A Personalized Multimedia Web-Based Educational System with Automatic Indexing for Multimedia Courses

O. Shata

Abstract— **This paper proposes a personalized web-based system with evolving automatic indexing that may be used in multimedia courses. Instructors may use the system to create and deliver different presentations, and individual learners may use it to retrieve resources personalized according to their different learning needs. Unlike similar systems which have a repository of fixed resources, this system uses a web crawler that harvests the web for multimedia resources to continuously update the links in its database and to continuously check the validity of the stored links. This makes the system always up-to-date. The importance of the system lies in its new approach to represent web resources, and in using continuously updated resources to provide responses adapted to users needs while saving the users the need to interact directly with the web.**

Index Terms— **Automatic indexing, human computer interaction, multimedia, web-based.**

I. Introduction

Many instructors depend on PowerPoint presentations for the illustration of teaching materials in their lectures. Such presentations may work well for many courses. However, in a course such as "Multimedia Systems", learners would expect more than just a static presentation. Usually students enroll in a multimedia course and they have some high expectations in advance based on their interaction in their lives with multimedia applications (e.g. computer games) and based on what they hear or read about multimedia applications, such as they facilitate interactive accessing of multiple media and which could be online via a network. Learners usually expect that they shall have a unique learning experience that makes heavy usage of animation, images, audio, and video technologies which facilitate tailored learning in real world context, probably with frequent instant access to information, resources, and events via the World Wide Web. However, giving learners what they expect is one of the major challenges an instructor may face. In this paper, we propose a personalized web-based system with evolving automatic indexing that may be used in multimedia courses by both instructors to prepare and teach multimedia presentations, and by learners to retrieve and learn multimedia information. Most of the adaptive web-based teaching and learning systems depend on selecting material from a repository of fixed resources according to some criteria that is supported by user models. Unlike those systems which have a repository of fixed resources, this system uses a database of links to multimedia resources. The links may point to local resources on a local server, or may point to other resources on the web. The system uses a web crawler that harvests the web for multimedia resources to continuously update the links in the database and to continuously check the validity of the stored links. This makes the system always up-to-date. We have developed the web crawler ourselves and this gave us the ability to control the indexing. The system has been developed to be used by various multimedia instructors and many multimedia learners. Different instructors may use the system to create and deliver different presentations, and individual learners may use it to retrieve resources personalized according to their different learning needs. The system provides many advantages to both instructors and learners. Instructors can create presentations in a short time from a combination of local materials and remote ones with links to some dynamic resources on the web. Learners do not have to search the web themselves and examine many unrelated documents to find useful resources, but rather they submit queries and get individualized answers delivered to them according to their individualized learning needs based on their learners' models. The answers are from a dynamic and continuously updated database of resources (many of them are on the Web fetched by a web crawler that was developed and programmed locally) and which are highly rated by instructors and peer learners. This saves them effort and time. The system was implemented to serve a course in "Multimedia Systems"; however, it may be adopted to serve other courses as well.

II. RELATED WORK

The work in this paper can be linked to work in the areas of adaptive hypermedia systems [12, 13], personalized e-learning [10, 11], user modeling [1, 2, 3, 5, 14, 15], semantic web modeling [6, 17, 18], and information retrieval [16]. Interested readers may refer to those references for detailed information.

III. SYSTEM ARCHITECHTURE

In order to make the system available to a large number of

Manuscript received February 27, 2007.

O. Shata is with Qatar Universitys, Doha, Qatar (phone: 974-485-1172; fax: 974- 485-2124; e-mail: sosama@qu.edu.qa).

students and instructors, it was designed as a web-based application that follows a three-tier client/server architecture: data tier, server tier, and client tier.

A. The Data Tier

The data tier consists of a file system and a database.

1. The File System

The file system consists of any presentations that have already been developed by instructors, in addition to multimedia files and resources of any type (audio, video, image, and text) and which exist locally on our server.

2. The Database

The database consists of the following:

- An index of multimedia resources. The index includes links to files in the file system, and links to other multimedia resourced on the web and which have been gathered by the web crawler. The index is not displayed to users since the system aims to provide users with resources and presentations adapted to their individual needs. Hence, it is incorporated into the searchable database. Each index entry consists of a hyperlink to a resource with its metadata (e.g. title, headings, type, size, keywords, etc.). The index is organized in an alphabetical order, but also allows for searching by synonyms (e.g. multimedia and multiple media), acronyms (e.g. Graphics Interchange Format and GIF), and phrase inversion. In addition, each entry in the index has some other information such as technical level (introductory, intermediate, and advanced), instructors rating, and learners rating. The ratings are updated whenever a user retrieves or uses the resource using a formula to reflex the accumulated ratings.
- A semantic network for multimedia keywords. A list of multimedia keywords has been developed and a semantic network is used to describe the relationships between the keywords (e.g. direction of relationship. Each keyword (node) in the semantic network has characteristics associated with it (e.g. level of difficulty, knowledge). This will be used to comprehend the user query and to provide an augmented reply.
- Personal information about the learners and which is needed to provide resources adapted to their needs (e.g. level of knowledge, learning preferences, etc.). At the beginning, this part of the database is empty. With our approach to provide users with adapted resources, it is necessary to continuously monitor their learning process and build a user model for each user based on his/her interaction with the system (e.g. keywords in queries, his /her rating of the reply). This model is used by the system to adapt its replies to a user when s/he uses the system subsequently. When a learner enters a query, usually in the form of multimedia keywords (e.g. mp3 compression), each keyword is checked against the semantic network of the multimedia keywords and which we have built. The system identifies the characteristics of each keyword in the query and the neighbor keywords (nodes) in the semantic network, consults the user model, and finally checks the index for the proper resources that suit that user, order them in a presentation and returns them to the user.
- Learners and instructors' assessment. All users are asked to assess the resources recommended to them by the system; the assessment provided by a learner is used in building the learner model. The assessments provided by both learners and instructors are used to construct a rating matrix for the resources. The rating matrix is used to filter the resources, cluster them into groups, deciding on those resources that must be removed from the database, focus on resources with high ratings by extracting more links from them, explore those links further in a breadth first techniques. Hence, the database is dynamic and continuously updated to better adapt the system to its users.

B. The Server Tier

This is the application tier. It consists of five main modules.

1. The Supervisor Module

This module is responsible for communicating with the users and for controlling the work of the other modules. It receives queries from users and decides how to deal with them. If the user is an instructor, then the Supervisor would ask for extra information about the required resources such as the types of resources s/he needs (text, image, audio file, etc.), level (introductory, intermediate, advanced), and content (keywords to be included). Then the Supervisor forwards the query to the Composer module. If the user is a learner, then the Supervisor forwards the query to the Composer and activates the Monitor module without asking for extra information. Upon receiving an answer from the Composer, in the form of files, it forwards those files to the Presenter module which decides on an order to glue those files together. The Presenter then returns the glued files as one component representing an answer to the user's query. The Supervisor finally forwards that answer to the user. This module is also responsible for receiving assessment of the user and updates the ratings of the resources accordingly. The Supervisor also sends a copy of the learner's assessment of the returned answer to the Monitor module.

2. The Composer Module

This module is responsible for returning an answer to the user query in the form of several multimedia files, which are then forwarded by the Supervisor to the Presenter. That answer is adapted to the user depending on his/her user model. Upon receiving a query from the Supervisor module with information about the user (instructor / learner) it works to find an answer. If the user is an instructor then it searches the index / database for resources that make best match with the instructor's request, extracts the links of the resources, and forward them to the Supervisor module to forward them to the instructor. If the user is a learner and the composer receives the query forwarded by the Supervisor, usually in the form of multimedia keywords (e.g. mp3 compression), the key words are checked against the semantic network of the multimedia keywords and which we have built, and which is represented in the database. The Composer identifies the characteristics of each keyword and the neighbor keywords (nodes) in the semantic network, consults the user model, and finally checks the index for the proper resources that suit that user, extracts their links, decides on an order to compose a presentation, and returns the representation to the Supervisor which forwards it to the user. The composer uses a set of formulas when evaluating a resource to be adapted to a specific learner such as:

total number of satisfied learners within the same category as the current learner / total number the resource was retrieved for learners in that category, and

total number this learner was satisfied with resources with the same difficulty level / total number of times resources were retrieved for that learner.

3. The Presenter Module

This module receives from the Supervisor the material prepared by the Composer, in the form of separate files, and decides on an order to glue those files together. The Presenter works as an automatic content management module. It uses metadata of the files and some defined rules to order the material. Such rules include:

- Order the retrieved files according to the order of the keywords in the user's query.

- For files that are ordered according to the same keyword, use the following order to activate those files: image, text files, animation, and finally video files. Audio files may be played only upon user's request.

The presenter returns the glued files to the Supervisor to be forwarded to the user. In addition to the order suggested by the Presenter, the supervisor also returns the files returned by the composer categorized by type (e.g. text, audio, video, animation, … etc.) and gives the option to the user to play them according to his / her preference.

In this version of the system, the Presenter is simple and depends on syntactic words and metadata. Later, we intend to use semantic data in the files for proper ordering and for inserting images in proper locations in text files.

4. The Monitoring Module

This module is responsible for building learners models to be used by the Composer in customizing returned answers. In particular, it:

- receives a copy of the query submitted by the learner from the Supervisor, analyses the keywords in the query by consulting the index and the semantic network,
- receives a copy of the learner's assessment of the returned answer via the Supervisor and analyses this assessment,
- also gathers other data about the learner's behavior (e.g. the time spent by the learner on a resource that s/he has assessed to be useful). The system uses a database to manage the multimedia files.

The Monitoring module uses the data gathered from the processes above to update the learner's model accordingly.

5. The Web Crawler Module

The basic design and working techniques for crawlers may be found in several seminal publications [4, 9]. The crawler builds an index that is saved in the database. The index is updated based on the ratings obtained from users. If a resource is rated as un-useful by instructors then the link to that resource is deleted from the database with other links obtained from that resource. When a resource is rated as valuable then the crawler focuses on that resource, extracts links from it, and proceeds further based on those links. Although there are commercial crawlers and even free crawlers available [4, 9, 19], however, we preferred to program a web crawler because this enables us to build our special-purpose index (i.e. based on content stored in multimedia files), maintain mirror sites for some identified Web sites (if needed), test web pages and links for metadata, and evaluate each fetched page to decide if it should be stored or not.

C. *The User Tier*

Users are of two types: instructors and learners. An instructor may submit a query asking for resources on a multimedia topic, usually in the form of multimedia keywords. The instructor also defines the level of difficulty of the required resources, maximum number of resources, and their type (e.g. text, image, audio, and video). In this sense the system works as a retrieval system. Instructors do not have user models stored. However, each instructor is responsible for assessing and rating the retrieved resources. A learner may submit a query about a multimedia topic, usually in the form of multimedia keywords. The Supervisor module receives a query and processes it, as described earlier, and returns a presentation in the form of links. The user clicks on each link to activate the corresponding part of the presentation.

IV. SYSTEM IMPLEMENTATION

The implementation of each component of the system is described next.

A. *The File System*

The file system consists of the multimedia files of any type such as GIF, JPEG, MP3, PDF, and MS-word which exist locally on our server. It also consists of any PowerPoint presentations that have been developed and saved on the server. For each file stored there is a link to it in the database as well as metadata that include (short description of content, keywords, type, size, level of difficulty, rating by instructors, rating by learners,… etc.).

B. *The Database*

The system uses a database to manage the multimedia files. Our institute has a license to use Oracle. However, we preferred to use the free and open source MySQL database because we wanted the system to be practiced by students; and the use of MySQL will give students a wide opportunity to examine the code, and to participate in developing applications fast. In addition, the database will be accessed via the World Wide

Web and MySQL adapts well to HTML. The interaction between a scripting language like PHP, which we used, and MySQL to store data is very smooth. We had two options to store the multimedia files in MySQL. The first was to use BLOB's which can be used to store all file types such as GIF, JPEG, MP3, PDF, and MS-word. However, this choice has the disadvantage that the database would grow so fast since multimedia data files are usually of large sizes. The second choice was to keep the multimedia files outside the database in the form of a file system, and only store the links to those files in the database. We preferred the second option because it allows: files to be loaded quicker, images to be cached easily on a user's machine, and provides one consistent method for storing links to resources on our server and on the Web. The database also stores learners' models and the index created for the web resources.

C. The Crawler

We have programmed a crawler in Java and which uses a centralized manager that decides which URLs are to be visited, and which URLs should be stored. Those URLs are analyzed, and selected ones are stored in the MySQL database to be used by instructors and learners. The choice of Java goes along with the idea of having our system platform-independent. Each fetched page is parsed to extract existing hyperlinks needed by the crawler to continue working, and to obtain other information required for indexing. The index is updated based on the ratings obtained from users. If a resource is rated as un-useful by instructors then the link to that resource is deleted from the database with other links obtained from that resource. When a resource is rated as valuable then the crawler focuses on that resource, extract hyperlinks from it, and proceed based on those hyperlinks for a longer depth. Initial pages required by the web crawler are provided by instructors. The web crawler visits the extracted links using a breadth-first technique which is believed to result in better results [8]. Links that are retrieved by the crawler are stored in two forms: first, as a semantic network (the nodes are the links, and the labels are labeled with the common words that exist in the metadata of the two documents connected by the link), and second, as an index for fast retrieval.

D. The Server

We used the Apache free web server. It is ideal for running PHP scripts which access the MySQL database. The Supervisor, Presenter, Composer and Monitoring modules were implemented in PHP.

E. The User Tier

Since the system was designed as a web-based application, users may access it from any computer using any commercial web-browser, e.g. Netscape Communicator or Microsoft Internet Explorer. A user enters a query that is received by the Supervisor module which processes it. The answer to the user's query is returned in two forms: first, as one presentation that has been prepared by the Presenter module, and which consists of the files prepared by the Composer glued together in some order, where the user may play the presentation by clicking on a button; second, in the form of a presentation that contains links to multimedia files categorized by their types and gives the user the option to select which file to play.

V. Conclusion and Future Work

This paper has described the design and implementation of a multimedia based personalized system for teaching and learning multimedia courses in an effort to improve the quality of teaching and learning and to provide adapted responses to the learners’ needs in multimedia courses. It integrates new principles and tools in the areas of e-learning, user modeling, adaptive hypermedia, and semantic web modeling. The system saves the users the need to interact directly with the web to search for resources relevant to their queries, but rather it updates its database of resources by checking the web regularly and by considering the ratings given by the users for the stored resources. The system also adapts its responses to users based on their models which are stored in the database. The system is flexible and allows for the integration of various tools and technologies. Based on a first evaluation of the system, we intend to improve the system by developing a natural language interface, building an editor to allow instructors to integrate the multimedia teaching material into one-unit presentation rather than just returning a set of links, and by improving the filtering capabilities of the crawler. In addition, we intend to enhance the Presenter module, which serves as an automatic content manager, to glue the different files that represent an answer to a user's query using semantic information in those files.

References

[1] A. Kobsa, Generic User Modelling Systems. User Modelling and User – Adapted Interaction 11: 49 – 63, 2001. Kluwer academic Publishers, Netherlands.

[2] C. Basu, H. Hirsh, W. Cohen, and C. Nevill-Manning, Technical paper recommendations: a study in combining multiple information sources. Journal of Artificial Intelligence Research, 1, 231-252. 2001.

[3] C. Boyle, and A.O. Encarnacion, MetaDoc: an adaptive hypertext reading system. User Models and User Adapted Interaction. 4, 1-19. 1994.

[4] D. Zeinalipour-Yazti and M. Dikaiakos, Design and implementation of a distributed crawler and filtering processor. In *Proc. of NGITS 2002,* volume 2382 of *Lecture Notes in Computer Science*, pages 58–74, 2002.

[5] F. De Rosis, B. De Carolis, and S. Pizzutilo, User tailored hypermedia explanations. INTERCHI'93 Conference Proceedings: Conference on Human Factors in Computing Systems, INTERACT'93 and CHI'93, Amsterdam, The Netherlands. 169-170. 1993.

[6] J. Geurts, S. Bocconi, J. Van Ossenbruggen , and L. Hardman,
Towards Ontology-driven Discourse: From Semantic Graphs to Multimedia Presentations. In Second International Semantic Web Conference (ISWC2003), Sanibel Island, Florida, USA, October 20-23, 2003.

[7] K.D. Bollacker, S. Lawrence, and C.L. Giles, A system for automatic personalized tracking of scientific literature on the web. In Proc. ACM Conference on Digital Libraries (DL 1999), 105-113. 1999.

[8] M. Najork and J.L. Wiener, Breadth-first search crawling yields high-quality pages. In *Proc. of 10th International World Wide web Conference*, Hong Kong, China, 2001.

[9] M. Najork and A. Heydon, High-performance web crawling. In J. Abello, P. Pardalos, and M. Resende, editors, *Handbook of Massive Data Sets*. Kluwer Academic Publishers, Inc., 2001.

[10] O.C. Santos, C. Barrera, E. Gaudioso and J.G. Boticario, ALFANET: an adaptive elearning platform. 2nd International Meeting on Multimedia and ICTs in Education (mICTE 2003). In: Méndez, A, Mesa, J.A., Mesa, J. (eds): Advances in Technology-Based Education: Toward a Knowledge-Based Society, 2003, pp. 1938-1942.

[11] O.C. Santos, J.G. Boticario and E.J.R. Koper, aLFanet. 2nd International Meeting on Multimedia and ICTs in Education (mICTE 2003). In: Méndez, A, Mesa, J.A., Mesa, J. (eds): Advances in Technology-Based Education: Toward a Knowledge-Based Society, 2003, p. 2014.

[12] P. Brusilovsky. Methods and techniques of adaptive hypermedia. In UM and User-Adapted
Interaction. Kluwer academia publishers, pp. 87–129. 1996.

[13] P. Brusilovsky, Adaptive hypermedia, UM and User Adapted Interaction, vol. 11(1), pp. 87–110. 2001.

[14] P. Brusilovsky, Adaptive hypermedia, User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2): 87-110. 2001.

[15] R.G. Paredes, and G. Ayala, A user model server for the personalization of digital services and collections, proceedings of the XII Congreso Internacional de Ingeniería Electrónica, Comunicacionesy Computadoras, IEEE Puebla, Acapulco, Mexico, pp.84-88. 2002

[16] S. Lin, M.C. Chen, J. Ho and Y. Huang, ACIRD: Intelligent Internet Document Organization and Retrieval, IEEE Transactions on Knowledge and Data Engineering, 14(3), 2002, pp. 599-614.

[17] T. Berners-Lee, J. Hendler, and O. Lassila, The semantic web. Scientific American, 284(5):35{43, 2001.

[18] T. Berners-Lee, R. Cailliau, A. Luotonen, H.F. Nielsen, and A. Secret, The World Wide Web, Communications of the ACM, 37(8). 1994.

[19] Wget's Website. http://wget.sunsite.dk/