance that modelbases can encounter incomplete information, for example, the type of coefficient is unknown. A lack of such information should not prevent users from composing a model. Representation of incomplete information has not been adequately addressed in the literature. However, the authors wish to argue that data model developers should consider this issue.

4. Conclusions

Table 1. Excerpt of a comparison of representation frameworks to characteristics of data model

Approaches/frameworks \ Characteristics		Graphical view	A formal meta model	Query processing support	Ease of modification/ customization	Model information at a glass box view	Model execution support	Support of interoperability	Support of mathematical equations	Operations on index set	Support of representation of incomplete information
Data-centric	**Fourer [5]**	Yes*	No	Yes	No	Poor	Indirect**	No	No	No	No
Structure-centric	**OOSML [11]**	No	Yes	Yes	Yes	Medium	Indirect**	Yes	No	No	No
Abstraction-centric	**RMT [14]**	No	Yes	Yes	Yes	Medium	Indirect**	No	No	No	No
Logic-centric	**PM* [13]**	No	Yes	No	No	Medium	N/A	No	Partial	Partial	No
Computation-centric	**LPL [10]**	No	No	No	No	Excellent	Yes	No	Yes	Yes	No

* Employ a Table metaphor ** Require extensive transformation to do so

Although a diversity of data modeling frameworks exist in the modelbase community, a more expressive conceptual data model is still lacking. The need for it is driven by the impact of Internet and Web technologies. Data model requirements—a check list to determine whether a framework apprehends essential characteristics of a good data model—have been presented.

As a result of comparisons, no single framework wins general overall acceptance. (Table 1 provides an except of the comparison). In order to satisfy the requirements, either a new data model must be developed or an existing framework could produce a specialized representation of itself, in order to be compatible with the modern modeling environment.

In conclusion, since the complexity of the Web, which influences modelbases, will continue to increase, exploitation of an expressive data model is a necessity. The proposed data model requirements are therefore a first step toward the development of a new breed of representation framework for modelbases.

5. References

[1] Bharadwaj, A., Choobineh, J., Lo, A., and Shetty, B., "Model Management Systems: A Survey", *Annals of Operations Research*, 38, 1992, pp. 17-67

[2] Bhargava, H., Krishnan, R., Roehrig,S., Casey, M., Kaplan, D., and Müller, R., "Model Management in Electronic Markets for Decision Technologies: A Software Agent Approach", Proceedings of the *30th Hawaii International Conference on System Sciences*, Maui, HI, 1997

[3] Blanning, R., "An Entity-Relationship Approach to Model Management", *Decision Support Systems*, 2, 1986, pp. 65-72

[4] Collaud, G. and Pasquier-Boltuck, J., "gLPS: Graphical Tool for the Definition and Manipulation of Linear Problems", *European Journal of Operational Reaearch*, 72, 1994, pp. 277-286

[5] Fourer R., "Database Structure for Mathematical Programming Models", *Decision Support Systems*, 20, 1997, pp. 317-344

[6] Fourer, R., Gay, D., and Kernighan, B., "AMPL: A Mathematical Programming Language", *Management Science*, 36, 1990, pp. 519-554

[7] Geoffrion, A., "An Introduction to Structured Modeling", *Management Science*, 33:5, 1987, pp. 547-588

[8] Geoffrion, A., "Indexing in Modeling Language for Mathematical programming", *Management Science*, 38:3, 1992, pp. 325-344

[9] Guide to Available Mathematical software (GAMS), NIST, http://gams.nist.gov

[10] Hurlimann, T., "LPL: A Mathematical Modeling Language version 4.42", Department of Informatics, *University of Fribourg*, working paper, 2001

[11]Kim, H., "An XML-based modeling language for the open interchange of decision models", *Decision Support Systems*, 31 , 2001, pp. 429-441

[12] Krishnan, K., and Chari, K., "Model Management: Survey, Future Research Directions and a Bibliography', *The Interactive Transactions of OR/ MS (ITORMS)*, Vol 3, 2000

[13] Krishnan, R., "A logic Modeling Language for Automated Model Construction", *Decision Support Systems*, 6 , 1990, pp. 123-152

[14] Kwon, O. and Park, S., "RMT: A modeling support system for model reuse", *Decision Support Systems*, 16, 1996, pp. 131-153

[15] Lazimy, R., "Object-Oriented Modeling Support System: Model Representation, and Incremental Modeling", Proceedings of *the Twenty-Sixth Annual Hawaii International Conference on System Science*, 1993, pp. 445-459

[16] Lin, S., Schuff, D., and Louis, R., "Subscript-Free Languages: A Tool for Facilitating the Formulation and Use of Models", *European Journal of Operational Research*, 123:3, 1998, pp. 614-627

[17] Mannino, M., Greenberg, B., and Hong, S., "Model Libraries: Knowledge Representation and reasoning", *ORSA Journal on Computing*, 2:3, 1990, pp. 287-301

[18] Maturana, S., "Issues in the design of modeling languages for mathematical programming", *European Journal of Operational Research*, 72, 1994, pp. 243-261

[19] Muraphan N., "Model Representation with a Resource Description Framework", *Asian Institute of Technology*, Thailand, thesis IM-99-7, 1999

[20] NEOS Server for optimizing, http://www-neos.mcs.anl.gov/neos/

[21] Universal Description, Discovery and Integration (UDDI), http://www.uddi.org

[22] Web Services Activity, W3C, http://www.w3.org/2002/ws/

[23] Web Services Description Language (WSDL), http://w3.org/TR/wsdl

Using WI Technology to Develop Intelligent Enterprise Portals

Ning Zhong, Hikaru Ohara, Tohru Iwasaki
Dept. of Information Eng., Maebashi Institute of Technology
460-1 Kamisadori-Cho, Maebashi-City 371-0816, Japan
E-mail: zhong@maebashi-it.ac.jp

Yiyu Yao
Dept. of Computer Science, University of Regina, Canada
Regina, Saskatchewan, Canada S4S 0A2
E-mail: yyao@cs.uregina.ca

Abstract

Web Intelligence (WI) presents excellent opportunities and challenges for the research and development of new generation of Web-based information processing technology, as well as for exploiting Web-based advanced applications. The paper investigates how to develop intelligent enterprise portals that enable e-business intelligence and deal with the scalability and complexity of real world, efficiently and effectively, by using WI technology.

1 Introduction

With the rapid growth of the Web, research and development on Web Intelligence (WI) have received much attention [16, 29, 35, 39, 40, 42, 43]. There is great potential for WI to make useful contributions to e-business (include e-commerce), e-science, e-learning, e-government, e-society, and so on. The WI technology revolutionizes the way in which information is gathered, stored, processed, presented, shared, and used by virtualization, globalization, standardization, personalization, and e-portals.

One of the most sophisticated applications on the Web today is *enterprise information portals*, which is a single gateway to personalized information needed to make informed business decisions, by operating with state-of-the-art markup languages to search, retrieve and repackage data. The enterprise portals are being developed into an even more powerful center based on component-based applications called Web Services [1, 22]. These portals solve business problems by offering advanced features that enable B2B transactions and automate business processes. Organizations are now turning portals outward as a means of enhancing relationships with customers and partners. In other words, portals are the cornerstones to success in making informed business decisions and in the move to e-business intelligence. They unify access to business contents, trading partners and customers need to do their jobs: Web data, workgroup information, business intelligence, front- and back-office applications, expertise and even data in legacy systems. Portals improve ROI (return on investment) through improved collaboration and communication, smarter decision-making, increased productivity, and easier access to business information, applications and expertise [26].

The rest of this paper is organized as follows. Section 2 discusses how to develop intelligent enterprise portals by using WI technology such as Web mining and semantic social networks. Section 3 describes how to offer advanced features that enable e-business intelligence such as targeted marketing that is a new business model by an interactive one-to-one communication between marketer and customer, as well as deal with the scalability and complexity of real world, efficiently and effectively. Finally, Section 4 gives conclusions.

2 How to Develop Intelligent Enterprise Portals?

An enterprise portal enables a company or an organization to create a *virtual enterprise* where key production steps are outsourced to partners. Many organizations are implementing a corporate portal first and are then growing this solution into more of an intelligent B2B portal. By using a portal to tie in back-end enterprise systems, a company can manage the complex interactions of the virtual enterprise partners through all phases of the value and supply chain.

2.1 Virtual Industry Park: An Example of Enterprise Portals

As an example for developing enterprise portals by using WI technology, we here discuss how to construct an intelligent virtual industry park (VIP) that has been developing in our group. The VIP portal is a website in which all of the contents related to the mid-sized/small-scale companies in Maebashi city can be accessed.

The construction process can be divided into three phases. We first constructed a basic system including the fundamental functions such as the interface for dynamically registering/updating enterprise information, the database for storing the enterprise information, the automatic generation/modification of enterprise homepages, and the domain-specific, keyword-based search engine. When designing the basic system, we also started by analyzing customer performance: what has each customer bought, over time, total volumes, trends, and so on.

Although the basic system can work as a whole-one, we now need to know not only past performance on the business front, but also how the customer or prospect enters our VIP portal in order to target products and manage promotions and marketing campaigns. To the already demanding requirement to capture transaction data for further analysis, we now also need to use the Web usage mining techniques to capture the clicks of the mouse that define where the visitor has been on our website. What pages has he or she visited? What is the semantic association between the pages he or she visited? Is the visitor familiar with the Web structure? Or is he or she a new user or a random one? Is the visitor a Web robot or other users? In search for the holy grail of "stickiness", we know that a prime factor is *personalization* for:

- making a dynamic recommendation to a Web user based on the user profile and usage behavior,
- automatic modification of a website's contents and organization,
- combining Web usage data with marketing data, product data, customer data, among others, to give information about how visitors used a website for marketers.

Hence, we need to extend the basic VIP system by adding more advanced functions such as Web mining, the ontologies-based search engine, as well as automatic email filtering and management.

Finally, a portal for e-business intelligence can be implemented by adding e-business related application functions such as customer relationship management (CRM), targeted marketing, electronic data interchange (EDI), and security solution.

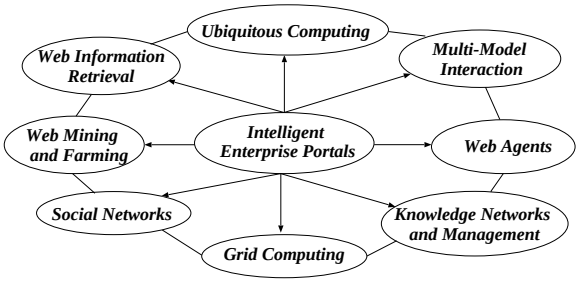


Figure 1. An intelligent enterprise portals centric schematic diagram of WI technology

2.2 An Intelligent Enterprise Portal Centric Schematic Diagram of WI Technology

From the example stated in the above subsection, we can see that developing an intelligent enterprise portal needs to apply results from existing disciplines of AI and IT to a totally new domain. On the other hand, the WI technology is also expected to introduce new problems and challenges to the established disciplines on the new platform of the Web and Internet. That is, WI is an enhancement or an extension of AI and IT.

In order to study advanced WI technology systematically, and develop advanced Web-based intelligent enterprise portals and information systems, we provide a schematic diagram of WI technology from a Web-based, intelligent enterprise portals centric perspective in Fig. 1, in which directed lines denote that the development of intelligent enterprise portals needs to be supported by vairous WI related techniques, and undirected lines denote that the component WI techniques are relevant each other.

2.3 Web Mining and Farming

The enterprise portal based e-business activity that involves the end user is undergoing a significant revolution [24]. The ability to track users' browsing behavior down to individual mouse clicks has brought the vendor and end customer closer than ever before. It is now possible for a vendor to personalize his product message for individual customers at a massive scale. This is called *targeted marketing* (or *direct marketing*) [27, 43]. Web mining and Web usage analysis play an important role in e-business for customer relationship management (CRM) and targeted marketing.

Web mining is the use of data mining techniques to automatically discover and extract information from large Web data repositories such as Web documents and services [12, 24, 30, 34]. Web mining research is at the crossroads of research from several research communities, such

as database, information retrieval, artificial intelligence, especially the subareas of machine learning and natural language processing. Web mining can be divided into four classes of data available on the Web:

- Web content: the data that constitutes the Web pages and conveys information to the users, i.e., html, graphical, video, audio files of a Web page.
- Web structure: the data that formulates the hyper-link structure of a website and the Web, i.e., various HTML tags used to link one page to another and one website to another website.
- Web usage: the data that reflects the usages of Web resources, i.e., entries in Web browser's history and Internet temporary files, proxy server and Web server logs.
- Web user profile: the data that provides demographic information about users of the website, i.e., users' registration data and customers' profile information.

Furthermore, Web content, structure, and usage information, in many cases, are co-present in the same data file. For instance, the file names appeared in the log files and Web structure data contain useful content information. One may safely assume that a file named "WebLogMining.html" must contain information about Web log mining. Similarly, the categories of Web mining cannot be considered exclusive or isolated from each other. Web content mining sometimes must utilize Web structure data in order to classify a Web page. In the same way, Web usage mining sometimes has to make use of Web content data and Web structure information.

A challenge is to explore the connection between Web mining and the related agent paradigm such as Web farming that is the systematic refining of information resources on the Web for business intelligence [9]. Web farming extends Web mining into an evolving breed of information analysis in a whole process of Web-based information management including seeding, breeding, gathering, harvesting, refining, and so on.

2.4 Semantic Social Networks for Intelligent Enterprise Portals.

Developing intelligent enterprise portals need to study both centralized and distributed information structures on the Web. Information/knowledge on the Web can be either globally distributed throughout the Web within multi-layer over the infrastructure of Web protocols, or located locally, centralized on an intelligent portal providing Web services (i.e. the intelligent service provider) that is integrated to its own cluster of specialized intelligent applications. However, each approach has a serious flaw. As pointed out by Alesso and Smith [1], the intelligent portal approach limits its uniformity and access, while the global semantic Web approach faces combinatory complexity limitations.

A way to solve the above issue is to develop and use the *Problem Solver Markup Language* (PSML), for collecting globally distributed contents and knowledge from Web-supported, semantic social networks and incorporating them with locally operational knowledge/databases in an enterprise or community for local centralized, adaptable Web intelligent services [39, 42].

The core of PSML is distributed inference engines that can perform automatic reasoning on the Web by incorporating contents and meta-knowledge autonomically collected and transformed from the semantic Web with locally operational knowledge-data bases. A feasible way to implement such PSML is to use existing Prolog-like logic language plus dynamic contents and meta-knowledge collection and transformation agents.

In our experiments, we use KAUS for representation local information sources and for inference and reasoning. KAUS is a knowledge-based system developed in our group which involves knowledge-bases on the basis of Multi-Layer Logic and databases based on the relational data model [18, 19, 28]. KAUS enables representation of knowledge and data in the first-order logic with data structure in multi-level and can be easily used for inference and reasoning as well as transforming and managing both knowledge and data.

By using this information transformation approach, the dynamic, global information sources on the Web can be combined with the local information sources in an enterprise portal together for decision-making and e-business intelligence.

3 Data Mining for Web-Based Targeted Marketing

An enterprise portals for business intelligence needs the function of Web-based targeted marketing, which is integrated with other functions of Web intelligence such as Web mining, the ontologies-based search engine, personalized recommendation, as well as automatic email filtering and management [42].

Targeted marketing aims at obtaining and maintaining direct relationships between suppliers and buyers within one or more product/market combinations. Targeted marketing becomes more and more popular because of the increased competition and the cost problem.

Furthermore, the scope of targeted marketing can be expanded from considering only how products are distributed,

to include enhancing the relationships between an organization and its customers [11] since the strategic importance of long-term relationships with customers. In other words, once customers are acquired, customer retention becomes the target. Retention through customer satisfaction and loyalty can be greatly improved by acquiring and exploiting knowledge about these customers and their needs. Such targeted marketing is called "targeted relationship marketing" or "customer relationship management (CRM)" [25].

3.1 The Market Value Functions (MVF) Model

Targeted marketing is an important area of applications for data mining, data warehousing, statistical pattern recognition, and intelligent agents [21]. Although standard data mining methods may be applied for the purpose of targeted marketing, many specific algorithms need to be developed and applied for direct marketer to make decisions effectively.

Let us consider now a typical problem of targeted marketing. Suppose there is a health club that needs to expand its operation by attracting more members. Assume that each existing member is described by a finite set of attributes. It is natural to examine existing members in order to identify their common features. Information about the health club may be sent to non-members who share the same features of members or similar to members. Other examples include promotion of special types of phone services, and marketing of different classes of credit cards. In this case, we explore the relationships (similarities) between people (objects) based on their attribute values. The underlying assumption is that *similar type of people tend to make similar decisions and to choose similar services*. Techniques for mining association rules may not be directly applicable to this type of targeted marketing. One may produce too many or too few rules. The selection of a good set of rules may not be an easy task. Furthermore, the use of the derived rules may produce too many or too few potential new members.

In order to solve the issue, we proposed a new model for targeted marketing by focusing on the issues of knowledge representation and computation of market values [29, 30]. More specifically, we assume that each object is represented by its values on a finite set of attributes. We further assume that market values of objects can be computed using a linear market value function. Thus, we may consider the proposed model to be a *linear* model, which is related to, but different from, the linear model for information retrieval.

Let U be a finite universe of objects. Elements of U may be customers or products we are interested in market oriented decision making. The universe U is divided into three pair-wise disjoint classes, i.e., $U = P \cup N \cup D$. The sets P, N and D are called *positive*, *negative*, and *don't know* instances, respectively. Take the earlier health club example, P is the set of current members, N is the set of people who had previously refused to join the club, and D is the set of the rest. The set N may be empty. A targeted marketing problem may be defined as finding elements from D, and possibly from N, that are similar to elements in P, and possibly dissimilar to elements in N. In other words, we want to identify elements from D and N that are more likely to become new members of P. We are interested in finding a market value function so that elements of D can be ranked accordingly.

Information about objects in a finite universe is given by an information table [20, 31]. The rows of the table correspond to objects of the universe, the columns correspond to attributes, and each cell is the value of an object with respect to an attribute. Formally, an information table is a quadruple:

$$S = (U, At, \{V_a \mid a \in At\}, \{I_a \mid a \in At\}),$$

where U is a finite nonemptyset of objects, At is a finite nonempty set of attributes, V_a is a nonempty set of values for $a \in At$, $I_a : U \rightarrow V_a$ is an information function for $a \in At$. Each information function I_a is a total function that maps an object of U to exactly one value in V_a. An information table represents all available information and knowledge. Objects are only perceived, observed, or measured by using a finite number of properties [20].

A market value function (MVF) is a real-valued function from the universe to the set of real numbers, $r : U \longrightarrow \Re$. In the context of information retrieval, the values of r represent the potential usefulness or relevance of documents with respect to a query. According to the values of r, documents are ranked. For the targeted marketing problem, a market value function ranks objects according to their potential market values. For the health club example, a market value function ranks people according to their likelihood of becoming a member of the health club. The likelihood may be estimated based on its similarity to a typical member of P.

We studied the simplest form of market value functions, i.e., the linear discriminant functions. Let $u_a : V_a \longrightarrow \Re$ be a utility function defined on V_a for an attribute $a \in At$. The utility $u_a(\cdot)$ may be positive, negative, or zero. For $v \in V_a$, if $u_a(v) > 0$ and $I_a(x) = v$, i.e., $u_a(I_a(x)) > 0$, then attribute a has a positive contribution to the overall market value of x. If $u_a(I_a(x)) < 0$, then a has a negative contribution. If $u_a(I_a(x)) = 0$, then a has no contribution. The pool of contributions from all attributes is computed by a linear market value function of the following form:

$$r(x) = \sum_{a \in At} w_a u_a(I_a(x)), \tag{1}$$

where w_a is the weight of attribute a. Similarly, the weight w_a may be positive, negative and zero. Attributes with

larger weights (absolute value) are more important, and attributes with weights close to zero are not important. The overall market value of x is a weighted combination of utilities of all attributes. By using a linear market value function, we have implicitly assumed that contributions made by individual attributes are independent. Such an assumption is commonly known as utility independence assumption. Implications of utility independence assumption can be found in literature of multi-criteria decision making [8].

The market value model proposes a linear model to solve the target selection problem of targeted marketing by drawing and extending result from information retrieval [29, 30]. It is assumed that each object is represented by values of a finite set of attributes. A market value function is a linear combination of utility functions on attribute values, which depends on two parts: *utility function* and *attribute weighting*.

The market value function has some advantages: firstly, it can rank individuals according to their market value instead of classifying; secondly, the market value functions is interpretable; thirdly, the system of the market value function can perform without expertise.

3.2 Multi-Aspect Analysis in Multiple Data Sources

Generally speaking, customer data can be obtained from multiple customer touchpoints [10]. In response, multiple data sources that are obtained from multiple customer touchpoints, including the Web, wireless, call centers, and brick-and-mortar store data, need to be integrated into a single data warehouse that provides a multi-faceted view of their customers, their preferences, interests, and expectations for multi-aspect analysis. Hence, a multi-strategy and multi-agent data mining framework is required [35, 36].

One of main reasons for developing a multi-agent data mining system is that we cannot expect to develop a single data mining algorithm that can be used to solve all targeted marketing problems since complexity of the real-world applications. Hence, various data mining agents need to be cooperatively used in the multi-step data mining process for performing multi-aspect analysis as well as multi-level conceptual abstraction and learning.

The other reason for developing a multi-agent data mining system is that when performing multi-aspect analysis for complex targeted marketing problems, a data mining task needs to be decomposed into sub-tasks. Thus these sub-tasks can be solved by using one or more data mining agents that are distributed over different computers. Thus the decomposition problem leads us to the problem of distributed cooperative system design.

In the VIP stated in Section 2.1, for instance, there are mainly three kinds of data sources considered, namely, customer database, products database, and Web framing database. Furthermore, in addtion to the MVF based data mining methed mentioned in Section 3.1, we have developed various data mining methods, such as the GDT-RS inductive learning system for discovering classification rules, the LOI (learning with ordered information) for discovering important features, as well as the POM (peculiarity oriented mining) for finding peculiarity data/rules, to deal with each of them, respectively [23, 30, 37, 41, 44].

However, when we try to integrate the three kinds of data sources together into the advanced VIP system, we must know how to interact with each of those sources in order to extract the useful pieces of information which then have to be combined for building the expected answer to the initial request. Hence, the core question is how to manage, represent, integrate and use the information coming from multiple data sources.

We proposed to use the RVER (Reverse Variant Entity-Relationship) model to represent the conceptual relationships among various types of interesting data mined from multiple data sources [43]. Furthermore, the advanced rules hidden inter-multiple data sources can be learned from the RVER model. Besides the RVER model, ontologies are also used for multi-data source description and integration [38].

Here we would like to emphasize that how to manage, analyse, and use the information intelligently from different data sources is a problem not only exists in the e-business field, but also exists in the e-science, e-learning, e-government, and all intelligent Web information systems [46]. The development of enterprise portals and e-business intelligence is a good example for trying to solve this kind of problem.

3.3 Building a Data Mining Grid

In order to implement an enterprise portal (e.g. the VIP discussed in Section 2.1) for Web-based targeted marketing and business intelligence, a new infrastructure and platform as the middleware is required to deal with large, distributed data sources for multi-aspect analysis. Our methodology is to create a grid-based, organized society of data mining agents, called a *Data Mining Grid* on the grid computing platform (e.g. the Globus toolkit) [2, 4, 7]. This means

- To develop various data mining agents for different targeted marketing tasks;
- To organize the data mining agents into a grid with multi-layer under the Web as a middleware that understands the user's questions, transforms them to data mining issues, discovers the resources and information about the issues, and get a composite answer or solution;

- To use the grid of data mining agents for multi-aspect analysis in distributed, multiple data sources;
- To manage the grid of data mining agents by a multi-level control authority.

That is, the data mining grid is made of many smaller components that are called *data mining agents*. Each agent by itself can only do some simple thing. Yet when we join these agents in a *grid*, this leads to implement more complex targeted marketing and business intelligence tasks.

4 Conclusions

Web Intelligence (WI) is a newly recognized, very dynamic field. It is the key as well as the most urgent research field of IT in the era of the World Wide Web, the wisdom Web, knowledge grid, intelligent agents, and ubiquitous computing and social intelligence. The WI technology will produce the new tools and infrastructure components necessary to create intelligent enterprise portals that serves its users *wisely* for e-business intelligence.

References

[1] H.P. Alesso and C.F. Smith: *The Intelligent Wireless Web* (Addison-Wesley, 2002)

[2] F. Berman: From TeraGrid to Knowledge Grid, *CACM*, 44, 27-28 (2001)

[3] T. Berners-Lee, J. Hendler, O. Lassila: The Semantic Web, *Scientific American*, 284, 34-43 (2001)

[4] M. Cannataro and D. Talia: The Knowledge Grid, *CACM*, 46, 89-93 (2003)

[5] David Shepard Associates: *The New Direct Marketing*, McGraw-Hill (1999)

[6] D. Fensel: *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce* (Springer, 2001)

[7] I. Foster and C. Kesselman: *The Grid: Blueprint for a New Computing Infrastructure* (Morgan Kaufmann, 1999)

[8] P.C. Fishburn: Seven Independence Concepts and Continuous Multiattribute Utility Functions, *Journal of Mathematical Psychology*, Vol. 11 (1974) 294-327

[9] R.D. Hackathorn: *Web Farming for the Data Warehouse* (Morgan Kaufmann, 2000)

[10] S.Y. Hwang, E.P. Lim, J.H. Wang, and J. Srivastava: *Proc. PAKDD 2002 Workshop on Mining Data across Multiple Customer Touchpoints for CRM* (2002)

[11] W. Klosgen and J.M. Zytkow: *Handbook of Data Mining and Knowledge Discovery* (Oxford University Press, 2002)

[12] R. Kosala and H. Blockeel: Web Mining Research: A Survey, *ACM SIGKDD Explorations Newsletter*, 2, 1-15 (2000)

[13] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins: The Web and Social Networks. *IEEE Computer* Special Issue on Web Intelligence, 35(11) 32-36 (November 2002)

[14] J. Liu, N. Zhong (eds.): *Intelligent Agent Technology: Systems, Methodologies, and Tools* (World Scientific, 1999)

[15] J. Liu, N. Zhong, Y.Y. Tang, P.S.P. Wang (eds.): *Agent Engineering* (World Scientific, 2001)

[16] J. Liu, N. Zhong, Y.Y. Yao, Z.W. Ras: The Wisdom Web: New Challenges for Web Intelligence (WI), *Journal of Intelligent Information Systems*, 20(1) 5-9 (Kluwer Academic Publishers, 2003)

[17] Z. Lu, Y.Y. Yao, N. Zhong: Web Log Mining. N. Zhong, J. Liu, Y.Y. Yao (eds.) *Web Intelligence*, 172-194 (Springer, 2003)

[18] S. Ohsuga and H. Yamauchi: Multi-Layer Logic - A Predicate Logic Including Data Structure as Knowledge Representation Language. *New Generation Computing*, 3(4) 403-439 (Springer, 1985)

[19] S. Ohsuga: Framework of Knowledge Based Systems - Multiple Meta-Level Architecture for Representing Problems and Problem Solving Processes. *Knowledge Based Systems*, 3(4) 204-214 (Elsevier, 1990)

[20] Z. Pawlak: *Rough Sets, Theoretical Aspects of Reasoning about Data*, (Kluwer, 1991)

[21] P. Van Der Putten: Data Mining in Direct Marketing Databases, W. Baets (ed). *Complexity and Management: A Collection of Essays* (World Scientific, 1999)

[22] P. Raghavan: Social Networks: From the Web to the Enterprise, *IEEE Internet Computing*, 6(1), 91-94 (2002)

[23] Y. Sai, Y.Y. Yao, and N. Zhong: Data Analysis and Mining in Ordered Information Tables, *Proc. 2001 IEEE International Conference on Data Mining (ICDM'01)* (IEEE Computer Society Press, 2001) 497-504.

[24] J. Srivastava, R. Cooley, M. Deshpande, P. Tan: Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations, Newsletter of SIGKDD*, 1, 12-23 (2000)

[25] R. Stone: *Successful Direct Marketing Methods*, 6th ed. NTC Business Books (1996)

[26] M. Wesker: Using a Portal to Solve Business Problems, KMWorld (July/August 2001)

[27] A.R. Simon, S.L. Shaffer: *Data Warehousing and Business Intelligence for e-Commerce* (Morgan Kaufmann, 2001)

[28] H. Yamauchi and S. Ohsuga: Loose coupling of KAUS with existing RDBMSs. *Data and Knowledge Engineering*, 5(4) 227-251 (North-Holland, 1990)

[29] Y.Y. Yao, N. Zhong, J. Liu, S. Ohsuga: Web Intelligence (WI): Research Challenges and Trends in the New Information Age. N. Zhong, Y. Y. Yao, J. Liu, S. Ohsuga (eds.) *Web Intelligence: Research and Development*, LNAI 2198, (Springer, 2001) 1-17

[30] Y.Y. Yao, N. Zhong, J. Huang, C. Ou, C. Liu: Using Market Value Functions for Targeted Marketing Data Mining, *International Journal of Pattern Recognition and Artificial Intelligence*, 16(8) 1117-1131 (World Scientific, 2002)

[31] Y.Y. Yao and N. Zhong: Granular Computing Using Information Tables, Lin, T.Y., Yao, Y.Y., Zadeh, L.A. (eds.) *Data Mining, Rough Sets and Granular Computing* (Physica-Verlag, 2002) 102-124

[32] Y. Ye, J. Liu, A. Moukas: Agents in Electronic Commerce. *Special Issue on Intelligent Agents in Electronic Commerce, Electronic Commerce Research Journal* (2001)

[33] N. Zhong, C. Liu, S. Ohsuga: Dynamically Organizing KDD Process in a Multi-Agent Based KDD System, J. Liu, N. Zhong, Y.Y. Tang, P. Wang (eds.) *Agent Engineering* (World Scientific, 2001) 93-122

[34] N. Zhong: Knowledge Discovery and Data Mining, *The Encyclopedia of Microcomputers*, 27(Supplement 6) 235-285 (Marcel Dekker, 2001)

[35] N. Zhong, Y.Y. Yao, J. Liu, S. Ohsuga (eds.): *Web Intelligence: Research and Development*, LNAI 2198 (Springer, 2001)

[36] N. Zhong, J. Liu, S. Ohsuga, J. Bradshaw (eds.): *Intelligent Agent Technology: Research and Development* (World Scientific, 2001)

[37] N. Zhong, J.Z. Dong, C. Liu, S. Ohsuga: A Hybrid Model for Rule Discovery in Data, *Knowledge Based Systems, An International Journal*, 14(7) 397-412 (Elsevier Science, 2001)

[38] N. Zhong: Representation and Construction of Ontologies for Web Intelligence, *International Journal of Foundations of Computer Science (IJFCS)*, 13(4) 555-570 (World Scientific, 2002)

[39] N. Zhong, J. Liu, Y.Y. Yao: In Search of the Wisdom Web. *IEEE Computer*, 35(11) 27-31 (2002)

[40] N. Zhong, J. Liu, Y.Y. Yao (eds.): *Special issue on Web Intelligence (WI), IEEE Computer*, 35(11) (November 2002).

[41] N. Zhong, Y.Y. Yao, J.Z. Dong, S. Ohsuga: "Gastric Cancer Data Mining with Ordered Information", J.J. Alpigini et al (eds.) *Rough Sets and Current Trends in Computing*, LNAI 2475, Springer (2002) 467-478.

[42] N. Zhong, J. Liu, Y.Y. Yao (eds.): *Web Intelligence* (Springer, 2003)

[43] N. Zhong, J. Liu, Y.Y. Yao: Web Intelligence (WI): A New Paradigm for Developing the Wisdom Web and Social Network Intelligence. N. Zhong, J. Liu, Y.Y. Yao (eds.): *Web Intelligence*, 1-16 (Springer, 2003)

[44] N. Zhong, Y.Y. Yao, M. Ohshima: Peculiarity Oriented Multi-Database Mining, *IEEE Transaction on Knowlegde and Data Engineering*, 15(4) (2003) 952-960.

[45] N. Zhong: Towards Web Intelligence, Ernestina, M., Javier, S., Piotr, S.S. (eds.) *Advances in Web Intelligence*, LNAI 2663, Springer (2003) 1-15.

[46] N. Zhong, J.L. Wu, C. Liu: Building a Data Mining Grid for Multiple Human Brain Data Analysis, *Proc. International Workshop on Knowledge Grid and Grid Intelligence*, Halifax, Canada (2003).

Racer: An OWL Reasoning Agent for the Semantic Web

Volker Haarslev[†] and Ralf Möller[‡]
[†]Concordia University, Montreal, Canada (haarslev@cs.concordia.ca)
[‡]University of Applied Sciences, Wedel, Germany (rmoeller@fh-wedel.de)

Abstract

Racer, which can be considered as a core reasoning agent for the semantic web, is briefly described. Racer currently supports a wide range of inference services about ontologies specified in the Ontology Web Language (OWL). These services are made available to other agents via network based APIs. Racer is currently used by various clients such as ontology editors, ontology development and visualization tools, and a first web-based prototype for exploration and analysis of OWL ontologies.

1 Introduction

The Semantic Web initiative defines important challenges for knowledge representation and inference systems. Recently, several standards for representation languages have been proposed. One of the standards for the Semantic Web is the Resource Description Framework (RDF [1]). Since RDF is based on XML it shares its document-oriented view of grouping sets of declarations or statements. With RDF's triple-oriented style of data modeling, it provides means for expressing graph-structured data over multiple documents (whereas XML can only express graph structures within a specific document). As a design decision, RDF can talk about everything. Hence, in principle, statements in documents can also be referred to as resources. In particular, conceptual domain models can be represented as RDF resources. Conceptual domain models are referred to as "vocabularies" in RDF. Specific languages are provided for defining vocabularies (or ontologies). An extension of RDF for defining ontologies is RDF Schema (RDFS [2]) which only can express conceptual modeling notions such as generalization between concepts (aka classes) and roles (aka properties). For properties, domain and range restrictions can be specified. Thus, the expressiveness of RDFS is very limited. A much more expressive representation language is OWL (Ontology Web Language) [3]. Although still in a very weak way, based on XML-Schema, OWL also provides for means of dealing with data types known from programming languages.

The representation languages mentioned above are defined with a model-theoretic semantics. In particular, for the language OWL, a semantics was defined such that very large fragments of the language can be directly expressed using so-called description logics (see [4]). The fragment is called OWL DL. With some restrictions that are discussed below one can state that the logical basis of OWL can be characterized with the description logic $\mathcal{SHIQ}(\mathcal{D}_n)^-$ [5]. This means, with some restrictions, OWL documents can be automatically translated to $\mathcal{SHIQ}(\mathcal{D}_n)^-$ T-boxes. The RDF-Part of OWL documents can be translated to $\mathcal{SHIQ}(\mathcal{D}_n)^-$ A-boxes.

2 Racer: An OWL Reasoner

The logic $\mathcal{SHIQ}(\mathcal{D}_n)^-$ is interesting for practical applications because highly optimized inference systems are available (e.g., Racer [6]). Racer is freely available for research purposes and can be accessed by standard HTTP or TCP protocols (the Racer program is subsequently also called Racer server). Racer can read OWL knowledge bases either from local files or from remote Web servers (i.e., a Racer server is also a HTTP client). In turn, other client programs that need inference services can communicate with a Racer server via TCP-based protocols. OilEd [7] can be seen as a specific client that uses the DIG protocol [8] for communicating with a Racer server, whereas RICE [9] is another client that uses a TCP protocol providing extensive query facilities (see below).

The DIG protocol [8] is a XML- and HTTP-based standard for connecting client programs to description logic inference engines. DIG allows for the allocation of knowledge bases and enables clients to pose standard description logic queries. As a standard and a least common denominator it cannot encompass all possible forms of system-specific statements and queries. Let alone long term query processing instructions (e.g., exploitation of query subsumption, computation of indexes for certain kinds of queries etc., see [10]). Therefore, Racer also provides a TCP-based interface in order to receive instructions (statements) and queries. For interactive use, the language supported by Racer is not XML- or RDF-based. The advantage is that users can spontaneously type queries which can be directly sent to a Racer server. However, the Racer TCP interface can be very easily accessed from Java or C++ application programs as well. For both languages corresponding APIs are available.

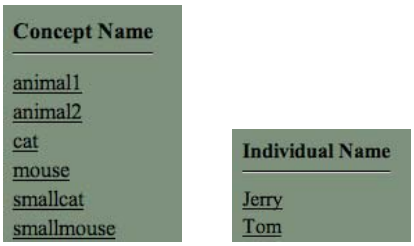


Figure 1: The lists of known concepts and individuals.

3 Some Supported Inference Services

In description logic terminology, a tuple consisting of a T-box and an A-box is referred to as a knowledge base. An individual is a specific named object. OWL also allows for individuals in concepts (and T-box axioms). For example, expressing the fact that all humans stem from a single human called ADAM requires to refer to an individual in a concept (and a T-box). Only part of the expressivity of individuals mentioned in concepts can be captured with A-boxes. However, a straightforward approximation exists (see [11]) such that in practice suitable $\mathcal{SHIQ}(\mathcal{D}_n)^-$ ontologies can be generated from an OWL document. Racer can directly read OWL documents and represent them as description logic knowledge bases (aka ontologies). In the following a selection of supported T-box queries is briefly introduced.

- Concept consistency: Is the set of objects described by a concept empty?
- Concept subsumption: Is there a subset relationship between the set of objects described by two concepts?
- Find all inconsistent concepts mentioned in a T-box. Inconsistent concepts might be the result of modeling errors.
- Determine the parents and children of a concept: The parents of a concept are the most specific concept names mentioned in a T-box which subsume the concept. The children of a concept are the most general concept names mentioned in a T-box that the concept subsumes.

Whenever a concept is needed as an argument for a query, not only predefined names are possible. If also an A-box is given, among others, the following types of A-box queries are possible:

- Check the consistency of an A-box: Are the restrictions given in an A-box w.r.t. a T-box too strong, i.e., do they contradict each other? Other queries are only possible w.r.t. consistent A-boxes.
- Instance testing: Is the object for which an individual stands a member of the set of objects described by a certain query concept? The individual is then called an instance of the query concept.
- Instance retrieval: Find all individuals from an A-box such that the objects they stand for can be proven to be a member of a set of objects described by a certain query concept.
- Computation of the direct types of an individual: Find the most specific concept names from a T-box of which a given individual is an instance.
- Computation of the fillers of a role with reference to an individual.

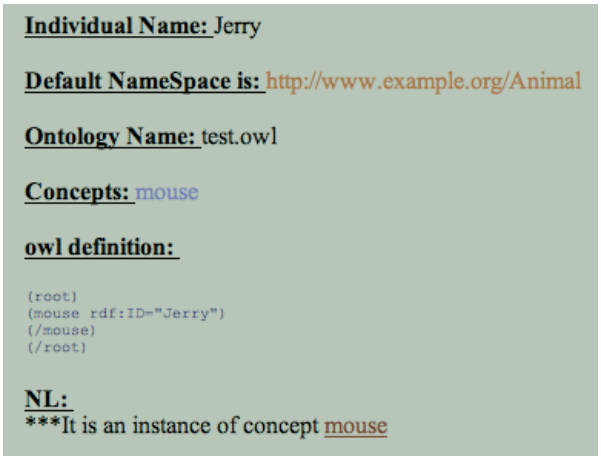


Figure 2: Information about the individual JERRY.

Given the background of description logics, many application papers demonstrate how these inference services can be used to solve actual problems with OWL knowledge bases. The query interface is extensively used by RICE and a tool for ontology exploration and analysis that is introduced in the next section.

4 Ontology Exploration and Analysis Tool

This section presents a first prototype for an ontology exploration and analysis tool designed for OWL. This tool parses OWL files and presents a “natural language” interface for exploring and analyzing ontologies. This is facilitated by using the inference services of Racer. We demonstrate this with a simple browsing scenario using a small ontology (for sake of brevity) about the cartoon characters Tom and Jerry.

Let us assume a corresponding ontology has been loaded. We start browsing the lists of all concept and individual names declared in this ontology (see Figure 1). We are interested in the individual JERRY (shown in Figure 2) and learn that JERRY is an instance of the class MOUSE. We know that cats usually eat mice, so we decide to inspect the description of CAT (see Figure 3) by clicking on the corresponding hyperlink in Figure 1.

Figure 3 shows a description of the class CAT. This description displays results from the inference services of Racer and consists of the following information.

Concept Name: cat

Default NameSpace is: http://www.example.org/Animal#

Ontology Name: test.owl

Ancestors: ((*TOP* TOP) (|http://www.example.org/Animal#animal2|) (|http://www.example.org/Animal#animal1|))

Descendants: ((*BOTTOM* BOTTOM) (|http://www.example.org/Animal#smallcat|))

Parents: ((|http://www.example.org/Animal#animal2|) (|http://www.example.org/Animal#animal1|))

Children: ((|http://www.example.org/Animal#smallcat|))

Roles used by this concept:
eat-mouse

Individuals of this concept:
Tom

owl definition:

```
(root)
(owl:Class rdf:ID="cat")
 (rdfs:subClassOf rdf:resource="#animal1")
 (/rdfs:subClassOf)
 (rdfs:subClassOf rdf:resource="#animal2")
 (/rdfs:subClassOf)
 (owl:Restriction)
  (owl:onProperty rdf:resource="#eat-mouse")
  (/owl:onProperty)
  (owl:minCardinality)
  1
  (/owl:minCardinality)
 (/owl:Restriction)
(/owl:Class)
(/root)
```

NL:
It is the subclass of animal1
It is the subclass of animal2
it has a filler in the role ' eat-mouse
and has at leaset 1 instance.

Figure 3: Description of class CAT.

- Concept (class) name
- Default name space
- Ontology filename
- The names of the ancestor classes (TOP, which is a synonym for THING, and ANIMAL1, ANIMAL2)
- The names of the descendent classes (BOTTOM, which is a synonym for NOTHING, and SMALLCAT)
- The parents (ANIMAL1 and ANIMAL2)
- The children (SMALLCAT)
- The names of the roles (properties) mentioned in this class definition (EAT-MOUSE)
- The individual names that are instances of this class (TOM)
- The OWL definition (for debugging purposes)
- A description of the class declaration in a formalized "natural language"

We learn that a cat has to be in the relationship (role) EAT-MOUSE with at least one individual. After clicking on the corresponding hyperlink, the description EAT-MOUSE is displayed in Figure 4.

Roles may be part of a role hierarchy. For instance, for this example we discover that EAT-MOUSE is defined as a child of role EAT-ANIMAL and a parent of role EAT-SMALL-MOUSE.

Most readers will agree that this kind of information is better readable and helpful in understanding ontologies than just reading the OWL specification. The final version of this web-based tool will offer more support for the exploration and analysis of (unknown) OWL ontologies with the help of the OWL reasoning agent Racer. It is planned to provide a more ad-

Figure 4: Description of property EAT-MOUSE.

vanced query support and better cross-referencing. The tool is implemented as a web server and can be used with any web browser. Currently the tool is designed as a reactive agent for understanding OWL ontologies. However, we also envision a more proactive version of this tool that would automatically notify users or agents if interesting information about ontologies becomes available. In the following section we describe a general interface supporting the proactive behavior of such a type of agents.

5 Accessing Retrieval Inference Services

The main examples for the Semantic Web use information retrieval applications involving one or more agents. In a full-fledged information retrieval scenario, an agent might consult a document management system provided by an agent host environment. The agent can ask for documents that match a certain query in a similar way as discussed above. This scenario can also be realized with Racer if documents are annotated with meta data formalized with RDF [12]. Information about documents can be represented using A-boxes. RDF annotations for documents are read by Racer and corresponding assertions are added to an A-box. Data types and values play an important role for describing documents (e.g., year, ISBN number etc.). Agents can retrieve documents by posing retrieval queries to A-boxes w.r.t. to specific T-boxes in the way exemplified above.

5.1 Publish/Subscribe Interface

If we consider an instance retrieval query Q w.r.t. an A-box A, then it is clear that the solution set for Q could be extended if more information is added to A over time (whoever is responsible for that, another agent or the agent host environment). It would be a waste of resources to frequently poll the host environment with the same query (and repeated migration operations). Therefore, Racer supports the registration of queries at some server w.r.t. to some A-box (Publish/Subscribe Interface). With the registration, the agent specifies an IP address and a port number. The corresponding Racer Server passes a message to the agent if the solution set of a previously registered instance retrieval query is extended. The message specifies the new individuals found to be instances of the query concept Q. We call the registration of a query, a subscription to a channel on which Racer informs applications about new query results. For details see the Racer manual [11].

Rather than considering a single query in isolation, a prac-

tical system should be able to consider query sets (as database systems do in many applications). With the publish/subscribe interface, multiple queries can be optimized by Racer. Instance retrieval queries can be answered in a faster way if the set of candidates can be reduced. In a similar way as for databases, the idea is to exploit results computed for previous instance retrieval queries by considering query subsumption (which is decidable in the case of the query language that Racer supports). However, this requires computing index structures for the T-box (the process is known as T-box classification) and, therefore, query subsumption is enabled on demand only. On the one hand, there are some applications, in which A-boxes are generated on the fly with few queries referring to a single A-box. On the other hand, there are applications which pose many queries to more or less "static" T-boxes and A-boxes (which are maybe part of the agent host environment). The Racer Server supports both application scenarios. As a design decision, Racer computes answers for queries with as few resources as possible. Nevertheless, a Racer Server can be instructed to compute index structures in advance if appropriate to support multiple queries.

5.2 Additional Features of the Racer System

Optimizations: Various optimization techniques for ontology-based query answering with respect to T-boxes, A-boxes, and concrete values have been developed, implemented, and investigated with the Racer System. One of the design goals of Racer is to automatically select state of the art optimization techniques that are applicable to the current input.

Persistency: In a similar way as in database systems, for query answering w.r.t. T-boxes and A-boxes complex data structures are computed and used internally by Racer. Internal structures of T-boxes and A-boxes being processed for query answering can be saved to disk for quick access and later reuse if the Racer Server is restarted.

Multi-User Support, Thread Safeness, Locking, Load Balancing: In a distributed systems context, there can be multiple agents connecting to a server at the same time. If they refer to the same A-boxes and T-boxes, requests must be synchronized. Thus similar problems as with databases such as thread safeness, locking, and load balancing have to be dealt with. For instance, if multiple Racer Servers are started, queries can be automatically directed to "free" Racer Servers. These problems are tackled by the Racer Proxy, which is supplied as part of the Racer System distribution.

6 Conclusion

This paper briefly described the OWL reasoning agent Racer and its services and demonstrated that Racer can cooperate with an ontology exploration and analysis tool. Description logic systems freely available for research purposes now provide for industry-oriented software integration and begin to ensure stable access in multi-user environments as can be expected in the context of the semantic web with its XML-based representation languages (RDF, OWL).

Acknowledgements

We gratefully acknowledge the work of Ying Lu, who is developing the ontology exploration and analysis tool.

References

[1] O. Lassila and R.R. Swick, "Resource description framework (RDF) model and syntax specification. recommendation, W3C, february 1999. http://www.w3.org/tr/1999/rec-rdf-syntax-19990222", 1999.

[2] D. Brickley and R.V. Guha, "RDF vocabulary description language 1.0: RDF Schema, http://www.w3.org/tr/2002/wd-rdf-schema-20020430/", 2002.

[3] F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein, "OWL web ontology language reference", 2003.

[4] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, *The Description Logic Handbook*, Cambridge University Press, 2003.

[5] F. Baader, I. Horrocks, and U. Sattler, "Description logics as ontology languages for the semantic web", in *Festschrift in honor of Jörg Siekmann*, D. Hutter and W. Stephan, Eds. 2003, LNAI. Springer-Verlag.

[6] V. Haarslev and R. Möller, "Racer system description", in *International Joint Conference on Automated Reasoning, IJCAR'2001, June 18-23, 2001, Siena, Italy.*, 2001.

[7] S. Bechhofer, I. Horrocks, and C. Goble, "OilEd: a reason-able ontology editor for the semantic web", in *Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence, September 19-21, Vienna*. LNAI Vol. 2174, 2001, Springer-Verlag.

[8] S. Bechhofer, R. Möller, and P. Crowther, "The DIG description interface", in *Proc. International Workshop on Description Logics – DL'03*, 2003.

[9] R. Möller, R. Cornet, and V. Haarslev, "Graphical interfaces for Racer: querying DAML+OIL and RDF documents", in *Proc. International Workshop on Description Logics – DL'03*, 2003.

[10] V. Haarslev and R. Möller, "Optimization stategies for instance retrieval", in *Proc. International Workshop on Description Logics – DL'02*, 2002.

[11] V. Haarslev and R. Möller, "The Racer user's guide and reference manual", 2003.

[12] Adobe Systems Inc., "Embedding XMP metadata in application files", 2002.

Object Database on Top of the Semantic Web

Jakub Güttner
Graduate Student, Brno Univ. of Technology, Faculty of Information Technology, Czech Republic
guttner@fit.vutbr.cz

Abstract

This article compares the structure of Semantic Web RDF (Resource Description Framework) to a worldwide distributed database. Reasons are given for treating such database as an object-oriented one. The article lists the similarities and differences between RDF and object databases, gives reasons for extracting object data from RDF and shows the structures that can be discovered in RDF graphs, discusses and unifies different approaches for querying and navigating object-oriented and RDF graphs and builds a simple formal model of an object-oriented DB on top of model-theoretic RDF semantics.

1. Introduction

Although the World Wide Web is practical and readable for humans, computers cannot process its semantics. For them, it is hard to tell that a document is a CV, which of the numbers contained in it is the date of birth and which link leads to the company where the person works.

The Semantic Web tries to address this problem and store information in a more organized way. This article shows that such effort is basically building a worldwide database, compares such database to object-oriented databases (OODBs), shows how to extract object data from it and gives an example of building the semantics for a simple OODB on top of the Semantic Web Resource Description Framework.

The following section, *Semantic Web and Objects*, shows the similarities between these two concepts, explains how an object database is suitable for storing the corresponding information and why it's useful to extract objects from the Semantic Web.

Third section, *Graph-based access*, shows how the Semantic Web and object structures differ in the way of obtaining information from graphs, and suggests how to reconcile the two approaches.

Fourth section, *An Object Database Over the Semantic Web*, gives a summary of a simple model-theoretical object database definition over RDF-based Semantic Web data.

2. Semantic Web and objects

2.1 The vision of the Semantic Web

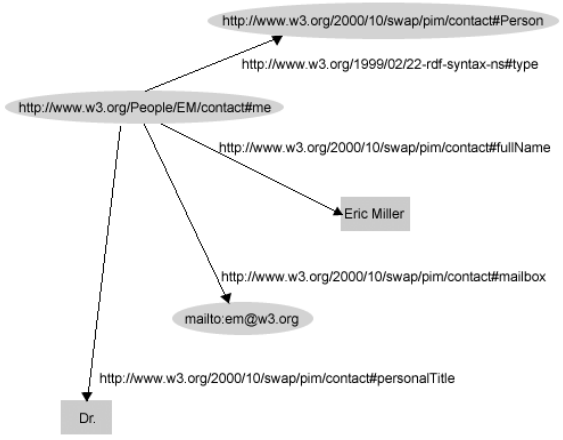


Figure 1. A RDF graph about Eric Miller [2]

The Semantic Web is a W3C activity that aims at extending the current World Wide Web with information that has well-defined meaning both to humans and computers. This would create an environment where "software agents roaming from page to page can readily carry out sophisticated tasks for users" [1]. The way to reach this goal lies in inference rules for the agents and presenting information in structured collections in contrast to today's WWW pages that only define formatting of text, not its meaning.

At the core of the Semantic Web lies the RDF — Resource Description Framework [2] that stores information in graphs where each edge represents a binary predicate (figure 1). For uniqueness across the whole Web, both edges and nodes are labeled with URI references (urirefs). RDF and RDFS (RDF Schema) also provide other features like a collection vocabulary, typing and subclassing, anonymous nodes and elementary datatypes. Semantics of new urirefs can be formally

specified using existing sets of references or model theory and RDF closures.

Examples of existing RDF vocabularies are Dublin Core Metadata for interoperable online metadata standards, RSS (RDF Site Summary) for site syndication, or OWL — Ontology Web Language.

2.2 Semantic Web as a database

"The Resource Description Framework (RDF) is a framework for representing information in the Web." [2]

RDF and the Semantic Web define how to store information. Although the Semantic Web is mostly mentioned from the viewpoint of knowledge management and artificial intelligence, managing a body of information is a typical database problem. From this perspective, the Semantic Web can and should be viewed as a worldwide database. Of course it is widely distributed and not centrally managed, often incomplete or inconsistent and very loose in its format — but it still is a database, albeit not a relational one. There are, however, other types of databases as well, ones whose structure is very close to the graph nature of the Web.

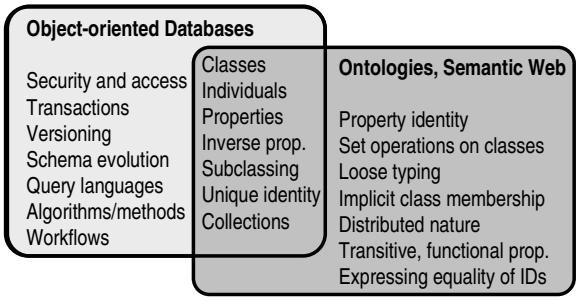


Figure 2. Comparing OODBs and the Semantic Web

An object-oriented database and the Semantic Web share several important concepts. Adding agents to the Semantic Web is then similar to adding deductive principles to an OODB. Some of the main similarities in structure and concepts are:

Unique identifiers — A strong requirement for every OODB are a unique identifiers (OID). In today's setting that means unique across the whole Internet, which is exactly what URI references do. Using RDF urirefs for OIDs would establish an understanding of the Semantic Web as a worldwide distributed database.

Graph-theoretic foundations — RDF uses model theory to interpret graphs. OODBs still do not have a common theoretical foundation [3], [4], but many models use graph theory, combining it with set theory [5], [6] or extending into category theory [7], [4], [8]. Graph theory in OODBs is suitable for modeling relationships between objects or inheritance hierarchies. Using a shared foundation for both RDF and object databases would connect these areas very closely.

Description logics — The RDF is built on existentially quantified first order logic. In deductive object-oriented databases, description logics (F-logic, Transaction logic, HiLog [9]) are used for describing a database schema. This is yet another formalism that connects the two together.

Other similarities between the Semantic Web and OODBs are suggested in figure 2 together with examples of specific areas where these two could enrich each other in the future.

2.3 Extracting objects from the Semantic Web

Why extract objects from the Semantic Web?

Application of algorithms — Many programming languages are object-oriented. It would be convenient to present them with Semantic Web data in the form of objects. Moreover, strongly typed objects present correct data to algorithms without the need for explicit checking of their structure.

Integrating database concepts — Semantic Web data processed as an OODB could use some of the well-developed database concepts like indexing, access control or query languages (see figure 2).

Efficient storage — Today the Semantic Web still works in small and manageable scale and there are no serious performance problems. In the future, the performance of physical storage of RDF data may become increasingly important. Looking at RDF data through the object database paradigm allows the use of common object storage techniques.

Below we list the basic features of an OODB according to the ODMG standard [10] and G2 Concept Definition Language [3]. In the final section of this article, we build a formal OODB model on top of RDF semantics that captures all of these areas. Here each of them is explained and its RDF counterpart is mentioned.

Objects — Everything in an OODB is stored in objects. An object has a unique OID (URI in RDF) and it contains either a tuple or a collection of attributes or references. All RDF nodes with a uriref can be considered objects.

Datatypes — Most OODB models have a set of elementary types that are used to construct composite types. These types cannot be further decomposed and their semantics is fixed. This is equivalent to RDF T-interpretations; one of the most commonly used sets of datatypes is XML Schema.

Attributes and relationships — A collection or tuple can either contain or reference a value. This is important for modeling, physical data storage and update semantics. When the value of an attribute is a uriref, it is

referenced, and when the value is either a datatype or a blank node (a tuple or a collection without a uriref) with one reference to it, its value is embedded within the parent object.

Tuples — A tuple has attributes labeled by property urirefs. Every blank RDF node that has at least one non-collection attribute can be considered a tuple. Adding a uriref to the node makes it a tuple object.

Collections — RDF has collection vocabulary without any formal semantic restrictions. All its objects need to have the same type. Any blank RDF node with collection attributes (eg. rdf:_1) is a collection and adding a uriref to the node makes it a full object. RDF can have objects that act as both tuples and collections, similarly to G2 CDL [3], and the model takes it into account.

Types — In most OODBs, an object has exactly one type that specifies its internal structure. However, a RDF node can have no types or multiple types. Having no type is equivalent to having the most general type ("Any") and having multiple types is similar to the concept of multiple inheritance and object roles. Nodes that conform to the structure prescribed by all of their types (through domain and range properties of object attributes) are labeled as "strongly typed" in the model.

Inheritance — In both RDF and OODBs, inheritance is a fundamental tool for constructing new concepts. The RDF definition of inheritance as a subset relationship on elements of the domain of discourse is suitable for its database counterpart, because it allows multiple inheritance and preserves the notion of "strongly typed" objects.

With these guidelines an object-oriented structure can be extracted from any RDF graph. Depending on the graph, the form of the result can either be quite loose or strongly typed.

The following section discusses some difficulties in navigating and querying RDF graphs as OODBs, while the final section gives an example of a formal OODB model built on top of RDF semantics.

3. Graph-based access

3.1 Navigating the Semantic Web graph

For the purposes of reasoning, a RDF graph is often perceived as a set of facts (triples — binary predicates). When retrieving information for deductions using first order logic, the most common query is finding a set of triples with a constant predicate — in Prolog syntax, this would be something like parent(X,oid1) or supervises(X,Y). From the viewpoint of retrieving data, the access point into a RDF graph is *the label of an edge*, therefore the graph does not need to be connected and it is not important whether all its nodes are reachable. This approach implies that from the viewpoint of physical data organization, a knowledge base is physically grouped by predicates.

While this is common from the deductive point of view, it presents an obstacle for viewing the RDF graph as an object database.

3.2 Navigating an object database

The navigation in an object database is different. In an OODB, data are physically grouped by objects. An object is accessed as a whole; it is unusual to retrieve all occurrences of a given attribute. In Prolog syntax, accessing the whole object can be expressed as X(oid1,Y). The main access point into the database is *the label of a node*, typically an extent — an object that stores a collection of instances of a given type.

The practical result is that an object database is navigated by traversing edges of its graph. In contrast to the Semantic Web approach, the direction of an edge is important and the whole graph must be reachable from the access points.

3.3 A common access model

To access the RDF graph as an object database, the whole graph needs to be reachable from certain access points. Figure 3 shows part of a RDF graph ("Peter likes cabbage"). To make such graph fully reachable, three changes (shown in gray) to the original structure need to be made:

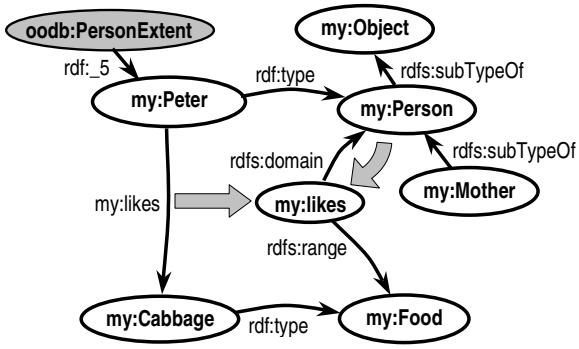


Figure 3. Reachability in part of a RDF graph

Adding extents of types. In a typical OODB, every object needs to be part of at least one extent. All extents are collections directly reachable from the system catalog (a single access point into the database), which in effect makes all the objects in the database accessible. In RDF setting, extents could actually be identified with types (oodb:PersonExtent and my:Person).

Reversing the direction of rdfs:domain. When finding out information about a certain type, it is useful to ask a question like X(my:Person,Y) to find out all the attributes of my:Person and their types. For this reason, it would be useful to define a predicate that has the same meaning as rdfs:domain, but its direction is reversed (see the curved gray arrow in figure 3).

Connecting edges to corresponding nodes. In RDF, it is natural that an object can act both as a predicate and as subject/object. To find out information about an attribute in an object database, one can usually examine the type of this object. However, the RDF model does allow extra attributes, therefore it should be possible to find the specifics of an attribute elsewhere. In figure 3, straight gray arrow indicates a new connection that needs to be made.

Traversing from access points together with removing property-driven random access into the graph certainly limit the freedom of finding data in the RDF graph, but a big advantage is that the data can be physically organized by objects and the usual OODB optimizations can be applied to things like attribute storage or type information.

4. An object database over RDF

The RDF standard is open to further extensions. New sets of URI references (vocabularies) can get formal meaning either by making statements about them using already defined predicates (like rdf:Bag or rdfs:subPropertyOf) or by redefining their model-theoretic interpretation.

This section presents direct model-theoretic semantics of a basic vocabulary that supports the elementary notions of an object-oriented database (as defined in sources like [10], [11], [5], [3]). A *"SODA"* prefix (Semantic Object-oriented DAtabase) is used for this vocabulary. With these extensions, object structures can be machine-entailed from arbitrary RDF graphs. This model can also be used for later implementation of OODB on top of RDF with automatic consistency checking using only the means supplied by RDF and for populating the database with RDF data.

4.1 Basic RDF semantics

Description of RDF semantics [2] states that for a set of URI references (a vocabulary) V that includes the RDF and RDFS vocabularies, and for a datatype theory T, a *T-interpretation* of V is a tuple I = < IR, IP, IEXT, IS, IL, LV > (which stands for Resources, Properties, Extensions, Semantic map, Literal map and Literal Values).

IR, *universe*, is a set of semantic images of urirefs from V

IP is subset of IR. It contains the images of the properties from V — urirefs that label edges of RDF graphs

IEXT: IP → IR×IR defines property extensions. It gives all object-subject pairs that make a property from V true.

IS: V → IR assigns semantic images in the universe to urirefs.

IL is a mapping from the set of typed literals to the set **LV**, a superset of all untyped literals and a subset of IR.

The RDF vocabulary interpretation also defines class extensions – for every class c, ICEXT(c) = {x ∈ IR | <x,c> ∈ IEXT (IS(rdf:type))}. The rdf: prefix is connected to http://www.w3.org/1999/02/22-rdf-syntax-ns# and rdfs: then stands for http://www.w3.org/-2000/01/rdf-schema#. Apart from the RDF and RDFS requirements, there are several more things that need to be true for every T-interpretation.

4.2 Obtaining the OODB graph

RDF is monotonic, which means that adding edges to a RDF graph cannot change the validity of previous entailments — anyone can publish RDF statements without harming other people's entailments. However, in some cases it is useful to "close" the RDF graph to additions. Otherwise, one thing that would not be possible is deciding whether an instance of an object structurally conforms to its type since the definition of a type cannot be guaranteed to be complete.

4.3 Part of model-theoretic OODB semantics

There is not enough space here to present the full definition of the model including a comprehensive overview of RDF semantics — only several sample definitions and features of the model are given. The full model is given in [12].

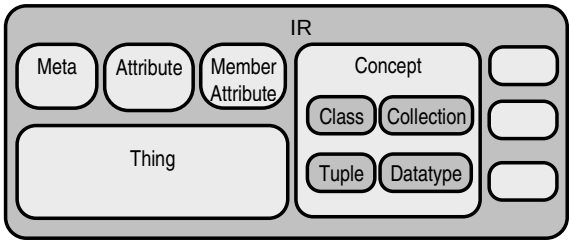


Figure 4. Division of the universe

SODA vocabulary: {soda:Concept, soda:Class, soda:Tuple, soda:Meta, soda:Thing, soda:Collection, soda:Attribute, soda:MemberAttribute} ∈ IC; {soda:collectionOf, soda:type, soda:subTypeOf, soda:member, soda:_1, soda:_2,...} ∈ IP. Most of the SODA vocabulary is subclassed from RDF with domain and range restrictions.

A **SODA-interpretation** is a T-interpretation that makes further conditions (and T-interpreted triples) true.

The universe is divided into disjoint sets like in strongly typed programming languages — see figure 4.

Blank nodes cannot represent class instances since they do not have a universal OID (uriref).

$$\text{\_xxx is blank node} \wedge \langle IS{+}A(\text{\_xxx}),y\rangle \in IEXT(IS(soda{:}type)) \Rightarrow y \in ICEXT(IS(soda{:}Tuple)) \cup ICEXT(IS(soda{:}Collection))$$

Strong typing: A tuple must have all the attributes that its type prescribes with an exception of undefined reference.

$$x \in I2EXT((IS(soda{:}Class)) \cup I2EXT(IS(soda{:}Tuple)) \wedge a \in ICEXT(IS(soda{:}Attribute)) \wedge \langle x,c\rangle \in IEXT(IS(soda{:}type)) \wedge \langle a,c\rangle \in IEXT(IS(rdfs{:}domain)) \wedge c \in ICEXT(IS(soda{:}Collection)) \cup ICEXT(IS(soda{:}Tuple)) \cup ICEXT(IS(soda{:}Datatype)) \Rightarrow \exists\, y{:}\ \langle x,y\rangle \in IEXT(a)$$

Data recursion: Collections, tuples and literals need to avoid being embedded in each other. Since only one attribute in the graph refers to them, there needs to be a path from some class concept to these concepts (fig. 5).

$$x_0 \in I2EXT(IS(soda{:}Collection)) \cup I2EXT(IS(soda{:}Tuple)) \cup I2EXT(IS(soda{:}Datatype)) \Rightarrow \exists\, n{\geq}1,\ x_1 \ldots x_n,\ a_1 \ldots a_n{:}\ \forall i \in \{1 \ldots n\}{:}\ \langle x_i,x_{i-1}\rangle \in IEXT(a_i) \wedge a_i \in ICEXT(IS(soda{:}Attribute)) \cup ICEXT(IS(soda{:}MemberAttribute)) \wedge x_n \in I2EXT(IS(soda{:}Class))$$

The resulting vocabulary contains the most important concepts from the area of OODB as specified by ODMG, O2 or Matisse projects (see sources). Moreover, the model has very close correspondence to the CDL (Concept Definition Language) of the G2 database [3] and semantic nuances of the vocabulary definition were formed according to this system.

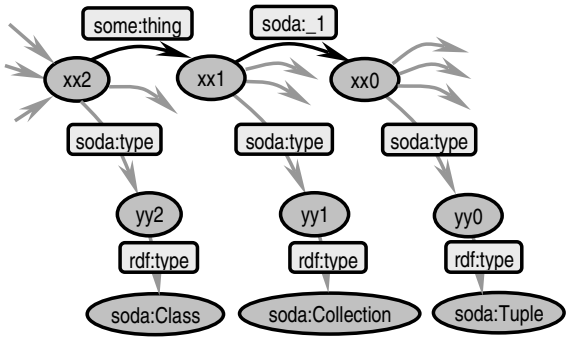


Figure 5. Part of a SODA RDF-based database

5. Conclusion

The Semantic Web with its graph structure and unique identifiers is very similar to an object database in both structure and philosophy. It is useful to extract object data from RDF graphs for algorithm application, using database-oriented data handling and more efficient storage.

The article showed the equivalents of fundamental OODB features in RDF graphs. Different ways of navigating the data graphs were presented and some guidelines for reconciling the were given. The article also presented part of a formal OODB model built on top of the RDF semantics that takes the RDF specifics into account.

Future work can further refine the model and bring together different aspects of the Semantic Web and OODBs that are currently used only in one of the areas (see figure 2).

This work was supported by the CEZ:J22/98: 262200012 project "Research in Information and Control Systems" and the grants GAČR 102/01/1485 "Environment for Development, Modelling, and Application of Heterogeneous Systems" and FRVŠ FR828/03/G1 "Dynamic Object Model in Interpreted Systems".

6. References

[1] T. Berners-Lee, The Semantic Web. In: *Scientific American*, USA, May 2001.

[2] P. Hayes, *RDF Semantics*. W3C Working Draft January 2003, http://www.w3.org/TR/2003/WD-rdf-mt-20030123/ .

[3] T. Hruška, M. Máčel, Object-Oriented Database System G2, In: *Proceedings of the Fourth Joint Conference on Knowledge-Based Software Engineering*, IOS Press Brno, Ohmsha Publishing, Czech Republic, 2000.

[4] C. Tuijn, *Data Modeling from a Categorical Perspective*. PhD thesis, Antwerpen University, Netherlands, 1994.

[5] C. Lécluse, P. Richard, F. Vélez, O2, an Object-Oriented Data Model. In: B*uilding an Object–Oriented Database System: The Story of O2*, Morgan Kaufmann Publishers, USA, 1992.

[6] J. Güttner, Object Attributes as Functions. In: *Proceedings od 6th International Conference ISIM 03 — Information Systems Implementation and Modeling*, MARQ Ostrava, Czech Republic, 2002.

[7] K.-D. Schewe, *Fundamentals of Object Oriented Database Modelling*. Clausthal Technical Univ. Germany, 1995.

[8] P. Kolenčík, Conflicts in Classes Defined by Multiple Inheritance. In *Information System Modeling Conference MOSIS'98*, Ostrava, Czech Republic, 1998.

[9] M. Kifer, Deductive and Object Languages: Quest for Integration. In: *DOOD '95*, Springer, Singapore, 1995.

[10] M. Atkinson et al, The Object-Oriented Database Systems Manifesto. In: *Deductive and Object-oriented Databases*, Elsevier Science Publishers, USA, 1990.

[11] R. D. D. Cattel, D. K. Berry, eds., *The Object Data Standard: ODMG 3.0*. Morgan Kaufmann Publishers, San Francisco, USA, 2000.

[12] J. Güttner, *An object-oriented database model on top of RDF*, www.fit.vutbr.cz/~guttner/ThesesModel.doc, 2003.

Structural Caching XML data for Wireless Accesses

Shiu Hin Wang
Department of Computing
The Hong Kong Polytechnic University
Hong Kong
852-97235397
hwshiu@mail.hongkong.com

Vincent Ng
Department of Computing
The Hong Kong Polytechnic University
Hong Kong
852-27667242
cstyng@comp.polyu.edu.hk

Abstract

Recent web cache replacement policies incorporate information such as document size, frequency, and age in the decision process. In this paper, we propose a new caching algorithm, StructCache, for wireless accesses of XML data. The algorithm is an enhancement of the Greedy-Dual-Size (GDS) policy and the Greedy-Dual-Frequency-Size (GDFS) policy. It would consider document sizes, access frequency and exploits the aging mechanism to deal with cache pollution. In addition, the structural information of XML is utilized to achieve better hit ratios. Experimental results show that the StructCache algorithm outperforms GDS and GDFS algorithms for queries which are sub-tree(s) in XML documents of precedent queries and queries of the same axis and node tests in XML documents with precedent queries.

1. Introduction

By virtue of the increasing processing power of embedded computers, wireless computing and Mobile Commerce (mCommerce) is the wave of the future [7]. Caching and prefetching of XML data in heterogeneous networks, especially for the mobile environment, reduce traffics and improve the performance of dissemination of XML data, which in turns improve the usability of the Internet as a large and distributed information system.

Very often, user access patterns are helpful for the customization for specific type of users. The relative importance of long-term popularity and short-term temporal correlation of references for web cache replacement policies has not studied thoroughly. This is partially due to the lack of accurate characterization of temporal locality that enables the identification of the relative strengths of these two sources of temporal locality in a reference stream [15].

Moreover, better cache policies are equivalent to several-fold increase in cache size. Efficient cache and prefetching algorithms reduce the needs of cache sizes to match the growth rate of web content. The gains from efficient cache and prefetching algorithms are compounded through a hierarchy of caches [15].

Existing web caching algorithms capture the characteristics and differences of paging in file systems. However, they do not consider the nature and properties of the objects themselves. In this paper, we try to propose a caching technique to improve the performance of query responses. It is our aim to improve the performance of XML queries against large XML files, which in turn may improve the usability of wireless applications.

In this paper, our main focus is the benefits brought from our proposed replacement algorithm to cache XML documents and the comparison of performance with other algorithms. The paper is composed of six sections. In section 2, we first review background study and previous related work. Section 3 consists of the structure and details our proposed XMLCache framework. Section 4 gives the details of our caching algorithm, StructCache, for caching of XML objects. The procedures and results of the experiment are presented in section 5. Section 6 summarizes our work and section 7 contains the references.

2. Background and Related Work

Existing web caching algorithms mainly consider individual documents as the individual objects to be cached. The larger the document, the greater the overhead when cache misses. This may pose a problem especially under a bandwidth and memory constrained environments such as wireless environments.

Traditional object caching algorithms that have not considered the syntactic and semantics characteristics of XML documents may not handle the HTML, XML contents in an efficient manner. Our suggested caching algorithms tries to exploit the syntactic structure of XML documents and the XML based quires to improve the caching performance in a high latency and low bandwidth environments.

We develop a caching framework that is used for caching of both XML and non-XML documents. The caching technique is done on client-side, which is supposed to be embedded in wireless computing devices with limited bandwidth communication connections. We will study the performance of our proposed caching algorithm that tries to exploit the schemas of XML and XML queries. It is also

expected that the algorithm will be more effective, especially in situations in which the object size to cache size ratio is high, high network latency and low transmission environment.

2.1. Cost Metrics

Cost metrics [1,5,6] are used as objective measurements of the effectiveness of the caching algorithms. Three cost metrics mostly employed are:

1) Bit Model: The cost of a cache miss equals to the size of the missing item. This measure provides an objective measure for the effectiveness of our proposed algorithms.

2) Cost Model: The cost of a cache miss is unity. This is used for evaluating the use of the heuristics of XML queries in improving the effectiveness of caching.

3) Time Model: The cost of a miss equals to the average time to load such an individual item. Here, it means the time to retrieve the whole object (time of retrieval due to page fault) or the user perceived response time (network Delay). For some wireless application, a user who has part of the results and progressively getting the remainings may be more crucial than obtaining the complete results at a minimum time even though the total time of retrieval is longer.

2.2. Related Work

Caching algorithms targeting for web have become more prevalent. GreedyDual [15] web caching algorithm is one of the typical examples. It is a generalization of GreedyDual-Size algorithm [5] and a development of the family of algorithms derived such as GreedyDual Frequency-Size. Trace driven simulation illustrates that it has superior performance when compared to other web cache replacement policies proposed in the literature[15].

There are many factors affecting the performance of a given cache replacement policy. GreedyDual caching algorithm exploits the size, miss penalty, temporary locality and long-term access frequency and captures both popularity and temporal correlation:

1) Size – Web objects are of various size and caching smaller objects usually results in higher hit ratios, especially given the preference for small objects [16].

2) Miss Penalty – The miss penalty varies significantly. Assigning higher preference to objects with a high retrieval latency can achieve high latency saving [14]

3) Temporary locality – Web access patterns exhibit the temporal locality [2]. Similar to the LRU replacement policies, GreedyDual also assign higher preference to recently accessed objects.

4) Long-term access frequency – The bursty behavior of the popularity of web objects were found over short times scales while it is more smooth over long times scales [3].

Apart from the researches in replacement algorithms targeting for Web, there are hierarchical and cooperative caching architectures such as the Harvest project [8], access driven cache [9], adaptive Web caching [10]. For distributed caching architecture, there are examples such as summary cache [11] and Internet cache protocol (ICP) [13]. With hierarchical caching, caches are placed at multiple levels of the network. A hierarchical architecture is more bandwidth efficient, particularly when some cooperating cache servers do not have high-speed connectivity. With distributed caching, most of the traffic flows through low network levels, which allows better load sharing and are more fault tolerable. However, a large-scale deployment of distributed caching may encounter the problems of high connection times, higher bandwidth usage and administrative issues [12].

Because of the relatively small bandwidth of mobile environment when compared with that of connected networks, the gains arising from efficient caching and prefetching in mobile environments are even apparent. However, conventional replacement algorithms still have room for improvements under the Internet and mobile environment when data objects are having syntactic and semantic implications.

3. The XMLCache Framewok

Our proposed framework utilizes the conventional caching algorithms for non-XML documents whilst handles XML data objects differently. The determination of which caching strategy is employed depends on the type of documents requests from clients. A different caching algorithm (StrcutCache) will exploit the coherency of similar requests made by clients with the idea of the structure of XML documents will have impacts on successive retrieval.

The framework shown in Figure 1 acts as a proxy between the mobile clients and remote data source. It mainly divides retrieval objects into XML and non-XML objects. For XML data queries, it is handled by our proposed caching algorithm called StructCache in the caching engine. For non-XML data queries, it is handled by existing web caching algorithms. Depending on whether the XML documents fetched from remote servers have the corresponding DTD (Document Type Definition) or not, the DTD may need to be extracted by the DTD Extractor.

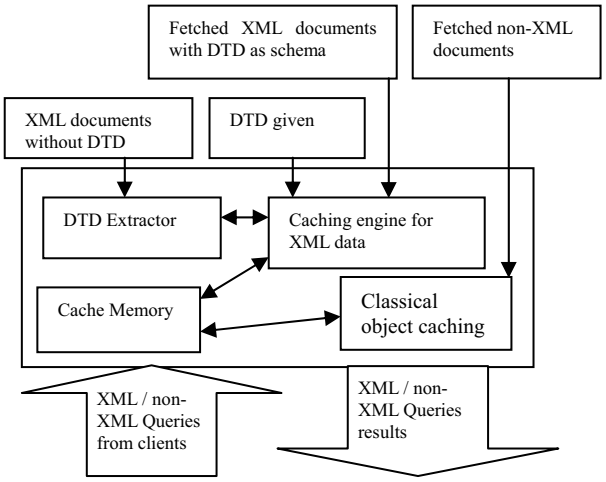


Figure 1. Internal architecture of StructCache framework

Every XML query sent from clients is evaluated, which triggers the retrieval of XML documents when the required fragments do not exist in the cache memory. The client returns the query result directly when the required objects are located in cache. Each data object of XML documents in cache memory is associated with a timestamp and the corresponding DTD. DTDs of the retrieved XML documents do not exist will trigger the generation of initial multiplication factors, which will affect the score updating process in later phase. Depending on the frequency of accesses and sizes of objects, a score for each of the data node of a given DTD will be updated in runtime. More details will be covered in Section 4.3.

In our framework, there are a number of assumptions:

a) Client computation power, power consumption and the complexity of the algorithm are not the limiting factors.

b) Since the cache is located on the client wireless devices (e.g. PDA), the size of the cache as well as the bandwidth of the wireless networks are the limiting factors.

c) Because of the high latency, and relatively low transmission rate of the communication channels, the overhead of retrieving objects from heterogeneous network is higher than that from the cache. We assume that the throughput and the latency are averaged constants throughout our model. We choose a typical transmission rate reported by mobile network services providers rather than that derived theoretically from a selected modulation methods, and carriers.

d) We assume caching can be performed either in the mobile clients or proxies. Details of switching of underlying cellular networks are intentionally abstracted to generalize our proposed algorithm for different kinds of carriers or application-specific uses.

3.1. Queries from clients

In general, requests from mobile computing devices expressed in URL-like query can be decomposed into two parts. The first part is the URL that locates the resource to be retrieved whilst the second part is an XPath expression. Figure 2 illustrates an example of a client request. In this example, the first part is the 'http://www.polyu.edu.hk?/article/author/name' and the second part is '[firstname='Joe']'.

The syntax of the client request:
[URL]?[XQL]
Key:
[URL] : Universal Resource Locator that locates the resources(XML / non-XML files)
[XQL] : XQL of the target XML document
Example:
http://www.polyu.edu.hk?/article/author/name[firstname='Joe']

Figure 2. An example request from client application

Our cache engine acts as a proxy between the client application and remote servers. Items fetched on behalf of client applications include XML and non-XML files. The fetched XML documents can also be classified into two types. The first type is those XML documents have the corresponding DTD given in advance` whilst the second type has no knowledge about the document's DTD. In the first case, our replacement algorithm can be directly applied. For the second case, since the corresponding DTD of a XML document not known in advance, a DTD extractor is used for the estimation of the structure of the DTD.

The XMLCache framework has 4 major modules:

a) DTD Extractor – This module is responsible for the extraction of DTD from a given XML document no DTD given in advance. It is employed when clients requests query XML documents that have no predefined DTD.

b) The StructCache Engine – Our proposed engine of caching XML data under mobile.

c) Cache Memory – The module has two parts. The first part is a set of mappings that maps the document identifier (URI) to a given timestamp (Document's Last Modification Time). The second part is the repository where individual cached objects are placed. For XML documents, it should be the fragments of XML documents whilst it will be complete binary image for graphics files.

d) Classical Cache Engine – This component encompasses conventional online replacement algorithms such as GDFS, which treats the whole file as an individual object to be cached. This is mainly used for caching of non-XML documents.

4. StructCache

StructCache is an algorithm used in the caching engine for XML data with DTD. The algorithm can be divided into two phases. The first phase is to determine the weighting factors, called multiplication factors for a group of XML documents with the same schemas which are used in runtime phase. During operations, the score of each node, which is depended on the multiplication factors calculated in the first phase, access frequencies, size of nodes as well as the XPath of the XML queries derived from subsequent client requests.

Comparing with other classical object caching algorithms, StructCache adopts an adaptive way rather than solely relying on frequency, size of objects in cache. It is adaptive in the sense that the initial multiplication factors depend on the structure of the associated DTD, and the object replacements process is affected by the multiplication factors, XPath, the heuristics of access of nodes or nodes sets as well as the size of a given node or nodes sets relative to the whole XML document. Although we do not quantify the relationship between the factors considered and their effects in this paper, it is observed that the factors concerned are correlated with long-term access patterns. It is because, very often, the design of schema mostly reflects the relative importance of fragments in XML documents and the temporal and spatial locality of accesses.

Normally, systems can either fetch a block in response to a cache miss (on-demand fetch), or it can fetch a block before it is referenced in anticipation of a miss (prefetch)[16]. Fine-grained objects will save more redundant space but sacrifice the computation power. Our goal is to find an online policy for on-demand fetching and caching of XML documents without knowing the sequences of references in advance. Instead of caching the whole XML document, the memory objects to be cached are fragments of XML documents derived from the results of XML queries in the request stream, which may be of various sizes. For a limited amount of cache in wireless computing devices, we try to exploit the heuristic as well as the semi-structured characteristics of XML documents. Our problem in fact is a general caching problem [12], when the pages have varying sizes and costs. This problem arises, among other places, in cache design for networked file systems or the world-wide web [12]. In web caching, popular online caching algorithms such as Least Recently Used (LRU) has heuristics justification that real-life sequences often exhibit the property that "the past predicts the future". They normally treats the whole file as an individual object [1]; by contrast, we try to apply a more fine-grained approach such that individual objects to be cached are XML fragments from the query results, with a view to minimize page faults, especially for wireless communication channels and documents with large size. The heuristics of our approach are based on the XML data queries generated by the client application(s) or requested from users.

StructCache is our caching algorithm dedicatedly for XML data with DTD. It is to work within the caching engine for XML data in our framework. The algorithm has two phases. The first phase is the determination of factors and initialization of variables from a newly fetched DTD. It is triggered by the cache miss and fetching of XML documents with undetermined DTD from remote server(s). The second phase is the runtime phase that fetches objects, invalidates cache, performs updating of variables and triggering the initialization phase for any fetched and undetermined DTD.

```
<!ELEMENT article(category, title, publisher, author*)>
<!ENTITY % Address "(#PCDATA)">
<!ELEMENT title(#PCDATA)>
<!ELEMENT title(#PCDATA)>
<!ELEMENT publisher(publishername,address)>
<!ELEMENT publishername(#PCDATA)>
<!ELEMENT address(#PCDATA)>
<!ELEMENT author(name,age,address?)>
<!ELEMENT name(firstname?,lastname?)>
<!ELEMENT age(#PCDATA)>
<!ELEMENT firstname(#PCDATA)>
<!ELEMENT lastname(#PCDATA)>
```

Figure 3. An example DTD

4.1. Initialization Phase

In this phase, multiplication factors are generated for each of the XML, which may affects the cost of updating and in turns that of replacement. The construction of the multiplication factors is determined by a DTD of XML document, which is then stored for use in later phase.

Each node of XML documents is associated with a score, which is initialized to zero and affected by the access patterns during runtime. The initial cache plan may affect the performance of caching as the cost updating process in runtime phase depends on the multiplication factors generated in this phase.

```
<article>
    <title>
    A Relational Model for Large Shared Data Banks
    </title>
    <publisher>
            <publishername>HKPolyU Publishing Co. Ltd.
            </publishername>
            <address>Honghom, Kowloon
            </address>
    </publisher>
    <author>
            <name>
                    <firstname>E.F.</firstname>
                    <lastname>Codd</lastname>
            </name>
            <age>54</age>
    </author>
</article>
```

Figure 4. An example XML document

The initial phase consists of three steps. The first step is the construction a directed graph from a DTD of that XML documents. By expanding all entities definitions within the directed graph, a tree is generated. Figure 3 shows a sample DTD and Figure 4 shows an instance of the DTD. Figure 5 shows the tree constructed from the DTD.

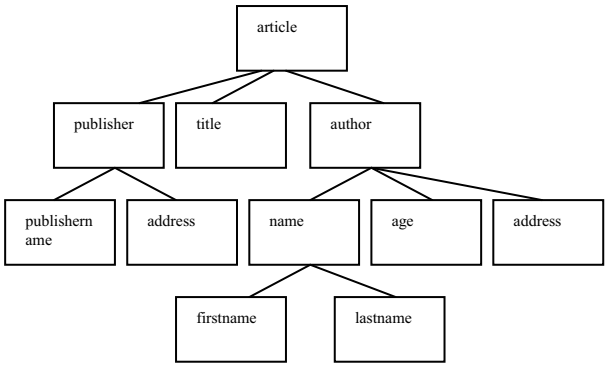


Figure 5. A directed graph constructed from the DTD

The second step is the assignment of weightings to each of the leaf nodes of the directed graph. Table 1 shows the relative weightings of properties observed from common DTDs for XML documents. The weights represent the relative strength of the relationship among nodes of a DTD from design view. The higher is the weighting, the stronger is the relationship.

The assignment of multiplier and replacement cost is based on the expected probabilities of occurrences of those nodes in instances of the DTD. Basic replacement cost depends on the occurrence notation of a node. The cost implies the relatively importance of that node derived from the DTD whilst multiplier is a factor that depends on the elements' content specifications.

We model the relationship with three kinds of factors. The first one is the common design characteristics of a DTD. An example is that a node with mandatory notation is more important than optional one. The second one is the probable inter-relationship among nodes. In this paper, we classify this kind of relationship into mutual exclusive, co-occurrence, sub-typing, and no relationship at all. The third one is the relative size of the actual instantiation of the nodes. Therefore, the model of relationship can be constructed as:

$$R \sim (T + I) / S$$

where R is the relative strength of importance, T is the common design characteristics of a DTD, I is the type of inter-relationship, and S is the relative size of the probable instantiation of a node.

For easier operation, the size of the probable instantiation of a node is replaced with the number of options of that node and the common design characteristics into either no implication, optional or mandatory, which is normalized to 1, 1/2 and 1 respectively. For the type of inter-relationship, we normalize the co-occurrence, sub-typing, no relationship and mutual exclusive relationship into numerical values 1, 1, 1/2, 1/n where n is the number of options.

Table 1. Assignment of weighting factors

Element with Occurrence Notation	Basic replacement cost(Q)
?(Optional)	1/2
*(Zero or More)	1
+(One or More)	number of options
|(OR)	1/number of options
No Notation	1

ATTLIST	Basic replacement cost(Q)
Element(s) with attribute ID	1
Element(s) with fixed attribute	1

Element's ContentSpec	Multiplier(T)
PI	1
ANY	1
Mixed	Sum of all possible weightings/number of options
Comment	1/2
Fixed	2
PCDATA	2

Basic replacement cost (Q_i) is multiplied by the corresponding multiplier(T_i) to obtain the multiplication factor that leaf node. Assignment process is then performed in a bottom-up manner. The score of a non-leaf node is the aggregated sum of its descendent nodes multiplied by the multiplier of that node. The assignment process iterates until the root node is reached. The multiplication factor for a given node W_1 is:

$$\begin{aligned} W_i &= T_i * Q_i \\ &= T_i * \Sigma W_{i-1} \\ &= T_i * \Sigma W_{i-1} * \Sigma W_{i-2} * \ldots\ldots * \Sigma W_1 \end{aligned}$$

The multiplication factor reflects the relatively importance of a given node within a DTD for a given XML document. The value of a factor derived from the DTD is a numerical representation for manipulation in the later phase. It is expected that nodes having higher values are more important as they appear more often in instances of that DTD. Figure 6 illustrates the weighted directed graph of the example DTD.

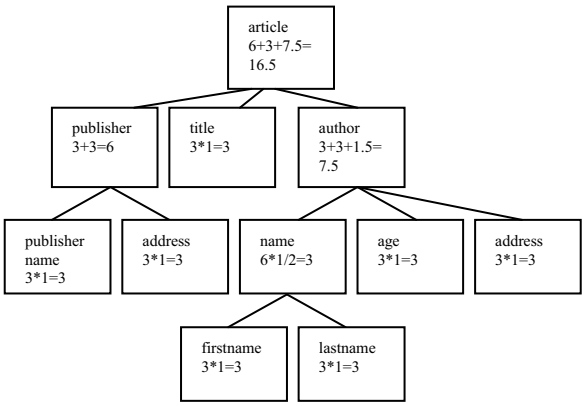


Figure 6. The weighted directed graph constructed from DTD

In step 3, the multiplication factors constructed in step 2 are derived. Apart from these factors, each node of the selected XML document has an associated score. Although both the score and multiplication factor are derived and initialized in this phase and used afterwards, the differences between them is that the latter reflects the schema characteristics of the DTDs while the former is a set of variables for manipulation in runtime phase. The associated score of each node C_i is initialized to zero for each DTD fetched from remote server.

$\forall C_i \{C_i = 0\}$

As such, a mapping of DTD and the multiplication factors, in additional to the set of initialized scores are generated in this phase.

4.2. Runtime Phase

4.2.1. Score Updating

For each of the XML query requested by the client, the timestamp of the corresponding local document is compared with that of remote server to check for data coherency. The invalidation of data triggered by the discrepancies of these two timestamps results in the retransmission of the corresponding XML document and updating of local timestamp. For a cache hit, the score of the selected node(s) C_i is/are re-calculated by the corresponding adjustment factor δ_i:

$$C_i = C_i + \delta_i$$

The updating of cost in each node is affected by the size of the element, access frequency and the difference between fan-out and fan-in of a node within the actual XML instantiations of the DTD. Hence, the cost, δ_i is defined as:

$$\delta_i = \alpha * ln(W_i * S/S_i * F_i/F * E_i/E)$$

where a is a constant, W_i is the Multiplication Factor derived from initial phase, S_i is the size of node(s), S is size of the XML object fragment, F_i is the access count of that node in the cache, F is the total hit count of the XML object fragment, and E_i is a fan-out factor which is determined by the number of edges connected to children nodes U_i and that of parent nodes V_i:

$$E_j = U_i - V_i \text{ iff } U_i - V_i >= 0$$
$$E_j = 1 \text{ if } U_i - V_i < 0$$

All other unaffected nodes $\{C_i\}$ are deducted by an adjustment factor δ_j:

$$C_j = C_j - \delta_j$$

The adjustment factor δ_j of unaffected node is determined by:

$$\delta_j = \sum \delta_i / |C_j|$$

4.2.2. Object Replacement

Object replacement process encompasses three steps:

a) When the required XML document fragments are stored in the cache memory, the local copy is returned to client and no object replacement occurs.

b) Whenever cache miss or cache incoherency occurs, the requested object(s) retrieved from remote servers will be stored in local cache memory as long as the maximum available cache size is not exceeded.

c) Whenever cache miss or cache incoherency occurs and the space available in local cache memory is not enough to accommodate the requested object(s) retrieved from remote servers, object replacement process will begin and each queries will be evaluated by accumulating the score of the nodes(C_i) across the axis for the corresponding XPath. The cached query and object pair having the least evaluated total value will be evicted and the process iterates until the space available can accommodate the executed query and fetched XML fragment.

In other words, the following two criteria must be met for the occurrence of object replacement:

$$(i) \quad \forall C_k \{ \sum C_i > \sum C_k \}$$
$$(ii) \quad \sum Size(C_i) <= \sum Size(C_k)$$

where $C_i \in$ {sets of the retrieved nodes} and $C_k \in$ {sets of the nodes of the path expression to be replaced}. Figure 7 is the summary of our proposed algorithm StructCache:

```
Find W_i for all nodes of a given XML document
C_i ← 0.0
For each request query p do
    If p is in cache
        then
        C_i=C_i + δ_i  where δ_i = α * ln(W_i * S/S_i * F_i/F * E_i/E) for all affected nodes
        C_j=C_j - δ_j  where δ_j = ∑δ_j / |C_i| for other nodes
    else fetch p
    While there is not enough free cache for p
    Evict fragment(s) with min{∑ C_i (q)|q} are the nodes of the axis of XQL data query
    in cache}
```

Figure 7. Summary of StructCache

GDFS employs the dynamic aging mechanism or inflation value to simulate the reference correlation of web traffic. Instead, our proposed algorithm uses the reward and punishment mechanism in updating and does not need to determine the base value during the reset step of cache hit. Utility value reflects the normalized expected cost saving if the object stays in the cache. Given the long-term reference pattern is stable, GDFS uses f(p) * c(p) / s(p) that consider the reference count, cost of fetching and size of object as well as the aging factor to approximate the utility value. By contrast, the StructCache algorithm considers the structure of XML document, in additional to the temporal locality, spatial locality, cost of fetching and size of object.

5. Experiment Results

To facilitate our evaluations of different caching strategies, we have performed a simulation to test the performance of StructCache against the GDS and GDFS algorithms. We model the clients' requests of XML by a list of predefined XML objects queries, which is executed sequentially. Different algorithms are implemented in proxy between the client and remote servers. The proxy is responsible for handling the clients' requests and returning the results to clients. In this experiment, our focus is on the performance gain by caching the queries' result of XML fragments instead of whole documents and the study the effects of the incorporation of the structure of DTD and the XPath in replacement algorithms.

During the experiment, the execution sequence of the batch of queries remains unchanged throughout the experiment. In our experiment, the predefined queries can be classified into the following four types:

1) Queries have similar XPath but different predicates.
   /article/author/name[firstname='Peter'] and /article/author/name[firstname='Tom'] are example of this type of queries
2) Queries have results that are subsets of results of previous queries
   /article/author and /article are example of this type of queries
3) Queries randomly select nodes and have no predicate
   /article/author/name/firstname and /article/publisher/publishername are example of this type of queries
4) Queries randomly select nodes and have arbitrary predicates
   /article/author/name[firstname='Tom'] and /article/publisher[publishername='ABC Publisher'] are example of this type of queries

We compare the effectiveness and relative gain of performance of StructCache with GreedyDual Size(GDS) and GreedyDual Frequency-Size (GDFS) algorithms in terms of the Bit Model and the Cost Model.

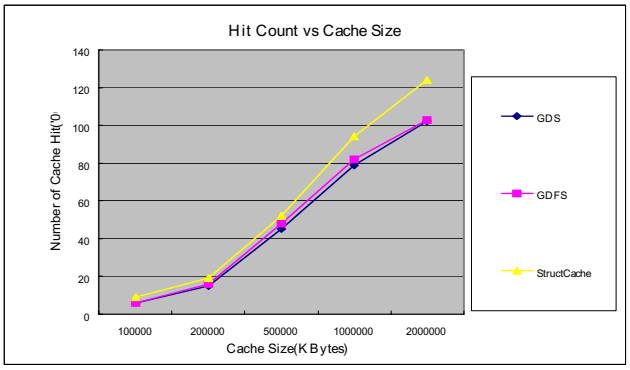


Figure 8. Hit counts versus cache Size for different evaluated algorithms

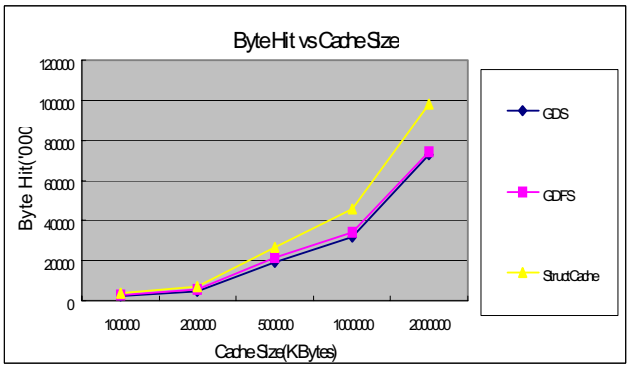


Figure 9. Byte hit versus cache Size for different evaluated algorithms

Figure 8 shows the plot of hit counts versus cache size for the three algorithms. Figure 9 shows the plot of the corresponding byte hit versus the cache size. Both results illustrate that larger cache sizes give higher hit counts and byte hits and the StructCache algorithm outperforms the GDS and GDFS algorithm in terms of the Bit Model and Cost Model for XQL queries. The improvement in hit count is up to 20% and 22% in byte hit. The gain of performance is highly related to the types of queries. The result is particularly apparent for XML documents with relatively large document size to cache size ratio. For retrieving of large size XML documents, the fetch cost is relatively large and caching of XML fragments not only reduces the size of cache objects, but also reduces the page faults.

The incorporation of syntactic features of DTD as a parameter in cost updating function and replacement algorithm gives additional information about the objects to be cached. One reason is the design of schema usually considers the relatively importance of a node and the relationship among various nodes. For the sub-typing relationship and the occurrence notation can be exploited. The stream of XQL queries are also exploited by our proposed algorithm. In StructCache algorithm, the XPath of XQL queries are used in score updating and the determination of objects replacement. We found that the

following two kinds of XQL queries are well handled with our proposed algorithm:

a) Subsequent queries are specific sub-tree(s) of precedent queries

b) Subsequent queries are of the same level and path with precedent queries

In other words, it performs well when the stream of queries exhibits the spatial locality characteristics and the user access preference is 'moving from general to specific'. Results also indicate that the performance is still comparable to traditional caching algorithms even though the two above criteria cannot be met.

6. Conclusion

In this paper, we present a XMLCache caching framework for XML data under the mobile environment. It takes care of both XML and non-XML data and the replacement algorithm considers the syntactic characteristics of the XML schema in additional to the access pattern of XML queries, the long-term access frequencies and fragment size. By using the Cost Model and the Hit Model as the metrics, preliminary experiments show that our proposed algorithm outperforms the GDS and GDFS for the same configurations of cache sizes and user queries.

Acknowledgement

The work reported in this paper was partially supported by Hong Kong CERG Grant – PolyU 5094/00E.

7. References

[1] Saied Hossenini-Khayat, "Replacement algorithms for object caching", Proceedings of the ACM symposium on Applied Computing, Atlanta, GA USA, Mar 1998.

[2] R. Wooster and N. Abrams. "Proxy caching that estimates page load delays". In Proceedings of the 6th International WWW Conference, 1997.

[3] Steve D. Gribble and Eric A. Brewer. "System design issues for Internet middleware services : Deductions from a large client trace". In Proceedings of the 1997 USENIX Symposium on Internet Technology and Systems, 1997.

[4] Pei Cao, Edward W. Felten, Anna R. Karlin and Kai Li. "A study of integrated prefetching and caching strategies", Proceedings of the 1995 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems, May 1995.

[5] Susanne Albers, Sanjeev Arora and Sanjeev Khanna. "Page Replacement for General Caching Problems", In Proceedings of the Tenth ACM-SIAM Symposium on Discrete Algorithms, 1999.

[6] S. Irani. "Page replacement with mult-size pages and applications to Web caching". Proceedings 29th Annual ACM Symposium on Theory of Computing, 701-710, 1997.

[7] Ed Sutherland. "Predicting M-Commerce Trends for 2002: Part II" . http://www.mcommercetimes.com/Industry/211, Jan 2002.

[8] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, and J. Worrel, "A hierarchical Internet Object Cache", Usenix'96, January 1996.

[9] J. Yang, W. Wang, R. Muntz, and J. Wang, "Access Driven Web Caching", UCLA Technical Report #990007.

[10] S. Michel, K. Nguyen, A. Rosenstein, L. Zhang, S. Floyd and V. Jacobson, "Adaptive Web Caching: towards a new caching architecture", Computer Network and ISDN Systems, November 1998.

[11] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z. Broder, "Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol", IEEE/ACM Transactions on Networking, Vol. 8 No.3, June 2000.

[12] Jia Wang, "A Survey of Web Caching Schemes for the Internet", Cornell Network Research Group(C/NRG), 2000.

[13] D. Wessels and K. Claffy, "Internet Cache Protocol(ICP)", version 2, RFC 2186.

[14] Anja Feldmann, Ram?n Cáceres, Fred Douglis, Gideon Glass, and Michael Rabinovich. "Performance of Web Proxy Caching in Heterogeneous Bandwidth Environments". AT&T Labs-Research, Florha Park, NJ, USA, 1999.

[15] Shudong Jin and Azer Bestavros, "GreedyDual Web Caching Algorithm - Exploiting the Two Sources of Temporal Locality in Web Request Streams", Boston University, 2000.

[16] Virgílio Almeida, Azer Bestavros, Mark Crovella, and Adriana de Oliveira. "Characterizing Reference Locality in the WWW". Department of Computer Science, Boston University, 1996.

Topic Distillation: Content-Based Key Resource Finding

K. L. Kwok
Computer Science Dept., Queens College, CUNY
kwok@ir.cs.qc.edu

Q. Deng
Computer Science Dept., Queens College, CUNY
peterqc@yahoo.com

N. Dinstl
Computer Science Dept., Queens College, CUNY
emc21@earthlink.net

Abstract

This paper describes an approach to the topic distillation task of TREC-2002 for finding key resources in a 1.25-million-page collection of .gov domain. Content weighting of page body, anchor texts and IR techniques were employed as a primary step. Additionally, page out-linked content and in-link counts were investigated for their ability to improve key resource detection.

Output from distillation is a rank list that may come from very few hosts. Users on the other hand might prefer answers that come from diverse sources that would provide a more balanced-view of a topic. We propose host diversification processes to adjust the answer list based on a host-repeat factor and host weights.

1. Introduction

In something like a dozen year's time, the World Wide Web has fundamentally transformed the way people search for needed information. This impact is realized because WWW provides: i) a virtual, global library with such an enormous scale that practically most topics, aspects of data and information is included in its coverage; ii) an internet infrastructure that has enabled users anywhere in the world to access this library without time and space constraints; iii) search engine software that allow users to locate relevant information with relative ease. The dream of the Memex machine [6] seemed to have been realized. Naturally, for such a young technology, there are problems and deficiencies that need to be solved, and in time they would be resolved. Some examples of such issues include: how to avail pre-digital print and other content material to be searchable on the Web economically and how to store ever increasing volumes of data efficiently (e.g. [2]); the need to provide fault-tolerant broadband network in the country for ever increasing object sizes (e.g. [9]); and improvement of the quality of search engines (e.g. [18]).

To facilitate users locate needed information, various models and methods have been developed. These include: identifying authority and hub pages for a user topic [15], page ranking for retrieval [4], community discovery [16], finding homepage/subdirectory page of a topic named in a query [11], modeling of the web [12], etc. to name a few.

Topic distillation ([7], [3], [1]) is another aspect of Information Retrieval (IR) that is of interest for Web users. When a user poses a query for retrieval, the returned page list from search engines may be mainly irrelevant. On other occasions, it could contain hundreds of pages that are all seemingly relevant, overwhelming the user. In such situations, it would be helpful to identify a small number of items that are of high quality, utility and with sufficient source diversity. These are pages that one could bookmark for later consultation for example. This investigation focuses on this topic distillation aspect of IR. Section 2 discusses the meaning of topic distillation and reviews the approaches that have been proposed. Section 3 presents the TREC-2002 environment used for this study. Section 4 details our approach, and Section 5 has our results. Section 6 contains our conclusions.

2. Topic distillation

2.1. Characteristics of the task

Topic distillation is described in [19] as the task of:

> "finding a list of key resources for a particular topic. A key resource is a page which, if someone built me a (short) list of key URLs in a topic area, I would like to see included."

Like many other notions in the information field (such as relevance, aboutness, etc.), 'key resource' does not have a clear-cut definition. It resonates with the user: 'I know it when I see it'. The Web-track Guideline has described characteristics of key resources via some possibilities, including e.g.: i) home page of a site, or main page of a sub-site, dedicated to the topic; ii) outstanding content pages on the topic; iii) a hub page with useful links to content pages for the topic; iv) a relevant service. Example ii) is high quality content pages. Example i) and

iii) are likely introductory pages with links that serve to gather or summarize relevant on-topic target pages into one convenient place – these pages may contain some but not necessarily detailed content themselves. Thus, "key resources are more than relevant pages".

2.2. Key resources

Examples of key resources, and their difference from relevant pages, are given in the Guideline for "obesity in the U.S." query. For example, although these pages:

www.surgeongeneral.gov/topics/obesity
/calltoaction/principles.htm
www.surgeongeneral.gov/topics/obesity
/calltoaction/factsheet02.pdf
www.surgeongeneral.gov/topics/obesity/calltoact
ion/fact_glance.htm

are good relevant answers to the query, the following page, which introduces the topic and points to all three, is considered a correct key resource answer:

www.surgeongeneral.gov/topics/obesity/

Other examples are:

www.lbl.gov/Workplace/patent/iplinks.html
(query: "intellectual property")
www.niddk.nih.gov
(query: "symptoms of diabetes")

They look like an index to organize and summarize good sources and sites concerning this topic, irrespective of whether they are within the same host or not.

2.3. Approaches to topic distillation

Topic distillation appears to originate in [7]: "Given a topic, the algorithm first gathers a collection of pages from among which it will distill ones that it considers the best for the topic". They employed the HITS algorithm [15] to compile resources among a set of candidate pages, but enhanced the adjacency matrix elements with text weights. Vectors of hub (H) and authority (A) pages mutually reinforce each other after a number of iterations of HITS: H=WA, and A=W'H with normalization of vectors in between, and the top n authority/hub pages become distilled answers. In [8], the authors refined the approach in their CLEVER project, like removing edges within same hosts. [3] noticed certain deficiencies with Kleinberg's approach, and proposed sharing authority or hub weight among edges from/to the same host, as well as using content to prune and regulate edge weights.

The above approaches employ Kleinberg's HITS as the primary algorithm and add content weighting as a refinement. Their answer pages are discovered based on mutual reinforcement of in-links and out-links, and is more of a consensus-finding tool among page builders' opinion. Moreover, since conferring authority (hub) value to a page is not always trustworthy; these approaches suggest having links from the same host be consolidated or down-weighted.

In the context of the Web Track distillation task, a page may become a key resource not only because of its content, but also because of the quality of its out-linked pages. This definition puts equal emphasis on the content of out-linked pages whether they are within-hosts (for content organization for example), or across hosts (actual references). Cross-host edges are much fewer than within-host edges in this .gov collection (see next Section). It is also not essential for a page to have many reinforcing in-links in order to satisfy key resource definition, such as the case for less popular topics or brand new creations. This suggests that key resource finding could use a different process other than HITS. We propose employing content weighting as the primary method because of the way key resources is defined, while adding link information in a more traditional way.

3. TREC-2002 web track .gov collection

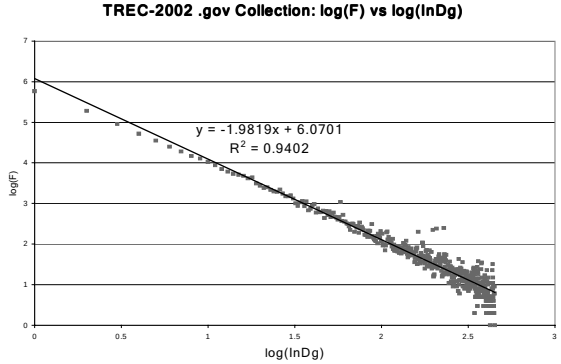
TREC has supported studies of Web retrieval in the past [21] by organizing experiments in a repeatable environment using static web collections. Participants are required to use the same document collection and a common set of topics (from which queries for retrieval are derived), and return retrieval lists according to pre-announced objectives. Construction of the collection has been described in the Guideline -- a breadth-first crawl, starting 2002, of the first million pages in the .gov domain, plus ¼ million non-html pages. There are 49 topics. Top sub-lists of the participant results are pooled and manually judged for distillation correctness by TREC assessors, centrally evaluated, and measures such as precision and recall are provided (see for example [20]). The advantages of the TREC approach include: a reasonably number of topics (49), unbiased assessors providing manual evaluation of good quality, better exhaustiveness because of the large number of participants (17) and runs (71), and the static nature of the collection allows investigators to do repeatable experiments and train their approaches. The topics come with three sections (title, description and narrative). By choosing to use either the title only, title with description, or all sections, one can experiment with distillation using varying query lengths.

Some statistics of the TREC 2002 .gov collection [10]:

# of pages	~	1.25×10^6
# of links	~	11.2×10^6
# unique source pages	~	1.07×10^6
# unique target pages	~	1.15×10^6
Avg. links/page	~	8.96
# of hostnames	~	8.00×10^3
# of cross-host links	~	2.50×10^6
avg # of cross-host links per host	~	300

Graphs of in-degree and out-degree distribution are shown in Figs.1a,b. The in-degree fit obeys the power law as found by other investigators, but with a smaller exponent of ~1.98 rather than the ~2.1 as given in [5], while the out-degree distribution has the characteristic 'droop' at small values. The average number of links ~ 9 is larger than 7 reported before. It is to be noted that others used much larger collections.

On average each host has $1.25x10^6$ /($8.00x10^3$) ~156 pages, and over 3/4 of the links (~$8.7x10^6$) are within same host. Cross-host links average to about 300 per host, or about 2 cross-host links per page on average. On the other hand, the average number of within-host links is $8.7x10^6$ /8.00 x 10^3 ~1000 per host or ~6.5 per page.



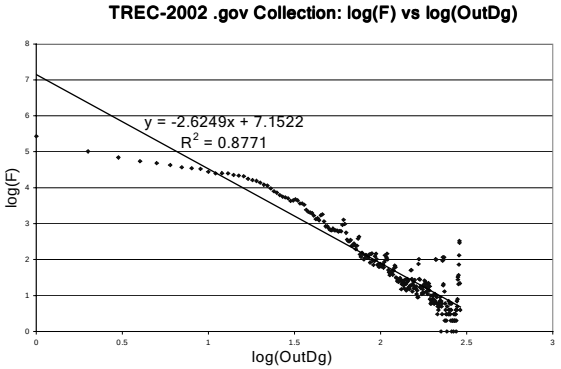


Fig.1a,b: Distribution of In-Degree and Out-Degree for the TREC-2002 .gov Collection

4. Content-based topic distillation

4.1. Defining key resource candidate pool

Some investigators have reported that for this task, the HITS algorithm did not perform well (e.g. [22], [23]). We take a more content-oriented to distillation -- do normal IR on the collection, follow with key resource finding procedures on the ranked retrieval list.

Each page from .gov collection is separated into four objects, viz.: a) .txt – containing the body full text; b) .href – incoming anchor texts linking to this page; c) .title – page title only; and d) .meta – containing texts under the meta-tags. The .txt and .href type documents are processed with Porter's stemming while the others do not. This is done with the aim to have higher precision when doing retrieval with the shorter documents. In particular, the .href object is independent of whether the target page has good content description or not.

Four collections were formed. Retrieval with our PIRCS system [17] employing one same query, stemmed or un-stemmed according to the collection type, results in four rank lists. Each page k has possibly four retrieval status values (RSV's) R_{ik}; they are linearly combined into one value S_k that defines a single resultant rank list:

$$S_k = \Sigma \alpha_i R_{ik}, \quad (1)$$

where i =1, .. 4, corresponding to the 4 retrieval lists;
$\alpha = <\alpha_1,\alpha_2,\alpha_3,\alpha_4>$ is a vector of combination coefficients.

This combines evidence from multiple sources and is often done in IR to boost retrieval effectiveness. The top p ranked items form a candidate pool. Key resource finding and a page's out-linked content are limited to this pool.

4.2. Key resource detection: combining self and out-linked content

By definition, key resource finding needs to characterize the content of each page and its out-linked pages. One method simply employs S_k, normalized by ΣS_k , as the content weight C_k of page k. The following transformation that preserves ranking is actually used since it often provides more consistent results:

$$C_k = \exp(a+b(S_0 - S_k)) / [1+ \exp(a+b(S_0 - S_k)] \quad (2)$$

where the constants (a,b) are chosen as (1.0,1.5).

To account for a page's out-linked content, we define for each page k a link weight L_k by averaging the immediate out-linked neighbors' content weight as follows:

$$L_k = \Sigma_{\text{outlink\_pages\_j}} C_j / m^e \quad (3)$$

where m is the out-link count,
e = 0.8 appears to work better than 1.0

The out-link count m includes all links pointing within .gov, not just those to the candidate pool. Page-links outside the candidate pool are considered to have no relevant content. This normalization by m^e may partially account for the density of good links for each page.

Finally, we combine both content and link weights to define a composite weight for page k, P_k, that may be representative of its overall distillation value:

$$P_k = \beta C_k + (1-\beta)L_k \quad (4)$$

Various coefficients β were tried, values between .5 to .8 seem appropriate. This page weight, P_k, is then employed

to rank candidate pages of the pool for their key resource status with respect to a given topic.

4.3. Key resource detection: adding in-degree

We also experiment with incorporating the influence of in-degree counts d on key resource detection by introducing a 'popularity factor' g(d). Every citing to a page may confer evidence that this target page is useful. [1] has shown that the in-link counts of a page is similarly useful for quality indication as authority in page collections that are from the same directory. We tried new weighting formulae C', P' as follows:

$$C'_k = C_k + g(d) \quad (5a)$$

$$P'_k = P_k + g(d) \quad (5b)$$

where $g(.) = c/[1 + \exp(-b*(d-D))]$, d=in-degree. (6)

constants are chosen as: c=0.25-1.0, b=0.25, D=20

In order to render different measures comparable, d was transformed by g (.), chosen as the sigmoid function, so that effects of low degrees are squashed to small ~0.0 values, while high degrees saturates to 1. Modifying parameter values in Eqn.6, g(d) can also simulate linear behavior.

4.4. Key resource answers with host diversification

In principal, returning a list of ranked key resources for a topic would be sufficient for topic distillation. However, the answer list may predominately come from very few hosts. For sharply focused topics such as: "US passport", source diversity is of little utility. For more social, controversial topics such as: "global warming", "liver cancer treatment", one would expect users to prefer results from a diversity of sources in order to gain a more complete, balanced view of the topic.

A simple way to bring on diversification is to suppress same host pages from the ranked answer list based on a host-repeat factor r, scanning down from the top-ranked. If one needs maximum diversity in the top 10 for example, the host-repeat factor is set to 1 and only the 10 best pages with *unique* hosts would be presented. For no diversity checking, the host-repeat factor is set to >10, and the original is the answer list.

Another approach is to additionally rank the hosts. The ultimate goal of topic distillation is to satisfy information needs of a user. Answers to topic distillation may not be just key resources, but pages coming from hosts that are relevant. Once within such sites, the user may also conveniently browse other pages for additional pertinent data. A key resource page coming from a host with rich pertinent context would confirm and confer reliability of the page in the mind of the user, compared to key resources that may be good but with few related context pages in its neighborhood. For this purpose, the candidate pool can be clustered into host groups based on the first URL segment. To rank hosts, we define host weight as an average of the content C_k of its pages. Once hosts and pages are weighted, one dispenses out r pages from each top host until the required output size is satisfied.

5. Results and discussion

5.1. Distillation based on information retrieval techniques

A basis approach to distillation is simply use the ranking outcome of a search engine. No out-link content is considered. For us, this rank list is obtained from four collections separately or by RSV combination (Eqn.1).

Table 1 shows the precision values at 10, 20, 30 pages (P10, .. etc) for separate retrieval lists as well as some representative combinations. Among the four separate collections, .txt has the best performance as expected because documents in .txt are longer. The others have much poorer results. The anchor text collection .href performs better than .title; .meta is non-competitive. However, when retrieval lists are combined in various ways, P10 results can surpass .txt alone by 12% to 32%. For short queries (~3 terms), the .title retrieval list combining with .txt using coefficients 1/3 and 2/3 produces the best P10 result of 0.2347. Combining .href with .txt also helps to produce a P10 value of 0.2163, but less than the synergistic effect of .title. This could be due to the asymmetric processing: .txt has stemming but .title does not; this may help boost the precision. However, when .href was processed without stemming, it produces worse results (not shown) by itself or in combination. Content carried by .title and .href could be quite different.

Table 1 also shows distillation results using medium length (~7.3 terms) and long queries (~16 terms). The behavior of the .txt collection by itself has the expected result: longer queries give better precision as in normal IR: P10 improves from .1776 to .2 for medium, and .2224 for long. The other collections separately have more erratic behavior. Combination of lists however produces the best precision for long query P10 value of .2510. Queries of different lengths require different parameters to obtain good results. The general observation is that short queries can achieve P10 range .21 to .23 using a combination coefficient of .65 to .7 for the .txt list and smaller values for the others. For long queries, using .txt coefficient of .75 or .8 can attain P10 of 0.23 to 0.25.

Long queries are unrealistic. However its superior result can show how much short queries may be missing. Medium length queries generally have the worst results and would not be reported in later sections.

Table 1: Results of Distillation: Information Retrieval Techniques for Three Query Types (Pnn =Precision at 10, 20, 30, <a1,a2,a3,a4> are combination coefficients)

	.txt (stemmed)	.href (stemmed)	.title	.meta	combine <3/4,1/4 0,0>	combine <2/3,0, 1/3,0>	combine <.7,.1, .2,0>	combine <.65,.1, .25,0>	combine <.7,.1, .1,.1>
Query Type: **short** (title)									
P10	**.1776**	.1245	.1082	.0612	.2163	**.2347**	.2347	.2265	.2184
P20	**.1510**	.0939	.0929	.0480	.1745	**.1857**	.1847	.1888	.1755
P30	**.1340**	.0755	.0748	.0381	.1456	**.1558**	.1605	.1592	.1510
Query Type: **medium** (title + description)					<.8.1.1,0>	<.75,.05,.2,0>			
P10	**.2000**	.1204	.0939	.0816	.2061	.2245	**.2347**	.2286	.2245
P20	**.1561**	.0888	.0878	.0510	.1714	.1806	**.1694**	.1786	.1786
P30	**.1327**	.0782	.0694	.0435	.1469	.1490	**.1503**	.1531	.1551
Query Type: **long** (all sections)					<.8,.2,0,0>	<.8,0,.2,0>			
P10	**.2224**	.1265	.0918	.0592	.2306	.2388	.2469	**.2510**	.2490
P20	**.1837**	.0949	.0765	.0439	.1847	.1878	.1857	**.1888**	.1939
P30	**.1612**	.0789	.0633	.0367	.1762	.1707	.1735	**.1707**	.1694

5.2. Distillation using Link Information

5.2.1. Accounting for out-linked content weight. As discussed before, a characteristic of key resource is that they may point to good content while not necessarily quality-rich themselves. Since pure IR technique (Section 5.1) does not use out-linked content, we study whether accounting for it as in Eqn.3,4 may attain better results.

The top n documents from the best retrieval list in Table 1 (for short or long queries) define a candidate page pool (pool size = n). This pool limits our search for key resources to ones with some reasonable content. Eqn. 4 defining the distillation value of a page, P_k, is used to re-rank the pages in the pool. Since the basis of the raw RSV from the PIRCS retrieval system is log odds values (unconstrained real numbers), it is more convenient to normalize them to values between 0 and 1. We try normalizing by division by the sum of retrieved RSV values, and by using Eqn.2.

Results in Table 2 shows that Eqn.2 RSV normalization is preferable to normalizing linearly (middle 2 columns). Adding out-linked weight (L_k) to source content (C_k) shows consistently slight improvements for long queries. Short queries show an improvement at P10 (pool=50) but negative for others. It seems that out-linked content, as suggested in key resource definition, has only small effects. It is possible that most answer pages in these experiments have good content and therefore linked content has little impact. Another reason may be that content weights C_k and link weights L_k are noisy (especially for short queries). We examine this issue further below.

Table 2: Results of Distillation: Accounting for Out-linked Content Weight (Pnn =Precision at 10, 20, 30)

	Best IR Result from Table 1	$P_k = C_k + L_k$ (linear norm)	$P_k = C_k + L_k$ (Eqn.2 norm)	$P_k = C_k + L_k$ (Eqn.2 norm)
Query Type: **short**		**pool size =300**	**pool size =300**	**pool size =50**
P10	**.2347**	.2026	.2327	**.2408**
P20	**.1857**	.1603	.1765	**.1827**
P30	**.1558**	.1359	.1517	**.1537**
Query Type: **long**		**pool size =300**	**pool size =300**	**pool size=400**
P10	.2510	.2388	.2531	**.2551**
P20	.1888	.1939	.1959	**.1939**
P30	.1707	.1673	.1769	**.1748**

The out-links of a page can be inaccurate because the page composer may insert references that are out of topical area (e.g. see the tropical fruit example in [8]). We try to account for this by using all out-link counts m in Eqn.3. In addition, the content weights returned by our IR system can also be inaccurate. For example, at a size of 100, the best retrieval lists in Table 1 have P100 less than 0.1. Assuming optimistically that all relevant content pages (not just key resources) may double the number of answers, the P100 value is still less than 20 relevant pages

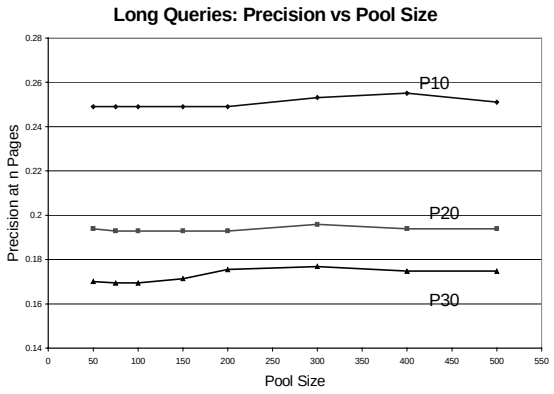
per 100. Hence L_k weight can be unreliable, especially for short queries: links to good resources may have low content weight, and vice versa. However, we may be able to demonstrate its effect by varying the pool size. Small pool sizes have better precision ratio for out-linked content, but ignore higher-ranked pages as candidate key resources. Large pool sizes have the opposite effect.

Fig.2a,b shows how Table 2 precision values (Eqn.2 norm) vary with pool sizes. Although variations are small, we see a trend that for long queries it is preferable to use pool sizes like 400, while for short queries there is a distinct rise in precision for smaller size like 50. Short queries return noisier ranked lists from IR; it is best to limit resource finding in the more signal-rich region of the retrieval.

5.2.2. Adding effects of in-link counts. In-link counts were shown to be as useful as authority measures under certain circumstances [1]. We employed in-link counts to augment key resource finding based on Eqn.5 to account for popularity for both content (C') only or for content and link (P'). Table 3 displays the results. Effects however are small, mostly negative compared to those from IR and Table 2.

5.3. Distillation with diversity of hosts

5.3.1. Diversity based on host-repeat factor. Host diversification is potentially beneficial for users to get more balanced-view answers. Table 4 shows host



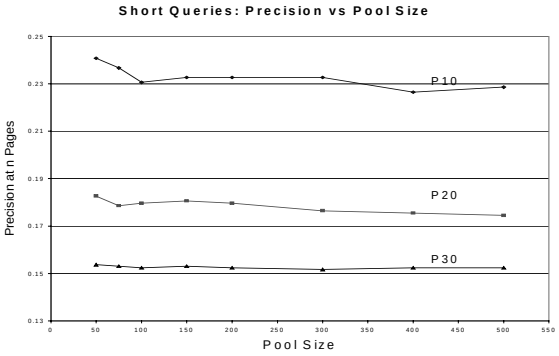


Fig.2a,b: Variation of Distillation Precision at 10, 20 and 30 Documents Retrieved with Pool Size

Table 3: Results of Distillation: Accounting for In-Link Counts (Pnn =Precision at 10, 20, 30)

	Best IR: Table 1	Eqn.5a C'	Eqn.5b P'	$P_k=C_k+L_k$ Table2	Eqn.5a C'	Eqn.5b P'
Query: **short**		pool size =300		pool size = 50		
P10	**.2347**	.2367	.2327	**.2408**	.2347	.2367
P20	**.1857**	.1857	.1776	**.1827**	.1857	.1816
P30	**.1558**	.1551	.1511	**.1537**	.1551	.1537
Query: **long**		pool size =300		pool size = 400		
P10	**.2510**	.2449	.2531	**.2551**	.2449	.2551
P20	**.1888**	.1898	.1929	**.1939**	.1898	.1929
P30	**.1707**	.1714	.1748	**.1748**	.1714	.1728

Table 4: Results of Distillation: using Host-Repeat Factor (P10 =Precision at 10 documents retrieved)

	Best IR: Table 1	$P_k = C_k+L_k$ Table 2.	Host-Repeat Factor =3	Host-Repeat Factor=2	$P_k = C_k+L_k$ Table 2.	Host-Repeat Factor=3	Host-Repeat Factor=2
Query: **short**		pool size =300			pool size = 50		
P10	**.2347**	.2327	.2204/ **.2184**	.2224/ **.2204**	.2408	.2204/ **.2204**	.2204/**.2204**
Query: **long**		pool size =300			pool size = 400		
P10	**.2510**	.2531	2571/**.2612**	2347/ **.2408**	.2551	2571/**.2612**	.2388/**.2408**

diversification effects with host-repeat factors of 3 and 2 on the P10 distillation results of Table 2. There are two values in each cell. The second value (bolded) restricts promotion of a page only if it is in the top 50 of the IR retrieval list to assure better quality. This additional restraint produces better precision for long queries. Host-repeat factor of 3 performs better than 2. Since evaluation does not take diversity into account, using repeat factor of 2 replaces too many relevant pages from top-10 that are repeat-hosts. For short queries, the effect is little change to negative – probably the quality of content ranking is not sufficiently accurate for this purpose. The example rank list on the right column shows the top 10 retrieved pages for query #551: "intellectual properties", without and with host diversification. Original short query retrieval of Table 2 returns 9 pages from the same host in the top 10. Using host-repeat factor of 3 (this section 5.3.1), only the top 3 from cybercrime.gov remains. New sources such as commdocs.house.gov, www.usdoj.gov, are included. Using host-repeat factor of 2 with host weighting (Section 5.3.2), more diverse answers are displayed. However, because better-ranked relevant pages are replaced by ones further down the list, host diversification does not mean that precision can be preserved. In this example, none of the replacing pages are answer pages.

Averaging over 49 queries, 2.5 pages per query are replaced in the top 10 with a host-repeat factor of 3.

5.3.2. Diversity based on host weight. Another method of realizing diversity is to organize the candidate pool into host groups, then rank hosts using average content weight C_k, with the pages within a host ranked by P_k as before. Top r pages are selected from top hosts down until 10 pages are reached. We further rank these 10 pages by P_k.

Previously, answers can come from a host that may not be too context-related. This host weighting approach is designed to remedy the situation. For example, a query concerning tax matters should prefer pages from IRS rather than pages from a tax preparer, assuming that IRS host has better content and ranks better among other hosts. The current Web Track evaluation is not designed to evaluate how appropriate (the surrounding pages of) a site is to support a key resource page as answer. We show in Table 5 the effect on P10 when host weighting is employed. The effect is small.

Short query 551 (Table 2 pool=50) Output Ranking:

Without Host Diversification	**With**	
Section 5.2.1	**5.3.1**	**5.3.2**
0 cybercrime.gov/ipmanual/indexa.htm	0	0
***1** cybercrime.gov/ipmanual2.htm	1	1
2 cybercrime.gov/ipmanual/05ipma.htm	2	
3 cybercrime.gov/ip.html		
4 cybercrime.gov/ipmanual/01ipma.htm		
5 cybercrime.gov/ipmanual/06ipma.htm		
6 cybercrime.gov/ipmanual/appa.htm		
7 cybercrime.gov/ipmanual/03ipma.htm		
8 cybercrime.gov/ipguidance.htm		
***9** www.customs.gov/impoexpo/protect.htm	3	2
commdocs.house.gov/committees/judiciary/ hju66710.000/hju66710_0f.htm	4	3
www.usdoj.gov/atr/public/guidelines/ ipguide.htm	5	4
www.customs.gov/nafta/ipr.htm	6	5
commdocs.house.gov/committees/judiciary/ hju66710.000/hju66710_0.htm	7	6
commdocs.house.gov/committees/judiciary/ hju63594.000/hju63594_0f.htm	8	
www.usinfo.state.gov/topical/econ/ipr/ipr-uspolpg.htm	9	7
www.ftc.gov/speeches/muris/intellectual.htm		8
www.ltg.ca.gov/programs/caip.asp		9

6. Conclusion

Key resource finding for topic distillation, by definition, appears to be more of a content-oriented process. Our experiments show that pure IR techniques can provide the bulk of distillation effectiveness (Precision at 10 values of 0.2347 short queries, and 0.2510 long). Adding out-linked content improves precision a little (0.2408 short, 0.2551 long). Short queries need to use a small candidate pool size of 50 (vs 400 for long queries). In-degrees of a page have not been found useful. Host diversification is proposed to give

Table 5: Results of Distillation using Host Weight (Pnn =Precision at 10)

	Best IR: Table 1	$P_k = C_k+L_k$ Table 2.	Host-Repeat Factor =3	Host-Repeat Factor=2	$P_k = C_k+L_k$ Table 2.	Host-Repeat Factor=3	Host-Repeat Factor=2
Query: **short**		pool size =300			pool size =50		
P10	**.2347**	.2327	.2184	.2102	.2408	.2224	.2245
Query: **long**		pool size =300			pool size =400		
P10	**.2510**	.2531	.2612	.2408	.2551	.2612	.2408

users a more balanced-view in their answer lists. For the top 10 pages after adjustment for better host diversification with at most 3 repeated host, long queries can give better precision at 10 (0.2612), but short query results suffer (0.22). It appears that for short queries, the quality of the original IR retrieval is not sufficiently accurate, and any further processing is not productive. Our best long and short query P10 values are competitive with results that have been reported for these experiments

It is possible that better ways of discriminating between content and irrelevant out-link pages may improve results based on out-links. Other ways of accounting in-degree weighting may be worth further investigation for their impact on key resource detection.

Acknowledgment

This work was partially supported by the Space and Naval Warfare Systems Center San Diego, under grant No. N66001-1-8912.

References

[1] Amento, B, Terveen, L & Hill, W, "Does "authority" mean quality? Predicting expert quality ratings of web documents". *Proc. SIGIR 2000*, (2000), pp.296-303.

[2] *American Memory*, http://memory.loc.gov/ammem/ftpfiles.html (2003).

[3] Bharat, K & Henzinger, M.R, "Improved algorithm for topic distillation in a hyperlinked environment", *Proc. of 21st Ann. Intl. ACM SIGIR Conf. on R&D in IR*, (1998), pp.104-111.

[4] Brin,S & Page, L, "The anatomy of a large scale hypertextual Web search engine", *Proc. 7th Intl. WWW Conf*, (1998).

[5] Broder, A, Kumar, R, Maghoul, F, Raghavan, P, Rajagopalan, S, Stata, R, Tomkins, A & Wiener, J, "Graph structure in the web", *Proc. 9th WWW Conf.* (2000), pp.309-320.

[6] Bush, V., *As we may think.* http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm (1945).

[7] Chakrabarti, S, Dom, B, Raghavan, P, Rajagopalan, S, Gibson, D. & Kleinberg, J, "Automatic resource compilation by analyzing hyperlink structure and associated text", *Proc. 7th WWW Conf*, (1998a), pp.65-74.

[8] Chakrabarti, S, Dom, B, S, Gibson, D, Kumar, R, Raghavan, P, Rajagopalan & Tomkins, A, "Experiments in topic distillation", *ACM SIGIR'98 Post Conf. Workshop on Hypertext IR for the Web*, (1998b).

[9] *Computer System Policy Project*, http://www.cspp.org/reports/networkedworld.pdf, (2003).

[10] Craswell, N & Hawking, D, "Overview of the TREC-2002 Web Track", in *The Eleventh Text REtrieval Conference, TREC 2002*. National Institute of Standards and Technology Special Publication 500-251. U.S.GPO (2003), pp.86-95.

[11] Craswell, N, Hawking, D & Robertson, S, "Effective site finding using link anchor information", *Proc. 24th ACM SIGIR 2001*, (2001), pp.250-257.

[12] Dill, S, Kumar, R, McCurley, K, Rajagopalan, Sivakumar, D & Tomkins, "A, Self-similarity in the web. ACM Trans". *Internet Technology, 2*, (2002), pp.205-223.

[13] Huberman, B.A, Pirolli, P.L.T, Pitkow, J.E & Lukose, R.M, "Strong Regularities in Word Wide Web Surfing", *Science, 280*, (1998), pp.95-97.

[14] Jeong, A & Barabasi, A.L, "Diameter of the World Wide Web", *Nature, 401*, (1999), pp.130-131.

[15] Kleinberg, J, "Authoritative sourcs in a hyperlinked environment", *Proc. of 9th ACM-SIAM Symposium on Discrete Algorithms*, (1998).

[16] Kumar R, Raghavan, P, Rajagopalan, S & Tomkins, A, "Trawling the web for emerging cyber-communities", *Proc. 9th WWW Conf*, (1999).

[17] Kwok, K. L., "A network approach to probabilistic information retrieval", *ACM Transactions on Office Information System*, (1995), 13:324-353.

[18] *Search Engine Watch*, http://searchenginewatch.com, (2003).

[19] *TREC-2002 Web Track Guidelines*, Available at http://trec.nist.gov, (2002).

[20] Voorhees, E, "Overview of TREC 2002", in *The Eleventh Text REtrieval Conference, TREC 2002*. National Institute of Standards and Technology Special Publication 500-251. U.S.GPO (2003), pp.1-16.

[21] *Web Track Tasks in Special Publications of NIST 2002 to 2000*. Available at http://trec.nist.gov

[22] Xu, H, Yang, Z, Wang, B, Liu, B, Cheng, J, Liu, Y, Yang, Z, Cheng, X & Bai, S, "TREC-11 experiments at CAS-ICT: filtering and web", in *The Eleventh Text REtrieval Conference, TREC 2002*. National Institute of Standards and Technology Special Publication 500-251. U.S.GPO (2003) pp.141-151.

[23] Zhang, M, Song R, Lin, C & Ma, L, "THU at TREC2002 web track experiments", in *The Eleventh Text REtrieval Conference, TREC 2002*. National Institute of Standards and Technology Special Publication 500-251. U.S.GPO (2003), pp.591-594.

Generating Rule-Based Trees from Decision Trees for Concept-based Information Retrieval

Ronnie Fanguy
Nicholls State University
is-raf@nicholls.edu

Vijay Raghavan
Center for Advanced Computer Studies
University of Louisiana at Lafayette

Abstract

Web-based information retrieval systems may result in poor levels of precision and recall when users are required to articulate their own queries. Concept-based information retrieval attempts to solve this problem by allowing users to select from concept definitions specified by experts. However, it is unrealistic to expect experts to define every concept which will be of interest to users. Therefore, we propose a system for generating concept definitions from decision trees.

1. Introduction

As the World Wide Web grows in importance as a medium for the exchange of information and ideas among various groups of individuals, the need for efficient and effective web-based information retrieval systems is heightened. One major challenge in this arena is retrieving relevant documents with high precision and high recall—with the ultimate goal being retrieval of all relevant documents (perfect recall) and only relevant documents (perfect precision). However, it is often difficult for users to articulate exactly the right query to achieve high levels of precision and recall. Therefore, retrieval requests often result in the user obtaining many irrelevant documents while missing documents that are relevant.

One attempt to solve this problem is found in the adaptation of concept-based information retrieval to web-based information retrieval systems. Most web-based search engines are based on the Boolean Retrieval Model, which requires a user to specify a query as some boolean combination of search terms [1, 2]. To ease the burden of query formulation and to deal with the disparity between a user's vocabulary and the vocabulary used in a document collection, researchers have proved the effectiveness of replacing the boolean search engine's interface with that of a concept-based information retrieval system [3, 4].

In concept-based information retrieval, a user issues a document retrieval request by selecting a concept, as opposed to specifying a boolean expression. The concept-based retrieval system uses the definition of the selected concept to generate one or more appropriate boolean search requests which are automatically submitted to the underlying boolean search engine. The documents retrieved from the underlying system are then merged and presented to the user. Such a system greatly relieves the user from the burden of specifying appropriate queries—as they simply select the concept of interest.

A key difficulty with concept-based retrieval is obtaining an adequate collection of concept definitions. What a user is allowed to retrieve is limited by the concepts that are defined. The number of concepts that are required to make the system usable may be quite large.

Another key difficulty exists when there is disagreement on the definition of a concept among users. If the system only allows a concept to be defined in one way, then the system will not be able to model these differences. Unless we move from a situation where concept definitions are presumed relevant for all users toward personalized concept definitions, precision and recall of our retrieval system will suffer. The interested reader is referred to [9] for an excellent discussion on the need for personalized information retrieval.

In this paper, we address these problems by presenting a system which generates personalized concept definitions from decision trees. We further explain concept-based information retrieval and rule-based concept trees in section 1.1. Then, in section 1.2, we discuss the structure and functioning of a decision tree classifier. In section 2, we present the overall framework of our system for generating rule-based concept trees and describe its components. Section 3 then presents our algorithm for converting a decision tree into a set of rule-based concept trees. Finally, sections 4 and 5 respectively evaluate the system and conclude the paper discussing future work.

1.1. Concept-Based Information Retrieval

The original concept-based information retrieval system was introduced in [5]—with their system called Rule Based Retrieval of Information by Computer

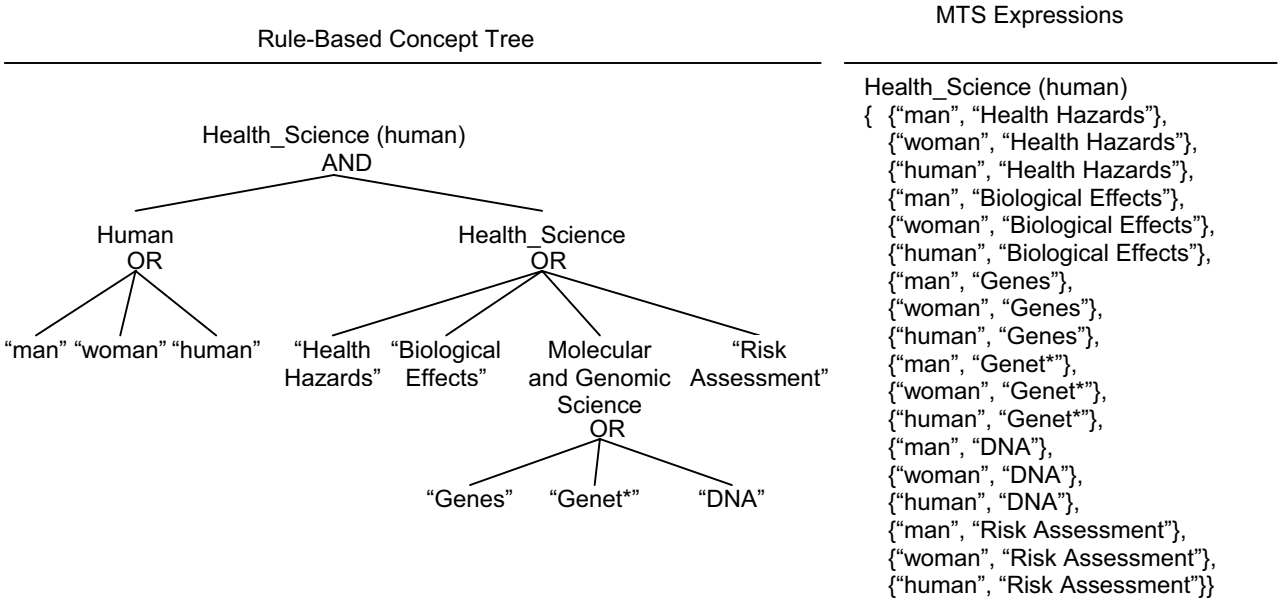


Figure 1. Definition of concept health_science (human).

(RUBRIC). With RUBRIC, a concept is represented by a tree structure. The tree structure used in this system, called a rule-based tree, represents a concept as a boolean (AND/OR) combin1ation of subconcepts and index terms. In this structure, subconcepts are included as a tree's intermediate nodes and are associated with either the boolean operator AND or the boolean operator OR. The terminal (leaf) nodes of a tree are each associated with an index term. Consider the rule-based concept definition of human health science shown in Figure 1. (This example is borrowed from [3]). Notice that human health science is defined as a conjunction of the subconcepts human and health science. Furthermore, these two subconcepts are further defined. The definitions of subconcepts continue until leaf nodes are reached. At this point, the definitions are based on the index terms from the document collection.

In RUBRIC, rule-based trees are used to retrieve documents by applying bottom-up processing for each document in the document collection. Initially, each of a tree's leaf nodes are assigned a value to indicate whether or not they are satisfied based on the contents of a document being considered for retrieval. A leaf node is satisfied if the document contains the leaf node's associated term. Otherwise, the leaf node is not satisfied. From the leaf nodes, values are propagated up the tree towards the root node to determine whether or not the concept is satisfied. Based on the value propagated, the system decides whether or not the document is retrieved (whether or not the document matches the root concept).

When propagating a value to an OR intermediate node, if any of its children are satisfied, then the OR node is satisfied. However, for an AND intermediate node, all of the children must be satisfied in order for the node to be satisfied. When the root node is satisfied, the concept associated with the tree is considered to be satisfied. Thus, documents for which the root node is satisfied are retrieved for the user.

It can be very expensive to retrieve documents in this way. Alsaffar et.al. [6] recognize this shortcoming and propose preprocessing the rule base to speed up retrieval requests. In their solution, an AND/OR tree is converted into a group of Minimal Term Set (MTS) expressions. A MTS expression is simply a minimal set of terms that may be used to represent the root concept. That is, if all of its terms are found within a document, then the document satisfies the root concept and should be retrieved. Any one of the MTS expressions in the group may be satisfied in order for the root concept to be satisfied. Only when none of the MTS expressions are satisfied do we say that the document does not match the root concept.

Figure 1 shows the group of MTS expressions for the human health science concept. Notice that {man, genes} is one MTS expression. Therefore, if the terms "man" and "genes" both appear in a document, then we know that the root concept human health science is satisfied. Reexamine the rule-based tree defining the human health science concept, and notice that the term "man" satisifies the subconcept human and the term "genes" satisfies the subconcept health science. Together these two subconcepts satisfy the root concept human health science. Satisfaction of any one of the MTS expressions listed will similarly result in satisfaction of the root concept human health science.

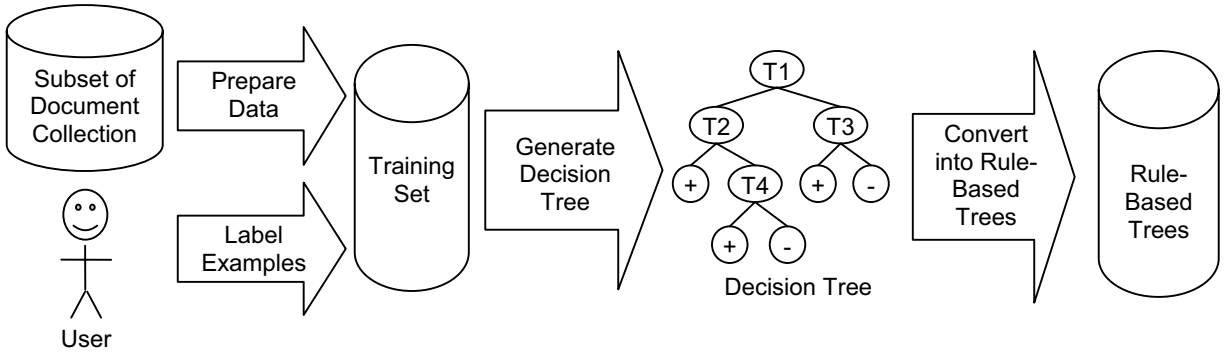


Figure 2. System for generating rule-based trees.

The usability of a concept-based information retrieval system is limited by its ability to collect concept definitions. Such systems will only be successful if the concepts that users are interested in retrieving are represented in the system's rule base. Therefore, a large number of rule-based trees is typically required of the system. Unfortunately, each of these rule-based concept definitions must be composed by experts who are familiar with the document collection and the index terms used therein and also with the mechanics of constructing rule-based trees. Expecting experts to define rule-based trees for every concept in which users may be interested is obviously unreasonable as the cost in terms of time and money is extremely prohibitive.

Kim, et. al. [7] recognize this problem and propose a solution for automatically generating concept definitions in the form of rule-based trees. However, their approach requires the existence of a machine-readable thesaurus. This may be appropriate for some domains; however, it will not be applicable in many others. Unless these limitations are overcome in a more general sense, concept-based information retrieval systems will by and large be limited to the academic and theoretical realm.

In this paper, we propose a solution to overcome these limitations by automatically constructing concept definitions from decision trees. Thus, when using our approach, the requirement may be translated into the ability to build a decision tree. Algorithms for the efficient induction of decision trees have been studied and are well understood. Such algorithms require as input a set of pre-classified training data. If we are able to gather a representative set of documents and elicit concept labels for each document in the set, then we can apply these algorithms. The result of which may be processed by the algorithms presented in this paper to generate a concept definition for the concept(s) of interest. Subsequently, the concept definition may be used by the system to retrieve documents which are relevant to the concept.

1.2. Decision Tree Classifier

A decision tree is a classification mechanism whereby an object is subjected to a series of tests. Each possible sequence of test outcomes yields an appropriate class label to associate with the object being classified. A decision tree is structured as a tree whose branch (non-terminal) nodes are tests and whose leaf (terminal) nodes represent class labels. Each branch node in a decision tree has a number of child nodes—one for each possible test outcome. For example, if a branch node is associated with the test Height > 5, then the possible test outcomes are {true, false}. This branch node will, therefore, have two children—one connected via an arc associated with the true outcome and one associated with the false outcome.

When classifying an object with a decision tree, we begin at the root node of the decision tree and apply its associated test. Depending on the result of the test, the appropriate arc corresponding to this result is traversed to the appropriate child node in the tree. If the subsequent node is a branch node, then its associated test is applied and the appropriate arc is again traversed. This process continues until a leaf node is reached. At this time, the concept label associated with the leaf node is returned as the appropriate concept for the object being classified.

We are interested in examining the features that cause an object to be assigned to a particular concept or category. The decision tree certainly contains information appropriate for distinguishing between sets of concepts. If we are able to extract this information from the decision tree, then we will be a step closer to creating concept definitions.

2. System Framework

Our system for generating rule-based trees for conceptual retrieval is shown in Figure 2. In this system, the set of rule-based trees is derived from decision trees.

Decision tree generators require a training set of data as input. Therefore, before we can generate decision trees, we must prepare a training set. In the current context, a training set is a set of labeled attribute vectors describing a subset of documents from the document collection. The attribute vectors describing each document include the index terms associated with the document collection and perhaps other document data that are available. The labels associated with each attribute vector are obtained from the user. The user will select an appropriate label to indicate which concept applies from a set of concept labels. If there is only a single concept of interest—the label will indicate whether or not each document is relevant with respect to that concept.

Once the training set is prepared, it will be used as input to the system component responsible for generating a decision tree. See5, the Windows version of the popular decision tree generator c4.5 [8], is the program we use to perform this step. By identifying the distinguishing features of documents in the training set, See5 constructs a decision tree which may be used to assign an appropriate concept label to a document.

3. Transforming Decision Trees into Concept Trees

We propose an algorithm which converts a decision tree into a number of concept definitions—one for each concept label used within the decision tree. The structure we use to store these concept definitions is RUBRIC's rule-base tree.

The rule-base tree structure may be used to represent the concepts stored within a decision tree. The path from the root node to a leaf node in a decision tree represents a series of tests that must all be passed in order for the concept associated with the leaf to be satisfied. That is, the conjunction of each of these tests (along with their appropriate results) gives us one way to satisfy the concept—the very essence of the AND operator. However, there may be many such paths through the tree for a single concept. Any one of these paths will yield satisfaction of the concept—the very essence of the OR operator. Hence, with an appropriate conversion mechanism, the distinguishing concept characteristics stored within a decision tree may be restructured using the AND and OR nodes of a rule-based tree.

However, the restructured version of the decision tree must support the same types of tests that are used within the decision tree. RUBRIC's rule-based concept trees have leaf nodes that are associated with terms. The condition associated with the leaf nodes may be interpreted as "the associated term is present in the document." However, this type of test is not flexible enough to support decision tree tests. Therefore, some modification must be made to either the rule-based concept tree structure or the information stored for each document.

By expanding the concept tree structure to allow more flexible types of conditions, we can represent the tests that are used within a decision tree. Instead of storing just a term, each leaf node may be extended to support conditions of the form **attribute op value**, where

- **attribute** is some property of set of objects being considered,
- **op** is an equality (==, !=) or relational (<, <=, >, >=) operator,
- **value** is some literal.

This extension will enable support for the types of tests commonly used within decision trees.

Alternatively, the result of the test conditions could be

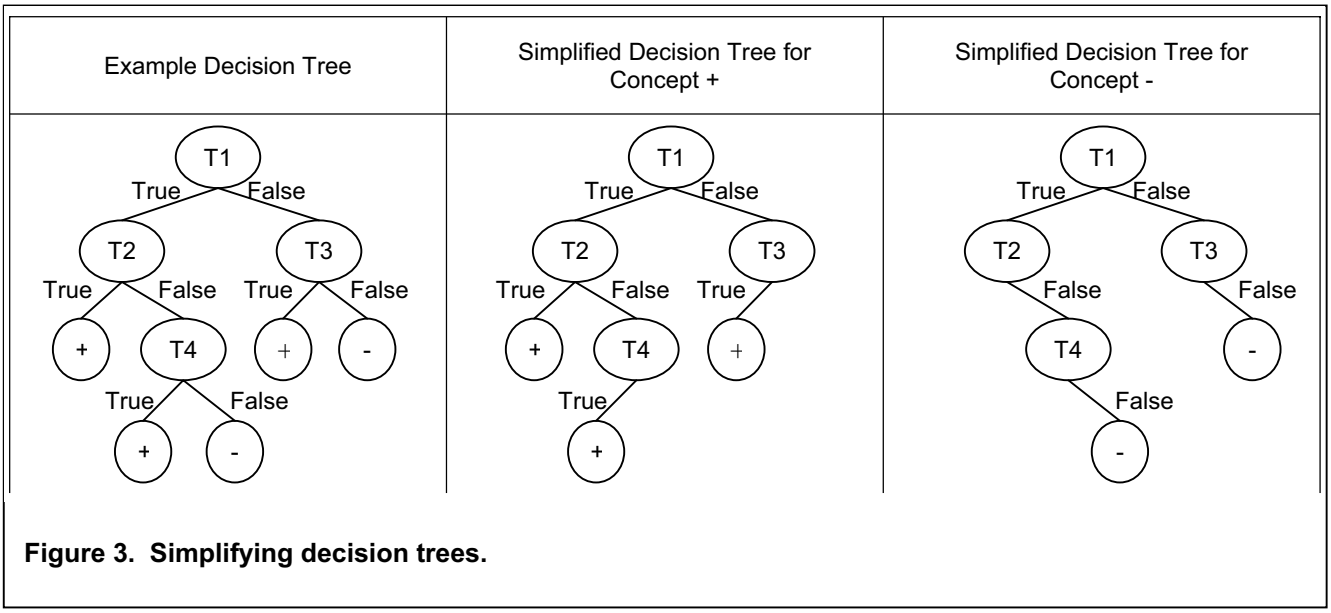


Figure 3. Simplifying decision trees.

stored as part of the document indexing structure. Thereby, the concept tree's leaf node may store the test condition as if it were a term. With this change, determining whether "the associated term is present in the document" will appropriately correspond to evaluating the decision tree test result. In this way, decision tree tests may be modeled while requiring no change to the existing rule-based tree structure.

We split the task of transforming decision trees into concept trees into two main parts—simplifying decision trees and inverting the simplified decision trees. Given a particular concept that is represented within a decision tree, we can perform these two steps to construct a rule-based tree that represents the concept.

3.1. Simplifying Decision Trees

The first step necessary to construct an AND/OR concept tree from a decision tree is to simplify the decision tree. In simplifying the decision tree, we consider only the paths that lead to a given concept label. The decision tree is simplified by removing any components (subtrees) that are not associated with the concept being defined—so that only those paths through the tree that lead to the concept being defined remain. An example is shown in Figure 3. Here, we see how a decision tree, which classifies objects as either + or –, may be simplified to decision trees containing only relevant information for each of these two concepts. In this example, each non-leaf test node results in either true or false. Notice that in the simplified tree for concept + only the components of the tree that lead to + leaf nodes remain in the tree. Similarly, only the components of the tree that lead to – leaf nodes remain in the simplified tree for the concept –. By performing this step, we are able to focus solely on the portion of the tree that is relevant to the concept being defined.

3.2. Inverting Simplified Decision Trees

After the decision tree is simplified so as to focus on only those paths that lead to the concept being defined, we proceed to invert the simplified decision tree. This process is demonstrated by the recursive function shown in Figure 4. This function takes two parameters as input:

- D—the current node within the simplified decision tree and
- ψ—the AND/OR concept tree being constructed.

Initially, D should refer to the root of the decision tree, and ψ should be an empty AND/OR concept tree. The function proceeds to process a single node of the decision tree at a time. It begins at the root node and relies on recursive calls to process the entire tree. As the function traverses the simplified decision tree, it constructs an appropriate AND/OR concept tree—which it returns as output.

As this algorithm processes each node in the simplified decision tree, it first considers the number of children at the current decision tree node (D). If there is more than one child, then there is more than one path through this node which will lead to the concept being defined. In this case, any one of the paths may be taken. Therefore, an OR node is needed in the concept tree.

Next, we proceed to iterate through each of the child nodes of the current node in the decision tree. In processing the children, we consider whether or not the child is a leaf node. If it is a leaf node, then we simply add a node to the concept tree which represents the test result that leads from the current node in the decision tree to the child leaf node. However, if the child is not a leaf node, then there are other nodes which must be processed along the current path through the decision tree. In this case, we must add an AND node because we must satisfy this test along with the other subsequent test nodes in the current path. Along with this AND node, we add a child that is responsible for handling the test that corresponds to the arc which leads from the current node to the non-leaf child within the decision tree. Other children of the AND node are added as the non-leaf child nodes of D nodes are recursively processed.

Notice also that the algorithm gives considerable attention to maintaining the current parent within the concept tree being constructed. This attention is required to ensure that when the processing of each child node is

```
ψ-AND/OR concept tree being constructed
η,η'-Current parent node in Concept Tree
D-Current node in the decision tree
C-Child node in decision tree

function ψ=ConstructConcept(D, ψ)
   η= ψ.CurrentNode
   If (D.NumChildren > 1)
      η= ψ.AddORNode()
   For each C ∈ D.Children
      If η != ψ.CurrentNode
         ψ.MakeCurrent(η)
      If (!IsLeafNode(C)) AND
         (!IsAndNode(ψ.CurrentNode))
         η'= ψ.AddANDNode()
         ψ.MakeCurrent(η')
      ψ.AddLEAFNode(D.Test, C.ParentTestResult)
      If (!IsLeafNode(C))
         ψ = ConstructConcept(C, ψ)
return
```

Figure 4. Algorithm to invert a simplified decision tree.

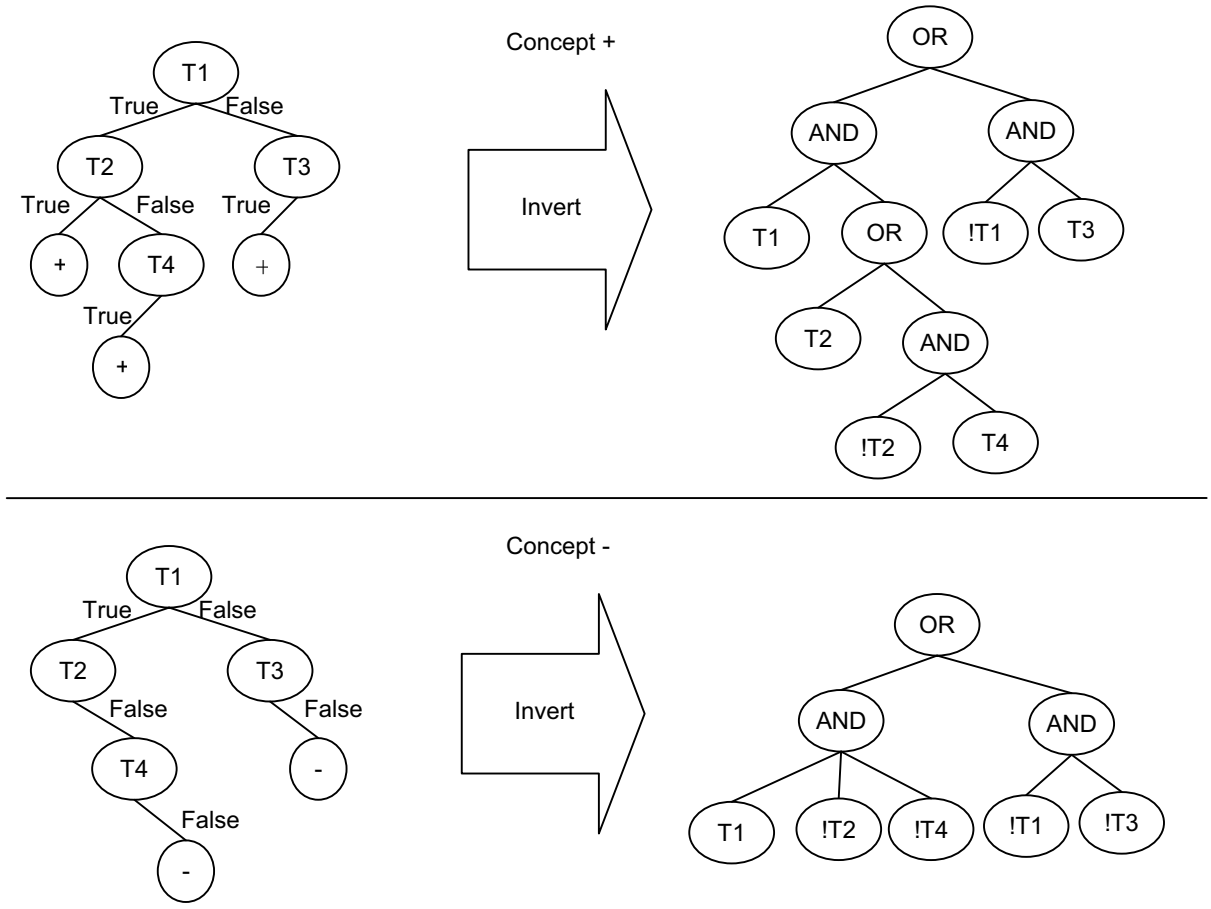


Figure 5. Inverting the simplified decision trees for concepts + and -

complete (after arbitrarily many recursive calls) we return to the correct positioning within the concept tree being constructed to continue appending other child nodes—if necessary. Figure 5 shows the result of applying this algorithm to the simplified decision tree for the concepts + and -.

4. System Evaluation

There are three major advantages of this approach over existing approaches to constructing concept definitions in the form of rule-based trees. The first and most obvious advantage is that a concept may be constructed in the absence of experts and thesauri. Any user who is willing to supply class labels is able to construct concept definitions.

Second, using the characteristics identified by the decision tree generator as key to distinguishing one concept from others has particularly valuable benefits to an information retrieval system. Rather than simply getting a definition which serves to characterize a concept, we obtain a rule-based tree which is focused on the characteristics that distinguish a concept from others within the document collection. Therefore, the resulting concept definition is composed of the features that allow a user to precisely retrieve the documents that are relevant to the concept.

When analyzing this advantage, the reader may be prone to question whether there is any advantage of the rule-based tree representation over the decision tree representation. Recall that a decision tree is a classifier which distinguishes between a number of different classes. A rule-based tree is different in that it defines a single concept. However, the rule-based tree representation is not just a classifier. It is a hierarchy which defines the root concept in terms of subconcepts. This is an important distinction because we are not only defining the concept associated with the root node of the tree, but we are also identifying some structure within that concept by the hierarchy of subconcepts. We do not make any claims regarding the quality of this structure. Presently, we are able to generate some structure with our

algorithm whose complexity is a function of the number of nodes in the decision tree—O(n). We are currently examining other more computationally expensive alternatives that generate trees which have greater structural quality.

Third, the concept definitions may be personalized to a much higher degree than existing approaches allow. Rather than offering only an expert's definition of a concept or relying on the word relationships stored in a thesaurus, our approach focuses on defining concepts that are tailored to a particular user's definition of the concept—based on their personal labeling of the training data. When the definition is not agreed upon, this characteristic is a major benefit.

However, these benefits do not come without a cost. The benefits are gained at the cost of greater user involvement in the construction of concept definitions. Instead of simply being given a concept definition, the user bears the burden of assigning concept labels to documents. However, this cost may be reduced in several ways. A user's personal concept definition may evolve during several iterations. Initially, a rough concept definition may be formed by evaluating a small subset of documents. Documents that are subsequently retrieved may also be evaluated to make the definition more precise. Over several iterations, the concept definition will more precisely meet the user's needs. Another mechanism which may be employed to ease the user's burden is to allow groups of users who share similar concept definitions to share labeled examples. In this way, the burden of labeling examples is shared among the group.

5. Conclusion/Future Work

In this paper, we have presented an approach to generating concept definitions from decision trees. Generating concept definitions without the help of an expert greatly improves the usability of a concept-based information retrieval system. Such systems consequently increase the likelihood of boolean search engine effectiveness, as the system does not rely on user-specified queries. Also, the user is likely to be pleased to be relieved of the burden of query articulation. Our system replaces this burden with the requirement of assigning concept labels to documents. We believe that the user will find this burden far lighter than the frustration which may develop from trial and error query development.

Future work on this research project involves carrying out experiments which quantitatively evaluate the quality of our system relative to the existing approaches, its implications for personalization, and its usefulness in aiding human understandability of concepts. Furthermore, rule-based trees may also be extended to include weighted arcs. We plan to extend our conversion algorithm to calculate these weights. Additionally, we are examining alternative means of automatically generating rule-based concept trees.

Additionally, we are exploring the potential of this approach to expand the usefulness of browsing directories—such as those found in Yahoo, Google, Altavista, etc. These directories enable users to find relevant web pages by allowing them to start at some high level category and traverse down to more and more specific categories. These directories can be thought of as trees which are defined by OR nodes, where each child is a specific subcategory of the general parent node—similar to an is-a relationship (generalization). By applying rule-based trees, we can model not only OR nodes but also AND nodes. This allows us to model aggregation (the part-of relationship) as well as generalization. Determining the usefulness of this extension will require additional research.

6. References

[1] G. Salton and M. McGill, *An Introduction to Modern Information Retrieval*, New York, NY: McGraw-Hill, 1983.

[2] K. Jones and P. Willett, "Introduction to Chapter Five," in *Readings in Information Retrieval*, San Francisco, CA, Morgan Kaufmann, pp.257-263, 1997.

[3] F. Lu, T. Johnsten, V. Raghavan, D. Traylor, "Enhancing Internet Search Engines to Achieve Concept-based Retrieval," *InForum 1999—Improving the Visibility of R & D Information*, Oak Ridge, TN, May 1999.

[4] A. Alsaffar, J. Deogun, V. Raghavan and H. Sever, "Concept Based Retrieval By Minimal Term Sets," *International Symposium on Methodologies for Intelligent Systems*, Warsaw Poland, June , 1999.

[5] B. McCune, R. Tong, J. Dean and D. Shapiro, "RUBRIC: A System for Rule-Based Information Retrieval," *IEEE Transactions on Software Engineering*, vol. 11:2, pp. 939-944, 1985.

[6] A. Alsaffar, J. Deogun, V. Raghavan, H. Sever, "Enhancing Concept-Based Retrieval Based on Minimal Term Sets." *Journal of Intelligent Information Systems*, vol. 14, pp. 155-173, 2000.

[7] M. Kim, V. Raghavan, "Adaptive Concept-based Retrieval Using a Neural Network." *Proceedings of ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval*, Athens, Greece, July 2000.

[8] J. Quinlan, *C4.5 Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[9] J. Pitkow, et al, "Personalized Search." *Communications of the ACM*, vol. 45, no. 9, pp. 50-55, 2002.

Mining for User Navigation Patterns Based on Page Contents

Yue Xu
School of Software Engineering and Data Communications
Queensland University of Technology
Brisbane, QLD 4001, Australia
E-mail: yue.xu@qut.edu.au

Abstract

This paper describes two novel methods that capture the most popular or preferred navigation sessions visited by web users. Most existing techniques used by web log mining are access-based approaches that statistically analyze the log data that reflects the way past users have interacted with the site and do not pay much attention on the content of the pages. The methods proposed in this paper take page contents into account for mining user navigation patterns. The page content is characterized as a topic vector. The first method determines the popular sessions by examining the deviation probability of a session as well as comparing the distance between topic vectors. The second method determines the popular sessions by using entropy of a session as an estimator. Experiments have been conducted and the results show that, by considering page contents, the sessions which contain irrelevant pages can be ruled out by the two methods.

1 Introduction

Navigating through a large web site for finding relevant information can be tedious and frustrating. Recently, there is a growing interest in developing adaptive web site agents that assist users in navigating web sites to find the information of their interests by recommending relevant web documents to the users [4, 6]. For achieving this goal, an essential problem should be solved that is to find the web documents which are relevant to the user interests. Recent years, an increasing number of researchers have focused their study on applying data mining techniques to web log data analysis for finding regularities in web users' navigation patterns. The user navigation patterns are then used to determine the documents that are relevant to the user's interests. When users interact with a web site, the user's navigation tracks are stored in web server logs and the log data is a good collection of data for being analysed to capture the user navigation behaviour patterns. Some approaches have been proposed that analyze previous users' web logs to discover user navigation patterns such as popular navigation sessions, page clusters, and popular paths between web pages [1, 5, 6, 8, 9, 11]. These patterns are then used to classify a new user into a category and the pages related to that category will be recommended to the user.

The data available for web-based systems has two forms: the content of the web site itself and the access patterns of users to the site. Most techniques used by web log mining are access-based approaches that statistically analyze the log data that reflects the way past users have interacted with the site and do not pay much attention on the content of the pages. One such approach is the statistical model proposed by Borges and Levene [1]. In this model, the user navigation information, obtained from web logs, is modelled as a hypertext probabilistic grammar(HPG). The set of highest probability sessions generated by the grammar corresponds to the user preferred navigation trails (also called association rules). However, the association rules are generated by only analysing web log data. Nothing of the page contents has been considered. In this paper, we propose a technique that takes page contents into account for finding user popular navigation sessions. The content of a page can be expressed by a conceptual description languange for describing the topics involved in this page. The conceptual language may be simple conjunctions of attributes or complex and cognitively inspired descriptions as in [7]. In this paper, associated with a web page there is a topic vector which characterizes the conceptual aspect of that page. Firstly, in section 2 we give a brief review to the HPG model proposed by Borges and Levene. Then, in section 3, we present two novel methods to find the most popular navigation sessions based on user access data and page content as well. Finally, section 4 summarizes the paper.

2 A Review of the HPG Model

A log file is an ordered set of web page requests made by users. The requests are stored in the order that the server receives them. If multiple users are browsing the site concurrently, their requests are interminingled in the log file. The page requests made by one user can be extracted from the log file. Since it is expected that a user may visite a web site more than once, a user navigation session is usually defined as a sequence of pages visited by the same user such that no two consecutive pages are separated by more than a certain amount of time, for example, 30 minutes as many authors have adopted [1]. Techniques to infer user navigation sessions from log data are described in [2]. A collection of user navigation sessions can be described by a hypertext probabilistic language [3] generated by a hypertext probabilistic grammar (HPG) [1] which is a proper subclass of probabilistic regular grammars [10]. A HPG is a probabilistic regular grammar which has a one-to-one mapping between the sets of nonterminal and terminal symbols. Each nonterminall symbol corresponds to a web page and a production rule corresponds to a link between pages. Moreover, there are two additional artificial states, S and F, which represent the start and finish states of the navigation sessions. The probability of a grammar string is given by the product of the probabilities of the productions used in its derivation. The productions with S on its left-land side are called *start productions* and the productions corresponding to links between pages are called *transitive productions*.

From the set of user sessions we obtain the number of times a page was requested, the number of times it was the first state in a session, and the number of times it was the last state in a session. The number of times a sequence of two pages appears in the sessions gives the number of times the corresponding link was traversed. The probability of a production from a state that corresponds to a web page is proportional to the number of times the corresponding link was chosen relative to the number of times the user visited that page. A parameter α is used to attach the desired weight of a state corresponding to the first page browsed in a user navigation session. If α =0, only states which were the first in an actual session have probability greater than zero of being in a start production. In this case, only those strings which start with a state that was the first in a session are induced by the grammar. If α=1, the probability of a start production is proportional to the overall number of times the corresponding state was visited. In this case, the destination node of a production with higher probability corresponds to a state that was visited more often.

The HPG can be formally defined as follows.

Definition 2.1 (HPG) *: An HPG is a five-tuple* $< H, M, S, F, \Gamma >$ *which is denoted as* $\Phi(H, M, S, F, \Gamma)$*, where:*

- H *is a finite set of nodes,* $H = \{h_1, \ldots, h_m\}$ *which represents the set of states involved in the HPG. Each state represents a page request.*
- $M = \{p(h_i, h_j)\}$*,* $h_i, h_j \in H$*, is a* $m \times m$ *matrix, that is,* $\forall h_i, h_j \in H$*,* $1 \geq p(h_i, h_j) \geq 0$*,* $\Sigma_{k=1}^{m} p(h_i, h_k) = 1$*.*
  The matrix M *is called the* **transition matrix** *of* Φ *and the probabilities* $p(h_i, h_j)$ *are called the* **transition probabilities** *of* Φ *which are calculated by the mapping function* Γ*.* $\forall h_i \in H$*,* $p(S, h_i)$ *is the probability that a user will start his/her navigation by visiting the page associated with* h_i*.* $\forall h_i, h_j \in H$*,* $p(h_i, h_j)$ *is the probability that a user who is browsing the page associated with* h_i *will next browse the page associated with adjacent state* h_j*.*
- S *is the start state of the HPG.*
- F *is the finish state of the HPG.*
- Γ *is a function from* $H \times H$ *to* $[0, 1]$*, which is used to calculate* Γ*, i.e.,* $\forall h_i, h_j \in H$*,* $p(h_i, h_j) = \Gamma(h_i, h_j)$*.*

In the model proposed by Borges [1], the probabilities of the transition matrix M are determined from statistical information collected from the logs. For simplicity, we use h_i to denote the page which is associated with h_i. For a single user, let R be the total number of page requests, R_i be the number of requests to page h_i, N_s be the total number of sessions, and N_{si} be the number of sessions with h_i being the first page. The probabilities $p(S, h_i)$ are determined by the following equation, where S is the start state, h_i is a state in $H - \{S\}$, $1 \geq \alpha \geq 0$:

$$p(S, h_i) = \alpha \frac{R_i}{R} + (1 - \alpha) \frac{N_{si}}{N_s} \tag{2.1}$$

Similarly, let R_j^i be the number of page requests to page h_j immediately after visiting page h_i, R^i be the total number of page requests immediately after visiting page h_i, the probabilities $p(h_i, h_j)$ are determined

Table 2.1: An example set of user's sessions

Session Number	User Sessions
1	$A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_4$
2	$A_1 \rightarrow A_5 \rightarrow A_3 \rightarrow A_4 \rightarrow A_1$
3	$A_5 \rightarrow A_2 \rightarrow A_4 \rightarrow A_6$
4	$A_5 \rightarrow A_2 \rightarrow A_3$
5	$A_5 \rightarrow A_2 \rightarrow A_3 \rightarrow A_6$
6	$A_4 \rightarrow A_1 \rightarrow A_5 \rightarrow A_3$

Table 2.2: The association rules obtained from the HPG in Figure 2.1 ($\lambda = 0.3$)

Session Number	User Sessions	Derivation Probabilities
1	$A_4 \rightarrow A_1$	0.5
2	$A_5 \rightarrow A_2 \rightarrow A_3$	0.45
3	$A_5 \rightarrow A_3$	0.4
4	$A_3 \rightarrow A_4$	0.4
5	$A_2 \rightarrow A_3 \rightarrow A_4$	0.3
6	$A_1 \rightarrow A_5 \rightarrow A_2$	0.3

by the following equation, where h_i, h_j are two states in $H - \{S\}$:

$$p(h_i, h_j) = \frac{R_j^i}{R^i} \tag{2.2}$$

The strings generated by the grammar correspond to the user navigation sessions. For each string $< h_{i1}, \ldots, h_{ir} >$ generated by the grammar, its derivation probability is defined as follows:

$$p(< h_{i1}, \ldots, h_{ir} >) = \prod_{k=1}^{k=r-1} p(h_{ik}, h_{i(k+1)}) \tag{2.3}$$

The aim of the HPG is to identify the subset of these sessions, which correspond to users' preferred sessions also called the association rules. A session is included in the grammar's language if its derivation probability is above a *cut-point* λ. The *cut-point* is responsible for pruning out strings whose derivation contains transitive productions with small probabilities.

Table 2.1 gives an example of a collection of user sessions. The collection of navigation sessions in the example contains a total of 24 page requests. The association rules generated by the HPG in the example are given in Table 2.2, where $\lambda = 0.3$.

3 Content-based HPG

As we have mentioned that HPG relies only on user access log data to discover the association rules. However, the data in web server logs may not really reflect user's navigation intention. One reason is that some data may be missing due to caching by the browser. This arises most commonly when the visitor users the browsers back button. For example, if the user returns back to a previous page p_i from the current page p_j and then goes to page p_k, this will appear in the log as $p_i \rightarrow p_j \rightarrow p_k$ but not $p_i \rightarrow p_j \rightarrow p_i \rightarrow p_k$. Another reason is that the user may visit some pages which are not relevant to his/her navigation interests and also recorded in the logs. These irrelevant pages in the user's navigation sessions can make great impact on the quality of the association rules. We argue that the page contents can be used to eliminate or alleviate the impact made by the missing data or the irrelevant pages.

3.1 A Modified Transition Matrix

From Equation 2.1 and Equation 2.2, we can see that the probabilities $p(h_i, h_j)$ in M are obtained from statistical information captured in logs only, none of the page content information has been used. In this subsection, we present a method to incorporate page content information into the probability calculation.

In this paper it is assumed that there are n topics involved in a web site, and that associated with each page in the site there is a n-dimensional vector which characterizes the relevancy of each topic to the page. The ith element in the vector represents the relevancy assigned to the ith topic. That is, $\forall h_i \in H$, there is a n-dimensional vector denoted as T_i, $T_i =< t_{i1}, \ldots, t_{in} >$, where t_{ij} represents the relevancy of the jth topic to the page h_i.

We observed that with a navigation goal in his/her mind, a user will visit the pages which are content relevant. This observation suggests that for two pages h_i and h_j whose topic vectors are $T_i =< t_{i1}, \ldots, t_{in} >$ and $T_j =< t_{j1}, \ldots, t_{jn} >$ respectively, if the distance between each pair of the corresponding vector elements t_{ik} and $t_{jk} (k = 1, \ldots, n)$, denoted as $D(t_{ik}, t_{jk})$, is big, the two pages can be thought of as irrelevant. The probability $p(h_i, h_j)$ should be increased if $D(t_{ik}, t_{jk})$ is small because h_j would be a good candidate to visit next after visiting h_i if the user wants to find some information which relevant to the information on page h_i. Under this consideration, we modify Equation 2.2 by taking the distance of page topic vectors into account to calculate the transition probabilities. Equation 3.1 below calculates the transition probability from two aspects. $p(h_i, h_j)$ is determined by Equation 2.2 which is the contribution from user access statistics. $\frac{\beta}{e^{2D(T_i, T_j)}}$ calculates the contribution from the page topics. In Equation 3.1, p_c denotes the modified probability and p is the probability calculated by Equation 2.2, $D(T_i, T_j)$ is the arithmetic average of the difference between T_i

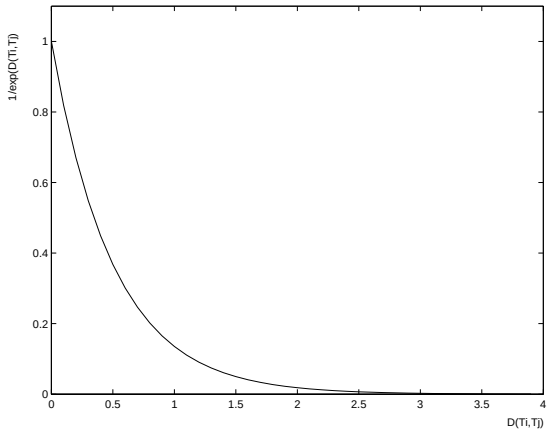


Figure 3.1: Function $\frac{1}{e^{2D(T_i,T_j)}}$

Table 3.1: Topic vectors

Page	Topic Vector
A_1	$T_1 =< 1, 1, 0, 0, 1 >$
A_2	$T_2 =< 1, 1, 1, 0, 0 >$
A_3	$T_3 =< 1, 0, 1, 1, 0 >$
A_4	$T_4 =< 0, 1, 1, 0, 1 >$
A_5	$T_5 =< 1, 1, 0, 1, 0 >$

Table 3.2: The association rules obtained after considering topics ($\beta = 0.3, \lambda = 0.3$)

Session Number	User Sessions	Derivation Probabilities
1	$A_5 \rightarrow A_2 \rightarrow A_3$	0.67
2	$A_4 \rightarrow A_1$	0.64
3	$A_5 \rightarrow A_3$	0.54
6	$A_1 \rightarrow A_5 \rightarrow A_2$	0.47
4	$A_3 \rightarrow A_4$	0.46
5	$A_2 \rightarrow A_3 \rightarrow A_4$	0.36

and T_j:

$$p_c(h_i, h_j) = p(h_i, h_j) + \frac{\beta}{e^{2D(T_i,T_j)}} \qquad (3.1)$$

$$where \quad D(T_i, T_j) = \frac{\Sigma_{k=1}^{k=m} |t_{ik} - t_{jk}|}{m}, 0 < \beta < 1$$

Figure 3.1 depicts the function $\frac{1}{e^{2D(T_i,T_j)}}$, from which we can see that the contribution from page topic vector to $p_c(h_i, h_j)$ is getting small as the distance between the two topic vectors gets larger.

Support there are five topics involved in the example above and each page in the example has a topic vetor as shown in Table 3.1, where value 1 indicates that the corresponding topic is involved in that page and values 0 indicates not. It can be figured out that the topic distance between page A_1 and A_4, A_5 and A_2, A_2 and A_3, A_5 and A_3 and A_1 and A_5 are the same, the distance between A_3 and A_4 is a bit larger. By using Equation 3.1, we get the association rules as given in Table 3.2. The probability of session 6 becomes larger than the probabilities of session 4 and session 5 after considering the impact of the page topics. The irrelevancy of A_3 and A_4 makes the probabilities of sessions 4 and 5 not increase much. This example shows that the consideration of the page content does have impact on the centainty of the association rules. If the cut-point λ increases, session 4 and session 5 will be ruled out from the set association rules before session 6 is.

3.2 Session Entropy and Pruning

In information theory, Shannon's measure of entropy is used as a measure of the information contained in a piece of data. For a random variable X with a set of possible values $< x_1, \ldots, x_n >$, having probabilities $p(x_i)$, $i = 1, \ldots, n$, if we had no information at all about the value X would be, the possibility for each value should be the same, i.e. $1/n$. In this case, X is in its most uncertain situation. According to information theory, the entropy of X reaches its maximum in this situation. On the other hand, if the entropy of X is close to zero, the value of X has few uncertainties. In this case, there should be a small set of values with high probabilities and others with very low probabilities. Based on this theory, we propose to use the entropies of topics in a session to prune the association rules.

$\forall h_i \in H$, its topic vector is $T_i =< t_{i1}, \ldots, t_{in} >$. Each topic can be treated as a random variable with two possible values: involved or not involved. The entropy of topic t_j to page h_i can be estimated by $H(t_{ij}) = -(p(t_{ij})logp(t_{ij}) + (1 - p(t_{ij}))log(1 - p(t_{ij})))$, where $p(t_{ij})$ is the probability of t_j being involved in h_i. Assume that $t_{i1}, \ldots, t_{in}$ are independent variables, according to entropy theory, we have $H(t_{i1}, \ldots, t_{in}) = \Sigma_{k=1}^{k=n} H(t_{ik})$, which estimates the certainty of topics $t_{i1}, \ldots, t_{in}$ being involved in h_i. Let $s_i =< h_{i1}, \ldots, h_{ir} >$ be a session which represents page sequence $h_{i1} \rightarrow h_{i2} \ldots \rightarrow h_{ir}$, $T_{ij} =< t_{ij_1}, \ldots, t_{ij_n} >$ be the topic vector of page h_{ij} with $1 \leq j \leq r$, $T_{s_i} =< t_{s_i1}, \ldots, t_{s_in} >$ be the topic vector of s_i, and $p(t_{ij_k})$ is the probability of the kth topic being involved in h_{ij}. The probability of the kth topic being involved in s_i denoted as $p(t_{s_ik})$ $(1 \leq k \leq n)$ can be estimated by the following equation:

$$p(t_{s_ik}) = \frac{\Sigma_{j=1}^{j=r} p(t_{ij_k})}{r} \qquad (3.2)$$

Table 3.3: Topic vectors of the sessions in the above example

Sessions	Topic Vector
s_1	$T_{s1} =< 0.5, 1, 0.5, 0, 0.5 >$
s_2	$T_{s2} =< 1, 0.67, 0.67, 0.67, 0 >$
s_3	$T_{s3} =< 1, 0.5, 0.5, 1, 0 >$
s_4	$T_{s4} =< 0.5, 0.5, 1, 0.5, 0.5 >$
s_5	$T_{s5} =< 0.67, 0.67, 1, 0.3, 0.3 >$
s_6	$T_{s6} =< 1, 1, 0.3, 0.3, 0.3 >$

Table 3.4: The association rules obtained from the HPG with entropies ($\lambda = 0.3$)

Sessions	User Sessions	Sesion Entropies
s_1	$A_4 \rightarrow A_1$	0.6
s_2	$A_5 \rightarrow A_2 \rightarrow A_3$	0.8
s_3	$A_5 \rightarrow A_3$	0.6
s_4	$A_3 \rightarrow A_4$	1.2
s_5	$A_2 \rightarrow A_3 \rightarrow A_4$	1.1
s_6	$A_1 \rightarrow A_5 \rightarrow A_2$	0.78

The entropy of a session can be estimated by the following equation:

$$H(s_i) = H(t_{s_i1}, \ldots, t_{s_in}) = \Sigma_{k=1}^{k=n} H(t_{s_ik}) \qquad (3.3)$$

where

$H(t_{s_ik}) = -(p(t_{s_ik})logp(t_{s_ik}) + (1-p(t_{s_ik}))log(1-p(t_{s_ik})))$

The entropy of a session estimates the certainty of the topics involved in the session. If the entropy is small, then there must be some topics with high probabilities and the others with very low probabilities. In this case, this session is a good candidate to be selected as an association rule. On the other hand, if the entropy is large, then the probabilities of the topics must be very close and low as well. In this case, this session shouldn't be selected as an association rule since the pages in this session may not focus on some certain topics. For the example used in previous sections, Table 3.3 gives the topic vector of each session and Table 3.4 gives the entropy of each session. Table 3.4 suggests similar result as Table 3.2 does. That is, the session 4 and session 5 should be ruled out from the association rule set since their entropy is high.

4 Conclusion

In this paper we have proposed two methods that find the most popular navigation sessions based on both user access data and page contents. The first method modified the HPG model proposed by Borges [1] by taking the page content into account. This modification makes the HPG model more robust because the use of the page contents can eliminate or alleviate the impact made by the missing data or the irrelevant pages in access logs. The second method determines popular sessions according to session entropy which is based on well-established Information Theory.

References

[1] J. Borges and M. Levene. Data mining of user navigation patterms. In *Proceedings of the Web Usage Analysis and User Profiling*, volume 1, pages 31–36, 1999.

[2] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, February 1999.

[3] M. Levene and G. Loizou. A probabilistic approach to navigation in hypertext. *Information Sciences*, 114:165–186, 1999.

[4] M. Pazzani and D. Billsus. Adaptive web site agents. *Autonomous Agents and Multi-Agent Systems*, 5:205–218, 2002.

[5] M. Perkowitz and O. Etzioni. Adaptive web sites: An ai challenge. In *Proceedings f IJCAI-97*, volume 1, 1997.

[6] M. Perkowitz and O. Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118:245 – 275, 2000.

[7] S.Hanson and M.Bauer. Conceptual clustering, categorization, and polynorphy. *Machines Learning*, 3:343–372, 1989.

[8] T. Toolan and N. Kusmerick. Mining web logs for personalized site maps. In *Proceedings of the International Conference on Web Information Systems Engineering*, volume 1, 2002.

[9] S. Tso, H. Lau, and R. Ip. Development of a fuzzy push delivery scheme for internet sites. *Expert Systems*, 16:103–114, 1999.

[10] C. S. Wetherell. Probabilistic languages: A review and some open questions. *Computing Surveys*, 12(4):361–379, 1980.

[11] T. W. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From user access patterns to dynamic hypertext linking. In *Proceedings of the fifth interntional world wide web*, volume 1, 1996.

An Effective Recommendation System for Querying Web Pages*

Chien-Liang Wu and Jia-Ling Koh†
Department of Information and Computer Education
National Taiwan Normal University
Taipei, Taiwan 106 R.O.C.
E-mail: jlkoh@ice.ntnu.edu.tw

Abstract

In this paper, a recommendation system for querying web pages is developed. When users query web pages through a search engine, the query keywords, browsed web pages, and feedback values are collected as user query transactions. A clustering algorithm based on bipartite graph analysis is designed to determine clusters of query keywords and the browsed web pages, called access preference clusters. Next, association rules of query keywords and web pages are mined for each access preference cluster. The feedback values of browsed web pages are incorporated into the calculation of the support and confidence for each association rule to reflect the subjective opinions. Based on the mined clusters and rules, the system applies the concept of collaborative filtering to recommend highly semantics relevant web pages in the access preference clusters partially covered by a user profile or a given query. The initial experiment result shows the system can improve the querying effect of searching engines.

1. Introduction

Data on Internet is increasing rapidly. The World Wide Web has become one of the important sources to obtain data. To help users find useful data from the World Wide Web in a fast and effective manner, search engines provide users query interfaces to search relevant Web pages links by inputting keywords or query terms. By applying the indexing strategies of Web pages on contained keywords, most search engines return a large number of satisfied page links .However, most of the time, only a small proportion among the returned results is feasible web pages for users, not limited to contain the queried keywords exactly, is an important issue in a search engine.

Data clustering strategies are often applied to perform data analysis, decision-making, data retrieving, image segmentation, etc.[2][4][8]. Recently, many researchers [3][6][7][9] also focused on the clustering methods for analyzing user browsing behaviors to obtain clusters of web pages or the associations among query keywords and web pages.

The correlation of query keywords and web pages was deduce in [9] by performing a clustering method. In this approach, semantic relevance of query keywords was analyzed according to query keywords and accessed web page links which represented the similar information to user feedbacks in a traditional IR environment.

User profiles are crucial information for analyzing user clusters in [11]. Mostly, a user profile consists of personal data, transaction data, transaction behavior, preferred data, and classification data of the user. In [12], not only the browsed pages, the browsing frequency, time, and order of browsing activities of users in a specific site were also taken into consideration to determine user groups of similar browsing behaviors.

A most popular technique applied in recommendation system is collaborative filtering. This type of recommendation systems [5][7][10] performed data clustering method to group users according to their preference or behaviors. The recommendations for a user are determined based on the behavior histories of users in the same group with the user.

Most related works discussed above used user browsing histories to show the preferences of users. In addition, collaborative filtering strategy was used to recommend web pages or documents most popularly. However, in the results of most clustering analysis on user preferences, one user only belongs to a certain cluster. For users having various kinds of preferences, the number of users in the same cluster may be very small, resulting in a limited range of recommendable data. That is, other recommendable information provided by users with partially similar preferences gets lost. Moreover, among the returned results of a query on a search engine, users usually browsed the first few page links. Therefore, only considering whether or not a returned page link is browsed is insufficient to determine the semantic association between a query keyword and the page.

In this paper, a recommendation system for querying web pages is developed. When users query web pages

* This work was partially supported by the Republic of China National Science Council under Contract No. 92-2213-E-003-012
† Author to whom all correspondence should be addressed.

through a search engine, the query keywords, browsed web pages, and feedback values are collected as user query transactions. A clustering algorithm based on bipartite graph analysis is proposed to determine clusters of query keywords and the browsed web pages, called access preference clusters. Next, association rules of query keywords and web pages are mined for each access preference cluster. The feedback values of browsed web pages are incorporated into the calculation of the support and confidence for each association rule. Based on the mined clusters and rules, the system applies the concept of collaborative filtering to recommend highly semantics relevant web pages in the access preference clusters partially covered by a user profile or a given query.

The organization of the rest of this paper is as follows. Definitions of the terms mentioned throughout the paper are introduced in Section 2. Section 3 describes the strategy for mining associations between query keywords and the browsed web pages. A recommendation method for querying web pages through collaborative filtering approach is presented in Section 4. In Section 5, the experiment shows the applicableness of the proposed system. Finally, Section 6 concludes this paper.

2. Basic definitions

In this paper, a recommendation system is designed to be cooperated with search engines to provide more feasible results for web documents searching. The basic idea is to mine the association between user queries and Web pages according to querying and browsing histories of users.

The system proposed in this paper consists of three major processing components. Data preprocessing is performed as described in this section. The other parts will be introduced in detail in the following two sections.

In this proposed system, a user is allowed to log in as a registered member or anonymous user. The system applies JSP sessions to distinguish access behaviors of users. A unique session number is given for each connection, which remains valid for 1.5 hours. Users log in the system to query the required Web pages by inputting keywords. Moreover, the registered members can assign feedback values after browsing the recommended Web pages. Therefore, in addition to the usual access information (such as the IP address, domain of requesting machine, time and date of request, access method, IP address of requested file, etc.), the member ID, session numbers, query keywords, and feedback values are also recorded in the log file of the recommendation server for each document access. First of all, data in log files is preprocessed to extract information relevant to browsing behaviors of individual users.

For each document access, the extracted information includes user ID, section number, query keyword, http of browsed web page, and feedback value. A *user query session* consists of a query keyword and the set of corresponding browsed web pages and feedback values in a user query session.

User query sessions collected by the system within certain period are transformed into user query transactions. A unique transaction id is assigned for each user query transaction. An example of user query session shown in Table 1 is transformed into user query transactions as shown in Table 2. It shows user "User1" inputs query keyword "q1", then browses the returned pages d1, d2, and d3, and gives relevant feedback values 1.0, 1.0, and 0.6 for these pages respectively. The constructed user query transaction is denoted as <q1> (d1 1.0) (d2 1.0) (d3 0.6). In addition, the system records user query sessions with the same User ID into profile of the corresponding registered user.

[Definition 1] Let UT be a set of user query transactions, Q be the set of query keywords in UT, and D be the set of web pages in UT. E is a relationship between Q and D such that $E \subseteq Q \times D$.

$\forall T \in UT$, T contains $\{< q_i > (d_j, f_j)\} \rightarrow (q_i, d_j) \in E$.

A *browsing association graph* G(UT)=(Q, D, E) represents a relationship of keyword set Q and web page set D, which is a bipartite graph. Figure 1 shows the browsing association graph for the running example shown in Table 2.

[Definition 2] Given a browsing association graph G(UT)=(Q, D, E). Vertices q_i and d_j are *adjacent* if $q_i \in Q$, $d_j \in D$, and $(q_i, d_j) \in E$. For each $q_i \in Q$, the *neighbors* of q_i are the set of web pages adjacent to q_i, denoted as **Neighbor q_i** , where $\text{Neighbor}(q_i) = \{d_k | d_k \in D \wedge (q_i, d_k) \in E\}$. In Figure 1, q_3 is adjacent to vertices d_1, d_2, and d_5. Thus, Neighbor(q_3)={d_1, d_2, d_5}.

[Definition 3] The similarity between keyword q_i and q_j, denoted as **Sim q_i, q** , is evaluated by the following formula:

$$\text{Sim}(q_i, q_j) = \frac{|\text{Neighbor}(q_i) \cap \text{Neighbor}(q_j)|}{|\text{Neighbor}(q_i) \cup \text{Neighbor}(q_j)|}.$$

The function is also extended for evaluating the similarity between two sets of keywords, in which the neighbor of a set of keywords Q is defined as:

$$\text{Neighbor}(Q) = \bigcup_{q_i \in Q} \text{Neighbor}(q_i).$$

Association rules are usually used to represent associations that occur between data items in a transaction database. In this paper, in order to provide feasible web pages searching, association between a keyword and the set of browsed pages is analyzed. In addition, data items are supposed to have various association weights in a transaction. In a user query transaction, the association weight for the query keyword is set to 1.0. Besides, the

Table 1. Example of user query session

UserID: User1	
aaa-BgcJm-5856	<q1> d1 1.0
	<q1> d2 1.0
	<q1> d3 0.6
	<q2> d2 0.8
	<q3> d1 0.8
	<q3> d5 1.0
	<q4> d4 0.8
	<q4> d5 0.8
	<q4> d6 0.2
	<q5> d5 0.6
	<q5> d6 0.2
UserID: User2	
adj-BhkSy-8118	<q1> d3 0.2
	<q2> d1 0.8
	<q2> d2 1.0
	<q3> d2 1.0
	<q3> d5 1.0
	<q4> d4 0.6
	<q4> d5 0.8
	<q5> d5 0.8
	<q5> d6 0.2
Anonymous	
abe-GnmOst-1758	<q1> d2 0.8
	<q1> d3 0.2
	<q2> d1 0.6
	<q2> d3 0.2
	<q3> d1 0.8
	<q3> d5 1.0
	<q4> d4 0.8
	<q4> d6 0.2
	<q5> d5 1.0

Transfer

Table 2. Example of user query

TID	User Query Transaction
1	<q1> (d1 1.0) (d2 1.0) (d3 0.6)
2	<q2> (d2 0.8)
3	<q3> (d1 0.8) (d5 1.0)
4	<q4> (d4 0.8) (d5 0.8) (d6 0.2)
5	<q5> (d5 0.6) (d6 0.2)
6	<q1> (d3 0.2)
7	<q2> (d1 0.8) (d2 1.0)
8	<q3> (d2 1.0) (d5 1.0)
9	<q4> (d4 0.6) (d5 0.8)
10	<q5> (d5 0.8) (d6 0.2)
11	<q1> (d2 0.8) (d3 0.2)
12	<q2> (d1 0.6) (d3 0.2)
13	<q3> (d1 0.8) (d5 1.0)
14	<q4> (d4 0.8) (d6 0.2)
15	<q5> (d5 1.0)

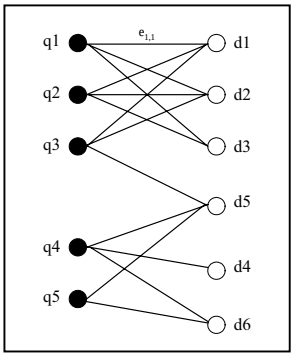


Figure 1. Bipartite graph

feedback value of each browsed Web page represents the association weight of the page in the transaction. For each data item X, X.weight(T_i) denotes the association weight of X in transaction T_i. X.weight(T_i)=0 if X is not contained in transaction T_i. In the example shown in Table 2, d_2.weight(T_2) = 0.8, d_2.weight(T_3) = 0 and q_2.weight(T_2) = 1.0

[Definition 4] Let **sup** denote the support of data item X in the database,

$$\sup(X) = \sum_{i=1}^{N} X.\text{weight}(T_i) \Big/ N$$

, in which N is the number of user query transactions in the database.

[Definition 5] Let **sup** , 1, , k denote the support of data itemsets $\{X, Y_1,\ldots,Y_k\}$.

$$\sup(X, Y_1,\ldots, Y_k) = \sum_{i=1}^{N} \left(X.\text{weight}(T_i) * \prod_{j=1}^{k} Y_j.\text{weight}(T_j) \right) \Big/ N$$

, in which N is the number of user query transactions in the database.

[Definition 6] The support and confidence of association rule $X \rightarrow Y_1,\ldots,Y_k$ ($k \geq 1$) are denoted as **sup** $(X \rightarrow Y_1,\ldots Y_k)$ and **conf** $(X \rightarrow Y_1,\ldots Y_k)$ respectively, which are computed by the following two formulas.

$\sup(X \rightarrow Y_1,\ldots Y_k) = \sup(X, Y_1,\ldots, Y_k)$, and

$$\text{conf}(X \rightarrow Y_1,\ldots Y_k) = \frac{\sup(X, Y_1,\ldots, Y_k)}{\sup(X)} \quad .$$

3. Association mining for query keywords and web pages

3.1. Access preference clusters

3.1.1. Clusters analysis. The similarity function formulated in definition 3 is used to calculate the similarity value between two query keywords. That is, the keyword clusters are evaluated according to the degree of neighborhood overlapping in the browsing association graph. If two query keywords have a similarity value no less than a predefined threshold value, these keywords are allocated in the same cluster.

Access Preference Clustering Algorithm
Input: browsing association graph G, *similarity-θ*, and *element-θ*
Output: Sets of access preference clusters
  Assign each query keyword in G to a cluster;
  Repeat
    Compute the similarity values for each pair of clusters;
    Select the largest Sim(C_i, C_j) among the cluster pairs;
    If Sim(C_i, C_j) is greater than *similarity-θ*
      Then Merge clusters C_i and C_j;
    Else Exit Repeat loop
  End Repeat
  Remove the clusters whose number of elements are below *element-θ*.

Figure 2. Algorithm of clustering analysis

Figure 2 is the pseudo code of the clustering algorithm for finding keyword clusters. Suppose similarity threshold value (denoted as *similarity-θ*) is set to 0.4, and threshold value of element numbers (denoted as *element-θ*) is set to 2. In the running example, two clusters of keywords C1={q_1, q_2, q_3}and C2={q_4, q_5} are obtained. The keywords in a cluster and the corresponding browsed web pages are jointly called an *access preference cluster* as the example shown in Figure 3. It is possible that a web

page is allocated in more than one access preference just like d5 shown in Figure 3.

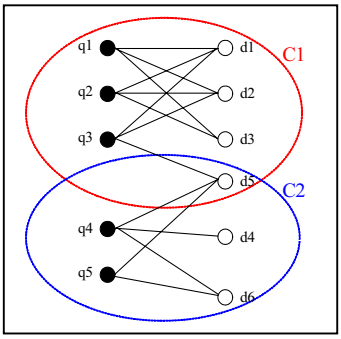


Figure 3. Example of access preference cluster

3.2. Association rules of keywords and web pages

3.2.1. User query transactions partitioning. After access preference clusters are found, user query transactions are correspondingly partitioned according to the clusters of query keywords. For example, in Figure 3, cluster C_1 includes query keywords q_1, q_2, and q_3, and cluster C_2 includes query keywords $q_4$and q_5. Therefore, the user query transactions are partitioned into two corresponding parts, SUT_1 and SUT_2, respectively. SUT_1 contains user query transactions TID {1, 2, 3, 6, 7, 8, 11, 12, 13} and SUT_2 contains {4, 5, 9, 10, 14, 15}.

In our approach, the access preference clusters are analyzed according to the relationship between query keywords and browsed web pages. Thus, it is possible that a user profile contains multiple preference clusters. In the running example shown as Table 1, the query keywords contained in the user profile of "User1" is {q_1, q_2, q_3, q_4, q_5}. It shows that the access preferences of User1 belong to both clusters C_1 and C_2. This approach reflects multiple access preferences of a user in order to better catch user preferences in actual cases.

Our approach aimed to determine association rules of the type $q_i \rightarrow D$, where q_i is a single query keyword and D is a set of web pages. In order to extract the representation information in a access preference cluster, frequent itemsets in the corresponding partition of user query transactions are mined by applying Apriori algorithm [1]. However, the support of a data item is calculated according to definitions 4 and 5. Since only the association between a single query keyword and web pages is considered, there is no need to find frequent itemsets that include more than one query keyword. Suppose the minimum support θ_s is 0.15. By following the example shown in Table 2, the frequent itemsets in partition SUT_1 is as shown in Table 3, in which L_{k-1} denotes the set of frequent itemsets consisting k-1 elements.

For each frequent itemset, an association rule is generated and rule confidence is calculated according to definition 6. Only the association rules with confidence no less than the minimum confidence θ_c are retained. Let the set of association rules deduced from the partition of query transactions of *i*th access preference cluster be denoted as $Rset_i$. Table 4 shows the set of association rules ($Rset_1$) deduced from the frequent itemsets shown in Table 10 (minimum confidence θ_c =0.4).

Table 3. Frequent ietmsets

L_1	Sup	L_2	Sup	L_3	Sup
q1	0.33	q1, d2	0.2	q3, d1, d5	0.17
q2	0.33	q2, d1	0.16		
q3	0.33	q2, d2	0.2		
d1	0.44	q3, d1	0.17		
d2	0.51	q3, d5	0.33		
d5	0.33	d1, d2	0.2		
		d1, d5	0.17		

Table 4. Association rule for SUT_1

Association Rule	Conf.
q1→d2	0.6
q2→d1	0.48
q2→d2	0.6
q3→d1	0.51
q3→d5	1.0
q3→d1, d5	0.51

4. Collaborative recommendation system

Two types of web pages recommendation methods are provided by this proposed system: (1) recommendation based on user profile, and (2) recommendation based on inputted query keywords.

4.1. Personalized recommendations

This type of recommendation method is provided for registered members. When a member submits a recommendation request, the system will find the access preference clusters that have keywords contained in the user's profile. The confidence of an association rule is used as the *recommendation value* of web pages in the rule. Thus, the association rules between keywords and web pages in such an access preference cluster are sorted by their confidences. The web pages in the first n rules, that the user has not yet browsed, are recommended in that order.

Figure 4 shows an example of access preference clusters C_1, C_2, and C_3 mined out, and their association rule sets for query keywords and web pages are $Rset_1$, $Rset_2$, and $Rset_3$, respectively. Suppose the user profile of "user3" is as shown in Table 5. After user3 submits a recommendation request, the user profile shows that user3 belongs to access preference clusters C_2 and C_3. The system then determines the n association rules with most significant confidences from $Rset_2$ and $Rset_3$, individually. Suppose n is set to 3. The set of rules $\{q_5 \rightarrow d_3, q_4 \rightarrow d_5, q_5 \rightarrow d_6\}$ from $Rset_2$, and $\{q_7 \rightarrow d_{11}, q_6 \rightarrow d_4, q_8 \rightarrow d_{10}\}$ from $Rset_3$ will be selected. Finally, the recommended web pages are d_3, d_6, d_{11}, d_4, and d_{10}. The

web page is ignored from the result because it has been browsed by user3.

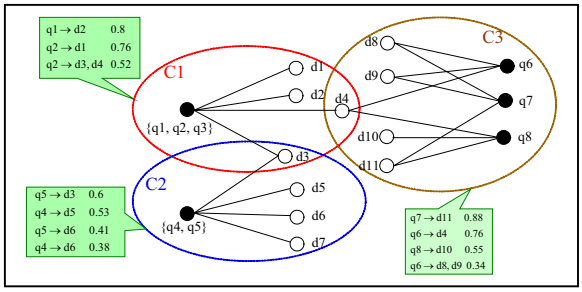


Figure 4. Example of access preference clusters

Table 5. Profile of User3

User ID: User3	
aaa-BgcJm-5856	<q4> d5 0.8
	<q4> d7 0.6
	<q5> d7 1.0
aks-HesBq-1475	<q6> d8 0.4
	<q6> d9 0.6
	<q7> d8 0.8
	<q7> d9 0.8

4.2. Recommendations based on query keywords

The second method of recommendation applies to both members and anonymous users. Each user submits a query keyword to the system for querying semantics related web pages. Based on the access preference cluster C_i the given keyword belongs to, the association rules in $Rset_i$ are sorted by confidence. Web pages in the first significant n rules are recommended.

Continue the example shown in Table 5. Suppose a user inputs query keyword q_6 and requires a recommendation search, the three rules with most significant confidences are selects from the association rules set $Rset_3$ because q_6 belongs to access preference cluster C_3. The web pages in these rules, namely d_{11}, d_4, and d_{10}, are then recommended. By applying this method, the semantics relevant web pages which do not contain the given keyword also can be obtained, such as d_{10} in this example.

5. System implementation and experiment

5.1. System implementation

The system is performed on a 800MHz Pentium III PC with 256MB of memory, running Windows 2000 Professional and Resin-2.1.0 Web Server. VB programming language is used to implement the internal training tasks of the system. JSP programming language is used to implement the functions of on-line querying recommendations.

Users log in the system as members or anonymous users. The system provides three functions for users: searching web pages through Chinese Yahoo searching engines, personalized recommendations of web pages, and recommendations of web pages by giving query keyword. The personalized recommendations are provided only for members since the corresponding user profiles are required for this function.

Users can submit query keywords to Chinese Yahoo searching engine and receive searched results through our system. Due to that a large amount of information may be returned, the system displays only the first 200 web page links returned to users. The system also provides a brief description for each returned page link. After clicking the title of a web page, users can further browse the full content of an article. Moreover, the feedback button located on the right ride of the title is used to assign a feedback value for the browsed page. This system uses the Liker's five-point scale to divide the feedback values into five levels (1 to 5). The corresponding feedback values for web pages are 0.2, 0.4, 0.6, 0.8, and 1.0. The default value is set to be 0.6. The system will record querying history of each user in the user profile, including query keywords, browsed web pages, and feedback values. The collected data is transformed to user query transactions. The clustering analysis and association mining are performed on these transactions, as introduced in the previous sections, to provide the other two functions.

5.2. Experiment

Due to privacy concerns, the access logs of searching engines cannot be obtained easily. The experiment is performed on the students in database laboratory to show the effects of query recommendation provided in this system.

In order to sure to find access preference clusters, query keywords used in this experiment were limited to data mining related terms, including “data warehouse”, “OLAP”, “data mining”, “sequential mining”, “decision tree”, “association rule”, “collaborative filtering”, and “personalization”. A total of 1861 query browsing records were collected, in which 1681 query records were served as training data and the other 180 query records were used as testing data.

The mean absolute error (MAE) is defined as the following formula to evaluate the feasibility of recommended web pages for a user’s query.

$$MAE_i = \frac{\sum_{s=1}^{r} |p_s - f_s|}{r}$$

in which p_s and f_s denote the recommendation value of the system and the feedback value given by the user for web page s, respectively. In addition, r denotes the number of web pages browsed by the user. The overall MAE value for a set of query recommendations is the average of the MAE values for these recommendations.

The smaller MAE value implies the recommendations fit the requirements of users.

When training the system, the values of *similarity-θ* and *element-θ* are set to 0.02 and 2, respectively. Two query preference clusters are extracted, in which the included keywords are shown as the follows Table 6.

Table 6. Result of clusters

Cluster 1	sequential mining, collaborative filtering
Cluster 2	data mining, data warehouse, OLAP

In addition, when mining the association rules in SUT_1 and SUT_2, the minimal supports are set to 0.15 and 0.09, and 65 rules and 43 rules mined out from $Rset_1$ and $Rset_2$, respectively. By changing the number of recommended web pages, the MAE values calculated from the testing data are shown in Figure 5. The result shows that the recommendations meet the requirements of users very closely. As the number of recommend web pages increases, the mean absolute error decreases even more.

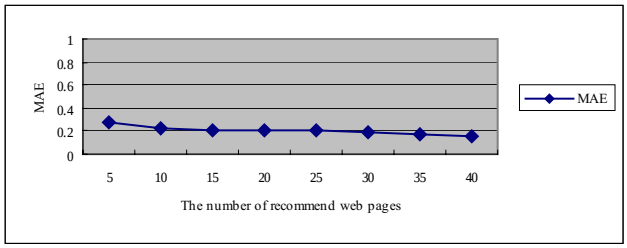


Figure 5. Result of MAE measure

6. Conclusion and future research directions

In this paper, a recommendation system cooperated with Chinese Yahoo searching engine is proposed. A clustering algorithm is designed to determine access preference clusters from previous user behaviors. These clusters represent the semantics related associations between query keywords and browsed web pages. According to the submitted query keywords or profiles of users, feasible results are recommended based on the association rules of keywords and web pages mined from user behaviors in each cluster. In this approach, a user with various preferences usually belongs to multiple access preference clusters. Therefore, the browsing behaviors of users with partially similar preferences are also used to provide recommendation information when applying collaborative filtering strategies. Moreover, the feedback values of web pages given by users are used in the computations of supports and confidences for association rules to reflect the subjective opinions. The initial experiment result shows the system can improve the querying effect of searching engines.

In the future, the plan is to obtain web server logs from search engines or collects user querying behaviors in much longer period in order to get more user query transactions for training system widely. In addition, the feedback values of browsed web pages are considered to be included in the similarity computation of keywords when mining access preference clusters. Furthermore, browsing time and frequency will be considered as weights and considered in the confidence computations of association rules in order to further improve the effectiveness of this system.

References

[1] R. Agarwal, and R. Srikant, "Fast Algorithm for Mining Association Rule in Large Databases," in Proceeding of The 20th International Conference on Very Large DataBases, 1994.

[2] D. Barbara, and P. Chen, "Using the Fractal Dimension to Cluster Datasets", in Proceeding of the 6th International Conference on Knowledge Discovery and Data mining, ACM SIGKDD, 2000.

[3] D. Beeferman, and A. Berger, "Agglomerative Clustering of A Search Engine Query Log," in Proceedings of the 6th International Conference on Knowledge Discovery and Data Mmining, ACM SIGKDD, 2000.

[4] C.H. Cheng , A.W. Fu, and Y. Zhang, "Entropy-based Subspace Clustering for Mining Numerical Data," in Proceeding of International Conference on Knowledge Discovery and Data Mining, ACM SIGKDD, 1999.

[5] X. Fu, J. Budzik, and K.J. Hammond, "Mining Navigation History for Recommendation," in Proceeding of International Conference on Intelligent User Interfaces, ACM, 2000.

[6] H.J. Kim, and S.G. Lee, "A Semi-Supervised Document Clustering Technique for Information Organization," in Proceeding of the 9th International Conference on Information and Knowledge Management, 2000.

[7] B. Mobasher, R. Cooley, and J. Srivastava, "Creating Adaptive Web Sites Through Usage-Based Clustering of URLs," in Proceedings of the Workshop on Knowledge and Data Engineering Exchange, 1999.

[8] K. Wang, C. Xu, and B. Liu, "Clustering Transactions Using Large Items," in Proceeding of the 8th International Conference on Information and Knowledge Management, ACM, 1999.

[9] J. Wen, J.Y. Nie, and H.J. Zhang, "Clustering User Queries of A Search Engine," in Proceedings of the 10th International World Wide Web Conference, 2001.

[10] Y.H. Wu, Y.C. Chen, and A.L.P. Chen, "Enabling Personalized Recommendation on the Web based on User Interests and Behaviors," in Proceeding of International Workshop on Research Issues in Data Engineering , RIDE IEEE, 2001.

[11] K.L. Wu, C.C. Aggarwal, and P.S. Yu, "Personalization with Dynamic Profiler," Advanced Issues of E-Commerce and Web-Based Information Systems, in Proc. of the 3th International Workshop on Computer Networks and Mobile Computing (WECWIS), 2001.

[12] J. Xiao, and Y. Zhang, "Clustering of Web Users Using Session-Based Similarity Measures," in Proceedings of International Conference on Computer Networks and Mobile Computing, 2001.

On User Recommendations Based on Multiple Cues

G. Dudek and M. Garden
Centre for Intelligent Machines, McGill University
3480 University St, Montréal, Québec, Canada H3A 2A7
{dudek,mgarden}@cim.mcgill.ca

Abstract*— *In this paper we present an overview of a recommender system that attempts to predict user preferences based on several sources including prior choices and selected user-defined features. By using a combination of collaborative filtering and semantic features, we hope to provide performance superior to either alone. Further, our set of semantic features is acquired and updated using a learning-based procedure that avoids the need for manual knowledge-engineering. Our system is implemented in a web-based application server environment and can be used with arbitrary domains, although the test data reported here is restricted to recommendations of movies.

I. Introduction

This paper outlines a recommendation system [1] that combines aspects of user-based and item-based methodologies to attempt to avoid some of the difficulties encountered with either approach in isolation. Web-based systems that attempt to recommend products to users based on knowledge of their personal preferences have become relatively commonplace. The preferences in question may simply be items in a current web-based shopping cart, but more typically they are based on historical shopping data or explicit responses to questions. As such, recommendation systems represent one of the most successful and unique application spaces of the world wide web; prior to the web, such recommendation systems were unheard of in this form and essentially infeasible.

Recommendation systems are typically based on one of two key paradigms: collaborative filtering, which makes recommendations to a user based on what similar users have done, or item-based filtering, which makes recommendation to a user based on inferred connections between the objects in the domain of interest without explicitly modeling other users (which are typically used implicitly to create the inter-object links). An idealized example of collaborative filtering would be to recommend to a user the films that his or her best friend is known to have enjoyed, and which the user has not seen before; in practice the "best friend" is computed using statistical methods, most commonly as a form of cross-correlation. A simple example of item-based filtering would be to recommend a film to a user based on its genre (e.g. comedy, family, action, etc.) given prior knowledge of the genre of the films the user has enjoyed in the past; in practice, the genre is one of several features that can be used, but choosing the features to use can be problematic.

[1] the system can be accessed at http://q.cim.mcgill.ca

Each of these standard paradigms has been demonstrated to be successful, but each suffers from several shortcomings. In addition, we have identified several shortcomings that seem to be shared by *both* approaches and which we are attempting to address.

Collaborative filtering suffers from the fact that the basis for user preferences is not modeled whatsoever: one user might like a film for its humor, another might like it for its action. As such, the correlation between their tastes may be incidental [1]. For pairs of users that have only rated a small number of items, the risk of incidental correlations based on insufficient data is significant. On the other hand, for users that have rated a very large number of items, the risk of incidental correlations is also substantial.

Item-based recommendations can be based on several approaches. We will restrict our attention to those based on semantic features such as the genre of the items. For such systems, the set of features used to classify the items is critical and it needs to be both expressive and tractable. If the features in question (e.g *action movie*) are too broad or subjective, then the recommendations will suffer (for example, a husband and wife might disagree on the definition of a "dramatic film" or a "violent film"; sometimes this disagreement can itself be dramatic). On the other hand, if the features are too narrow to be applicable to many items, then either the rating matrix will become too sparse or ratings will be interpreted inconsistently in order to force them to apply more broadly.

Finally, it is often the case that the recommendations suitable to a user will be context dependent. The context of a user's search often has a significant bearing on what should be recommended, yet to our knowledge is has not been considered in the context of filtering applications. For example, if one is looking for a book, it is often for a specific purpose, be it light reading, self-improvement, or academic research, and a book that is appropriate to a reader in the light reading context will probably not be useful for research. Similarly, if one is looking for a film, then those that are appropriate for viewing with one's family may be different from those one would view when alone.

In this paper we outline the design of a recommender system that combines aspects of collaborative filtering and item-based recommendation. Our collaborative filtering algorithm and item-based components are based on statistical pattern matching methodologies. In the development of our item-

based model, however, we use a non-deterministic selection mechanism to define the attributes of interest that relate items to one another. Our approach is domain independent and should be suitable for most domains, but in the context of this paper we will use movie recommendations as our example domain.

A. Outline

In the remainder of this paper we discuss related work, and the problem of providing personalized recommendations. We continue with an exposition of the methodology and architecture, provide an illustrative example, and discuss some of the realized benefits and challenges. Finally, we close with a discussion of open problems, directions for future work and conclusions/findings from our work.

II. Related Work

Several different approaches have been considered for automated recommendation systems. Very broadly, the bulk of these can be classified into three major categories: those based on user-to-user matching and referred to as collaborative filtering, those based on item content information, and hybrid methods.

Collaborative filtering was first developed and identified as a methodology in the Tapestry email and Usenet filtering system [2]. In that work the emphasis was on the use and transmission of manually generated annotations of articles. In the GroupLens system the collaborative filtering paradigm was automated to provide automatic filtering of Usenet news articles [3].

Recommender systems are often described as being either memory- or model-based. Memory-based systems make predictions based on the entire raw data set, while model-based systems perform predictive calculations based on a version of the data which has been reduced in size [4]. A memory-based system might calculate nearest neighbors for each user and make predictions based on the preferences of those neighbors. In such systems, the similarity between users is often defined in terms of Pearson Correlation or the Vector Similarity measure used in Information Retrieval [4], [5]. Examples of model-based systems include Goldberg's "Eigentaste" framework [6] and Canny's "Mender" system [5], both of which map ratings data to a lower-dimensional subspace before making predictions. Systems such as the probabilistic Personality Diagnosis use a hybrid of the memory and model-based approaches [7].

In most recommender systems the overall opinion of an item is given as a integer value on some discrete scale. In the Entree recommender system, however, the user indicates a feature in which the item is lacking [1]. The system then determines which items are rich in the qualities being searched for, based on the responses of other users.

Due to the large number of items in most recommender systems, the ratings data will tend to be very sparse [8]. In such situations knowing a user's opinion of an arbitrary item may not help in determining the user's relationship with other users [9], [10]. An important aspect of a recommender system therefore is having a principled way of suggesting items to be rated. Approaches include using Partially Observable Markov Decision Processes [11] or Expected Value of Information [9] to determine which ratings will provide the most information. A simpler approach is to prompt the user to rate items which have been rated with a high variance [6].

In developing our recommender system we have taken all of these issues into consideration.

III. Approach

Typically in a recommender system a user will be given a list of items and prompted to indicate his or her preference for each. Preferences are indicated with a value in some range of integer values, e.g. $[1,10]$, or by choosing "like" or "dislike". In Burke's Entree restaurant recommender system, instead of a numeric rating, the user selects a *semantic rating* which describes some aspect of the item [1]. For instance the user can select a rating such as "less expensive" or "nicer" to indicate an attribute he or she is looking for but feels is lacking in the current restaurant. In that work the set of possible semantic ratings a user can choose from is predefined when the system is created. Additionally, distances representing the similarity between the ratings are determined in advance by knowledge-engineering (i.e. manual intervention).

Other systems use information regarding the content of items in order to infer reasons behind a user's preferences. For instance if a user consistently exhibits a preference for movies which the system knows are classified under the action genre, then the system will automatically infer that the user enjoys action films.

In our system, we wish to have information about which items a user prefers, but also to collect information about which features of the item the user liked or disliked, i.e. which features contributed to the user's preference. In addition, we wish to learn a large set of features suitable for classification. To do this, we allow the user to suggest arbitrary features to the system at their discretion, and to classify items using these new *ad hoc* features. This avoids the need to guess all suitable features in advance. It also allows for the use of specialized features for specific sub-domains where specialized vocabulary may be appropriate. Finally, it permits new jargon to be introduced as it develops (for example in technical domains, or in domains related to popular culture). The obvious disadvantage of this is that (a) obscure or useless features may be introduced, (b) feature selection may become onerous due to the excess of available features and (c) redundant features may be introduced (such as both "funny" and "amusing" in the context of movies). We address these issues below.

In our system, the user indicates overall liking or disliking of an item with an integer rating on a scale of 1 to 10 (where 1 indicates an extremely negative opinion and 10 indicates an extremely positive opinion). To complete the rating the user is required to specify at least one feature of the item which was important to his or her overall rating. The user is presented with a list of possible features for rating purposes,

and is also given the opportunity to add new features to the system. For each feature chosen, the user specifies whether the feature contributes positively or negatively to the overall rating of the item, again on a scale of 1 to 10.

Both item-based and collaborative filtering-based recommendations have advantages [12]. Aside from the empirical data, item-based methods can be used to provide recommendations when the number of viewers (in the case of films) is too small to use collaborative filtering reliably (for example for new films). On the other hand, collaborative filtering can sometimes provide recommendations for items where the features have not been clearly defined, or in cases where existing features are not adequate descriptors. Thus, we have selected a hybrid approach that combines the two approaches as described above, with a weighting factor between 0.1 and 0.9 (i.e. one source has a weight of α and the other has a weight of $1-\alpha$).

A. Infrastructure

The system consists of a website implemented using the Zope application server to provide a dynamic HTML front end. User and item data is supported with a MySQL database. The more intensive computations are implemented in C and are connected to Zope and the website via Python. The movie database used is the EachMovie data set [2] (only the movie information was used).

The structure of the system, including the Zope web presentation, the database schema, and the recommendation code have all been designed to be domain-independent, so that domains other than movies can be easily used.

B. User similarity

The collaborative filtering component of our methodology hinges on the ability of define the similarity between a target user for whom we are to make a recommendation, and any other user. Once this similarity is defined it allows us to find users who are nearby to each other in the sense of preferences. The preferences of nearby users can then be used to compute a prediction for the unseen preferences of the user requesting recommendations. We use Pearson Correlation to compute similarity, and since we collect data both regarding the items themselves and the features of the items, we can compute similarity in either item or feature rating space.

Each user j will input a set of ratings in which $r_{ij} \in [1,10]$ is his or her overall rating of item i. The system contains a set of user-specified features $F = \{\phi_1, \phi_2, ..., \phi_t\}$. For each feature k felt to be important to the overall rating of item i, user j specifies a rating f_{ij}^k in the range $[1,10]$.

Given this ratings set, computing user similarity consists of two parts. First, we compute user similarity based purely on the "overall" ratings. The similarity between users p and q with respect to overall ratings is defined as the Pearson Correlation between their ratings:

$$s_r(p,q) = \frac{\sum_i (r_{ip} - \bar{r}_p)(r_{iq} - \bar{r}_q)}{\sqrt{\sum_i (r_{ip} - \bar{r}_p)^2 \sum_i (r_{iq} - \bar{r}_q)^2}} \quad (1)$$

where $\bar{r}_p$ and $\bar{r}_q$ are the mean overall item ratings for all items rated by users p and q, respectively, and the summations are over each item i rated by both users p and q.

Next, we compute the similarity between users based on their preference for features. We calculate the mean rating $\bar{f}_j^k$ given to feature k by user j, and $\bar{f}_j$, the mean feature rating assigned by user j across all ratings. Then we compute the similarity between two users p and q as the Pearson correlation between these feature rating statistics:

$$s_f(p,q) = \frac{\sum_k (\bar{f}_p^k - \bar{f}_p)(\bar{f}_q^k - \bar{f}_q)}{\sqrt{\sum_k (\bar{f}_p^k - \bar{f}_p)^2 (\bar{f}_q^k - \bar{f}_q)^2}} \quad (2)$$

where the summations are over all features k which have been used by both users p and q.

Given s_r and s_f and a weight $\alpha \in [0.1, 0.9]$ we can compute the hybrid similarity between users p and q as

$$s_\alpha(p,q) = (1-\alpha) s_r(p,q) + \alpha s_f(p,q) \quad (3)$$

Using equation 3 we can define the nearest neighbors of user j to be N_j^α, the set of n users with the highest similarity s_α to user j for a given α. The effect of α is to allow us to give more or less weight to the overall ratings or the feature ratings when determining user similarity.

Having chosen a value of α and computed the set of nearest neighbors to user j, we can predict the rating user j will give to item i:

$$r_{ij}^p = \bar{r}_j + \kappa \sum_{u \in N_j^\alpha} s_\alpha(j,u)(r_{iu} - \bar{r}_u) \quad (4)$$

where κ is a normalizing term.

C. User-specified features

In preliminary tests of the system a limited user population rapidly increased the number of features to over 100. Having to wade through this list to make each rating would be unacceptable to most users. Furthermore, several of the features only apply to certain classes of film, and some are too esoteric to be of interest to most users. Our approach, instead, is to select a useful subset of the features at any given time and present these to the user. On the other hand, it is worthwhile to collect *some* data on even those features of little apparent utility since new features need an opportunity to become useful, and even less useful features provide useful cues to inter-item associations. Thus, while the user is able to access and choose from the complete list of features, should that be desired, the system attempts to make the choice easier by probabilistically selecting a subset of features to be suggested for use. To suggest features for a particular item, the system computes the variance in ratings for each feature which has been used to rate the item. The features with the highest variance are considered to be informative because while users tend to agree the feature

[2] http://research.compaq.com/SRC/eachmovie/

is applicable to the item, they disagree about whether it is a positive or negative aspect of the item. On the other hand, low variance features are considered to be less informative. The highest variance features will always be suggested to the user, while a given low-variance feature t with variance σ_{it}^2 for item i will be suggested with the following probability:

$$P_i(\phi_t|\sigma_{it}^2) = \frac{1}{e^{g(\sigma_{it}^2)}} \quad (5)$$

where g is a normalizing function.

While the system automatically matches misspelled or similar features when they are added to the system, it is likely that users will enter features which are *semantically equivalent* to features already in the system. These features will be redundant and should be merged. By building and using a table of synonyms, we believe we can reduce the effect of these mutually redundant features. Such a table could be built in part by examining correlations in feature use as in Table II, i.e. features which are used for the same items very often can be flagged as possible synonyms by the system. The problem of redundant features may also admit more complex solutions and seems to be amenable to a correlation-based analysis and may be linked to issues of granularity [13].

IV. Preliminary Results

At the time of this writing, our results remain rather tentative as we are collecting further data. One difficulty we faced is that since our rating system is novel, any existing dataset we use can only be used for item information, and not for rating information. As a consequence the results presented below are preliminary, being based on the rating information we have been able to collect so far. At the time of writing the system consists of 48 users, 1016 ratings entered using 125 features, and the database contains 1641 movie entries.

We have estimated the optimum weight for our limited user population using leave-one-out cross validation. That is, for each user we ran a series of trials. In each trial, all but one of the items rated by the user were used to calculate a predicted rating for the remaining item (if possible), and then the error between the predicted and actual rating was calculated. We repeated this process for each user in the system, and calculated the mean of all absolute errors (MAE) over these trials. For cases where the number of nearest neighbors (as per Equation 4) is 12, the cross-validation error as a function of the weight given to feature information, α is illustrated in Fig. IV. It appears that optimum performance is achieved for a weight of 0.3. For some parameter settings, it appears that the optimum weight takes on different values, including cases where the optimum recommendation is produced by maximizing the influence of the feature-based recommendations. Due to the limited size of the user population these results should be regarded as tentative, but they suggest that item-based and collaborative filtering-based recommendations can be combined profitably. Further, the eclectic features suggested by a sample user population seem to provide effective recommendations, despite redundancy and some features of minimal utility.

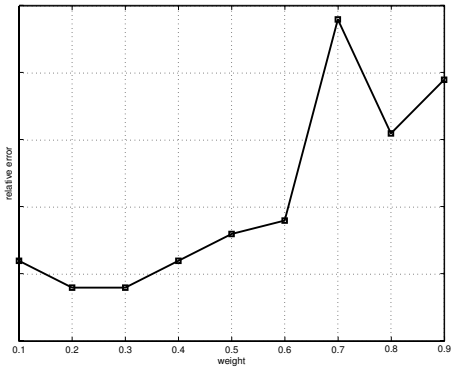


Fig. 1. Mean Absolute Error (MAE) of cross-validation test as a function of the weight α given to features (versus other users) when making predictions.

Presented in Table I is a list of some of the more popular features and the frequency with which they were used.

TABLE I

A selected sample of user-created features and a relative frequency indicator measuring how often each had been used.

Feature	Frequency of use
comedy	55
action	37
sci-fi	36
stupid	34
violence	22
long	15
black comedy	15
social commentary	13
dark	12
Ahnold-esque one-liners	11
complex	7
Ingrid Bergman	1

The features in the system consist of information about genre, actors, directors, as well as more high-level judgments, for instance “stupidity” and “complexity”. Information about genre, actors, directors, and even plot, can be obtained mechanically from the description of the item (i.e. databases such as the EachMovie set contain genre information), and most people would agree about whether such a feature applies to an item or not. On the other hand, the higher-level features would typically only be available from subjective reviews of the item, and would be subject to debate.

Some of the items in the list might appear useless at first. Take for instance the feature “stupid”. We expected that all users who used “stupid” to describe an item would use it negatively, but in fact it was used by many users, and as both a negative and positive feature. The same behavior was true for the feature “complex”. In followup interviews that were conducted with a subset of users, some people indicated that they genuinely enjoyed movies with a “stupid” component (much to the amazement of one of the authors of this paper).

We also examined the correlation between pairs of features (i.e. pairs of features which were both used to describe the same item). Table II lists some of the more highly correlated and interesting combinations. Such a table will help a human

system administrator to identify semantically equivalent features since they will presumably be used to describe the same items.

TABLE II
A SAMPLE OF PAIRS OF USER-CREATED FEATURES WHICH ARE USED TO RATE THE SAME ITEMS.

horror	grotesque
space wars	futuristic
artsy	stylized
complex	Time Travel
complex	interesting plot
woody allen	intellectual
children	amusing
animation	children
historical	complex content
atmospheric	photography
surreal	dystopia
action	violence
artsy	lots of questions unanswered
very moving	character drama

V. DISCUSSION

In this paper we have outlined the design of a recommender system we have developed that combines collaborative filtering with continuously modifiable user-suggested feature-based recommendations. In preliminary trials it appears to provide good recommendations and addresses some issues with standard systems in terms of how items are rated.

One issue we are currently addressing is the need to adjust the recommendation processes conditionally as a function of the context in which the recommendation will be used. We are currently evaluating allowing users to define the context in which a recommendation should be considered. This will allow us to conditionally rate items and, hence, to subsequently make conditional recommendations (i.e. "this film would be good to watch with your children"). The absence of existing data sets has hampered the quantitative evaluation of this approach. Further elaboration is outside the scope of this paper and so at this time we can only suggest that it appears promising.

Examining the features used to classify films, it was clear to us that several of the features suggested by users were ones we would not have inserted ourselves, and yet they proved appealing to users and useful; in fact one might speculate that these whimsical features improved the level of user satisfaction (although we have no firm data to corroborate that). Further, the mere fact that users could add additional features to address perceived shortcomings in the system seems to enhance the sense of satisfaction and community, and reduce frustration – all important factors in a recommendation system that uses collaborative filtering.

Our results based in stochastic presentation of features based on utility seems to allow us to construct an item-based recommendation component of our system that uses a rather large number of user-defined features, while only requiring users to select from a limited list when indicating their preferences. This seems to lead to good performance of the recommendation system. On the other hand, users occasionally expressed frustration (in followup interviews) because features they had expected to find based on past experience were not present sometimes. We have addressed this on an interim basis by allowing users to manually type in additional features, and if these match an existing feature then that feature is used.

REFERENCES

[1] R. Burke, "Semantic ratings and heuristic similarity for collaborative filtering," in *AAAI Workshop on Knowledge-based Electronic Markets*, pp. 14–20, AAAI, 2000.

[2] D. Goldberg, D. Nichols, B. M. Oki, and D. B. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. CACM 35, no. 12, pp. 61–70, 1992.

[3] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," in *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, (Chapel Hill, North Carolina), pp. 175–186, ACM, 1994.

[4] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 43–52, 1998.

[5] J. Canny, "Collaborative filtering with privacy via factor analysis," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 238–245, ACM Press, 2002.

[6] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[7] D. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, UAI 2000*, (Stanford, CA), pp. 473–480, 2000.

[8] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the tenth international conference on World Wide Web*, pp. 285–295, 2001.

[9] C. Boutilier and R. S. Zemel, "Online queries for collaborative filtering," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics* (C. Bishop and B. Frey, eds.), 2003.

[10] S. Dasgupta, W. Lee, and P. Long, "A theoretical analysis of query selection for collaborative filtering," *Machine Learning*, vol. 51, pp. 283–298, 2003.

[11] C. Boutilier, "A POMDP formulation of preference elicitation problems," in *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pp. 239–246, 2002.

[12] J. Herlocker, J. Konstan, A. Borchers, , and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in infofrmation retrieval*, pp. 230–237, 1999.

[13] P. Paulson and A. Tzanavari, "Combining collaborative and content-based filtering using conceptual graphs," in *Modeling with Words* (J. Lawry, J. Shanahan, and A. Ralescu, eds.), Lecture Notes In Artificial Intelligence Series, Springer-Verlag, 2003.

On Graph-based Methods for Inferring Web Communities

Jianchao Han
Dept. of Computer Science
California State University
Dominguez Hills
Carson, CA 90747
jhan@csudh.edu

Xiaohua Hu
College of Computer and
Information Science
Drexel University
Philadelphia, PA 19104
thu@cis.drexel.edu

Nick Cercone
Faculty of Computer Science
Dalhousie University
Halifax, NS B3H 1W5
Canada
nick@cs.dal.ca

Abstract

We summarize and analyze the graph-based approaches to inferring emergent web communities in this paper, especially focusing on the notions (definitions) of web communities used in these approaches. The advantages and disadvantages of these notions as well as associated search algorithms for web communities that can be represented in these notions are discussed, and then a new flexible notion of web communities is proposed. The flexibility of our notion is illustrated by showing that our notion covers most existing notions. With this novel notion, we present an efficient algorithm to find web communities from the WWW and discuss its implementation.

1. Introduction

Finding useful information or patterns efficiently on the World Wide Web (WWW) is becoming more and more important as the WWW grows immensely over time. The search for patterns on the WWW is a task with many facets, and depends largely on the underlying structure of the WWW. The web is a highly dynamic system, and several million new pages are created and added everyday. The creators of web pages come from a variety of backgrounds and have a variety of motives for creating the web contents, and typically link to existing pages on the web that are related to the new page. As a result, the link structure of the WWW contains a considerable amount of information about the relationships among those web pages. Many authors have experimented and claimed that the WWW can be modeled as a graph with each web page (or web site) being a vertex and each hyperlink being an edge connecting two web pages [2, 12]. The research on the network of links in this graph has been focusing on the improvement of web search engines as well as the discovery of useful patterns [1, 3].

One of these useful patterns is Web Community, which is understood as a set of related web pages. Several approaches to finding web communities on the web graph have been developed [5, 6, 7, 10, 11, 15, 17]. The main difference among these approaches is the definition of web communities. Different notions of web communities lead to different search algorithms to find web communities from the web graph. These notions reflect the opinions and needs of the authors. However, most existing notions and associated algorithms discover many web communities that are superfluous.

In this paper, we discuss these existing notions of web communities as well as their associated algorithms and analyze their advantages and disadvantages, and then propose a more flexible notion that covers the most existing definitions of web communities. A novel algorithm for finding web communities based on our new notion of web communities will be presented.

The rest of the paper is organized as follows: In Section 2, the most important existing notions of web communities based on the web graph and associated algorithms are summarized, and their advantages and disadvantages are analyzed. In Section 3, the connectivity structure and communities on the web graph are discussed. We propose a new notion of web communities and discuss the flexibility of our notion by indicating that some existing notions are the special cases of our notion in Section 4. Then, we present an algorithm to find web communities from the WWW in terms of our definition in Section 5. Finally, Section 6 is the conclusion with summary and future work.

2. Existing Methods

For simplicity, we abstract web pages as vertices (or nodes) and links as edges (or arcs) by ignoring the text and other content in web pages. A graph G is simply a set of vertices V and a collection E of pairs of vertices from V, called edges. This graph is denoted as G(V,E). Edges in a graph are either directed or undirected. A subgraph of a graph G(V,E) is a graph G'(V',E') such that $V' \subseteq V$ and $E' \subseteq E$. G'(V',E') is said to a node-induced subgraph of G(V,E) if $E'=\{(u,v) \in E \mid u,v \in V'\}$, that is, E' contains all the edges of E whose both end vertices are from V'. A web community is a subgraph of the web graph that satisfies some conditions.

Several methods based on the link structure of the WWW have been developed to infer emergent web communities. However, what a web community is has not been agreed. The authors of these existing methods propose their own definitions of web communities. Though these methods are designed based on different notions of web communities, the common thread amongst these notions proposed is that they model the web as a graph based on the link structure of the web pages and search for subgraphs of the graph deemed to indicate a web community. We will survey these different notions of web communities and the corresponding methods for determining them and analyze their limitations in this section.

Perhaps the most obvious property of the web graph that would indicate the presence of a community would be a clique. A clique is a complete node-induced subgraph of the web graph, that is, in the node-induced subgraph, there is an edge between any pair of vertices. There are two main barriers to finding cliques in the graph to determine communities [18]. The first barrier is the computation issue, since the problem of finding the maximum clique in a graph is NP-hard, even finding a good approximation to the maximum clique is also hard. The second barrier to using cliques to determine communities is that the connectivity required in a clique is too strong. It would be too much to expect all members of a community to be linked each other.

Therefore, existing methods are focusing on finding a weak property of the web graph to indicate the presence of a web community. Most methods use a bipartite subgraph to indicate a web community. A bipartite core is a graph whose vertices are partitioned to two parts: "hubs" and "authorities", and all edges are directed from "hubs" to "authorities". These methods can also be identified as HITS-based methods, because their implementations are based an algorithm, called HITS (Hyperlink-Induced Topic Search), that was originally proposed to improve web search engines [7, 10].

Another method views a subgraph as a community in which every vertex has more links to vertices within the community than to vertices outside the community. Finding such a community is actually to determine a minimum cut of the network graph.

The above two categories of methods to determine web communities only consider the link structure of the web graph and ignore the content of web pages. Weight link-based methods argue that the content of web pages, the textual similarity between web pages, and the usage statistics of web pages are also important factors for determining communities. Thus, the links are weighted with a function of these factors.

2.1. Highest weighted hubs and authorities as a web community

The first notion of web community stems from the HITS (Hyperlinked-Induced Topic Search) algorithm proposed by Kleinberg [10]. HITS is concerned with the identification of authoritative hypermedia sources for broad-topic information discovery and built on two premises. The first premise is that the implicit annotation provided by human creators of hyperlinks contains sufficient information to infer a notion of "authority"; and, based on the first premise, the second premise is that sufficiently broad topics contain embedded communities of hyperlinked pages. There are two distinct but interrelated types of pages: authorities on the topic that are highly referenced pages, and hubs that are pages that point to many of the authorities and serve as strong centers from which authorities are conferred on relevant pages. Hubs and authorities exhibit what could be called a mutually reinforcing relationship: a good hub points to many of good authorities, and a good authority is linked to by many of good hubs. The HITS algorithm computes an authority weight and a hub weight for each node in the web graph.

The authors of [7] declare the 10 pages with the highest authority weight together with the 10 pages with highest hub weight to be the core of a community. The authors also note that the number 10 is more or less arbitrary and is essentially for a manageable size. Thus, this notion of web community can be defined as follows:

Definition 1 (HITS-notion): A web community can be characterized by the n authority pages with the highest authority weights and the m hub pages with the highest hub weights, where the authority weights and hub weights are calculated by the HITS algorithm and n and m are arbitrary.

The reason that we identify this notion of web community as HITS-notion is that the web community is determined by the outcome of the HITS algorithm. The n highest weighted authorities and the m highest weighted hubs characterize the core of the web community, and the authority weights and hub weights are computed by invoking the HITS algorithm. In the authors' experiment, n = m =10.

2.2. Strongly connected bipartite subgraphs as web communities

Kumar et al [11] argue that web pages that should be part of the same community frequently do not reference one another because the authors (companies) may be competitive, do not share the same point of view, or be not aware of each others' presence. Linkage between related pages can be nevertheless established by repeated co-citation and co-reference, which are originated in the bibliometrics literature. The main idea is that pages that are related are frequently referenced together or cite the same references. The citing pages function as "hubs", while the referenced pages are actually "authorities", corresponding to the concepts in HITS, both of which together form a bipartite graph. The web communities can be determined by the dense directed bipartite subgraphs.

A bipartite graph G(V,E) is a directed graph where the vertex set V can be partitioned into two subsets L (left set) and R (right set). For any directed edge (u,v) $\in$ E, the first vertex u must be in L and the second vertex v must be from R. G(V,E) is complete if E contains all possible edges between a vertex from L and a vertex from R. Figure 1 illustrates two bipartite graphs, where Figure 1 a) shows a general bipartite graph and Figure 1 b) is a complete bipartite graph. Note that edges within the left set and the right set are allowed, which is different from the traditional definition of bipartite graphs in the graph theory.

A bipartite graph is dense, if many of the possible edges between the left set *L* and the right set *R* are present. For any random bipartite graph *G* with the left set *L* and the right set *R*, there exist *i* and *j* such that, with high probability, *G* contains a complete bipartite subgraph with *i* vertices from *L* and *j* vertices from *R*, where *i* and *j* are left unspecified. This (*i*,*j*)-sized complete bipartite subgraph is called the core of *G*, which determines a community.

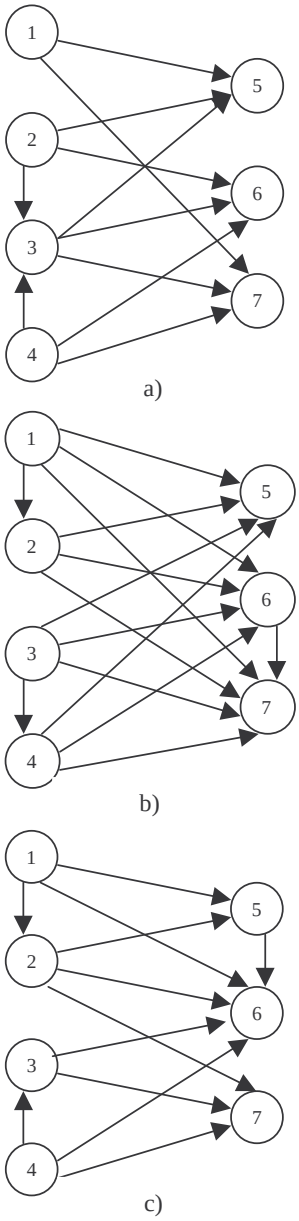


Figure 1. Bipartite graphs, where *L* = {1, 2, 3, 4} and *R* = {5, 6, 7}. a) a bipartite graph, b) a complete bipartite graph, and c) a dense bipartite graph with α=β=2.

Definition 2 (CBG-notion): A (*i*,*j*)-sized complete bipartite subgraph (CBG), with i and j unspecified, of the web graph characterizes the core of a web community.

One can see that this complete bipartite graph CBG-notion of web community is actually stronger than the HITS-notion of web community defined in Definition 1, because it requires all the hubs to point to the same authorities.

The authors further make a hypothesis and show that a random large enough and dense enough bipartite directed subgraph of the web almost surely has a core. Thus, a web community can be found by finding its core and using the core to find the rest of the community, and large numbers of web communities can be found by enumerating all (*i*,*j*)-sized cores in the web graph for small values of *i* and j.

While the Bipartite-notion of web communities with the corresponding trawling algorithm to finding communities scales the HITS-notion of web communities to large graphs such as the entire web graph, it does have some limitations. First, as pointed out in the previous paragraph, the Bipartite-notion requires that the bipartite subgraph be complete, which is too strong to discover some interesting communities. Moreover, the trawling algorithm is based on a vague hypothesis that a random large enough and dense enough bipartite directed subgraph of the web almost surely has a core.

The second limitation of the trawling algorithm is that once a page is detected to be a part of a community, it is eliminated from further consideration. This implies that a page can only belong to one community. However, any web page may in practice participate in multiple communities.

The third limitation of the Bipartite-notion of web community is that it represents a very authoritarian view of the world. If communities are required to contain clear authorities and hubs, then more "democratic" type of communities would be ruled out [18]. For example, a set of vertices in which each vertex is linked to a somewhat random subset of the vertices in the set would not be considered as a community. Especially, considering a directed clique structure (all vertices point to all other vertices), there may be no clear notion of hubs and authorities.

2.3. Densely connected bipartite subgraphs as Web communities

The complete bipartite graph notion of a web community core defined in Definition 2 is too strong to identify some interesting web communities, as pointed out by some researchers [17]. In order to overcome this problem, Reddy and Kitsuregawa [] propose a relaxed notion of a web community, called the dense bipartite graph (DBG) notion.

A bipartite graph *G* with left set *L* and right set *R* is called a **dense** bipartite graph (DBG) if each vertex from *L* has at least α links pointing to the vertices from *R* and each vertex from *R* has at least β links pointing to the vertices from *L*, where $\alpha \leq |L|$ and $\beta \leq |R|$ are the linkage thresholds. Figure 1 c) illustrates a dense bipartite graph with $L=\{1,2,3,4\}$, $R=\{5,6,7\}$, and $\alpha = \beta = 2$. The dense bipartite graph is denoted as $DBG(L,R,\alpha,\beta)$, and can be used to identify the core of a web community. Thus, the DBG-notion of web community is formally defined as follows.

Definition 3 (DBG-notion): The left set *L* of a dense bipartite subgraph $DBG(L,R,\alpha,\beta)$ with unspecified *L* and *R*, and fixed α and β, of the web graph contains the members of a web community.

The DBG-notion of the core of a web community attempts to capture the linkage denseness between the left set (L) and the right set (R) of a bipartite graph and uses a dense enough bipartite graph to identify a web community. With this definition, the authors of [17] present an algorithm to search all potential web communities that contain a dense bipartite subgraph.

One can see from above that any (i, j)-sized complete bipartite graph must be a dense bipartite graph as long as i and j are not less than the corresponding linkage thresholds. Therefore, the authors claim that the outcome of their algorithm includes the outcome of the trawling algorithm as a subset. However, the DBG-notion of web community identifies a web community only using the left set of a dense bipartite graph. This seems to be inconsistent with the original CBG-notion of web community, which uses both left set and right set of a complete bipartite graph to identify the core of a web community.

2.4. The Max-Flow-Min-Cut Notion of Web Community

Flake et al [5, 6] present a different notion of web community from the notions we have discussed so far. Given a set of crawled pages on some topic, a community on the web is defined as a set of web sites that have more links (in either direction) to members of the community than to non-members. Members of such a community can be efficiently identified in a maximum flow/minimum cut framework, where the source is composed of known members and the sink

consists of well-known non-members. Thus, this notion of web community, denoted Cut-notion, is formally defined as follows.

Definition 4 (Cut-notion): In the web graph with vertices set *V*, a web community is a vertex subset *C* $\subset V$, such that $\forall\ v \in C$, *v* has at least as many edges connecting to vertices in *C* as it does to vertices in (*V*-*C*).

Unfortunately, the most generic approaches to solve the balanced minimum-cut problem is NP-complete [5, 6]. To overcome this problem, the authors of [5, 6] proposed an algorithm to find approximate communities.

3. The Connectivity Structure and Communities on the Web

We discussed some existing notions of web community, along with the corresponding algorithms for finding communities on the web in the previous section. We also discussed the weakness of each notion. In this section, we turn to discuss the connectivity (link) structure of the web as well as the relationships between the structure and the community notion. Our purpose is to justify how the web community should be defined and characterized in terms of the properties of the web structure.

The connectivity on the web graph plays an important role in defining web communities. As pointed out in [18], one might consider proposing connected components of the web graph as a natural property to determine communities, that is, finding a set of web pages each of which is reachable from another page along the links. This notion of community, however, would in many cases generate communities of several million web pages. The recent research on the web graph that contains 200 million pages reveals that there is a single strongly connected component at the core of the web that contains about 56 million nodes, while the second largest strongly connected components are small in comparison, at about 100,000 nodes [2]. In addition, there is a large set of approximately 44 million nodes that have paths leading into the strongly connected central core component, and another set of roughly the same size whose nodes are pointed to by the strongly connected central core. Verbeurgt also analyzes another two notions of connectivity on the web graph: biconnectivity and alternating connectivity [18]. The biconnectivity applies to the undirected graph, and is defined as follows: two nodes *x* and *y* are biconnected, if there is no third node *z* such that *z* is on every path between *x* and *y*. Under this definition, the authors find that the web graph contains a "giant biconnected component" that generally contains all of the top hubs and authorities computed by the HITS algorithm. On the other hand, an alternating connectivity is defined as follows: two nodes *x* and *y* are alternating connected if there is an alternating path from *x* to *y*. An alternating path is a path where the directions of links strictly alternate between forward and backward. This connectivity is significantly related to the concepts of hubs and authorities.

Using these existing notions of connectivity to define web communities leads, more or less, to the notion of hub-authority community. The weakness of these notions used as the definitions of web communities is that a web community may contain too many web pages. On the other hand, defining a web community as a clique, a completely connected component of the web graph, is too strong that there are very few communities on the web. Moreover, finding the maximum clique is NP-hard. Thus, we expect a new notion to define web communities such that each web community represents a reasonable fraction of web pages, and the search for these communities from the web graph is not too hard.

4. A New Notion of Web Community

We propose a new notion of web community based on the connectivity structure of the web graph in this section. Our notion is motivated by the p-quasi complete graph proposed in [15].

Definition 5 (**p-quasi complete graph**): A *p*-quasi complete graph *QC = (V,E)* is an undirected graph such that deg(*v*) >= [p*(|V|- 1)], for all *v* $\in$ *V, p is the connectivity (0<=p<=1)*.

One can easily verify that if *p*=1, then the *p*-quasi (1-quasi) complete graph is a traditional complete graph; and all graphs are a *p*-quasi complete graph with $p = 0$ (0-quasi complete graph). Figure 2 shows a *p*-quasi complete graph with $p = 0.5$.

Our goal is to propose a new notion of web community based on the web graph structure. This notion should be neither too strong to cover most useful web communities, such as those that can be identified or characterized by the notions of web community defined in Definitions 2 through 4, and nor too weak such that many superficial web groups are included. To this end, we use a p-quasi complete graph to characterize a web community. For simplicity, we do not include the web page contents at this stage, and only consider the link structure of the web graph.

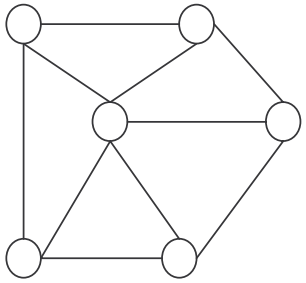


Figure 2. A p-quasi complete graph with p=0.5

Definition 5: A n-sized p-quasi complete graph, denoted QCG(n,p), is a p-quasi complete graph with n vertices.

Clearly, Figure 2 illustrates a 6-sized 0.5-quasi complete graph. A n-sized p-quasi complete subgraph can be used to indicate the core of a web community.

Definition 6 (QCG-notion): A n-sized p-quasi complete subgraph of the web graph identifies a web community.

Proposition 1: The web communities that can be characterized by the CBG-notion can also be identified by the QCG-notion.

Justification: By Definition 2, the core of a web community that can be characterized by the CBG-notion is a complete bipartite graph. Assume the core is an (i,j)-sized complete bipartite graph. Let n=i+j, and $p \le \frac{\min(i, j)}{i + j - 1}$, then the (i,j)-sized complete bipartite graph is a QCG(n,p). That is, the core of a web community characterized by the CBG-notion is actually a QCG(n, p). QED.

Proposition 2: The web communities that can be characterized by the DBG-notion can also be identified by the QCG-notion.

Justification: The members of a web community that is identified by the DBG-notion compose a dense bipartite graph DBG(L,R,α,β), by Definition 3.

Let n = |L|+|R|, and $p \le \frac{\min(\alpha, \beta)}{|L| + |R| - 1}$, then DBG(L,R,α,β) is a QCG(n,p). That is, the web communities that can be characterized by the DBG-notion can also be identified by the QCG-notion. QED.

Proposition 3: The web communities that can be characterized by the Cut-notion can also be identified by the QCG-notion.

Justification: By Definition 4, a web community defined by the Cut-notion is a minimum cut, C, whose each vertex has more links to other vertices inside the cut than to vertices outside the cut. Assume the number of vertices inside the cut is M, and the minimum degree of vertices inside the cut is m, that is M = |C|, and m = min{deg(v) | v∈C}. Let n=M, and $p \le \frac{m}{M - 1}$, then the minimum cut C is a QCG(n,p). Thus, the web community that is represented as a minimum is actually a QCG(n,p). QED.

From Propositions 1 through 3, one can see that our new notion, QCG-notion, of web communities covers the web communities that are identified by CBG-notion, DBG-notion, and Cut-notion. Therefore, any algorithm that can find n-sized p-quasi complete subgraphs in the web graph will find all web communities that can be found by the algorithms that search for web communities based on the CBG-notion, DBG-notion and Cut-notion.

5. An Algorithm for finding web communities

Our notion of web communities based on p-quasi complete graph is very flexible to represent diverse communities by choosing connectivity *p*. For large *p* values, the identified communities are strongly connected, while for small *p* values, the communities are loosely coupled.

In order to find web communities from the WWW, we should be able to efficiently find p-quasi complete subgraphs of the web graph. In this section, we present our graph-based neighborhood approximation approach to find p-quasi complete subgraphs.

We have proposed two different approaches to manipulate web graphs. One approach is to approximate the complete "neighborhood function" for the graph, similar to ANF [16], which is suitable for an extremely large web graph. The other approach is to represent the graph as an adjacency matrix, which can be treated as a set of bitmap indexes, and compress those bitmap indexes based on bitmap compression techniques [8, 9]. Then, the compressed bitmap indexes can be loaded into memory for efficient vector calculation, which is suitable for a medium or large network graph. In this paper we focus on the first approach.

To discover p-quasi complete subgraphs from the web graph is to identify the central most densely connected subgraphs. Matsuda et al [15] proved that finding the 0.5-quasi complete graph is NP-complete. The 1-quasi complete graph problem is identical to the CLIQUE problem that is also NP-complete. But it is unknown whether the general p-quasi complete

graph problem is NP-complete or not. Consequently we will develop some approximation algorithm for this problem. Our approach relies on the fast and high accurate methods provided in the ANF algorithm [16]. In order to find the p-quasi complete graph, we start the trawling from the potential hub vertices, which are the highly connected vertices in the web graphs. Then we search around the neighbor vertices within certain distance h from the hub vertices for the p-quasi complete graph. Our method is a heuristic approach and cannot guarantee to find all the p-quasi complete graphs but we believe it will find many p-quasi complete graphs in an efficient and effective way. Before we present the algorithm in a formal way, some definitions borrowed from [4, 16] are reintroduced below to make the paper self-content.

Definition 7: The individual neighborhood function for vertex u at distance h is the number of nodes at distance h or less from u. IN(u,h)= |{v: v ∈ V, dist(u,v) <= h}|.

Definition 8: The neighborhood function at h in web graph G is the number of pairs of nodes within distance h:

N(H) = |{u,v} : u∈ V , v ∈ V, dist (u,v) <=h| or

N(h) = $\sum_{u \in V}$ IN(u,h).

Previously Lipton and Naughton [13] presented an $O(n\sqrt{m})$ algorithm for estimating the transitive closure that uses an adaptive sampling approach for selecting starting nodes of breadth-first traversal, a graph traversal effective accesses the edge file in random order but it is shown experimentally that their approach does scale to the large web graph. When web graphs are too large to be processed effectively in main memory, we cannot perform any graph traversal. ANF makes it possible to deal with this problem that would have been at least infeasible, if not impossible before. With the highly accurate and efficient approximation tools from ANF, it is feasible to implement our heuristic algorithm to discover the p-quasi complete graph from the large web graph. In order to find the p-quasi complete graph of size n, it first calculates the neighborhood function N(h) of each node in the web graph, where h is set to $\log\log n/k$ (based on the experimentally study of large scale-free network graph, the diameters of such graph is in proportion to $\log\log n$ with n vertices). Then it filters out those nodes whose neighbor is not dense enough, which significantly reduce the search space for the potential p-quasi complete graph. At the last step, for each hub and its associated neighbor vertices, eliminate those vertices whose degree is less than p*n.

Our proposed algorithm is described in the following algorithm.

Algorithm: Finding web communities of size n
Input: G: web graph
P: connectivity ratio
n: number of vertices in the p-quasi complete graph
k: some constant factor
Output: A set of web communities

Step 1: Set h = *loglogn/k*
Step 2: Calculate the N(h) in G
Step 3: Remove those nodes v where N(v,h) is less than n, and put the remaining vertices in the hub queue HQ
Step 4: For u ∈ HQ, remove those vertices in IN(u,h) whose degree less than p*n.
Step 5: Check each remaining N(u,h) to remove those non p-quais complete graphs

6. Conclusion and Future Work

In this paper, we summarized and analyzed the graph-based approaches to inferring emergent web communities, especially focusing on the notions (definitions) of web communities used in these approaches. We discussed the advantages and disadvantages of these notions as well as associated algorithms of searching for web communities that can be represented in these notions. Unfortunately, most existing approaches to finding web communities from the web graph generate many web communities that are superfluous, and finding strong web communities such as CLICQUEs is hard.

We proposed a new flexible notion of web communities. In our definition, a web community is indicated by a n-sized p-quasi complete subgraph. We illustrate the flexibility of our notion by showing that our definition covers most existing notions. We also present an efficient algorithm to find web communities from the WWW based on our definition and discuss its implementation.

We haven't implemented our notion of web communities and associated search algorithm, which will be one of our future works. We will also consider the inclusion of web page contents into the web community definition.

10. References

[1] S. Brin and L. Page. *The Anatomy of a large-scale hypertextual web search engine*. Computer Networks and ISDN Systems, 30(1-7):107-117, 1998.

[2] A. Broder,R. Kumar, F. Maghoul, P. Raghavan and R. Stata., *Graph structure in the web*. In proceedings of the 9th International World Wide Web Conference, pp 247-256, 2000

[3] D. Cook and L. Holder, *Graph-Based Data Mining*, IEEE Intelligent Systems 15(2):32-41, 2000.

[4] J. P. Eckmann and E. Moses. Curvature of Co-links uncovers hidden thematic layers in the World Wide Web, Proc. Natl. Acad. Sci. USA 2002, 99:5825-5829.

[5] G. W. Flake, S. Lawrence, C. L. Giles, Efficient identification of web communities, Proc. of KDD, 150-160, 2000

[6] G. W. Flake, S. Lawrence, C. L. Giles, F. M. Coetzee, Self-organization and identification of web communities, IEEE Computers, 35(3):66-71, March 2002.

[7] D. Gibson, J. M. Kleinberg, P. Raghavan, Inferring Web Communities from Link Topology, {UK} Conference on Hypertext, 225-234, 1998

[8] X. Hu and J. Han, *Discovering clusters from large scale-free network graph*, in the ACM SIGKDD 2nd Fractals, Power Laws and Other Next Generation Data Mining Tools Workshop, Washington, DC, August, 2003.

[9] X. Hu, T.Y. Lin and E. Louie, *Bitmap techniques for optimizing support queries and association rule algorithms*, In the Proc. of the 2003 International Databases Engineering and Application Symposium, Hongkong, July 17-23, 2003

[10] J. M. Kleinberg. *Authoritative Sources in a Hyperlinked Environment*, Journal of the ACM, 604-632, 1999.

[11] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, Trawling the Web for emerging cyber-communities, Computer Netweorks, 31(11-16):1481-1493, 1999

[12] S.R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins and E. Upfal. The Web as a graph. In ACM SIGMOD-SIFACT-SIGART Symposium on Principles of Databases Systems, pp 1-10, 2000

[13] R.J. Lipton and J.F. Naughton, *Estimating the size of generalized transitive closures*. In Prod. of 15th VLDB, pp 315-326, 1989

[14] N. Matsumura, Y. Ohsawa, and M. Ishizuka. *Future directions of communities on the web*. First International Workshop on Chance Discovery, 17—20, 2001.

[15] H. Matsuda, T. Ishihara, A. Hashimoto, Classifying molecular sequences using a linkage graph with their pairwise similarities, Theoretic Computer Science 210: 305-325, Elsevier, 1999.

[16] C. R. Palmer, P.B. Gibbons and C. Faloutsos, *ANF: A fast and scalable tool for data mining in massive graphs*, in Proc. of SIGKDD 2002, Edmonton, Canada.

[17] P. K. Reddy, M. Kitsuregawa, Inferring web communities through relaxed cocitation and dense bipartite graphs, Proc. Of Data Engineering Workshop, 2001.

[18] K. Verbeurgt, Inferring Emergent Web Communities, International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, e-Medicine, and Mobile Technologies on the Internet, SSGRR 2003W, L'Aquila, Italy, 2003.

Collaborative Information Delivery: Issues of design and implementation

Samuel Kaspi
School of Information Systems
Victoria University
Melbourne, Victoria, Australia
Email: samuel.kaspi@vu.edu.au

Wei Dai
School of Information Systems
Victoria University
Melbourne, Victoria, Australia
Email: wei.dai@vu.edu.au

ABSTRACT

This paper investigates and proposes solutions framework for integrating and leveraging existing information sources through web-based decision support. Software architectures for enabling the information service delivery have been studied, particularly from their design and implementation perspectives. The potential of the web-based solutions framework has been analysed at practical levels involving the modern technologies of data communications, SOAP, XML and SUN's J2EE.

1. Introduction

The use of web-based information delivery service is driven by users needs. Based on the user profile information, data and information services can be delivered and presented intelligently. For example, appropriate levels of service features (available within the software products) are presented to the users based according to their sophistication and skills. This will generate a user oriented decision-making viewpoint dynamically at any stage of the interactive request-delivery process. The intended use of the data influenced by user profiles can assist in determining how the data can best be stored and presented, and serving users effectively. The degree of data access and connectivity that such a system would require can be best delivered by the Web, hence our decision to focus on the development of a web-based decision support system. The above considerations lead us to explore possibilities on the effective use of the Web's unique potential to meet the new requirements. The information service delivery can be effectively utilised to serve various business needs, such as in a decision-making process, once the associated knowledge is elicited, managed and applied.

2. Architecture Overview

To meet the fundamental requirements, a general architecture has been proposed which is based on the previous work [1] and described in Figure 1. The framework has been designed to interact with a range of the information resources such as database systems, Enterprise Information Systems (EIS) and legacy systems via the web in order to deliver user specific solutions. Users requests are generated through a client application (normally within a user's work environment) e.g. a client User Interface (UI), as shown in figure 1. The request is transmitted and received by the framework Interface Manager that activates the services offered by the selected back-end components.

The knowledge Management component offers the framework the capability of interacting with the client applications/request intelligently as well as directing client tasks accurately to the responsible targets across the network. The Data Management component will ensure that the framework is capable of dealing with a wide range of external database systems regardless of their locations or formats. The integration of knowledge management and data management delivers the framework the power of processing large volume of information in a real-time manner. Having a powerful business logic processing capability supported by all the required conventional computing services such as data management services, opens up a new solution perspective to real-time transactions delivery via the Web. The framework is based on a java implementation environment.

An important aspect of our framework is its ability to connect geographically dispersed systems. This is illustrated by figure 2. In figure 2, both the organization's servers and users are geographically distributed. This dispersion can be in either fixed or mobile. For example a laptop on an airplane can access the system as shown by *User 4* in Fig 2. As well, authorized users from outside the organization can access those parts of the system to which they have authorization. Similarly, the web based decision support system itself may reside on one sever and access the other parts as required or it may be distributed over several or all of the servers. However this geographical distribution is transparent to the user to whom the system appears logically the same.

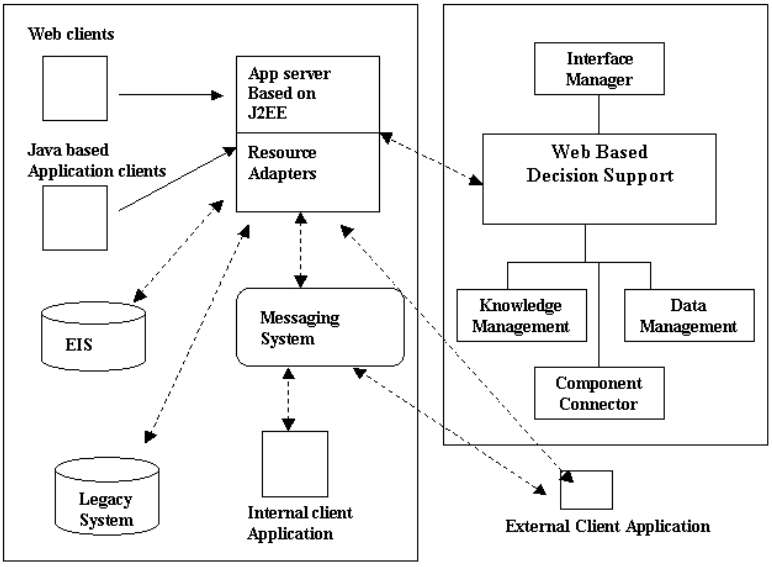


Figure 1: An overview of the composition of the Web-based Decision Support System (source) [6]

Our framework also caters for a federation of organizations of the type depicted in Figures 2. That is, each organization in the federation may have its own geographically dispersed system with each organization's system linked to the other organizations in a federation. In this federation, the federal web based decision support system may consist of each organization's decision support system linked together in a federation or it may be a decision support system designed for the federation and distributed over the servers appropriate for the situation. This ability to connect geographically dispersed systems is facilitated by our framework's integration with the web. Where there is a need for communication over geographically dispersed locations, the web offers extremely significant advantages – it is cost effective, its geographical coverage is almost omnipresent, support tools and technologies are ubiquitous and use of the browser is almost universally understood and accepted.

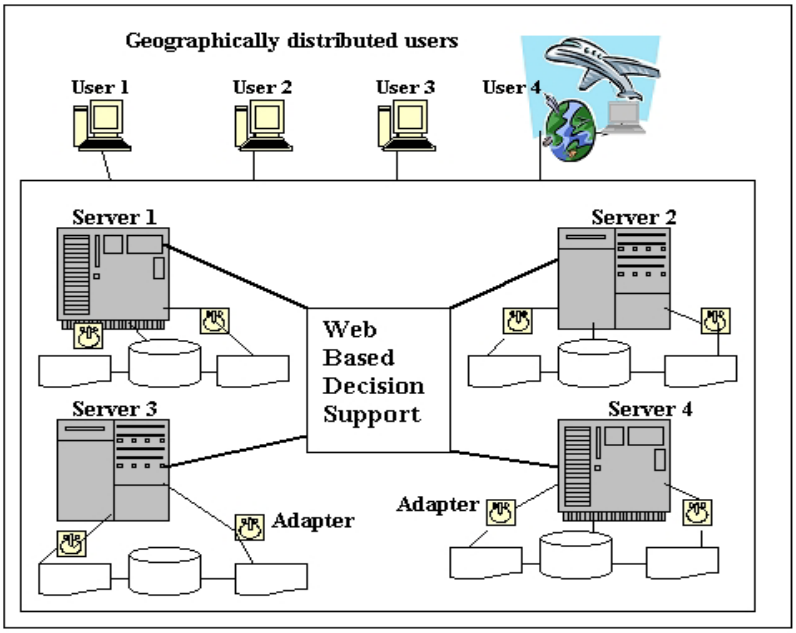


Figure 2: Distributed servers and distributed users

3. Related Work

The demand for integration has led to the development of Enterprise Resource Planning (ERP) and Enterprise Application Integration (EAI) systems. An overview of ERP and EAI and their technical challenges can be found in [2] and [3]. To address the integration challenges, Web Services proposes to address the issues of data and systems interoperability as well as process integration. However, technical obstacles remain; mainly in the areas of transaction processing and real-time delivery of information services and these obstacles can directly affect the success of e-business. To tackle these challenges, a knowledge driven Web Services delivery framework was proposed in [4]. However, the work was focussing more on the conceptual issues with little implementation detail supplied. In this paper, we expand previous work by giving a software development perspective, discussing the issues and alternatives required in delivering the proposed framework at the practical level.

4. Deployment Considerations

We have selected SUN's J2EE as the candidate technology to implement the above configurations. Access to EIS's and legacy systems is to be via adapters using custom sockets, java's JMS technology, custom access libraries or J2EE's connector interface as appropriate. An overview of the technologies to be used by our framework is shown in Figure 1. As is shown in Figure 1, our decision support system (individual, distributed or federated) is composed of four parts, an interface manager, a knowledge manager, a data manager, and a component connector. These sit in a J2EE application server. While initially our web based decision support system is to be implemented using J2ee, future versions may be able to integrate decision support systems written under various languages. This integration could be achieved by using CORBA orbs or using other connector technologies as they become available (it is anticipated that there will soon be connectors to bridge EJB and Microsoft's COM [5].

Another pillar of our framework is XML, which is to be the language used to represent data and knowledge in our framework. The choice of XML is natural since it is capable of rich expression, is almost universally accepted and used, is well supported by Java technology and is also supported by a myriad of tools. Naturally, each server in any organization may also be a client of another server in the same organization or another server in another organization in the federation.

5. Data Communications: SOAP and XML

One of J2EE's underlying technologies is RMI, which is the native J2EE/Java technology for implementing remote procedure (RPC) calls. Both Entity and Session beans implement the remote interface and as such RPC via RMI is natural and relatively easy to implement in a J2ee implementation. An alternative to RPC is messaging where instead of a client invoking a method on a server, the client sends a message to the server and receives a response. Messaging has some advantages over RMI- it is largely language and platform independent and so messaging implemented in the Java/J2ee environment by using Java's JMS technology could send a message from a Java program to a COBOL program running on a mainframe. As well, messaging is better suited for communication between systems operating on disparate time frames for example where a batch system communicates with a real time system. A third method of implementing remote communication is to implement RPCs' or messaging via the SOAP protocol (the ability to integrate messaging, SOAP and J2EE is quite recent and is enabled by the SOAP with Attachments API for Java (SAAJ) [7]). SOAP uses XML to encode requests and results and works naturally with common internet transport protocols such as HTTP, SMTP and POP3. As well, unlike RMI, SOAP is not a Java only solution and thus its use makes it easier to expand a system to include non–Java applications. A nice SOAP implementation is available from Apache at http://xml.apache.org/soap.

Despite SOAP's relative slowness, its advantages as an internet protocol using XML make it our mechanism of choice for implementing communication particularly between organizations. The use of RMI would be restricted to local intra-organization communication. Similarly in general, the use of JMS would be restricted to local intra-organization communication. However as discussed below, JMS could also have a role in connecting legacy systems to our framework. The role of the Messaging System module would be to control these communications and protocols.

6. The Connector Architecture

As indicated earlier our framework allows for connection to both EIS' and legacy systems. As well, as indicated earlier the options for achieving this connection include writing custom sockets and

terminal emulations, using a custom library or using J2EE's recently developed Connector Architecture. A detailed specification of the connector architecture can be found in [6] with a more general overview provided in [7] and [8]. Where possible, using the connector architecture is our preferred method for connection to both EIS's and legacy systems. The Connector Architecture has the advantage that it encourages EIS vendors to develop adapters since once developed it can be used by any application server that supports the connector architecture. This should ensure that a wide variety of adapters should soon be available. Indeed most of the major EIS vendors already supply or have indicated a willingness to supply resource adapters for the Connector architecture (the list of contributors to the development of the connector specification JSR 112 include BEA, Bull, Ericsson Infotech, Fujitsu Limited, Hewlett-Packard, IBM, Bahwan Cybertek, Inprise, IONA, MicroFocus, NEON Systems, Inc., Oracle, SAP AG, Siemens, Silverstream Software, Softwired AG, Sun Microsystems, Sybase, TIBCO Software Inc, Unisys, WebMethods Corp [6].

As well, many mainframe systems support various type of messaging systems, which can be accessed via JMS. The Connector architecture treats the JMS API provider as a resource adapter with the fact that it is JMS being transparent to end-users. Thus, a large number of legacy systems can also be accessed via the connector architecture.

7. Common Client Interface (CCI)

From the EJB developer's point of view, the most important part of the Connector API is the Common Client Interface (CCI) which is analogous to JDBC in that just as JDBC provides a standard way of making connections to DBMS' and creating statements to access the DBMS and retrieving results from the DBMS, so CCI provides a standard way to connect to EIS and to send and receive data from them. Like JDBC, CCI is relatively easy to use as indicated in Figure 7 below. As well, especially important from our framework's point of view, CCI supports XML thus allowing interaction with EIS using XML. Where connector adapters don't exist the use of custom libraries such as IBM's gateway for its CICS system is preferred over the writing of custom sockets. The writing of custom sockets is avoided where possible since mainframe applications are notoriously unfriendly to Unix, Linux or windows applications.

8. Knowledge Management Facilities

The knowledge management module contains the domain knowledge expressed as XML documents. This knowledge representation could encompass knowledge expressed as rules, cases, frames, and scripts etc. all of which are amenable to representation in XML. The inferencing mechanisms which process the knowledge would also reside in this module and would naturally have to be able to process XML. The parsing of documents would be done with either J2EE's DOM parser or J2EE's SAX parser as appropriate. The trade off is ease of use versus speed with DOM being easier to use and SAX being faster. In general for the processing of complex knowledge DOM would be preferred since a SAX implementation would require tortuous navigation. An alternative to DOM or SAX is the use of technologies that use java-specific XML tags such as JOX or KBML. However we forgo these technologies since it is unlikely that you would find many partners willing to use java-specific tags particularly if you later want to extend the system to non-java users.

9. Support for Interoperability

The use of EJB in data management is well understood and our framework does not add anything unusual to the field. As is standard practice, we use entity beans to represent persistent objects and session beans to implement the business logic and interact with the session beans. We prefer that in the entity beans persistence be bean-managed (BMP) rather than container managed (CMP). This is because while writing CMP code is easier than writing BMP; the quality of the CMP is directly dependant on the persistence manager implementation. As well, communication problems may arise in a federated system, with different partners using different CMPs'.

The framework is likely to contain a very large number of beans (components). Determining which combination of components is required to satisfy a particular application is the role of component connector in our system. This part of our framework will make use Java's reflection package and factory pattern to dynamically load classes as required. Figure 8 below, gives an example of an implementation that accepts a class name as an argument and then runs it.

10. Interface Manager

The Interface Manager module will, as its name implies manage interfaces between the users and

the system. As would be expected for a web-based system, this part of our framework uses JSP and XSL. XSL of course is a language that is used to translate XML from one format to another and can be used to process XML data and render a web page. "A *JSP page* is a text document that contains two types of text: static template data, which can be expressed in any text-based format, such as HTML, SVG, WML, and XML, and JSP elements, which construct dynamic content [7]. That is JSP can render a web page and in addition can also fetch data which XSL cannot. However to render a document expressed in XML, JSP needs to apply an XSL style sheet to the XML document. Thus for an XML based system such as ours, the two technologies are complementary.

11. Evaluation and Applications

So far, this paper has discussed implementation issues in relation to our frameworks components. Here we would like to briefly discuss the current state of the framework and to take another look from application perspectives. Because of our framework's flexible knowledge handling capability, the ease of incorporating problem-solving information into our framework's components and its adequate data management support, software components can be re-configured or re-built on demand to suit different application needs. One example use of this framework capability is the deployment of multi-agent team to support knowledge management infrastructure [9]. These agents can be tailored to perform the designated tasks across the web driven by data communication facilities and agent knowledge bases that are supported by the framework. The knowledge management component of the framework has been delivered to the computer labs for teaching purposes. At the present stage, we are focussing more on the J2EE support of our implementation tasks. Preliminary study on Microsoft .NET connection and implementation strategy is currently underway.

12. Summary and Conclusion

In this paper, we have proposed web-based decision support architecture and outlined its general aims and components functionality. We examined the framework specifically from implementation perspectives, aiming to deliver cost-effective and robust solutions. The concepts of SUN's J2EE and its facilities [10] were extensively borrowed and used within the framework. We aim to use this paper to help practitioners such as application developers to focus on the appropriate software development issues. The framework has been designed to work with existing information resources (such as database systems and software application packages). Currently, several lightweight client applications are being developed to support the demands coming from different domain areas. The framework is gearing up for web-based decision support with industrial strength.

13. References:

[1] W. Dai and S. L. Wright. 1996. Strategies for integrating knowledge-based system techniques within conventional software environments. International Journal of Intelligent Systems, Vol. 11, No.11, 989-1011, John Wiley & Sons.

[2] W. Hasselbring. 2000. Information System Integration. Communications of ACM. Vol. 43. No. 6. June. Pp 33- 38.

[3] Lee, J., Siau, K., and Hong, S. 2003. Enterprise Integration with ERP and EAI. Communications of ACM. Vol. 46. No. 2. February. Pp 54-60.

[4] Dai, W. Shen, N. Hawking, P. 2003. A Knowledge Driven Approach in Leveraging Web Services Delivery. In proceedings of the 2003 IEEE Int'l Conf. On Information Reuse and Integration. Oct., Las Vegas, USA (to appear).

[5] Wutka, M. 2001. Special Edition Using Java™ 2 Enterprise Edition. Que. USA.

[6] Sun Corporation. 2003. J2EE Connector Architecture Specification Version: 1.5, Page 27. URL: icd.ics.purdue.edu/~bseib/training/j2ee_specs/j2ee_connector-1_5-pfd2-spec.pdf.

[7] Sun Corporation. 2003. The J2EE™ 1.4 Tutorial. May. URL: http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html

[8] Ng, T. 2001. J2EE™ Connector Architecture: Overview and Roadmap. JavaONE Developers Conference. URL: http://java.sun.com/javaone/jp2001/pdfs/517.pdf.

[9] Dai, W., Rubin, S. H. and Chen, C. 2003. Supporting Knowledge Management Infrastructure: A Multi-Agent Approach. Proceedings of the 2003 Int'l Conf. On Intelligent Agent Technology, IAT'2003. Halifax, Canada. Oct. (to appear).

[10] Shin, S. 2003. J2EE™ Overview. Sun Microsystems, Inc., Oct.

Quantitative Analysis of the Difference between the Algebra View and Information View of Rough Set Theory

J.J. An, L. Chen, G.Y. Wang Y. Wu
Institute of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, 400065, P. R. China
crssc@cqupt.edu.cn wanggy@cqupt.edu.cn

Abstract

The attribute core of a decision table is often the start point and key of many decision information system reduction procedures based on rough set theory. The algebra view and information view are two main views and methods of rough set theory. In this paper, based on the problem of calculating the attribute core of a decision table, we will study the relationship between the algebra view and information view. Through simulation experiment, we find the quantitative difference of the attribute core of a decision table in both views of rough set theory. Especially, we find that their difference will be maximized in decision tables containing much inconsistent information.

1. Introduction

Rough set theory has been applied in such fields as machine learning, data mining, etc., since Professor Z. Pawlak developed it in 1982 [1, 2]. Reduction of decision table is one of the key problems of rough set theory. The attribute core of a decision table is always the start point of information reduction. There are two main views and methods about rough set theory, that is, the algebra view and information view [3, 4, 5, 6]. In this paper, we will study the problem of calculating the attribute core of a decision table. Based on former research results on this problem in the algebra view and information view of rough set theory [7-11], we will further study their relationship. Our simulation experiments give a quantitative answer of the difference between these two views of rough set theory. Especially, we find that there will be great difference of the attribute cores of an inconsistent decision table in these two views. This result is much useful for uncertain information system reduction

.

2. Basic Concepts

For the convenience of discussion, we introduce some basic notions about attribute reduction and attribute core of rough set theory at first.

Definition.1 An information system is defined as S=<U, R, V, f >, where U is a finite set of objects and R=C∪D is a finite set of attributes. C is the condition attribute set and D is the decision attribute set, $V=\cup V_a$ is a union of the domain of each attribute of R. Each attribute has an information function $f:U\times R\rightarrow V$.

Definition.2 Given an information system S=<U, R, V, f>, let X denote a subset of elements of the universe U($X\subseteq U$). The lower approximation of X in B($B\subseteq R$) is defined as $B_{-}(X)$, the upper approximation of X in B is defined as $B^{-}(X)$,

$$B_{-}(X)=\cup\{Y_i|(Y_i\in U|IND(B)\wedge Y_i\subseteq X)\},$$
$$B^{-}(X)=\cup\{Y_i|(Y_i\in U|IND(B)\wedge Y_i\cap X\neq\varnothing)\},$$

where, $U|IND(B)=\{X|(X\subseteq U\wedge\forall_x\forall_y\forall_b(b(x)=b(y)))\}$.

Definition.3 Given an information system S=<U, R, V, f> and an attribute $r\in P(P\subseteq C)$, if $IND(P-\{r\})=IND(P)$, r is said to be dispensable, otherwise indispensable.

Definition.4 Given an information system S=<U, R, V, f>, the C-positive region $POS_C(D)$ is $POS_C(D)=\{x\in U:\exists X[X\in U/IND(D)\wedge x\in C_{-}(X)]\}$.

Definition.5 Given an information system S=<U, R, V, f>, an attribute set $T\subseteq P(P\subseteq R)$ is a relative reduction of P with reference to an attribute set Q if T is relatively orthogonal with reference to the attribute set Q and $POS_T(Q)=POS_P(Q)$.

We use the term $RED_Q(P)$ to denote the family of relative reductions of P with reference to Q.

Definition.6 $CORE_Q(P)=\cap RED_Q(P)$ is called the Q-core of P.

3. Attribute Core in the Algebra View

Hu developed a method to calculate the attribute core of a decision table based on Skowron's discernibility matrix [7].

Definition.7 For a set of attributes $B\subseteq C$ in a decision table $S=(U, C\cup D, V, f)$, the discernibility matrix can be defined by $C_D(B)=\{C_D(i, j)\}_{n\times n}$, $1\le i, j\le n=|U/IND(B)|$,

where

$$C_D(i,j)=\begin{cases}\{a_k \mid a_k\in P \mid a_k(X_i)\neq a_k(X_j), X_i, X_j\in U\} & d(X_i)\neq d(X_j)\\ 0 & d(X_i)=d(X_j)\end{cases}$$

for i, j=1, 2, …, n.

Hu drew the following conclusion in [7]: $|C_D(i, j)|=1$ if and only if the attribute in it belongs to $CORE_D(C)$. This conclusion is used in many later documents.

Ye proved Hu's conclusion is not true in inconsistent decision tables by giving a counterexample. He developed an improved method to calculate the attribute core through improving the definition of discernibility matrix in the following way.

Definition.8 For a set of attributes $B\subseteq C$ in a decision table $S=(U, C\cup D, V, f)$, the discernibility matrix can be defined by $C'_D(B)=\{C'_D(i, j)\}_{n\times n}$, $1\le i, j\le n=|U/IND(B)|$,

where $C'_D(i,j)=\begin{cases}C_D(i,j) & ,\min\{|D(x_i)|,|D(x_j)|\}=1\\ \varnothing & ,\text{else}\end{cases}$

for i, j=1, 2, …, n, $|D(x_i)|=|\{d_y|y\in[x_i]_C\}|$.

In [8], Ye drew another conclusion: $|C'_D(i, j)|=1$ if and only if the attribute in it belongs to $CORE_D(C)$. Through comparing the attribute cores of the algebra view and information view, we find that Ye's method could be used to calculate the attribute core of a decision table in the algebra view of rough set theory [9].

4. Attribute Core in the Information View

For the convenience of later discussion, we introduce some basic concepts and theorems about the information view of rough set theory here [3].

Definition.9 Given an information system $S=<U, C\cup D, V, f>$, and a partition of U with classes X_i, $1\le i\le n$. The entropy of attributes $B(B\subseteq C\cup D)$ is defined as $H(B)=-\sum_{i=1}^{n}p(X_i)\log(p(X_i))$, where, $p(X_i)=|X_i|/|U|$.

Definition.10 Given an information system $S=<U, C\cup D, V, f>$, the conditional entropy of D $(U/IND(D)=\{Y_1, Y_2, \ldots, Y_m\})$ given $B\subseteq C$ $(U/IND(B)=\{X_1, X_2, \ldots, X_n\})$ is $H(D|B)=-\sum_{i=1}^{n}p(X_i)\sum_{j=1}^{m}p(Y_j|X_i)\log(p(Y_j|X_i))$, where $p(Y_j|X_i)=|Y_j\cap X_i|/|X_i|$, $1\le i\le n$, $1\le j\le m$.

The following 3 theorems are proved in [4].

Theorem 1 Given a relatively consistent decision table $S=(U, C\cup D, V, f)$, an attribute $r\in C$ is relatively reducible if and only if $H(D|C)=H(D|C-\{r\})$.

Theorem 2 Given a relatively consistent decision table $S=(U, C\cup D, V, f)$, the attribute set C is relatively independent if and only if $H(D|C)\neq H(D|C-\{r\})$ for all $r\in C$.

Theorem 3 Given a relatively consistent decision table $S=(U, C\cup D, V, f)$, attribute set $B\subseteq C$ is a relatively reduct of condition attribute set C if and only if $H(D|B)=H(D|C)$, and the attribute set B is relatively independent.

Definition.11 Given a decision table $S=(U, C\cup D, V, f)$, attribute set $B\subseteq C$ is a relatively reduct of condition attribute set C if and only if

(1) $H(D|B)=H(D|C)$, and

(2) $H(\{d\}|B)\neq H(\{d\}|B-\{r\})$ for any attribute $r\in B$.

Definition.12 $CORE_Q(P)=\cap RED_Q(P)$ is called the Q-core of attribute set P.

The attribute core of an inconsistent decision table in the information view can be calculated with the following theorem and algorithm.

Theorem 4 Given a decision table $S=(U, C\cup D, V, f)$, where $D=\{d\}$, attribute $r\in C$ is a core attribute if and only if $H(\{d\}|C)<H(\{d\}|C-\{r\})$.

In [9], Wang developed the following algorithm for calculating the attribute core of a decision table in the information view of rough set theory.

Algorithm 1:

Input: A decision table $S=(U, R, V, f)$.

Output: The attribute core of S in the information view, $CORE_D(C)$.

Step 1. $CORE_D(C)=\phi$.

Step 2. For each condition attribute r in C, do
If $H(\{d\}|C)<H(\{d\}|C-\{r\})$, then
$CORE_D(C)=CORE_D(C)\cup\{r\}$.

Step 3. Stop.

5. Difference of the Attribute Core of a Decision Table in the Algebra View and Information View

In the algebra view of rough set theory, a condition attribute is reducible if and only if the lower approximation of at least one decision class of the decision table will be changed after deleting it. That is, a condition attribute is reducible if and only if the

consistent part of the decision table will be changed after deleting it.

In the information view of rough set theory, a condition attribute is reducible if and only if the conditional entropy of the decision table will be changed after deleting it. However, the conditional entropy of the consistent part of a decision table is always 0. All conditional entropy of a decision table results from its inconsistent part. Thus, a condition attribute should be reducible in the information view if and only if the probability distribution of the whole decision table will not be changed after deleting it.

Let the attribute core of a decision table in the algebra view is $CORE_1$, and $CORE_2$ in the information view, we can draw the following conclusions [9]:

（1） If a decision table is consistent, its attribute core in the algebra view is equivalent to that in the information view, that is, $CORE_1 = CORE_2$.

（2） If a decision table is inconsistent, its attribute core in the algebra view is included by that in the information view, that is $CORE_1 \subseteq CORE_2$.

In addition, Ye drew a further conclusion about the difference between the algebra view and information view of rough set theory in [10]: if a decision table is partially inconsistent, that is, there exists an object, say $X_k \in U$, having $D(X_k)>1$, but $\min\{D(X_i),D(X_j)\}=1$ for any pair of objects of the decision table, its attribute core in the algebra view should be equivalent to that in the information view, that is, $CORE_1 = CORE_2$. This result is consistent with Ye's method in definition 8.

6. The Quantitative Difference of the Attribute Cores between the Algebra View and Information View

For comparing the attribute cores in the algebra view and information view of rough set theory, we have done the following four simulation experiments.

Table 1 Experiment Parameters

Experiment No.	1	2	3	4
Number of Condition Attribute	4	9	7	5
Domain of Condition Attribute	0,1	0,1,2	0,1,2,3	0,1,2,3,4
Number of Decision Attribute	1	1	1	1
Domain of Decision Attribute	0,1	0,1	0,1,2	0,1,2

In each simulation experiment, a lot of decision tables with different scale (number of samples) and complexity (uncertainty degree) are tested in both views. Table 1 shows the four sets of experiment parameters. For example, in experiment 1, there are 4 condition attributes and 1 decision attribute, the domain of each condition attribute is {0, 1}, and the domain of each decision attribute is {0, 1}. When the scale of decision table is small, a lot of decision tables are randomly generated and tested in order to get more accurate statistic result. The maximal number of decision tables with the same scale is 10000, whereas when the scale of decision table is large, due to the limitation of the memory capability of computer, lesser decision tables are randomly generated and tested. The minimal number of decision tables with the same scale is 5. In our experiments, all attribute cores in the algebra view are calculated using the method of Ye [8], while the attribute cores in the information view are calculated using algorithm 1 in section 5. The average number of the core attributes of these decision tables is taken as the statistic value for the number of core attributes under each scale of decision tables. In our experiments, the number of samples of a decision table is set to be 1 to 1000000. The processes of the other 3 experiments are the same. The results of these 4 experiments are show in figure 1 to 4. In these figures, the horizontal axis indicates the logarithm of the number of samples of a decision table (decision table scale), the vertical axis indicates the average number of the core attributes of the decision tables in the same scale.

In these figures, blue dot lines (- - -) indicate the result of the algebra view of rough set theory ($CORE_1$), red dash lines (— — —) indicate the result of the information view ($CORE_2$), black solid lines indicate that red line overlays blue line, that is, $CORE_1 = CORE_2$

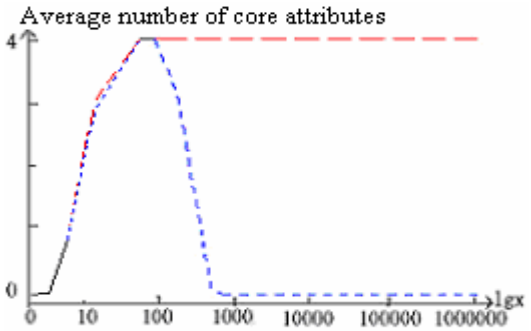


Figure 1 Experiment result 1

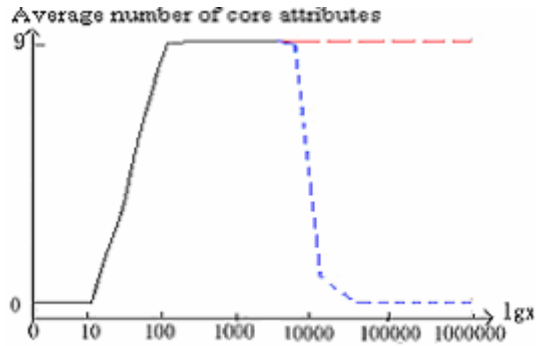


Figure 2 Experiment result 2

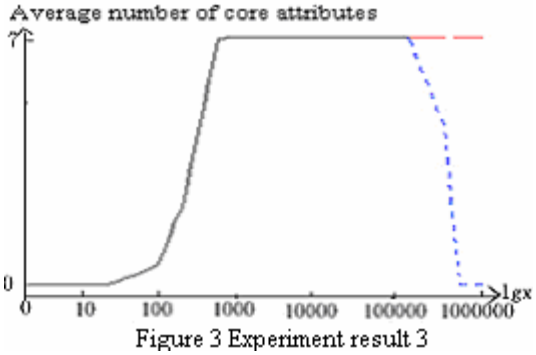


Figure 3 Experiment result 3

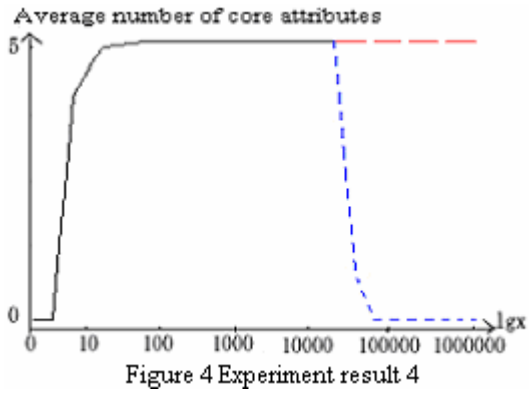


Figure 4 Experiment result 4

According to the above experiment results, we can draw the following conclusions:

（1）When the scale of a decision table is very small, the decision table is consistent and does not contain any inconsistent information. The redundancy degree of its condition attributes is high. Its attribute cores in the algebra view and information view are all empty. Obviously, there is no difference between them.

（2）When the scale of a decision table increases gradually, the inconsistent information of the decision table increases, and the redundancy degree of its condition attribute decreases gradually. The number of core attributes of the decision table increases gradually, at the same time, the attribute core of a decision table in the information view includes that in the algebra view.

（3）When the scale of a decision table increases to some degree, all condition attributes of the decision table are no more redundant, both the attribute cores in the algebra view and information view are all the full set of its condition attributes. There is no difference between them.

（4）When the scale of a decision table is large, the inconsistent information of the decision table is lager also. The lower approximation and the number of core attributes of the decision table decrease gradually in the algebra view, while in the information view, the attribute core is still all the full set of its condition attributes, since every condition attribute influences the distribution of samples of the decision table.

（5）When the scale of a decision table increases very large, the decision table is totally inconsistent. The lower approximation of the decision table is empty. Its attribute core in the algebra view is empty too. However, its attribute core in the information view is still the full set of its condition attributes. Every condition attribute influences the distribution of samples of the decision table.

7. Conclusion

Calculation of the attribute core of a decision table is the key of rough set theory, and the base of information reduction. In this paper, based on the difference between the definitions of attribute core of a decision table in the algebra view and information view, we quantitatively analyze their difference under different conditions. Our simulation experiment results prove that the attribute core in the information view includes that in the algebra view. Especially, we find that there will be great difference of the attribute cores of a decision table in these two views if it contains

much inconsistent information. This result is much useful for uncertain information system reduction

Acknowledgements

This paper is partially supported by National Science Foundation of China (No.69803014), National Climb Program of China, Foundation for University Key Teacher by the State Education Ministry of China (No.GG-520-10617-1001), Scientific Research Foundation for the Returned Overseas Chinese Scholars by the State Education Ministry of China, and Application Science Foundation of Chongqing.

References

[1] Pawlak Z, Grzymala-Busse J, Slowinski R, Ziarko W. Rough sets. Communication of the ACM, l995, 38(11): 89- 95

[2] Wang G Y. Rough set theory and knowledge acquisition. Xi'an: Xi'an Jiaotong University Press, 2001 (in Chinese)

[3] Wang G Y. Algebra view and information view of rough sets theory. Proceedings of SPIE, 2001, 4384: 200-207

[4] Wang G Y, Yu H, Yang D C. Decision table reduction based on conditional information entropy. Chinese Journal of Computers, 2002, 25(7): 759-766 (in Chinese)

[5] Y.Y. Yao. Constructive and Algebraic Methods of the Theory of Rough Sets. Information Sciences 109 (1998) 21-47

[6] Y.Y. Yao. Two Views of the Theory of Rough Sets in Finite Universes. International Journal of Approximate Reasoning 1996, 15:291-317

[7] Hu X H, Cercone N. Learning in relational databases: a rough set approach. Computational Intelligence, 1995, 11(2): 323-337

[8] Ye D Y, Chen Z J. A new discernibility matrix and the computation of a core. Acta Electronica Sinica, 2002, 30(7): 1086-1088 (in Chinese)

[9] Wang G Y. Calculation Methods for Core Attributes of Decision Table. Chinese Journal of Computers,2003,26(5)：611-615(in Chinese)

[10] Ye D Y and Chen Z J. Inconsistency Classification and Discernibility-Matrix-Based Approaches for Computing an Attribute Core. Proceedings of Springer, LNAI 2639: 269-273

[11] Wang G Y, Yu H, Yang D C, Wu Z F. Knowledge reduction based on rough set and information entropy. In: Proc World Multiconference on Systemics, Cybernetics and Informatics, Orlando, FL, 2001, 555-560

Purchasing the Web: an Agent based E-retail System with Multilingual Knowledge

Maria Teresa Pazienza, Armando Stellato, Michele Vindigni
DISP - University of Rome "Tor Vergata", Italy
{pazienza, stellato, vindigni}@info.uniroma2.it

Abstract

The more than enthusiastic success obtained by e-commerce and the continuous growth of the WWW has radically changed the way people look for and purchase commercial products. E-retail stores offer any sort of goods through evermore appealing web pages, sometimes even including their own search-engines to help customers find their loved products. With all this great mass of information available, people often get confused or simply do not want to spend time browsing the internet, loosing themselves into myriads of available e-shops and trying to compare their offers. E-retail systems are the natural solution to this problem: they place at people's disposal user-friendly interfaces, helping the customers in finding products from different e-shops that match their desires and comparing these offers for them.

Inside CROSSMARC, (a project funded by the Information Society Technologies Programme of the European Union: IST 2000-25366) different techniques coming from the worlds of NLP, Machine Learning-based Information Extraction and Knowledge Representation have been considered and conjoined to give life to an agent-based system for information extraction (IE) from web pages, which operates in a wide range of situations involving different languages and domains.

In this paper we describe the main components that realize the CROSSMARC architecture, together with their specific role in the process of extracting information from the web and presenting them to the user in a uniform and coherent way.

1. Introduction

Following the growing demand for user-friendly systems, dedicated to help people (not necessarily skilled with computers) solve everyday life tasks in an easier and convenient way, e-retail portals are becoming even more competitive than before: nowadays a wide variety of commercial agent-based systems currently guide many users in choosing online products, helping them to select features they may recognize as important for their needs and comparing on these basis the different product offers, in order to reach optimal satisfaction for the customer. In some cases, these agents simply access to a list of confederated sites which adhere to some standard in presenting their offers, in other cases, they have to mine relevant data from human-readable on-line product descriptions, extracting the information requested by the customer and presenting it in a synthetic and coherent way. Even those agents belonging to this latter category, typically don't use natural language technologies, and hence process strictly structured texts only, where product names, prices, and other features always appear in a fixed (or at least regular) order, making possible to use the page structure and/or mark-up tags as content delimiters.

Unfortunately, this kind of structured information is not what we expect to find in the web, where organization of web-pages usually tends more towards providing immediate human readability and giving emphasis on presentation of the products, than caring about how information can be easily extracted from automatic systems. Under this perspective, images and texts both contribute to the relevant information, being combined in a sometimes indivisible informational unit hard to disclose with ordinary web-mining techniques.

Things get even more complicated if we think about the possibility of examining and comparing offers from various countries, as we had to deal with different languages (and, consequently, with different character encodings used to represent their specific idioms); as a last consideration, technology and fashion push ahead very fast, seeing old concepts being unused and other ones emerging in the ever-evolving domains, moreover, even old recognized features may loose or gain importance in the aim of comparing products, or simply change the way we had to consider them (e.g. evaluation of prices or performances).

With this in mind, it's clear how standardization of the data structure that enclose the knowledge of a system, and strong decoupling of this data from the processing components, are necessary requisites to achieve optimal adaptivity towards different scenarios and applications.

We will describe here our contribution in building CROSSMARC, an e-retail product comparison multi-agent system, currently under development as part of an EU-funded project, aiming to provide users with product information fitting their needs. Inside CROSSMARC, technologies have been developed for extracting information from domain specific web pages, exploiting

language processing methods, machine learning techniques and a solid knowledge representation model in order to facilitate porting of the system to new domains. CROSSMARC also features localization methodologies and user modeling techniques in order to provide the results of extraction in accordance with the user's personal preferences and constraints.

2. System Architecture

The overall CROSSMARC architecture (see below Fig. 1) is realized through distributed but interoperable agents who communicate each other via a dedicated XML language in order to actuate, and coordinate at the same time, the different tasks that characterize the system's behavior.

Individual monolingual agents using XML to communicate each other have been plugged in: each partner in the project contributes with his autonomous agents, exchanging information through a common vocabulary provided by a domain specific ontology.

Agent roles in the architecture are primarily related to three main tasks:

- Implement a user interface, to process users' queries, perform user modeling, access the database and supply the user with product information.
- Extract Information from the WEB: here various processing steps are coordinated to find, retrieve, analyze and extract information from the Internet.
- Store the extracted information in a database, in order to feed the system with the data to be presented to the use

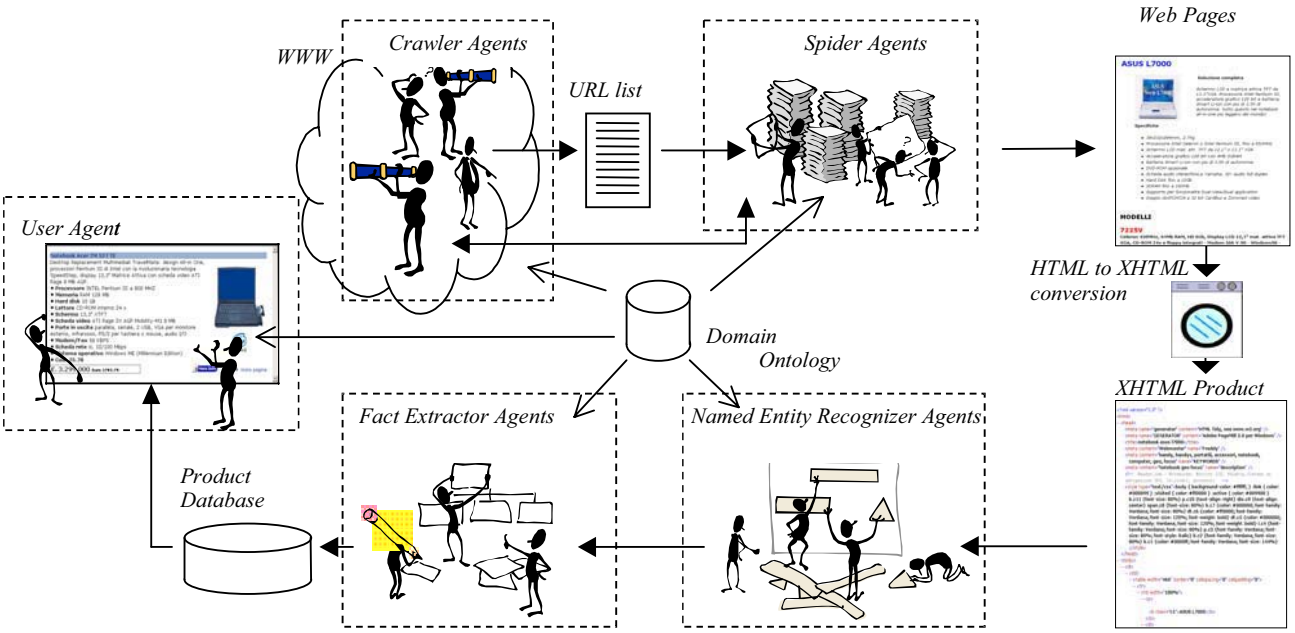


Fig. 1: Overall CROSSMARC Architecture

In particular, considering the main role of the Extraction agents, they are broadly divided into two categories, depending on their specific tasks:

- **Information retrieval agents (IR),** which identify domain-relevant Web Sites (*focused crawling*) and return web pages inside these sites (*web spidering*) that are likely to contain the desired information;
- **Information Extraction (IE) agents** (two separate agents for each language) which process the retrieved web pages. There are specific roles for every step of the extraction process: *Named Entity Recognition and Classification* (NERC), i.e. recognition of concepts pertaining to the domain of interest inside the web pages, *Fact Extraction* (FE), which consists in the identification of the number of products and how they are distributed in the web pages (*products demarcation*), and in the extraction of all the characteristics of these products.

All the agents share a common Knowledge model, which can easily be customized to new Domains, Languages and Extraction Templates. The customization process can be easily performed through an application (developed inside CROSSMARC) based on the APIs of the ontology editing tool Protégé-2000 [4], from the university of Stanford; an XML version of these ontologies and the FE XML Schema for every domain are then automatically derived from it through a specially designed plug-in that has been developed for this purpose inside the CROSSMARC Project.

Now we'll give more details regarding specific CROSSMARC components and how their work is coordinated to extract relevant information from the web.

3. Web Pages Collection

The process of collecting domain-specific web pages articulates in two different and complementary sub-processes:

- **focused crawling** to identify Web sites (e-retailers web sites) relevant to a specific domain (e.g. electronic products/computer goods)
- **domain-specific spidering of a Web site** to navigate through a specific Web site (e.g. retailer of electronic products), retrieving Web pages of interest (e.g. product descriptions).

Interesting Web sites are initially identified by an external focused crawling process. Then each site is spidered, starting at the top page, scoring the links in the page and following "useful" links.

4. Multilingual NERC and Name Matching

The Multilingual NERC subsystem architecture is shown in Fig. 2 where the individual components are autonomous agents, which need not to be installed all on the same machine.

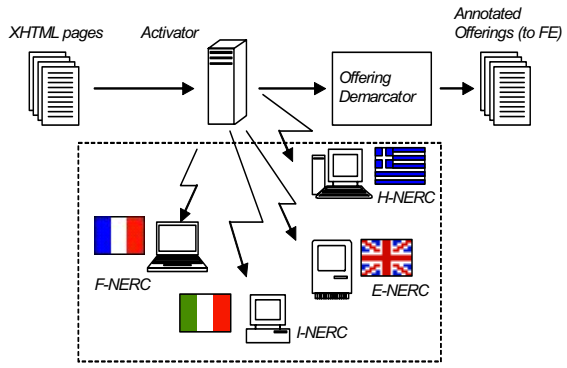


Fig. 2: architecture of the Named Entity Recognizer Component

We'll give now a inside view of the Italian Named Entity Recognizer and Classificator.

4.1 The Italian NERC

The I-NERC agent receives, from previous modules in the chain, the Web pages as XML structures containing description of one or more products from Italian retailers, then processes and enriches it by adding markup tags for domain specific information (i.e., product name, manufacturer, cost, etc.). In order to mark up relevant named entities, the I-NERC exploits two kinds of evidences:

- *Internal evidences*, that is information related to the entity itself.
- *External evidences* provided by the context in which the entity appears, which can in turn be divided into:
  i. *Structural evidences*, provided by the document organization (tables, bullet lists, or, at a lesser extent, formatting properties as bold, italic, etc.).
  ii. *Semantic evidences*, provided by its content (as in the case mentioned above).

The overall I-NERC agent is implemented as a sequence of processing steps driven by XSLT transformations over the XML input structure, by using a XSLT parser with a number of plugged-in specific extensions. A sequence of linguistic processes is activated applying a pipeline of transformations on the page

1) The **Normalizer** transformation provides pre-processing at character-level of the source document, to deal with bad word separators, wrong punctuation and word-level misspelling or ungrammaticalities.
2) The **Tokenizer** transformation applies to the page content, segmenting the text into atomic tokens, classified as word, number and separators and included in XML appropriate tags.
3) The **Terminology** transformation recognizes terminological expressions as well as simple constituents and expresses them in their standardized form.
4) **Lexicon-lookup** matches lexical rules and entries against the input. This phase relies on lexical knowledge, represented by an Italian lexicon, and additional lexical tables for specific information (for instance, measurement units).
5) **Unit-matcher** activates numerical expressions recognition in order to identify currencies, dates, lengths, and other domain specific quantities.
6) **Ontology-lookup** matches identified entities against the ontology and categorizes them accordingly.

5. Multilingual and Multimedia Fact Extraction

From a black-box point of view, the overall architecture of the Multilingual and Multimedia Fact Extraction (FE) component is analogous to the one of the NERC, being the input again represented by XHTML web pages, in this case enriched by NERC semantic annotations.

The first processing step performed by the Fact Extractor agents is *Product Demarcation*: NERC output is transformed by the *Product Demarcator module* (PD) of the Fact Extractor which analyzes the semantic categories identified by the NERC, tries to find

correlations between them and subsequently aggregates them into separate products.

Following these steps, the FE component exploits the NERC + PD annotations in order to identify which of the recognized semantic entities fill a specific fact slot inside a product description, according to the above mentioned XML FE Schema.

Inside CROSSMARC, different FE components have been realized from the four partners involved in the project. The common characteristic of all the Fact Extractor components is that all of them implement wrapper induction techniques for extracting only the analyzed information pertaining to the products recognized inside the pages. In particular a first version of the English Fact Extractor was based on Boosted Wrapper Induction [3], the Greek version of the Fact Extractor module is based on STALKER [1], while the Italian one is a customized implementation of the Whisk algorithm [2].

Obviously the scenario depicted is quite different from many of the typical wrapper induction approaches, in this case, strong semantic analysis performed by other linguistic processors modifies the search space of pure wrapper induction modules, limiting the number of valid extractions that a wrapper can make to those which maintain coherency with the structure of the pages and of the products presented inside them.

5.1 Italian Fact Extractor Component

As previously outlined, the Italian FE System has its core in a customized implementation of Soderland's WHISK algorithm of Wrapper Induction.

WHISK takes as input a set of hand-tagged instances, using them as a pool (the *training set*) for induction of IE rules, expressed in the form of regular expressions. These rules are induced top-down, first finding the most general rule that applies with success to the considered training instance, then producing new extended rules by adding terms one at a time and testing their behavior against the whole training set. The candidate extended rule that performs best against the whole training set is thereby chosen and examined for new possible extensions. The process is then reiterated until all the candidate extensions do not perform better than the rule produced in the previous step.

WHISK is capable of learning both single-slot and multi-slot rules, though we considered only single-slot rules because of the wide heterogeneity of combinations in which products can be presented: even in pages with similar structure or inside a single page, different products may vary in the number of characteristics they show or in the disposition of these characteristics among the description of the offer. For this reason, Italian FE relies on localization of the products inside the pages provided by the previous Product Demarcation component.

5.1.1 Whisk adaptation to CROSSMARC's needs

Soderland's original algorithm has been customized to meet the specific needs of the CROSSMARC environment, through the following aspects:

a) Ontology Lookup. WHISK uses the notion of Semantic Class to address disjunctive sets of terms that can be considered as equivalence classes. At the same time the concept of Semantic Tag is added to wrap concepts that may appear in a multitude of different aspects.

Both these two options have been adopted in CROSSMARC implementation of the WHISK algorithm, since all the NE tags (enriched by Product Demarcator attributes) are dynamically imported as Semantic Tags, while sets of Semantic Classes are defined on the basis of elementary concepts present in the ontologies.

b) Limiting Search Space of Induction when adding terms. WHISK original algorithm was conceived to operate on specific instances (i.e. pre-separated portions of the text containing the information to be extracted) while all the FE components developed inside the CROSSMARC project operate on the entire web pages. Heuristics that rely on semantic information provided by previous modules have been designed to limit the search space of induction [5].

c) Laplacian Expected Error versus Precision: rule appliance strategy. The Laplacian Expected Error Rate (i.e. $(e+1)/(n+1)$, where e is the number of wrong extractions and n is the overall amount of extractions made), originally adopted by Soderland for evaluating rules, has been preserved as the performance measure for evaluation of the temporary rules created during rule expansion, as it expresses a good trade-off between rule precision and recall while pure Precision is stored as an attribute for every rule, as it is necessary to establish which rules take precedence when they are applied outside the training phase.

5.1.2 Evaluation of Italian Fact Extractor component

At present time, CROSSMARC project is still to be concluded, but we made specific evaluation of FE components in the 4 different languages; in table 1 evaluation results for the domain covering laptop computer offers reports precision and recall statistics for all of the considered characteristics. The table below exposes black-box evaluation of the FE components, assuming optimal input from the previous processing steps (NERC and Product Demarcation);

Table 1: Evaluation results for the Laptop Computers Offers Domain on 4 different languages

FEATURE	ENGLISH		FRENCH		GREEK		ITALIAN	
	PR	RC	PR	RC	PR	RC	PR	RC
MANUFACTURER	0.89	1	0.99	1	1	1	1	0.99
PROCESSOR.	0.99	1	1	1	1	1	0.99	1
OP. SYSTEM	0.78	0.98	0.82	0.94	0.92	0.98	0.78	0.99
PROC.SPEED	0.86	0.99	0.95	0.98	0.85	1	0.95	0.98
PRICE	0.99	1	1	1	1	1	1	1
HD CAPACITY	0.99	0.94	0.94	0.80	0.96	0.96	1	0.88
RAM CAPACITY	0.82	0.97	0.95	0.94	0.90	0.80	0.96	0.89
SCREEN SIZE	0.85	0.98	0.70	0.99	0.95	0.98	0.92	0.99
MODEL NAME	0.99	1	1	0.99	1	1	0.99	1
BATTERY TYPE	1	0.86	0.97	0.63	0.97	0.76	1	0.5
SCREEN TYPE	0.82	0.98	0.81	0.96	0.99	1	0.86	0.99
WEIGHT	0.98	1	0.96	1	1	1	0.92	1
AVERAGE VALUES	**0.91**	**0,97**	**0.93**	**0.94**	**0.96**	**0.96**	**0.95**	**0.90**

6. The Knowledge model of CROSSMARC

All of the components described so far, are driven by a community of agents sharing common informational resources and semantics defined at different levels (i.e. lexical, ontological and task oriented). An ontological architecture has been developed upon this definition and has thus been organized around three different layers:

- a *meta-conceptual layer*, which represents the common semantics that will be used by the different components of the system in their reasoning activities
- a *conceptual layer* where relevant concepts of each domain are represented, and
- an *instances layer* where language dependent realizations of such concepts are organized.

The current ontology structure is maintained through a CROSSMARC application based on the APIs of Protégé 2000 [4], an ontology engineering environment that supports ontology development and maintenance. Protégé-2000 adopts a frame-based knowledge model, based on classes, slots, facets, and axioms.

The meta-conceptual layer of CROSSMARC defines how linguistic processors will work on the ontology, enforcing a semantic agreement by characterizing the ontological content according to the adopted knowledge model.

The Protégé metaclasses hierarchy has been extended introducing a few metaclasses. These are used in the Conceptual level to assign computational semantics to elements of the domain ontology. Basically the metaclass extension provides a further typization to concepts, adding a few constraints for formulating IE templates.

The instance layer represents both domain specific individuals and lexicalizations of these individuals into the adopted languages. It instantiates classes in the domain ontology; these instances fill the values for attributes of the domain templates.

7. Conclusions

The difficulty in building complex adaptive systems is represented by an unavoidable trade-off between how much experience and task-oriented skill must be put inside the system on one side, and how it must satisfy a certain degree of generality and the required openness versus possible extensions.

CROSSMARC aims to fulfil its requirements through a solid Knowledge Model provided with the necessary level of abstraction, which constitutes the main fabric through all of the agents that control its components can communicate, share information and cooperate to reach their tasks.

The neat separation between the Knowledge Model (the Meta-Conceptual Layer), the Domain Model (the Conceptual Layer) and Domain Instances and Languages (Values in the Ontology) permits easy plug-ability of different system resources and processors.

Following these premises, two techniques coming from two completely different approaches to IE have been integrated: Ontology and Language Driven Named Entity Recognition and Classification and Wrapper Induction Based Fact Extraction.

This way, the system takes the benefits of both the approaches: from a maintenance cost point of view, it is freed from the need for technical experts for customisation versus new domains and languages, leaving to knowledge engineers and domain experts the task of creating/updating ontologies and annotating other sites, at the same time, this combination permits to add the expressive power of Concept Recognition to Wrapper Induction processors, whose extracted elements would, in other case, remain meaningless strings.

10. References

[1] I. Muslea, S. Minton C. Knoblock "STALKER: Learning extraction rules for semistructured Web-based information sources". *AAAI-98*. Madison. Wisconsin.

[2] Sonderland S. "Learning Information Extraction Rules for semi-structured and free text." *Machine learning*. Volume 34 (1/3), 1999, pp. 233-272.

[3] Freitag D., Kushmerick N., "Boosted Wrapper Induction". In the Proceedings of the *7th National Conference on AI*, Austin, Texas, 2000.

[4] N. F. Noy, R. W. Fergerson, & M. A. Musen. "The knowledge model of Protege-2000: Combining interoperability and flexibility". *2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000)*, Juan-les-Pins, France,2000.

[5] M.T. Pazienza, A. Stellato, M. Vindigni, "Combining Ontological Knowledge and Wrapper Induction techniques into an e-retail System", *ATEM2003 Workshop on Adaptive Text Extraction and Mining* 22 Sept. 2003 Cavtat-Dubrovnik, Croatia.

Towards a Representation Framework for Modelbases

Thadthong Bhrammanee and Vilas Wuwongse
Computer Science & Information Management program,
Asian Institute of Technology
{ttb, vw}@cs.ait.ac.th

Abstract

This XML-based representation framework for modelbases supports a modeling life cycle as well as promotes shareable and reusable decision models over the Web. It comprises two representation layers: Model Ontology and Model Schema, the first of which plays a crucial role in defining an agreement for the use of concepts among representation layers as well as users/applications of modelbases. Model Schema provides a general description for each set of common decisions. Supportive capabilities of the proposed representation framework demonstrate its adequacy.

1. Introduction

The newly developed representation framework for modelbases is strongly motivated by the need for convenient use of a decision model across the Web as well as promotion of model reuse. A modelbase—which stores decision models—is a vital component of model-driven Decision Support Systems (DSS). The term "decision model" refers to a quantitative model used in Management Science and Operations Research (MS/OR).

There exists much research on the representation framework for modelbases. From the perspective of a decision model structure, there are five approaches to model representation [4]—ranging from those which ignore the algorithmic aspect to those with high emphasis on executable systems. The *data-centric* approach employs a traditional database data model as its design principle, the *structure-centric* approach views a decision model in terms of a definitional system, the *abstraction-centric* approach aims to reduce the complexity of models by hiding all but relevant data, the *logic-centric* approach is based on logic-based theory and the *computation-centric* approach presents an algebraic form of a decision model and aims to provide an executable modeling language.

On the other hand, from the research perspective on the forms specifying decision models, there are at least four approaches to represent them [4]: The *graphic* approach graphically represents the relations among parts of a decision model, the *text* approach provides a textual representation of a decision model, the *algebraic* approach represents a decision model in a way which is close to an algebraic representation and the *schematic* approach restricts the structure and content of a decision model to a certain schema. Note that various approaches can be combined in a single representation framework.

Up to the present time, there exist at least three major aspects which affect the development of a modelbase framework:

User aspect- a user's unawareness of an available decision model on the Internet and incompatibility of the user's machine [3].

Technologies- a frog leap advancement of computation via the Web, an increasing proficiency of DSS technologies such as mobile devices and the emerging Web services framework [18].

New business model- value added business services which make a decision model available for seven days and twenty-four hours.

Those aspects clearly display that the Internet is changing the way of exploiting decision models. Unfortunately, current representation frameworks are neither adequately Web-accessible nor sufficiently flexible to exchange models across platforms [4].

The proposed new framework employs XML-based representation and thus facilitates model exchange throughout the web as well as promotes interoperability, simplicity and extensibility. It comprises two representation layers: Model Ontology and Model Schema. The first provides a shared terminology which users or applications of modelbases can jointly understand and employ; the second contains a model description of a generic decision model. A model instance can be later represented as a specialization of a model described in the Model Schema layer. All layers are seamlessly interoperated by means of the Web

Ontology Language (OWL) [9] and RDF Declarative Description (RDD) (the Appendix provides a brief introduction to RDD) [1]. OWL provides a simple ontological-modeling facility. RDD enhances OWL expressive power with a computation mechanism and extends OWL elements with variables.

Sections 2 and 3 present Model Ontology and Model Schema, respectively, Section 4 discusses the framework's capability and Section 5 concludes.

2. Model Ontology

Model Ontology is a central vehicle for the reuse of knowledge in modelbases. Purposes of Model Ontology are:

- Definition of the meaning of terms/concepts and provision of shared terminologies, e.g., the term "activity" refers to the concept which requires a "resource"
- Embedment of quantitative problem knowledge, e.g., the taxonomy of mathematical problems.
- Implementation of a set of axioms, e.g., a stochastic model always contains random variables. The axioms will enable the model ontology to find automatically answers to various questions regarding the decision model domain.

Existing ontologies related to MS/OR have some of the necessary concepts and relations which can serve as shared conceptualization of decision models. Unfortunately, none of them have all necessary ones. For example, GAMS [8] covers most of the top concepts of mathematical problem; however, it does not include many basic concepts of the real world domain such as "resource" and "activity". Therefore, the newly developed Model Ontology reuses parts of existing ontologies and classification schemes. The related ontologies are:

GAMS [8]: The Guide to Available Mathematical Software (GAMS) is the NIST *de facto* tree-structured classification system for mathematical and statistical problems.

OZONE [15] provides a framework for constructing an appropriate domain model. Its application areas cover scheduling, manufacturing, space and transportation logistics. The five basic concepts in OZONE ontology are demand, product, resource, activity and constraint.

Process ontology [14]**:** The NIST standard of exchanging process information among manufacturing applications. The wide range of application areas which can employ this ontology include process planning, scheduling, business process reengineering, etc. Key concepts are activity, activity occurrence, time point, and object.

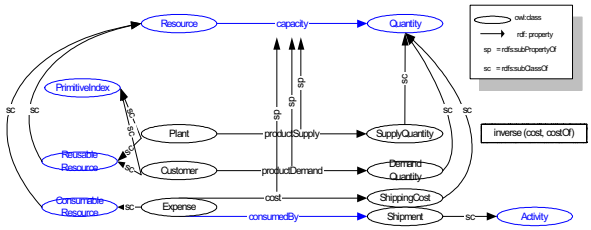


Figure 1. A fragment of terminologies defined in Model ontology

Enterprise ontology [17] is supported by various parties including IBM, and the UK Department of Trade and Industry. It aims to provide various parties—from business managers to software engineers—with a shared understanding of the aspects of a business enterprise. The major concepts in Enterprise ontology are activity, organization, strategy, marketing and time.

2.1 Concepts and properties in Model Ontology

Model Ontology comprises four main parts:

Problem Taxonomy contains the taxonomy of decision problems. "Model" is the top level concept/class. Problems at the lower level of the hierarchy become subclasses. For example, "Transportation Problem" is a subclass of "Constrained Optimization" which is a subclass of "Optimization Problem".

Mathematical Model Elements provide the indispensable elements of a quantitative model. Major elements are "Object", with the subclasses "Variable" and "Index", and "Relation" [7]. "Variable" has "Dependent Variable" and "Independent Variable" as subclasses. "Primitive Index" and "Compound Index" are kinds of "Index". "Relation" has "Objective" and "Constraint" as subclasses. "Constraint" has "Arithmetic Constraint" and "Logical Constraint" as subclasses [7].

World Elements are objects in the real world which relate to a decision problem. They also correspond to Model Elements in some sense. "Time", "Resource", "Activity", "Quantity", "State", and "Event" are examples of the subclasses of "World Element". "Time" can be specialized to "Time Point" and "Time Interval", "Resource" to "Consumable Resource" and "Reusable Resource".

Relations, Functions, and Operators- A "Relation" is a property which denotes a link between model elements or entities, for example, "Equal" and "Less Than". A "Function Type" denotes a predefined procedure, for example, "Sum" and "Mean". An "Operator" denotes a symbol which operates between functions, for example, "Minus" and "Time".

Figure 1 shows some of the concepts defined in Model Ontology. The Model Schema layer can use these concepts in a model description. The example illustrates the concepts related to a logistic domain. Here, "Resource" is an entity which supports or enables execution of "activity" [15]. "Shipment" is considered to be "Activity" in a logistic domain. An "Activity" uses a "Resource" to achieve or fulfill some "Events". "Plant" is a kind of "Reusable Resource". Each plant has a certain capacity, denoted as "Product Supply", which points to "Supply Quantity". "Plant" corresponds to a "Primitive Index". Plant is also known as a shipping origin. "Expense" or money is a "Consumable Resource" consumed by a "Shipment". "Shipping Cost" is a numerical quantity of the "Expense" of a "Shipment".

In addition, the Model Ontology layer also stores Ontology axioms. They are specified by RDD clauses. Axioms provide rules, including structural knowledge of a decision model. A "structural knowledge" is referred to in Krishnan et al. [12] as "model construct knowledge"—an important component of knowledge-based model formulation.

3. Model Schema

Model Schema describes a generic decision model, whence the Model Schema layer stores various model descriptions. Every decision model has both internal and external views, defined in model description (Figure 2a): *Model Profile* serves as a black box view of a model, while *Model Configuration* serves as a glass box view describing the internal configuration of a model.

Different decision model classes (e.g., optimization and simulation) can define their skeleton—major model constructs—inside a "Construct" element. For example, model constructs of all models in the optimization problem class consist of objective function, constraint, decision variable and coefficient.

Consider the example of a transportation problem; it inherits all constructs of an optimization problem. However, a transportation problem extends the model description to be more specific to the problem by selectively drawing logistic-related concepts from the Model Ontology. Figure 2b depicts a fragment of the model "Construct" element of a generic transportation model. Clearly, detail semantic of each concept specified here is traceable in the Model Ontology. For example, "Plant" is a "Primitive Index" and "Resource", "Route" is "Compound Index", etc.

In addition, the definition of model can be specifically defined for a more specific problem situation by specialization of a generic model stored in the Model Schema layer. This allows different problems with a similar problem domain situation to share a single model schema.

4. The supportive capabilities

The framework is found to support *essential data model characteristics for modelbases* indicated in [4]. The term "data model" refers to a particular language for modeling a decision model. In summary:

Model creation/formulation support: Various users of modelbases [2] have different objectives and need different supports to formulate a model. The representation layers used here appropriately satisfy the needs of each user type. *Model builders* maintain the ontology so that model descriptions can be properly created by *analysts*. *Decision makers* can create a decision model with instances by supplying data to the model description. In addition, the framework provides a graphical view of a decision model by employment of an RDF graph [10], with some extensions to suit an OWL expression.

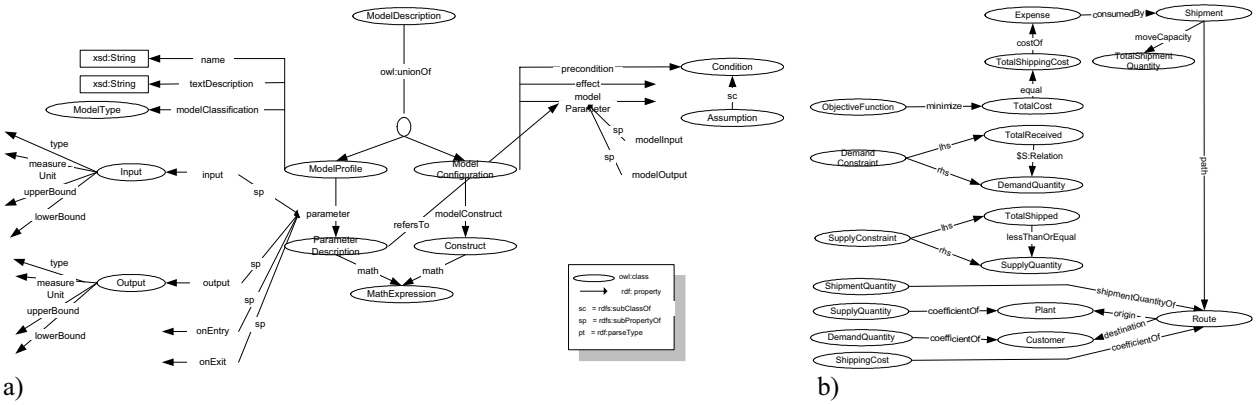


Figure 2. a) Excerpt of a general model schema b) A fragment of generic transportation model schema

Model advertisement/registration support: A standard online register for Web services [16] is friendly with Web Services Description Language (WSDL) [19] and DAML Services (DAML-S) [5]. The proposed framework then provides some equivalent features to WSDL and DAML-S, in order to satisfy the standard model registry.

Model discovery/selection: The proposed framework employs an RDD clause in order to generate the selection result for a complex model selection. By encoding the model in RDD language (see example in Figure 3), the bodies of the clause specify the condition. The head drives a selected decision model.

Model modification/customization support: Firstly, *support of problem type modification* (Decision problems in the same hierarchy can be modified from one to another), secondly, *support of model transformation across the modeling paradigm*s by employment of a transformation language such as Extensible Stylesheet Language Transformation (XSLT), finally*, integrity constraint support* by formalizing an integrity constraints as RDD clauses.

Model composition and integration support: Conflict resolution [11] is a requirement during performance of model composition/integration. By means of explicit specification of concepts which are stored in Model Ontology, *naming conflicts* can be solved. By means of integrity constraint support and the computation mechanism of RDD, *granularity* and *dimensional/unit of measurement conflicts* can be solved.

Model execution support and support of representation of mathematical equations: The framework actually allows representing MathML—a mutual way to provide a mathematical equation readable by human and machine—in the Model Schema. MathML representation of a decision model can be sent to a MathML-recognized solver for execution. Alternatively, XSLT can be applied to model description in order to transform an existing model into a representation of a modeling framework which owns a model solver such as AMPL [6].

Support of interoperability: Use of XML-based language to represent a decision model ensures interoperability, since XML is a *de facto* exchange standard among industries.

Support of indexing: The proposed framework is considered a symbolic subscript-free language, hence, provides ease of model formulation and ease for those who not originally create a model to understand it [13]. In addition, common index set functions define a dynamic subset (e.g., a subset of, union of, intersection of and complement of) are allowed.

Support of representation of incomplete information: By employment of RDD variables, the model containing unknown information (e.g., an unknown objective function and an unknown relation) can be formed as a non-ground RDD expression and stored in a model repository. Users can later on apply specialization to a model—when the information is known—in order to turn it into a complete model (ground RDD expression). Figure 2b shows an employment of RDD string-variable ($S) to define the unknown "relation" as *$S:Relation*.

```
<owl:Individual rdf:about="Answer">
     <ModelName rdf:resource="$S:ModelName"/>
</owl:Individual>
     ←
     <owl:Class rdf:about="$S:ModelName">
          <$I:ObjectPropertyA rdf:about="minimize">
               <rdfs:domain>
                    <owl:Class rdf:about="ObjectiveFunction"/>
               </rdfs:domain>
               <rdfs:range>
                    <owl:Class rdf:about="$S:SomeBusinessResource"/>
               </rdfs:range>
          </$I:ObjectPropertyA>
          <owl:Class rdf:about="$S:Shipment">
               $E:ShipmentElements
          </owl:Class>
               $E:OtherElements
     </owl:Class>,
     <owl:Class rdf:about="$S:Shipment">
          <owl:subClassOf rdf:resource="#Activity"/>
     </owl:Class>,
     Equal (<value>$S:Shipment<value><value>Shipment</value>).

% The XDD clause defines a selection query which returns name of a model
% which its objective is to "minimize"  some kind of "business resource",
% and containing "shipment" as an "activity" in a model.
```

Figure 3. A model selection

5. Conclusions

A new framework of model representation is proposed as a means of supporting modeling life cycle and handling current and future environmental impacts. The framework utilizes the ontological discipline to define terms which can be communicated across to modelbase users, including people and applications.

The merit of the metadata description facility–OWL– and the expressiveness of RDD drives a uniform representation of all three representation layers for modelbases. Thus, each layer is ready to exchange information items. In addition, the framework is feasible to integrate into a Web services framework and enjoy computation via the Web which also abates a user's unawareness of decision models and machine configuration incompatibility problems. The development of a prototype XML-based model representation under the proposed framework is underway.

6. References

[1] Anutariya, C., Wuwongse, V., Akama, K., and Nantajeewarawat, E. "RDF Declarative Description (RDD): A Language for Metadata". *Journal of Digital Information*, 2:2, 2001

[2] Balasubramanian, P. and Lenard, M., "Structuring Modeling Knowledge for Collaborative Environments", *Proceedings of the 31st Hawaii International Conference on System Sciences*, 1998, pp. 464-475

[3] Bhargava, H., Krishnan, R., Roehrig, S., Casey, M., Kaplan, D., and Müller R., "Model Management in Electronic Markets for Decision Technologies: A Software Agent Approach", *Proceedings of the 30th Hawaii International Conference on System Sciences*, Maui, HI, 1997

[4] Bhrammanee, T. and Wuwongse, V., "Requirements of a Data Model for Modelbases", *Proceedings of Workshop on Applications, Products and Services of Web-based Support Systems*, Halifax, Canada, 2003 (to appear)

[5] DAML Services (DAML-S), http://www.daml.org/services/

[6] Fourer, R., Gay, D., and Kernighan, B., "AMPL: A Mathematical Programming Language", *Management Science*, 36, 1990, pp. 519-554

[7] Greenberg, H., "The Role of Software in Optimization and Operations Research", Chapter 6.5 in *Encyclopedia of Life Support Systems*, Oxford, UK, 2002

[8] Guide to Available Mathematical software (GAMS), NIST, http://gams.nist.gov

[9] Harmelen, F., Hendler, Jim., Horrocks, I., McGuinness, D., Patel-Schneider, P., and Stein, L., "OWL Web Ontology Language Reference". W3C Working Draft 31 March 2003, http://www.w3.org/TR/owl-ref/

[10] Klyne., G. and Carroll, J., "Resource Description Framework (RDF): Concepts and Abstract Syntax", W3C Working Draft 23 January 2003, http://www.w3.org/TR/rdf-concepts/

[11] Krishnan, K., and Chari, K., "Model Management: Survey, Future Research Directions and a Bibliography", *The Interactive Transactions of OR/ MS (ITORMS)*, Vol 3, 2000

[12] Krishnan, R., Li and, X., and Steier D., '" Knowledge-based mathematical model formulation system", *Communications of the ACM*, 35:9, 1992, pp. 138-146

[13] Lin, S., Schuff, D., and Louis, R., "Subscript-Free Languages: A Tool for Facilitating the Formulation and Use of Models", *European Journal of Operational Research*, 123:3, 1998, pp. 614-627

[14] Schrlenoff, C., Gruninger, M., Tissot, Florence., Valois John., Lubell, Josh., and Lee, Jintae , "The Process Specification Language (PSL) Overview and Version 1.0 Specification", *NISTIR 6459, National Institute of Standards and Technology,* Gaithersburg, MD, 2000

[15] Smith, S., and Becker, M., "An Ontology for Constructing Scheduling Systems", to appear in working notes for 1997 *AAAI Spring Symposium on Ontological Engineering*, Stanford, CA, March 1997 (AAAI Press)

[16] Universal Description, Discovery and Integration (UDDI), http://www.uddi.org

[17] Uschold, M., King, M., Moralee, S., and Zorgios, Y., "The Enterprise Ontology", *The Knowledge Engineering Review*, Vol. 13, Special Issue on Putting Ontologies to Use, 1998

[18] Web Services Activity, W3C, http://www.w3.org/2002/ws/

[19] Web Services Description Language (WSDL), http://w3.org/TR/wsdl

Appendix

RDF Declarative Description (RDD) [1] is an RDF-based knowledge representation, which extends ordinary well-formed RDF elements by incorporation of variables for an enhancement of expressive power and representation of implicit information into so called *RDF expressions*. Ordinary RDF elements – RDF expressions without variable – are called *ground RDF expressions*. Every component of an RDF expression can contain variables, e.g., its expression or a sequence of sub-expressions (*E-variables*), tag names or attribute names (*N-variables*), strings or literal contents (*S-variables*), pairs of attributes and values (*P-variables*) and some partial structures (*I-variables*). Every variable is prefixed by '$*T*:', where *T* denotes its type; for example, $S:value and $E:expression are *S*- and *E*-variables, which can be specialized into a string or a sequence of RDF expressions, respectively.

An *RDD description* is a set of *RDF clauses* of the form:

$$H \leftarrow B_1, \ldots, B_m, \beta_1, \ldots, \beta_n,$$

where $m, n \geq 0$, H and the B_i are RDF expressions, and each of the β_i is a predefined *RDF constraint* – useful for defining a restriction on RDF expressions or their components. The RDF expression H is called the *head*, the set $\{ B_1, \ldots, B_m, \beta_1, \ldots , \beta_n\}$ the *body* of the clause. When the body is empty, such a clause is referred to as an *RDF unit clause* and the symbol '←' will often be omitted; hence, an RDF element or document can be mapped directly onto *a ground RDF unit clause*. Given an RDD description D, its meaning is the set of all RDF elements which are directly described by and are derivable from the unit and non-unit clauses in D, respectively.

Index of Authors

Published by
Department of Mathematics and Computing Science

Saint Mary's University

Halifax, Nova Scotia, Canada

Technical Report Number: 2003-03 October, 2003
ISBN: 0-9734039-1-8