PENS: A Personalized Electronic News System

Mehran Nadjarbashi-Noghani, Jie Zhang, Hossein Sadat K.M. and Ali A. Ghorbani
Faculty of Computer Science
University of New Brunswick
Fredericton, NB, E3B 5A3, Canada

E-mail: {m.nadj,jie.zhang,hossein.sadat,ghorbani}@unb.ca

Abstract

A framework has been developed as a basis for design and implementation of adaptive Web systems. This framework and the respective architecture support the idea of high-level synthesis and facilitate different types of adaptation. In this paper, we describe main components in the framework, and introduce an example application from e-News domain, a personalized electronic news system (PENS). PENS is implemented as the proof of concept and to demonstrate how Web pages are synthesized with different attributes from the same description, and to show adaptation based on users' behavior and client-side characteristics.

keywords: Adaptive Web Systems, Automated High-level Synthesis, Web Information Management, E-News.

1 Introduction

As the World Wide Web becomes larger in size and more complex in structure, it becomes more crucial for websites to guide the users to the relevant information, taking into account their background, interest, and goals. Adaptive Web systems are systems that adapt their response based on the context in which they are used [13]. The context is a set of system's parameters and variables that can change at any point of run-time. In adaptive Web systems, three main sources of information that constitute the context have been identified: user, environment, and technology [7]. The user preferences and browsing behavior are referred to as *user model*, which is the focus of most of the adaptive Web systems. The environment-related parameters such as the location of the user, or the time, might be considered as additional context information. Finally, technology-related parameters, such as client device characteristics and network bandwidth, are the other important elements of the context in an adaptive Web system.

The context information is used to provide the most relevant results for a user upon her request. For instance, the user model may determine the level of the user's knowledge based on which appropriate content is chosen to be presented. The client device characteristics, such as screen size, supported color depth, and supported markup language, can be used to generate the appropriate response for that device. This can be accomplished by scaling and converting images and text to compatible format. The context information can be obtained through monitoring the users' browsing behavior, mining the Web server log, querying the device characteristics, and so on. Besides, various AI techniques are used to extract useful and usable information from this raw data.

Adaptive Web systems use various context information to tailor the response of the system to the user's request through 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 (structure) adaptation adds, removes, hides, sorts, and 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, falls in this class of adaptation. Finally, presentation adaptation reformats the information fragments to achieve the appropriate final presentation for the current context. Resizing the images for a mobile device falls into this kind of adaptation.

Not all adaptive Web systems make use of all information context. Neither do all of them apply all three types of adaptation. However, it is expected that a general framework for adaptive Web systems provide facilities for the authors of such systems to harness the context information, to the extent that they need, to make a website as adaptive as possible.

The rest of the paper is organized as follows. In Section 2 some of the related works are reviewed. Section 3

explains the motivation of our work. Then, in Section 4, the architecture of the proposed framework is described. Section 5 describes an implemented example application based on our framework. System components designed for the mentioned application are discussed in Section 6. Finally, the conclusions and future work are presented.

2 Related Work

There are quite a few adaptive Web systems developed by researchers during the last decade (see [15]). These systems vary in application domain, platform, development methodology, levels of adaptivity, and so on. Some of them are specialized for a particular domain (e.g. online course), whereas others offer a more general framework. In this section, some of these systems are reviewed.

AHA! [5] is an adaptive hypermedia framework for developing educational applications. AHA! provides content adaptation (conditional fragments) and link adaptation (link coloring). In this framework, the domain is modelled through concepts and relationships between them. Concepts can be related to a resource (e.g. a page). AHA! adapts the pages based on the user model. It has a predefined page structure in the sense that the pages are not synthesized, however, the author can include conditional fragments so that if some conditions hold, the fragment is not shown.

InterBook [14] is a tool for authoring online adaptive textbooks. It uses a domain model of concepts and a user model to provide adaptivity. In addition to regular navigation support (back and forward), InterBook provides an adaptive set of links between the textbook and the glossary based on the current user's knowledge. It also provides visual cues about each link (adaptive annotation) and direct guidance about the suggested next page the user should visit.

SETA [4, 3] is a prototype toolkit for building adaptive Web stores. It dynamically generates the pages of a Web store catalog and selects the content of the pages based on user interest and familiarity with the products. Also, the system sorts the available items for a product class based on user preferences.

SeAN [2] is a an adaptive system for personalized access to news. It uses a structured hierarchy to represent news. Each piece of news is a composite entity that has several attributes that define its components. For example, title, abstract, full text, author, pictures, or video can be attributes of a piece of news. Based on this representation, different detail levels can be used for news according to the user model. SeAN adapts to the context in three ways; it selects the news topics relevant to the user; it presents an appropriate level of detail of the news based on the user model; and it provides advertisement most relevant to the current page and the user.

Automated Website Synthesis [8] is an approach for design and maintenance of website applications, which is based on computational logic. The website code is derived from its content via automated synthesis. The synthesizer is arranged in three different levels; it starts from a high level description of the website and produces an intermediate representation for the application using a domain-dependent formal language. Then the website code is automatically generated from the intermediate representation. The key idea in this approach is separating information content from its presentational form. Hence, different visualization descriptions are supported for the same site description.

3 Motivation

The result of any sort of adaptation, i.e. content adaptation, navigation adaptation, or presentation adaptation, should be eventually reflected in the pages that are returned to users' Web clients. This requires any adaptive Web system to prepare pages dynamically based on adaptation rules and context information. Many systems (see Section 2) use pre-styled templates and populate them with actual data in order to render the final page. The modifications related to adaptation can range from changing the attribute of a page component, such as changing the color of a link that is being accomplished in AHA!, to adding or removing a composite page component, like what is being performed in the page generation process of SeAN.

Although most of the adaptive systems and frameworks are designed for a particular application domain such as e-Tailer or e-News, the website designer should still think and design in terms of hypermedia concepts, e.g. Web pages and hyperlinks, and manage the Web system in terms of Web components. The adaptation rules are also specified to act on the page and page components. Separation of the role of application designer and manager from the role of Web designer requires automated Web synthesis for adaptive hypermedia, where a complete synthesis process starts from a high-level description and results in dynamically-produced Web pages. Because of the adaptation needs, some website features, like layout of the pages, should be generated on-the-fly and some other features, like the structure of the website (to be shaped by inter-page links), should be alterable.

We believe that presentation adaptation must be the least concern of the website designer. Each synthesis process should consider the capabilities as well as limitations of the client browser, and perform some automatic adaptation accordingly. As an example, the page that is generated for a desktop browser is not appropriate for showing on a hand-held device; the hand-held devices have screen size and page size limitations, and many of them require a different markup language coding such as WML. The page syn-

thesizer should be able to generate code for different user clients from the same site description.

4 Framework for Automated Synthesis

PENS has been developed based on a framework for adaptive Web systems [9].This framework provides an architecture based on which adaptive Web systems, for different domains, can be built. The core components in the architecture are the Synthesis Engine, the User Front-End, and the Conceptual Tasks. The architecture also defines the communication protocol between these components. In the following sections, these component are explained.

4.1 User Front-End

The User Front-End (UFE) component is the initial point of contact where HTTP requests arrive. This front controller provides a centralized entry point that controls and manages Web requests from different clients. It is designed based on the core J2EE Front Controller design pattern [16], in which the rationale behind such a centralized point of access is well explained. The UFE basically transforms a Web request into a synthesis request and sends it to the Synthesis Engine. Then, the response of the synthesis engine, which includes the page and its components, is retrieved and stored to be sent back to the browser upon individual requests. Hence, the UFE creates a level of abstraction between the Web tier and the adaptation server. Moreover, the UFE keeps track of sessions and generates user and device IDs for the new users and devices that connect to the system.

4.2 Conceptual Tasks

Conceptual Tasks are sources of data that shape the dynamic aspect of the system. The SE uses data provided by CTs in order to make decisions for adaptation. It also might be used for populating the under-construction page. CTs are application-specific in general, and can be provided as ready-to-run applications plugged-in to the system or can be generated by the compiler from the high-level description. Some CTs, like the User Profile Managers, are application-independent and present in almost all adaptive Web systems.

4.3 Synthesis Engine

Synthesis Engine composes a Web page based on the incoming request, current context, and various information sources. For each synthesis process, the start point is finding the correspondence between the incoming URI and a concept in the Site Description (SD). When the URI contains solely the name of website (like the one which is usually received as the first request from a user) then the Synthesis Engine will look for a page with the "Default Page" property, and start the page composition from that point. If the incoming URL contains a specific requested page, then the SE will query the SD for that particular page definition.

The SE knows only semantics that are defined in the Intermediate Format Vocabulary (IFV), including the hypermedia concepts and the defined relationships. It is presumed that the application website is defined in terms of the IFV. The SE starts with the 'base' concept (which is often a page concept) and explores all the defined properties and relationships of that concept down to the point that all constituting components of that concept are known and all page-population data is present. At this point, all properties of all page components have final values except for those properties that are related to the page layout.

As the SE goes through the mentioned process, it constructs a graph with a hierarchy pretty much the same as the concept hierarchy defined in the SD. Despite the SD, this constructed graph does not have any uncertain relationships (relationships that define the alternatives in realization of a component or bind the presence of a component to a constraint) and represents a page that observes the defined constraints. If data of a component (e.g. a text component) is supposed to be provided by a CT, then that CT will be contacted to retrieve the data. Note that all constraints and all system operations (like those that specify how and which CTs should be contacted) are parts of the SD.

Page layout generation process encompasses solving two essential problems:

1. Placement: the relative position of each page component, in other words, the order the components are going to be placed in the page.

2. Formatting: issues related to the appearance of each component. For example, for text components, the appropriate values for properties like font, color, size and so on are decided. For an image component, this may include properties like resizing factor, converting to B&W, and so forth.

A page might include one or more tables to implement the placement algorithm. The tables can appear at any level of page hierarchy, and they can be nested. There is a set of constraints in the IFV that are related to page-layout. These constraints are used to describe guidelines for the layout generator to determine the appropriate place of each component in the respective table.

4.4 Site Description in Intermediate Format

The Intermediate Format Vocabulary provides the author or compiler with the schema to describe the structure

and content of a website in terms of hypermedia concepts and relationships. A typical site description contains user model, user group model, content model, navigation model, and presentation model. The IFV provides the following classes of concepts and relationships:

- Concepts and attributes related to hypermedia building blocks. Page, text area, image, link, table, and page meta data are examples of concepts; and color, length, size, and format are examples of attributes.
- Concepts, relationships, and constraints that concern page-layout generation: Alignment, PlaceNotLeft, KeepTogether, and style, to name a few.
- Concepts and relationships related to system operations that instruct the SE how to achieve data required for adaptation decisions and page population.
- Concepts and relationships related to presentation and final code generation. Structure and tags of different markup languages fall into this category.
- Concepts and relationships for user modeling and user group modeling. Only those parts that are shared across all application domains will be introduced into the IFV.

The IFV and site description are described using Resource Description Framework (RDF) serialized in XML. RDF and RDF Schema [17] are extensible and scalable standards intended for description of meta-data as well as machine understandable semantics.

5 Example Application

Our system partially imitates the NEWS@UNB website, which news items are gathered from. Two different types of pages, the front page and the full news page, are generated by the system. As shown in Figure 1, the front page lists three most recent news items with their titles and first few statements in the "LATEST NEWS" section and four other news items with only titles in the "MORE TOPICS" section. Clicking the title of the news will lead the user to the full news page. As shown in Figure 2, the full news page presents full information of a news item, including its topic, publication date, correspondent, and news body. The news banner is located at top of both front page and full news page.

Users may see different pages because of adaptation provided by our system. Different criteria have been used for adaptation, including user location, user navigation history, and different types of devices. Detailed explanation of the adaptation is presented as follows.

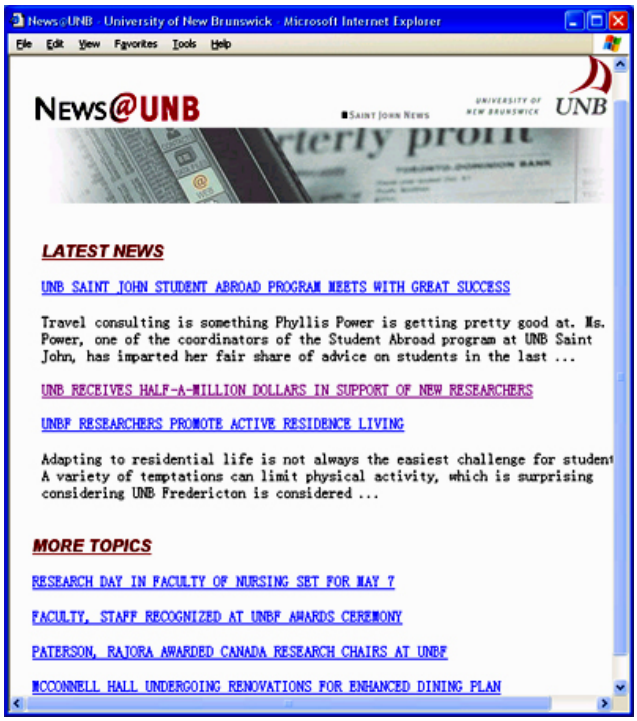

Figure 1. The Front Page

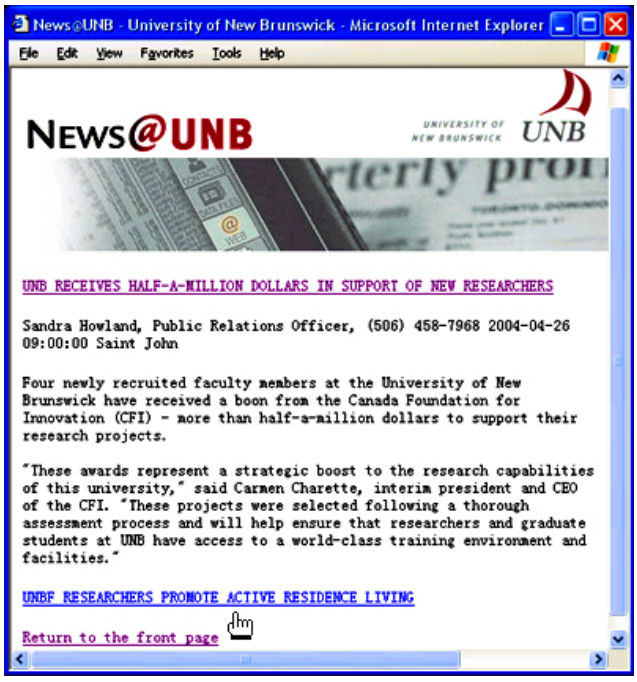

Figure 2. The Full News Page

5.1 Adaptation to User Location

User location is detected by one of the conceptual tasks, the IP Locator. Our system detects users from UNB's two major campuses, the Fredericton campus and the Saint John campus. Users from different campuses will see different front pages. The first difference is the news banner. Without adaptation, news items on the front page are sorted based on their publication date. Since we can assume that users from the Fredericton campus are more interested in Fredericton news, those news items related to the Fredericton campus will be placed above news related to the Saint John campus if the user connects from the Fredericton campus. Similarly, users from Saint John will see news related to the Saint John campus at top of the "LATEST NEWS" section.

5.2 Adaptation to User Navigation History

User navigation history keeps track of news items that the user has read. It also indicates how much the user is interested in each news category. We provide two types of adaptation based on user navigation history. In the "LATEST NEWS" section on the front page, the first few statements of the news item that the user has read will not be shown any more. This adaptation is based on the assumption that users would be not interested in reading the first few statements again if they have already read the full body of the news. The example is the second news item in the "LATEST NEWS" section as shown in Figure 1. Another adaptation is news recommendation, which is an example of navigation adaptation. As indicated by the cursor in Figure 2, there is a link to a related news item at the bottom of the full news page. This news item is the recommendation provided by the Association Miner (to be explained in Section 6.5).

5.3 Adaptation to Different Devices

Mobile devices have a smaller display compared to the desktop PCs. Therefore, the components of a page should be adapted to the client device, so that the final presentation remains elegant. Besides, some devices accept pages in different markup languages such as WML. Thus, if a device's preferred language is WML, then the final presentation will be generated in WML. On the other hand, recent mobile devices also support XHTML pages, so that richer content can delivered to the device. In either case, the images in the presentation need to be converted to the proper format. For instance, devices that only accept WML code, cannot show images in formats other than WBMP. Hence, the system dynamically converts all the images to WBMP at runtime. In addition, the images are resized to fit the device's screen. Figure 4 shows a Nokia device that prefers WML

Figure 3. The front page on a Blackberry device

page. The banner images have been converted to WBMP and delivered along with the WML page to the device. Figure 3 shows a Blackberry device accessing the front page of the sample application. Since Blackberry prefers content in XHTML format, the page has been delivered in XHTML. Also, the banner images have been resized.

6 Conceptual Tasks of e-News Application

6.1 IP Locator

The user's geographical location, as part of context information, can be used in Web applications for various purposes such as geographical access control and content and advertisement targeting. In adaptive Web systems, the location of the user can be used to provide more relevant information. For instance, the sample application provides each user with the most relevant news items at the top of the page. The IP Locator conceptual task provides a service through which the location code for an IP address can be retrieved.

6.2 Hand-held Profile Manager

In contrast to desktop PCs, which have all the necessary means to support rich Web features, hand-held devices suffer from hardware (and as a consequence, software) limitations, some of which are small screen size, limited processing power, and small memory. The need for Web support for hand-held devices on the one hand, and the existing device limitations on the other hand, have pushed researchers

Figure 4. The front page on a Nokia and also an Openwave device

and developers to develop new technologies to enable Web access for hand-held devices. Different protocols and languages have been developed during recent years to fulfill Web access requirements for hand-held devices.

In order for Web content servers to provide content for a wide range of devices with different capabilities, the capabilities of the target device should be taken into account when generating the content. However, there are too many types of devices to keep track of, and there are many new devices appearing in the market. Therefore, it is not practical to store different versions of content to be served for different devices. The solution is to inquire about device capabilities from the device itself so that an application can optimize the content based on the stated capabilities of the device.

The User Agent Profile (UAProf) [6] is the proposed standard, by the WAP Forum, for device capabilities specifications. The UAProf describes the capabilities of the hardware, software, supported technologies, and the browser of the device. This information is communicated in an XML document to the server to be used to tailor the response for the device.

The Hand-held Profile Manager (HPM) component, requests information from the client device UAProf and stores the attributes of the device in the database. Later on, the HPM responds to the requests of the SE for device attributes.

6.3 User Profile Manager

The User Profile Manager (UPM) is responsible for providing information about users and their navigation patterns. The UPM is also in charge of instantiating users' profiles from the user model and the default values, and later on, updating the profile. In the initialization stage, the UPM reads the description of the user model and the default values, creates database tables for the user model, and inserts default values into the database. It also provides a list of services, such as checking the existence of a user, creating and deleting a user profile, updating and retrieving a property value, and so on.

The user model is described in intermediate format for describing concepts and relationships existing in the application based on the User Model Vocabulary. The vocabulary is domain dependent but application independent. It contains ontology in a specific domain (e-News domain in our case).

6.4 Usage Group Handler

The Usage Group Handler (UGH) is composed of two components, the Usage Group Miner and the Group Matcher. Like the user model, the group model and the default values are described in RDF format based on the Group Model Vocabulary for the e-News domain. The UGH reads the description of the group model and creates database tables for storing user group information. Then, the UGH groups users together through the Usage Group Miner and assigns a user into an existing group through the Group Matcher based on the evaluation of the user's distance from the groups' centers.

As mentioned, the Usage Group Miner is responsible for grouping existing users together according to their interest in different news categories. The number of news items in each news category that have been read by the user represents the user's interests. Moreover, the earlier the news item has been read, the less weight it will have in the category that the news story belongs to because the user's interest might change over time. A new clustering technique called K-means+ algorithm is used in the implementation of the Usage Group Miner. K-means+ algorithm has been developed based on the K-means algorithm [10, 11]. The number of user groups will be generated by the Usage Group Miner. As explained earlier, a user's interests are represented by a vector of which each element is the number of news items that the user has read in each category. The Group Matcher 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.

6.5 Association Miner

The Association Miner (AM) here is used to recommend news items strongly related to the news stories that users are currently reading. Recommendations are provided based

on association rules discovered by the AM. The association rules are mined from visiting histories of the users in the same group. User's visiting history is made up of visited news items in all previous sessions. The association rule here indicates, given a particular news item read by users in a group, which other items the users have also read. The Apriori algorithm [1, 12], an unsupervised learning algorithm, has been implemented to mine the rules. The support and confidence values of the association rules must be above a threshold to be considered. The extracted rules are ranked based on a measure that is calculated from the support and confidence values. Moreover, some popular news items or the latest news items might also be recommended if the number of qualified news items for recommendation is not enough.

6.6 News Feeder

The News Feeder (NF) is in charge of providing news items to the Synthesis Engine. It also has a component called the News Retriever, which continuously retrieves news stories from news resources, extracts news features, and stores those features in the news database. News items are provided in XML format based on RSS specification. The features extracted from news items include name of correspondent, title of news items, date of publication, and location, as well as the full body. News resources will be reread after a certain period of time and new news items will be stored in the database.

7 Conclusions

We introduced a framework and architecture as a basis for design and implementation of adaptive Web systems. We also discussed those parts of the framework that have been implemented, including the User Front-End, Conceptual Tasks, the Synthesis Engine and the Site Description. An example application, PENS, was implemented and demonstrated as the proof of concept. It currently adapts to user location, user navigation history and different devices. Conceptual tasks of PENS, including the IP Locator, the Hand-held Profile Manager, the User Profile Manager, the Usage Group Handler, the Association Miner and the News Feeder, have been also described. Currently, we are working on the language for high-level description, and developing a compiler for that language is on our research agenda.

8 Acknowledgments

This work was funded by the Atlantic Canada Opportunity Agency (ACOA) through the Atlantic Innovation Fund (AIF) and through grant RGPIN 227441 from the National Science and Engineering Research Council of Canada (NSERC) to Dr. Ali A. Ghorbani.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference of Very Large Databases (VLDB)*, pages 487–499, Santiago, Chile, September 1994.

[2] L. Ardissono, C. Barbero, A. Goy, and G. Petrone. Adaptive web stores. In *Agents'99 Workshop: Agents for Electronic Commerce and Managing the Internet-Enabled Supply-Chain, Seattle, WA, May 1999*, pages 9–13.

[3] L. Ardissono, L. Console, and I. Torre. An adaptive system for the personalized access to news. *AI Commun.*, 14(3):129–147, 2001.

[4] L. Ardissono, A. Goy, R. Meo, G. Petrone, L. Console, L. Lesmo, C. Simone, and P. Torasso. A configurable system for the construction of adaptive virtual stores. *World Wide Web*, 2(3):143–159, 1999.

[5] P. D. Bra and N. Stash. AHA! Adaptive Hypermedia for All. In *the SANE 2002 Conference, Maastricht*, pages 411–412, 2002.

[6] M. H. Butler. Implementing content negotiation using CC/PP and WAP UAProf. Technical Report HPL-2001-190, Information Infrastructure Laboratory, HP Laboratories Bristol, August 2001.

[7] M. Cannataro, A. Cuzzocrea, and A. Pugliese. XAHM: an adaptive hypermedia model based on XML. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, pages 627–634. ACM Press, 2002.

[8] J. M. B. Cavalcanti and D. S. Robertson. Synthesis of web sites from high level descriptions. In *Web Engineering, Software Engineering and Web Application Development*, pages 190–203. Springer-Verlag, 2001.

[9] I. A. S. group. Adaptive web sites (AWS) framework high-level design document. Technical Report TR03-102, Intelligent and Adaptive Systems Research Group, Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada, December 2003.

[10] Y. Guan, N. Belacel, and A. A. Ghorbani. Y-means: a clustering method for intrusion detection. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE-2003)*, pages 1083–1086, May 4-7 2003.

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

[12] M. Joshi and V. Kumar. Tutorial on high performance data mining. In *Fifth International Conference on High Performance Computing (HiPC'98), Chennai, India; and at Joint International Parallel Processing Symposium/Symposium on Parallel and Distributed Processing (IPPS/SPDP)'99, San Juan, Puerto Rico*, 1999.

[13] 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.

[14] J. E. Peter Brusilovsky and E. Schwarz. Web-based education for all: a tool for developing adaptive courseware. In *Computer Networks and ISDN Systems, Vol. 30, Nos. 1–7*, pages 291–300, 1998.

[15] H. Sadat and A. A. Ghorbani. On the evaluation of adaptive web systems. In *WSS04, The Second International Workshop on Web-based Support Systems in conjunction with AI 2004*, pages 127–136, Beijing, China, September 2004.

[16] Sun Microsystems Inc., http://java.sun.com/j2ee/. *Java 2 Platform, Enterprise Edition (J2EE)*.

[17] W3C, http://www.w3.org/RDF/. *Resource Description Framework (RDF)*.