Adaptive Hypermedia Services for E-Learning

Peter Brusilovsky, Sergey Sosnovsky, Michael Yudelson

School of Information Sciences
University of Pittsburgh, Pittsburgh PA 15260, USA
{peterb, sas15, mvy3}@pitt.edu

Abstract. This paper discusses the opportunity of offering adaptive hypermedia functionality in E-Learning context through adaptive hypermedia services. We think, that this option bridgies the gap between the complexity of authoring and the necessary easy of use of adaptive hypermedia. This paper presents an open architecture for adaptive E-Learning that makes adaptive hypermedia services possible and briefly describes two adaptive hypermedia services QuizGuide and NavEx that stands between E-Learning portal and re-usable interactive content providing additional value for teachers and students who use this content through the portal. The value added by QuizGuide and NavEx is the ability to provide navigation support for each student without an overhead for a teacher. Teachers can bypass the time-consuming process of selecting relevant content for each course lecture while students receive better guidance because it is adapted to their learning goals and knowledge.

1 Introduction

A number of pioneer adaptive hypermedia systems such as ELM-ART [18], 2L670 [11], and KBS-HyperBook [13] have demonstrated the benefits of serving full-scale Web-based courses adaptively. Several adaptive hypermedia authoring systems and platforms [5; 12; 19] have been developed to assist the production of Web-based courses equipped with adaptive hypermedia. Yet, nearly all Web-based courses are now developed and served through so-called learning management systems [6] such as Blackboard [1] or WebCT [17]. Learning management systems (LMS) are powerful integrated systems that support a number of activities performed by teachers and students during the E-Learning process.

A quick comparison may tell why LMS and not adaptive hypermedia (AH) authoring systems is the dominated way to develop Web-based courses. First, LMS support a whole range of teacher activities while AH authoring systems focus mostly on content organization and presentation. Second, adding a new content fragment to an LMS requires no overhead, while any fragment added to AH course has to be indexed with domain model concepts [3].

How in this context can we bring the power of AH to thousands of real users? How can we bridge the gap between AH and LMS? In our earlier paper [7] we have argued that current generation of LMS systems can't support full-scale adaptive hypermedia.

Should we focus then on developing a new generation of LMS with embedded AH functionality as pioneered by ALE and a few similar systems [15]?

The authors argue that there is another way: embedding AH functionality into modern Web-based courses using *adaptive hypermedia services*. We think that the major bottleneck of using full-scale adaptive hypermedia in classroom is the ability of a teacher to create a complex AH course – even with the best possible authoring tools. As the experience of LMS use in universities demonstrates, a typical teacher is able to locate useful resources and attach them to the course Web site. However a teacher is rarely able to assemble a full-scale course from multiple small components even without indexing required by adaptive hypermedia. While we can expect that large course development teams and even some motivated and dedicated teachers can embrace and use adaptive LMS of the future, the real question is how we can help the rest. Our answer to this question is simple – we suggest offering adaptive hypermedia services that can be considered and attached to courses as a single re-usable component by practical teachers.

The idea of services allows bridging the gap between the complexity of authoring and the necessary easy of use of adaptive hypermedia. In our model adaptive hypermedia services can be designed and maintained by professional teams (that we call service providers) and can be used by anybody in the context of their courses. The idea of adaptive hypermedia services was pioneered in APLES .[8; 9; 10; 16]. In our work we are trying to expand the concept of AH services as well as to develop several practical services that can be used by anybody. We are also working on an open architecture for adaptive E-Learning that makes adaptive hypermedia services possible. This paper presents our architecture and briefly describes two adaptive hypermedia services that we have developed recently.

2 The KnowledgeTree Architecture

KnowledgeTree is a distributed architecture for adaptive E-learning based on the re-use of intelligent educational activities [4]. Capitalizing on the success of integrated LMS, KnowledgeTree aims to provide one-stop comprehensive support for the needs of teachers and students who are using E-Learning. It doing so it attempts to replace the current monolithic LMS with a community of distributed communicating servers (or services). The architecture assumes the presence of at least four kinds of servers: activity servers, value-added services, learning portals, and student model servers (Figure 1). These kinds of servers represent the interests of three main stakeholders in the modern E-Learning process: content and service providers, course providers, and students.

A *learning portal* represents the needs of course providers - teachers (trainers) and their respective universities (or corporate training companies). A portal plays a role similar to modern LMS in two aspects. First, it provides a centralized single-login point for enrolled students to work with all learning tools and content fragments that are provided in the context of their courses. Secondly, it allows the teacher responsible for a specific course to structure access to various distributed fragments according to the needs of this course. Thus, a portal is a component of the architecture

that is centered on supporting a *complete course*. Replicating the familiar functionality of an LMS, it provides a course-authoring interface for the teacher and maintains a runtime interface for the student. The difference between this and LMS an architectural separation of the unique *course structure* created by the teacher or course author from reusable course content and services. In KnowledgeTree, both learning content and learning support services (together called *activities*) are provided through the portal by multiple distributed *activity servers* (services). A portal has the ability to query activity servers for relevant activities and launch remote activities selected by students or by the portal itself.

Figure 1. Main components of the KnowledgeTree distributed architecture.

An *activity server* is a component that focuses on the prospects needs of content and service providers. It is centered on reusable *content and services*. It plays a role similar to an educational repository in modern courseware reusability approaches, in the sense that it hosts reusable learning content. The difference between this and a traditional learning repository is twofold. First, unlike repositories that are pools for storing simple, mostly static, learning objects, an activity server can host highly interactive and adaptive learning content. It can also host interactive learning services such as discussion forums or shared annotations. Secondly, an activity server assumes a different way of re-using its "content". While simple learning objects are re-used by being copied and inserted into new courses, an activity is re-used by referencing and is then delivered by its server.

These activities just can't be copied as files, they have to be served from a dedicated Web servers maintained by the content providers. The duty of an activity server is to answer the portal's and value-added service's requests for specific activities and to provide complete support for a student working with each of the activities residing on the server. The concept of reusable activities encourages content providers to develop highly advanced, interactive learning content and services. In particular, content and service activities delivered by a server can be intelligent and adaptive. Each activity can obtain and up-to-date information about each student from the student model server and thus provide a highly personalized learning experience.

It also monitors student progress, changes student goals, knowledge, and interests; then sends updates to the student model server.

A value-added service combines features of a portal and an activity server. It is able to "pass through" itself the "raw" content and services adding some valuable functionality to it - such as adaptive sequencing, navigation support, visualization, or content integration. Like a portal, it is able to query activity servers and access activities. Like an activity server, it can be queried and accessed by a portal. Value-added services are maintained by service providers. Since these services are course-neutral, they can be re-used in multiple courses providing larger building blocks for a teachers assembling an E-Learning system with the help of a portal.

The *student model server* is a component that represents the needs and the prospects of students in the process of E-Learning. This kind of server allows distributed E-Learning to be highly personalized. Ideally, a student model server can support student learning for several courses. It can be maintained by a provider (i.e., a university) or by the students themselves. It collects data about student performance from each portal and each activity server and provides information about the student to adaptive portals and activity servers that are then able to adapt instructional materials to their students' unique personalities and present development.

We anticipate that in the context of pure Web-based education, a student model server can reside on the student's own computer and support just one user. Using this method, the server can also serve as a tool for the user to monitor his or her own progress within various activities and courses. In the context of classroom education, the server can reside on a computer maintained by the educational establishment. Here it also supports the teacher's need to monitor the progress and the performance of the whole class. These arrangements can help to solve a number of privacy and security problems associated with student modeling.

With the KnowledgeTree architecture, a teacher develops a course using one portal and many activity servers and services. The student works through the portal serving this course, but interacts with many learning activities served directly by various activity servers. The student model server provides a basis for performance monitoring and adaptivity in this distributed context. The KnowledgeTree architecture is open and flexible. It allows the presence of multiple portals, activity servers, and student modeling servers. The open nature of it allows even small research groups or companies to be "players" in the new E-learning market. It also encourages creative competition between developers of educational systems - i.e., competition based on offering better services, not by monopolizing the market and resisting innovation. An activity server that provides some specific innovative learning activities can be immediately used in multiple courses served by different portals. A newly created portal that offers better support for a teacher or answers better to the needs of a specific category of course providers can successfully compete with other portals since it has access to the same set of resources as other portals. An new kind of student model server that provides better precision in student modeling or offers a better support for student model maintenance can successfully compete with older servers. Overall the architecture reflects the move from a product-based to a service-based Web economy.

While working with KnowledgeTree over the last two years we have clearly realized the need for adaptive hypermedia services. A teacher developing a course

with a portal faces large volume of educational resources. For example, we have developed a large amount of interactive questions and examples for programming that are available through our activity servers QuizPACK [14] and WebEx [2]. Potentially, a teacher may be interested to allow the student to use a large amount of resources. However, it's a duty of the teacher to provide at least some minimal *guidance* making sure that the student is not get lost in the long list of resources, but was able to see the right things in the right time. What the teachers are typically doing to provide this guidance is structuring resources – adding each resource to a specific part of the course like lecture or topic. However, even this simple structuring is a lot of work when the number of available resources is more than a hundred. Here is where adaptive hypermedia service can help. Our model is to use adaptive navigation support service as a layer between the course and the "raw content". The availability of this service will allow a teacher simply to add a whole bunch of activities to the course. Instead of providing navigation support manually for each activity, the teacher can simply rely on adaptive navigation support to advise the students what is the right activity to try at any given time. To explore this model we have created two adaptive hypermedia services. NavEx provides and adaptive access to interactive examples developed with WebEx [2] and – QuizGuide provides and adaptive access to QuizPACK quizzes [14].

2. The NavEx Service

NavEx (Navigation to Examples) that was designed to explore the idea of providing adaptive navigation support for accessing programming examples. NavEx is adaptive interface to our older system WebEx [2]. WebEx can serve interactive examples, but it has no knowledge about the examples and can't decide when a specific example is appropriate to explore. NavEx analyze the context of each example, identifies programming concepts behind it, and use traditional "traffic light" adaptive navigation support [3] to guide students to most relevant examples.

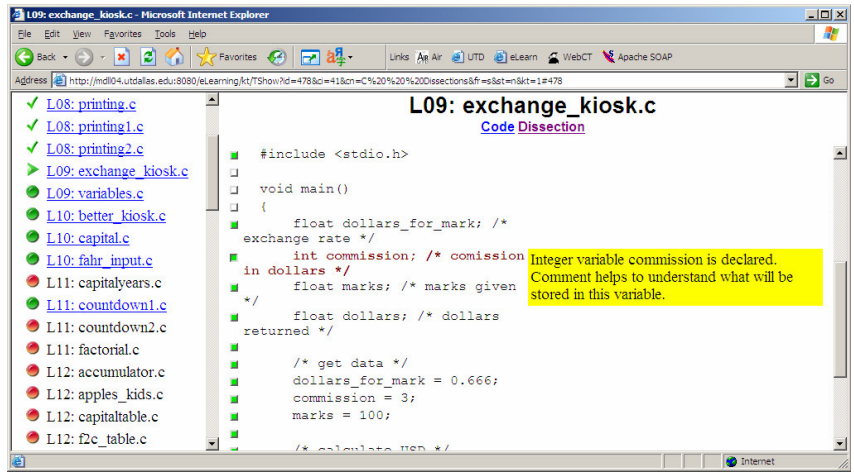

Figure 2: The interface of NavEx.

The interactive window of the NavEx system is divided into 3 frames (Figure 2). The leftmost frame contains a list of links to all examples/dissections available for a student in the current course. The links are annotated with colored bullets. Red bullet means that the student has not mastered enough prerequisite concepts to view the example. The link annotated with the red bullet is thus disabled. Green bullet means that the student has enough knowledge to view the example. Green check mark denotes example that has already been seen by the student. Green "play" bullet denotes the example that is currently being viewed. The order of links to examples is fixed, so students can find them at the same place in spite of their progress in the course.

The upper frame displays the name of the current example. Underneath are two links: one loads the source code of the example to the central frame (to be copied, compiled, and explored), the other – loads interactive example *dissection* (served directly by the WebEx system that is now a component of NavEx). Dissection is the same source code commented by the author. These comments address the meaning and the purpose of this line of the code and help the student to understand the example better. Extended comments are shown to the left form the code and can be activated by clicking on the bullet next to the line of the code. If the comment is available the bullet is green and is white otherwise.

3 The QuizGuide Service

QuizGuide, an adaptive service that help students selecting most relevant quizzes offered by QuizPACK system [14]. Similarly to WebEx, QuizPACK can deliver an activity, but has no idea when this activity is appropriate to a student. QuizGuide adds a layer of navigation support between a portal and the raw content using another an

innovative kind of adaptive annotation to show every student which topics are currently most important and which require further work.

The student interface of the QuizGuide system consists of two main parts: the quiz navigation area and the quiz presentation area (Figure 3). In the quiz presentation area students answer the questions and receive feedback. There are two types of questions: "*what is the final value of the marked variable?*" and "*what will be printed?*" The feedback indicates whether the answer is correct or not; if it is not, the correct answer is presented. After answering a question students are provided with two options to continue. They can move the next question in the quiz or repeat the previous one (with a different value parameter). The student work with the quiz area of QuizGuide is supported by original QuizPACK [14].

The quiz navigation area (left on Figures 3) provides hyperlinks to quizzes grouped by topics. When a student clicks on the topic name, the links to quizzes available for this topic pops out. The student cannon open several topics if s/he wants. A click on an opened topic collapses the list back. A click on a quiz link loads the first question of this quiz in the quiz presentation area. Altogether QuizGuide provides access to more then 40 quizzes organized in 20 topics. In this context our use of stretchtext technology decreases the information load on the student by removing link to the unrelated quizzes while allowing them to see “the whole picture”.

Adaptive navigation support is provided in the navigation area by adaptive icons shown to the left of each topic. QuizGuide adapts to two most cricial characteristics of the user: the knowledge level and the learning goal. To reflect both the goal and knowledge relevance of each topic in one icon, QuizGuide uses the “target-arrow” abstraction (Fugure 3). The number of arrows in the target reflects the level of knowledge the student has for the topic: the more arrows the target has, the higher the level is. The target color shows the relevance of the topic to the current learning goal: the more intense this color is, the more relevant the topic is. Topics, which are not ready to be studied are annotated with the crossed target. Hence we have four levels of knowledge (from zero to three arrows) and four levels of goal relevance (not-ready, important, less-important and non-important).

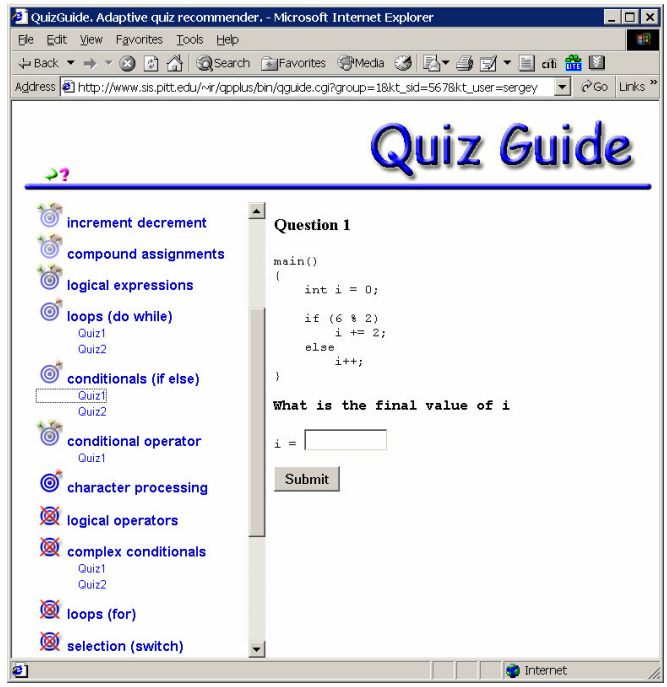

Figure 3: Student interface of QuizGuide.

QuizGuide system is implemented as a value-added service standing between the original QuizPACK system and the learning portal. QuizGuide generates the list of annotated hyperlinks to quizzes and uses QuizPACK to deliver the quizzes through the quiz presentation frame. When generating the adaptive quiz navigation part, QuizGuide sends a request to the central UM server CUMULATE [4] and gets the current values of the knowledge level and the learning goal for all topics.

4 Summary

We have developed two adaptive hypermedia services QuizGuide and NavEx that stands between E-Learning portal and re-usable interactive content providing additional value for teachers and students who use this content through the portal. The value added by QuizGuide and NavEx is the ability to provide navigation support for each student without an overhead for a teacher. With our services, teachers can bypass the time-consuming process of selecting re-usable content for each course lecture that meet goals and prerequisite restrictions. Instead, they can add a whole bunch of content as one package and rely on the navigation support provided by the services. At the same time, the students receive better guidance in selecting content then a

traditional list of resources for a specific lecture because it is adapted to their learning goals and knowledge.

An adaptive service like QuizGuide and NavEx is a relatively advanced product. However, a team with a good expertise in adaptive hypermedia and knowledge of the subject to be taught should be able to develop this kind of services relatively fast. Once developed, such a service can be used in many practical courses thus bringing the power of adaptive hypermedia to thousands of real students.

References

[1] Blackboard Inc.: Blackboard Course Management System, Blackboard Inc. (2002) available online at http://www.blackboard.com/

[2] Brusilovsky, P.: WebEx: Learning from examples in a programming course. In: Fowler, W. and Hasebrook, J. (eds.) Proc. of WebNet'2001, World Conference of the WWW and Internet, Orlando, FL, AACE (2001) 124-129

[3] Brusilovsky, P.: Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools. In: Murray, T., Blessing, S. and Ainsworth, S. (eds.): Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software. Ablex, Norwood (2003) 377-409

[4] Brusilovsky, P.: KnowledgeTree: A Distributed Architecture for Adaptive E-Learning. In: Proc. of The Thirteenth International World Wide Web Conference, WWW 2004 (Alternate track papers and posters), New York, NY, ACM Press (2004) 104-113

[5] Brusilovsky, P., Eklund, J., and Schwarz, E.**:** Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems. **30**, 1-7 (1998) 291-300

[6] Brusilovsky, P. and Miller, P.: Course Delivery Systems for the Virtual University. In: Tschang, T. and Della Senta, T. (eds.): Access to Knowledge: New Information Technologies and the Emergence of the Virtual University. Elsevier Science, Amsterdam (2001) 167-206.

[7] Brusilovsky, P. and Vassileva, J.: Course sequencing techniques for large-scale web-based education. International Journal of Continuing Engineering Education and Lifelong Learning **13**, 1/2 (2003) 75-94

[8] Conlan, O., Wade, V., Bruen, C., and Gargan, M.: Multi-model, metadata-driven approach to adaptive hypermedia services for personalized eLearning. In: De Bra, P., Brusilovsky, P. and Conejo, R. (eds.) Proc. of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002), Málaga, Spain (2002) 100-111

[9] Conlan, O., Wade, V., Gargan, M., Hockemeyer, C., and Albert, D.: An architecture for integrating adaptive hypermedia services with open learning environments. In: Barker, P. and Rebelsky, S. (eds.) Proc. of ED-MEDIA'2002 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, Denver, CO, AACE (2002) 344-350

[10] Dagger, D., Conlan, O., and Wade, V. P.: An architecture for candidacy in adaptive eLearning systems to facilitate the reuse of learning Resources. In: Rossett, A. (ed.) Proc. of World Conference on E-Learning, E-Learn 2003, Phoenix, AZ, USA, AACE (2003) 49-56

[11] De Bra, P.: Teaching Through Adaptive Hypertext on the WWW. International Journal of Educational Telecommunications **3**, 2/3 (1997) 163-180

[12] De Bra, P. and Calvi, L.: AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia **4** (1998) 115-139

[13] Henze, N. and Nejdl, W.: Extendible adaptive hypermedia courseware: Integrating different courses and Web material. In: Brusilovsky, P., Stock, O. and Strapparava, C. (eds.) Adaptive Hypermedia and Adaptive Web-Based Systems. Lecture Notes in Computer Science, Springer-Verlag, Berlin (2000) 109-120

[14] Sosnovsky, S., Shcherbinina, O., and Brusilovsky, P.: Web-based parameterized questions as a tool for learning. In: Rossett, A. (ed.) Proc. of World Conference on E-Learning, E-Learn 2003, Phoenix, AZ, USA, AACE (2003) 309-316

[15] Specht, M., Kravcik, M., Klemke, R., Pesin, L., and Hüttenhain, R.: Adaptive Learning Environment (ALE) for Teaching and Learning in WINDS. In: Lecture Notes in Computer Science, Vol. 2347. Springer-Verlag, Berlin (2002) 572-581

[16] Wade, V.: Dynamic adaptive composition of adaptible Web services. Eindhoven University of Technology, Budapest, Hungary (2003) 53-62

[17] WebCT: WebCT Course Management System, Lynnfield, MA, WebCT, Inc. (2002) available online at http://www.webct.com

[18] Weber, G. and Brusilovsky, P.: ELM-ART: An adaptive versatile system for Web-based instruction. International Journal of Artificial Intelligence in Education **12**, 4 (2001) 351-384, available online at http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/weber.html

[19] Weber, G., Kuhl, H.-C., and Weibelzahl, S.: Developing adaptive internet based courses with the authoring system NetCoach. In: Bra, P. D., Brusilovsky, P. and Kobsa, A. (eds.) Proc. of Third workshop on Adaptive Hypertext and Hypermedia, Sonthofen, Germany, Technical University Eindhoven (2001) 35-48, available online at http://wwwis.win.tue.nl/ah2001/papers/GWeber-UM01.pdf