GUMSAWS: A Generic User Modeling Server for Adaptive Web Systems

Jie Zhang
School of Computer Science
University of Waterloo
Waterloo, ON, Canada
j44zhang@uwaterloo.ca

Ali A. Ghorbani
Faculty of Computer Science
University of New Brunswick
Fredericton, NB, Canada
ghorbani@unb.ca

Abstract

In this paper we focus on the architecture, design and implementation of a generic user modeling server for adaptive web systems (GUMSAWS), reaching the goals of generality, extendability and replaceability. GUMSAWS acts as a centralized user modeling server to assist several adaptive web systems (possibly in different domains) concurrently. It incrementally builds up user models, provides functions of storing, updating and deleting entries in user profiles, and maintains consistency of user models. Our system is also able to infer missing entries in user profiles from different information sources, including direct information, groups information, association rules and general facts. We further evaluate its inference performance within the context of e-commerce. Experimental results show that the average accuracy of inferring user missing property values from different information resources is found to be almost 70%. We also use a personalized electronic news system to demonstrate the example of our system in use.

Keywords: Generic User Modeling Server, Adaptive Web Systems, Inference, Information Sources.

1 Introduction

Web sites are becoming more and more popular and convenient for providing information over a broad number of topics. They are used in diverse systems, such as educational systems, online information systems, online help systems, information retrieval systems and e-commerce systems. Problems occur as the use of web sites increases, their size gets larger, and their structure becomes more complex. The rich link structure of a web site can cause users to get easily overwhelmed by the large number of navigation choices, and they may become unable to navigate effectively. This is referred to as the "lost-in-hyperspace" problem. Web sites provide (from the point of view of a user) relatively static content, though they are viewed by diverse users. This may cause difficulties for those who have less background, may be redundant for those who are familiar with the information, and may present more uninteresting than interesting material for others. This is a "one-size-fits-all" problem common to non-adaptive web sites.

To address these problems, there is a demand for web sites to be automatically adapted to reach the goals of personalization, recommendation, selection, and usage analysis [9]. These web sites are called adaptive web sites. They are able to transform a page request into a final page response by considering information about the requested page, about the user, about the way the system has been used, about the environment of the site, and about the environment of the user. They may provide three different types of adaptation: content adaptation, navigation adaptation, and presentation adaptation. While there is some overlap between these types of adaptation, they are different in what they target. Content adaptation adds and/or removes information fragments to/from the page based on the current context. Navigation adaptation adds, removes, hides, sorts links, and/or changes the color of the links in a page, in order to provide the best navigation structure in the current context for the user. Adding a recommended item at the end of a page, also falls in this class of adaptation. Finally, presentation adaptation reformats the information fragments to achieve the appropriate final presentation for the current context.

Adaptive web systems require the support of user modeling technology to provide information of an individual user or group of users. We propose a generic user modeling server for adaptive web systems (GUMSAWS). It acts as a centralized user modeling server to assist several adaptive web systems concurrently. It consists of different components, such as the User/Group Model Description, the Authoring Tool, the Modeling Interface, the User Profile Manager, the Usage Group Handler, the Association Miner, the Inference Engine, and the Profile Editor. Together, they offer the user modeling functions of building up user or group models; storing, retrieving, updating and deleting entries in

Fifth Annual Conference on Communication Networks and Services Research(CNSR'07)
0-7695-2835-X/07 $20.00 © 2007 IEEE

profiles; and inferring missing user property values. Our system also allows users to review and modify their profiles.

We design and implement GUMSAWS to reach the goals of generality, extendibility and replaceability. GUMSAWS is applicable to multiple application domains. It can assist more than one applications (possibly for all applications with which users interact) and has the capability of domain-independent user modeling. It also allows extensions, so that new components can be developed from other fundamental ones. For example, new information sources can be generated for the purpose of inference by adding a new type of miner. Different machine learning and data mining techniques have been implemented in the system. They can be replaced by other techniques. For example, the K-means+ clustering algorithm [6] is implemented to group users with common interests. Other clustering algorithms, such as K-means, is also able to replace K-means+ and be integrated into the system.

GUMSAWS is examined within the two application domains, E-tailer and E-news. An example application in E-tailer domain can be an online store. An electronic news system is an example application in E-news domain. The accuracy of inferring user property values from different information resources, such as direct information, groups information, association rules and general facts, is evaluated by the Knowledge Discovery and Data Mining competition (KDDCUP2000) data in the E-tailer domain. KDDCUP2000 data is provided by the leading Data Mining and Knowledge Discovery competition in the world for web mining tasks. Experimental results show that the average accuracy of inferring user property values from different information resources is found to be almost 70%. PENS (Personalized Electronic News System) [15], an example application in the E-news domain, illustrates the basic user modeling functions provided by GUMSAWS. PENS also makes use of GUMSAWS to adapt to user navigation history.

The rest of this paper is organized as follows. In Section 2, we clarify the definitions of user profile, user model and user modeling, and emphasize on the importance of a generic user modeling server. In Section 3 we describe the framework of our system and the implementation of some system components. Experimental results are presented in Section 4. Section 5 demonstrates an example of our system in use. Finally, we conclude the current work and propose future work in Section 6.

2 User Modeling Overview

In this section, we clarify several terms, including user profile, user model and user modeling. We also briefly summarize user modeling history from user modeling components to user modeling shell systems and servers, to emphasize on the importance of a generic user modeling server.

2.1 User Profile

A user profile is defined as a collection of information about a user, including demographic information (name, age, location, to name a few), usage information (for example, pages visited, frequency of visit), and interests or goals (either explicitly stated by the user or implicitly derived by the system) [9]. It is an instance of a user model for a particular user. There are really two types of user profile data: those that describe individuals and those that describe groups of users.

2.2 User Model

Most adaptive web systems represent users via a user model. As defined by Kobsa [11], a user models is a collection of information and assumptions about an individual user (as well as a user group), which is needed in adaptation processes. To distinguish it from a user profile, we define it as an abstract representation which contains explicit assumptions on all aspects of users that may be relevant for the behavior of the system. It represents both individual users and groups into which users are classified. A user model combines user preferences with the stated goals or interests and the behaviors performed by that user, and uses this information to deduce the perceived current goals and interests of the user. Systems build a user model for describing individual or group users and distinguishing them in order to provide different services for different users.

There are two main types of user models that may exist in adaptive web systems: the overlay model and the stereotype user model [3]. An overlay model represents an individual user's information of each attribute defined for this user. For example, AHA! 2.0 [4] keeps every concept and associated attribute in the domain model of the application into the overlay user model. However, an overlay model has the problem of initialization because of the difficulty of collecting detailed user information. Stereotype user model distinguishes several typical or "stereotype" users. It is simpler than the overlay mode and is generalized from overlay attributes. For example, MetaDoc [1] uses stereotypes (novice, begin, intermediate, and expert) to represent a user's knowledge. The problem with the stereotype model is that many efficient adaptation techniques require a more fine-grained overlay model. One way to solve this problem is to provide a mapping from a stereotype to an overlay model. Because of the problems of two types of models, it may be better to combine them in the following way: stereotype modeling is used at the beginning to classify a new user and to set initial values for the overlay model, then a regular overlay model is used. Many systems, for instance web based adaptive education systems [16], use the combination of these two types of models.

2.3 User Modeling

User modeling is the whole process of constructing user models, and creating, updating or deleting user profiles. It contains the functions which are to incrementally build up a user model, to store, update and delete entries in instantiated user profiles, and to maintain the consistency of the model. We introduce two important processes of user modeling [3]: collecting data about users and processing the data to build or update user models.

User data may be gathered from a client side, a server side or from a proxy. There are also various ways for collecting data from a user. The traditional way is to let the user directly provide information (for example, age, location, gender, occupation, income) by filling in a form. However, the user may withhold information because of privacy issues. Usage information is the information that can be tracked for observing users' behavior. It is perhaps the most important user data, and is extracted from a web server log, which is the primary source of data in which the activities of web users are captured. Usage information may be described in terms of simple page views, transactions (which are "significant" events, and may combine multiple page views), and sessions (which are combinations of page views or transactions that together represent users' behavior) [18]. In addition to the simple sequence of events, information about time of access and frequency of access can also be captured as usage information. However, usage information is not fully reliable. The page clicked by a user does not guarantee that the user attentively reads its content. To make user modeling simpler and more reliable, it is necessary to involve the user in the process of user modeling to acquire additional information from the user.

Machine learning techniques are used to build a user model. Techniques that are widely used include linear models, TFIDF-based models, Markov models, neural networks, classification and clustering techniques, rule induction techniques, and Bayesian theory-based techniques. Data mining techniques such as association rule mining and maximal frequent sequence mining [17], are also used when building user models. In practice, many systems use various approaches and techniques to build up user models. Techniques and approaches should be chosen according to specific cases and needs.

2.4 Generic User Modeling Server

Another line of user modeling research focuses on generality of user modeling. In the early work, user modeling components were embedded into application systems, and were not distinguished from the components that perform other tasks. These systems used various machine learning techniques to construct different types of user models. The embedded user modeling components lack reusability and are only applicable to the adaptive systems that they belong to.

General user modeling systems and user modeling shell systems were first introduced by Finin in 1986 and Kobsa in 1990 [12], aiming at the development of integrated representation, reasoning, and revision tools that form an "empty" user modeling mechanism to meet the requirements of generality, expressiveness, and strong inferential capabilities. When filled with application-dependent user modeling knowledge, these shell systems would fulfil essential functions of a user modeling component in an application system. Some major user modeling shell systems for academic purposes have been developed, such as BGP-MS (Belief, Goal, Plan Management System) [13], GUMS (General User Modeling Shell) [5], UMT (User Modeling Tool) [2], and um [7].

User modeling shell systems become part of an application after being filled with application-dependent user modeling knowledge. They receive information about a user from the application only and supply the application with assumptions about the user. Many commercial user modeling shell systems have been developed using a client-server based architecture. They are not integrated into any applications, but communicate with applications through a network. User modeling servers are centralized user modeling components for more than one application (possibly for all applications with which the user interacts) and seem to have the capabilities of domain-independent user modeling [10]. These commercial user modeling servers abstract user models from application systems, and build them as a user model server so that more than one application with a similar domain can access the information from it. Typical examples of user modeling server include the Personalization Server, LMS (Learner Modeling Server) [14], and Personis [8]. Other commercial user modeling servers include Group Lens, LikeMinds, Frontmind and Learn Sesame [12].

3 GUMSAWS

We propose the GUMSAWS framework, as illustrated in Figure 1. In this figure, files are represented as rounded rectangles, databases are represented as columns, engines or components in the system are represented as rectangles, and arrows represent data flow between system components or between system components and data repositories. System components are grouped into four sub-systems, which are represented as dashed rectangles.

The Model Description (MD) consists of two components, the User Model Description (UMD) and the Group

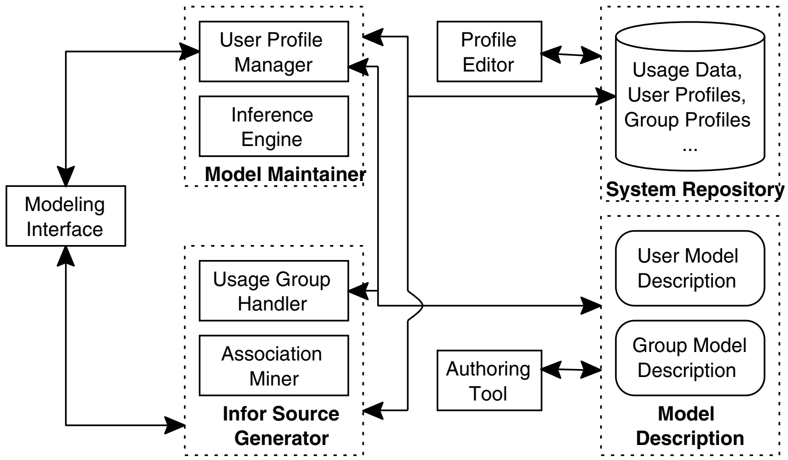

Figure 1. GUMSAWS Framework

Model Description (GMD). These are data repositories that store domain-dependent Intermediate Format Vocabularies (IFVs) and application-dependent description for user and group models. The IFV is the schema in intermediate format for describing concepts and relationships related to an individual user or group of users existing in adaptive web systems.

The Model Maintainer (MM) offers user modeling functions, including instantiating a user or group profile; storing, retrieving, updating and deleting entries in profiles; and inferring user missing property values. Two components, the User Profile Manager (UPM) and the Inference Engine (IE), are grouped into this sub-system.

The Information Source Generator (ISG) generates the information sources of groups information and association rules for the MM to update users' missing property values. Two components, the Usage Group Handler (UGH) and the Association Miner (AM), are included into this sub-system. User groups are generated by the UGH according to users' visiting history. Association rules are discovered by the AM. Rules indicate that amongst all properties of existing users, which values of other user properties also exist, given values of some particular user properties.

The System Repository (SR) is used to store usage data, user profile data, and group profile data. It also stores information sources of direct information, groups information, association rules, and general facts. The SR also connects some system components together. It is constructed in the initialization stage by some system components, such as the UPM and the UGH. It will also be updated by them during the system activity.

Note that some system components are not grouped into any of the four sub-systems. They provide interfaces for interactions. For example, the Modeling Interface (MI) is an interface between adaptive web systems and GUMSAWS. The main function of the MI is to forward adaptive systems' requests to components of GUMSAWS. The communication between the MI and the sub-systems is through a network. The Authoring Tool (AT) accesses model description and provides an interface for authors (i.e. system administrators) to specify the application-dependent user or group model description. Through the AT, authors may define user and group models, and default user and group profiles (referred to as "general facts" in this paper). The Profile Editor (PE) is implemented as an interface to allow users to see information held about them, and to modify their information. Requests from users received through the PE will be handled directly by the UPM as direct information about users. The PE makes adaptive web systems transparent in that users have full control on their information. In the following sections, we describe some of the system components in the sub-systems.

3.1 User Profile Manager

The UPM is responsible for providing adaptive web systems with information about users and their navigation patterns. The UPM is also in charge of instantiating users' profiles from user models and default values of user properties, and later on, updating profiles according to directly provided information from users. In the initialization stage, the UPM reads the User Model Description, creates database tables for user models, and inserts default property values (general facts) into the database. It also provides a list of services, such as checking the existence of a user, creating and deleting a user profile, and updating and retrieving a user property value.

3.2 Usage Group Handler

The UGH is responsible for generating user groups. A group model and default values of user groups are described in the Group Model Description. The UGH reads the description of a group model and creates database tables for storing user group information. The UGH then groups existing users together according to their interests, and assigns a user into an existing group based on the evaluation of the user's distance from the groups' centers. Users' interests are extracted from their visiting history and are represented by the number of pages that users have visited in each category. Moreover, the earlier the page has been visited, the less weight it will have in the category that the page belongs to, because users' interests might change over time. Therefore, a user's interest is represented by a vector of which each element is the number of pages that the user has visited in each category. The UGH determines how close the user is to each group center by calculating the Euclidean distance between the vector of the user's interest and the vector of each group center, which is the mean of all users' interest vectors in the group. Finally, the user will be assigned to the group whose center is the closest to the user.

3.3 Association Miner

The AM is responsible for providing the information source of association rules. It mines association rules from information about users. Association rules are amongst all properties of all users. They indicate that given values of particular user properties, which values of other user properties also exist. These association rules will be used to infer users' missing property values according to their existing property values. The support and confidence values of the association rules must be above a threshold. The extracted rules are ranked based on a measure that is calculated from the support and confidence values. A higher ranking implies that the association rule has the higher priority.

3.4 Inference Engine

The IE is in charge of inferring users' missing property values according to the four information sources and their reliability. Direct information about users is collected directly from the users of adaptive web systems through the PE and the UPM. Groups information is generated by the UGH. We believe that users who have similar interests will have properties in common. The most common property values are found for each group. If some property values of a user in a group are missing, they can be filled by the most common property values in this group. Association rules are discovered by the AM, and general facts are specified by authors through the AT.

Reliability of the information sources is defined as follows:

$$direct > groups > association\ \ rule > general\ \ facts$$

Direct information has the highest reliability because it is directly provided by users. We assume users would provide reliable information about themselves for the purpose of obtaining relevant responses from adaptive web systems. Groups information is more reliable than the information of association rules because groups information is generated within the scope of groups, whereas association rules are discovered within the scope of all existing users who communicate with the adaptive web systems. The information of general facts has the lowest reliability because they are found based on the statistics over property values of users who communicate with not only the clients of GUMSAWS, but also other adaptive web systems. Inference results from less reliable information sources can be overridden by that from more reliable information sources.

For a new user, property values are initialized according to general facts. These values will be updated if this user directly provides information about herself. To infer the user's property values, the IE first checks whether there are user properties whose values are not from the user's direct information. It will infer values of these user properties from the groups information according to which group this user belongs to, and which property values are the most common in that group. Before inferring user property values from association rules, the IE needs to check again whether there are user properties whose values are determined by the general facts. The values that are the most relevant for the user will be therefore assigned to the user.

4 Experimental Results

We examine GUMSAWS in the E-tailer domain to evaluate the performance of inferring users' missing property values. Our dataset is extracted from the KDDCUP2000 data, which contains clickstream and purchase data from Gazelle.com, a legwear and legcare web retailer that closed their online store on 8/18/2000. Information about 234954 user sessions and values of 296 properties for each user and each session are included in the dataset. Figure 2 shows the information about one of the user sessions and the user who performed this session. Records are separated by the ',' sign. Each record represents the value of its corresponding user property. For example, the first record in the figure represents the value of the property, 'Which Do You Wear Most Frequently'. The property values represented by '?' mark or 'NULL' indicate that no information has been collected for these properties.

We select the properties that have a large number of values provided by users. The four selected properties are 'pur-

Table 1. Information about Selected Properties

Property	Possible Values	# of Users	Most Common Value
purchase frequency	once a year, each week, every 6 months	2459	every 6 months
marital status	Inferred Married, Single, Inferred Single, Married	2974	Married
working or not	True, False	3527	False
gender	Female, Male	2423	Female

chase frequency', 'marital status', 'working or not', and 'gender'. Table 1 presents information about the possible values of these properties, total number of users who have provided information about each property, and the most common value for each property. The most common values and their corresponding properties represent general facts. For example, one of the general facts indicated in the table is that 'Female' is the most common value of the property 'gender'. General facts are mined from the whole dataset.

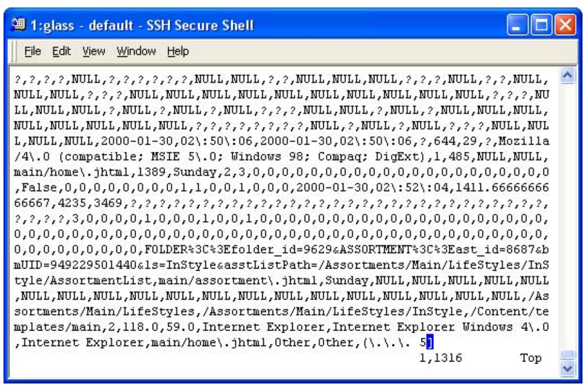

Figure 2. One User Session in the Original Dataset

The original dataset is preprocessed by extracting users who have registered and provided information about the four properties. After preprocessing, 1246 users are chosen to be involved in the evaluation. The values of those four properties and the information about users' navigation history (the number of pages visited in each category of products) are also extracted. To test the accuracy of inferring users' missing property values, we randomly set aside 10% of property values for each property. We repeat this process 10 times to produce 10 datasets. The rest is for training. From the training data, groups are generated by the Usage Group Handler, and association rules are discovered by the Association Miner. The Inference Engine will infer missing property values from groups information, association rules, and general facts. Property values that exist in the testing and training data are considered as users' direct information.

Accuracy of inference is calculated as the average ratio of the number of correctly inferred values for each of the four properties to the total number of missing values for this property. Results are presented in Table 2. The average accuracy is 67.6%, which is calculated after setting aside the highest and lowest values.

Table 2. Accuracy of Inferring Users' Missing Property Values

purchase frequency	marital status	working or not	gender	Accuracy
0.736	0.616	0.544	0.840	0.684
0.760	0.600	0.536	0.824	0.680
0.688	0.592	0.576	0.864	0.680
0.704	0.592	0.584	0.784	0.666
0.664	0.600	0.560	0.792	0.654
0.760	0.576	0.512	0.856	0.676
0.728	0.584	0.600	0.864	0.694
0.760	0.648	0.480	0.800	0.672
0.696	0.616	0.592	0.864	0.692
0.664	0.600	0.512	0.824	0.650
Average				0.676

We also carry out experiments to compare the inference performance by using different combinations of information sources. We define three notions as follows:

- **DI&GF**: the combination of direct information and general facts;
- **DI&GI&GF**: the combination of direct information, groups information and general facts;
- **DI&GI&AR&GF**: the combination of direct information, groups information, association rules, and general facts.

Experimental results are presented in Table 3. The results indicate that the combination of direct information, groups information, association rules, and general facts provides the best performance. The inference performance produced by the combination of direct information, groups information and general facts is better than that produced by the combination of only the direct information and the general facts.

Table 3. Comparison of Different Combinations of Information Sources

Test	DI&GF	DI&GI&GF	DI&GI&AR&GF
1	0.622	0.658	0.680
2	0.570	0.632	0.666
3	0.594	0.622	0.654
4	0.616	0.662	0.676
5	0.618	0.664	0.694
Average	0.604	0.648	0.674

5 Example of GUMSAWS in Use

The Personalized Electronic News System (PENS) [15] is an adaptive web news system. PENS presents news to users, taking advantage of context information such as users' location and behavior. It also adapts the presentation of news based on device characteristics, such as screen size and color capabilities. It is implemented to demonstrate the use of GUMSAWS. The use of our system here is to provide data sources that shape the dynamic aspect of PENS. The data provided by our system about users is used to make decisions for adaptation and for populating the under-construction page.

PENS partially imitates the NEWS@UNB website from which news items are gathered. News items are acquired from the news feed in the Rich Site Summary (RSS) format provided by the NEWS@UNB website. Three different types of web pages, the front page, the category based news page and the full news page, are generated by PENS. As shown in Figure 3, the front page has two parts. The left part lists news categories in the "NEWS SECTIONS" section. Clicking on one of the category names in this section will lead the user to the category based news page which shows the recent news items in this category. The right part lists three most recent news items with their titles and first few statements in the "TOP NEWS" section and four other news items with only titles in the "MORE TOPICS" section. Clicking on the title of the news will lead the user to the full news page.

Although the original NEWS@UNB website is static and does not provide any adaptation to users, PENS provides three types of adaptation, content adaptation, navigation adaptation, and presentation adaptation. It provides navigation adaptation based on user navigation history tracked by GUMSAWS. GUMSAWS keeps track of user navigation history which is composed of news items that the user has read. User navigation history also indicates how much the user is interested in each news category. User's interest in each category is determined by the number of news items in this category that have been visited by the user. For example, in the "NEWS SECTIONS" section on the front page, news categories are sorted and listed based on user interests in each category. As shown in Figure 3, the category "ACADEMIC" is on the top of the link list, which indicates that the user is more interested in the academic category than other categories.

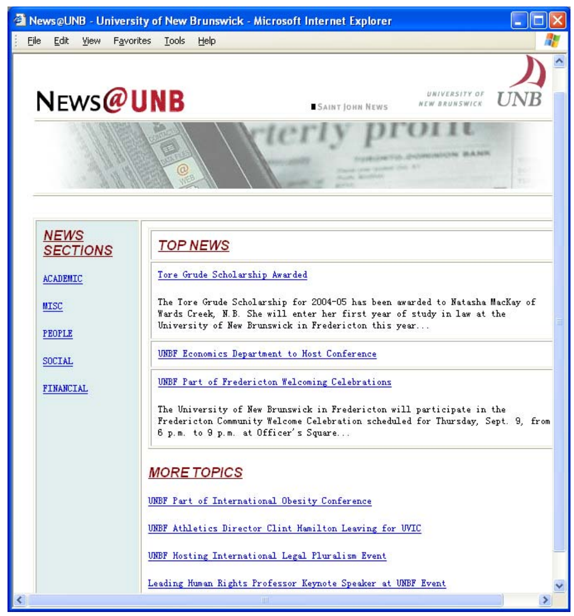

Figure 3. The Front Page

6 Conclusions and Future Work

We begin our work by clarifying the terms of user profile, user model and user modeling and summarizing the user modeling history from user modeling components to user modeling shell systems and servers. We then propose a generic user modeling server for adaptive web systems. Our system has been designed and implemented to provide basic user modeling functions, to have the capabilities of domain-independent user modeling, and to communicate with the adaptive web systems through a network. Overall, our system assists diverse adaptive web systems developed to provide concise guidance for users navigating web systems with very rich structure. It maintains and makes use of different information resources to perform better in inferring users' characteristics based on their navigation history. All in all, this research assists in providing personalized web service to users, of benefit in a variety of real world applications.

In future work, we will aim at developing a richer authoring tool. First, through the current Authoring Tool, authors define application-dependent user and group models and specify the default user and group profiles, based on predefined user and group vocabularies (definitions of concepts and relationships about users or groups). However, the user and group model vocabularies might not be complete enough to cover all user and group concepts and relationships in a specific domain. We will allow authors to define user and group concepts and relationships that do not currently exist in the predefined vocabularies. Second, the current Authoring Tool defines application-dependent user and group models for an adaptive system, independent from domain models of the system. However, user model definitions cannot be totally separated from domain models. For example, the overlay user model keeps every concept and associated attribute that are defined in the domain model of the adaptive system. It would be useful if authoring tools can import the domain models of the adaptive systems and allow authors to define user models from the concepts and associated attributes in their domain models.

The Profile Editor is implemented to make adaptive web systems transparent. However, the current PE can only allow users view and modify their profiles. It should be able to allow users to define which parts of the user model are allowed to be released for particular adaptive web systems. Users will have full control on the information sources that should be made available to each adaptive system. Users should also be allowed to view inference explanation for each inferred property value and to define which parts of the user model can be used for inference processes.

Currently, the communication between our system and adaptive web systems is through a protocol, which is not secure. Other possible secure protocols can be implemented in the system, such as SSL.

7 Acknowledgments

The authors generously acknowledge the funding from the Atlantic Canada Opportunity Agency (ACOA) through the Atlantic Innovation Fund (AIF) and through grant RGPN 227441 from the National Science and Engineering Research Council of Canada (NSERC) to Dr. Ghorbani.

References

[1] C. Boyle and A. O. Encarnacion. Metadoc: An adaptive hypertext reading system. *User Modeling and User-adapted Interaction*, 4, 1994.

[2] G. Brajnik and C. Tasso. A shell for developing non-monotonic user modeling systems. *International Journal of Human-Computer Studies*, 40:31–62, 1994.

[3] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modelling and User-Adapted Interaction*, 6(2-3):87–129, July 1996.

[4] P. De Bra, A. Aerts, D. Smits, and N. Stash. AHA! version 2.0, more adaptation flexibility for authors. In *Proceedings of the AACE ELearn'2002 conference*, October 2002.

[5] J. Fink, A. Kobsa, and A. Nill. *GUMS: A General User Modeling Shell*, pages 411–430. Springer, Berlin, Heidelberg, 1989.

[6] Y. Guan, A. A. Ghorbani, and N. Belacel. K-means+: An autonomous clustering algorithm. *Submitted to Pattern Recognition*, 2004.

[7] J. Kay. The um toolkit for cooperative user modelling. *User Modelling and User-Adapted Interaction, Kluwer*, 4(3):149–196, 1995.

[8] J. Kay, B. Kummerfeld, and P. Lauder. Personis: A server for user models. In *Proceedings of Adaptive Hypertext 2002*, pages 203–212. Springer-Verlag, 2002.

[9] M. Kilfoil, D. A. Ghorbani, W. Xing, Z. Lei, J. Lu, J. Zhang, and X. Xu. Toward an adaptive web: The state of the art and science. In *Proceedings of Communication Network and Services Research (CNSR) 2003 Conference*, pages 108–119, Moncton, NB, Canada, May 15–16 2003.

[10] A. Kobsa. User modeling: Recent work, prospects and hazards. In *M. SchneiderHufschmidt, T. Khme and U. Malinowski (eds.): Adaptive user interfaces: Principles and practice. Amsterdam: North-Holland*, 1993.

[11] A. Kobsa. Supporting user interfaces for all through user modeling. In *Proceedings of the Sixth International Conference on Human-Computer Interaction*, volume 1, pages 155–157, 1995.

[12] A. Kobsa. Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11(1-2):49–63, 2001.

[13] A. Kobsa and W. Pohln. The user modeling shell system bgp-ms. *User Modeling and User-Adapted Interaction*, 4(2):59–106, 1995.

[14] I. Machado, A. Martins, and A. Paiva. One for all and all in one - a learner modelling server in a multi-agent platform. In *the Seventh International Conference on User Modelling, Judy Kay (ed.), Springer Verlag*, pages 211–221, 1999.

[15] M. Nadjarbashi-Noghani, J. Zhang, H. Sadat, and A. A. Ghorbani. Pens: A personalized electronic news system. In *Proceedings of the Conference on Communication Networks & Services Research (CNSR 2005)*, Halifax, Canada, 2005.

[16] M. Virvou, V. Tsiriga, and M. Moundridou. Adaptive navigation support in a web-based software engineering course. In *Proceedings of the 2nd International Conference on Technology in Teaching and Learning in Higher Education*, pages 333–338, 2001.

[17] Y. Xiao and M. H. Dunham. Efficient mining of traversal patterns. *Data and Knowledge Engineering*, 39(2):191–214, November 2001.

[18] J. Zhang and A. Ghorbani. The reconstruction of user sessions from a server log using improved time-oriented heuristics. In *Proceedings of Communication Network and Services Research (CNSR) 2004 Conference*, pages 315–322, Fredericton, NB, Canada, May 19–21 2004.