

# 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 advantages of offering adaptive hypermedia functionality in the e-Learning context through *adaptive hypermedia services*. We think that this option bridges the gap between the complexity of authoring and the ease-of-use necessary for successful adaptive hypermedia. This paper presents an open architecture for adaptive e-Learning that would make these adaptive hypermedia services possible. It 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 want to use content through an e-Learning portal. The value added by QuizGuide and NavEx is the ability to provide navigation support for each student without causing overhead for the teacher. Teachers can bypass the time-consuming process of selecting relevant, reusable content for each course lecture while students receive better guidance when e-Learning is adapted to their learning goals and knowledge.

## 1 Introduction

A number of pioneer adaptive hypermedia systems such as ELM-ART [21], 2L670 [13], and KBS-HyperBook [15] have demonstrated the benefits of adaptively serving full-scale web-based courses. Several adaptive hypermedia authoring systems and platforms [7; 14; 22] have been developed to assist in the production of these specialized web-based courses. Nevertheless, nearly all web-based courses are now developed and served through so-called Learning Management Systems [8] such as Blackboard [3] or WebCT [20]. 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 us why LMS rather than adaptive hypermedia (AH) authoring systems are chosen as the primary 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. Secondly, adding a new content fragment to an LMS requires no overhead, while any fragment added to an AH course must be indexed with regards to the domain model's concepts [5].

In this context, how can we bring the power of AH to thousands of real users? How can the gap be bridged between AH and LMS? In our earlier paper [9] we argued that the current generation of LMS systems can't support full-scale adaptive hypermedia.

Should we now focus on developing a new generation of LMS which have an embedded AH functionality (as pioneered by ALE and a few similar systems [18])?

The authors present here another method: embed AH functionality into modern web-based courses using *adaptive hypermedia services*. The major bottleneck which prevents the use of full-scale adaptive hypermedia in the classroom appears to be the complexity a teacher faces when trying to create a complete AH course – even when being given the best authoring tools in existence. As demonstrated by current LMS use in universities, a typical teacher is able to locate useful resources and attach them to their course web site. However, teachers are rarely able to assemble a full-scale course from multiple instances of these small components, even if they skip the indexing step required of adaptive hypermedia. While we can expect that large course development teams and even some motivated and dedicated teachers will embrace and be able to use the adaptive LMS of the future, the real question is how we can help everyone else. Our answer to this question is simple – we suggest offering adaptive hypermedia services that can be chosen by the classroom teacher and attached to courses as a single reusable component.

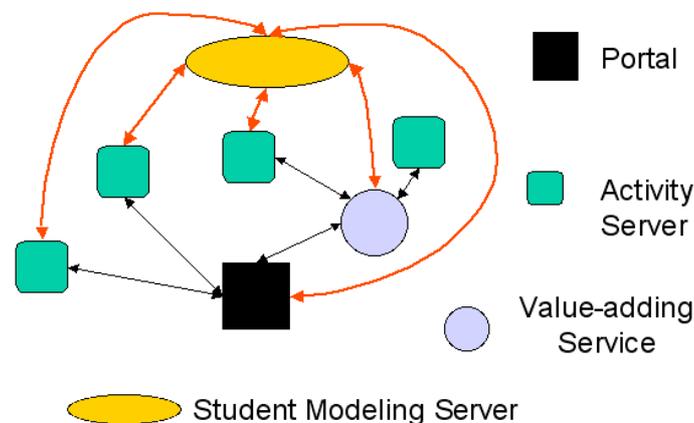
AH Services allow one to organize the complexity of authoring while maintaining the ease-of-use necessary for adaptive hypermedia. In our model, AH Services can be designed and maintained by professional teams (which we call *service providers*) and should be able to be used by anyone teaching a web-based course. The idea of AH Services was pioneered in APeLS [10; 11; 12; 19]. In our work we want to expand the concept of AH services as well as to develop several practical services that anyone can use. We are also working on an open architecture for adaptive e-Learning that makes AH Services possible. This paper presents our architecture and briefly describes two AH 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 [6]. 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. In 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: learning portals, activity servers, value-added services, 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. The 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 his or her course. Thus, the portal is the 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 is 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 projected needs of the 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, the activity server uses a different way to deliver its reusable "content": While simple learning objects are merely copied and inserted into new courses, an activity is re-used by referencing it and then delivered through the server.

These activities can't be merely copied as files, but have to be served from 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 any 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 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 and changes that take place in 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 pipe through itself "raw" content while 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. Thus they provide 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, the student model server would 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. A 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 [17] and WebEx [4]. Potentially, a teacher may be interested in allowing the student to use a large amount of resources. However, it's a duty of the teacher to provide at least some minimal *guidance* to make sure that the student is not getting lost in the long list of resources, but is able to access the right things in the right time. What teachers are typically doing to provide this guidance is structuring resources – adding each resource to a specific part of the course, such as a lecture or weekly topic. However, even this simple structuring grows overwhelming when the number of available resources is more than a hundred. At this point, an adaptive hypermedia service can help. Our model uses adaptive navigation support service as a layer between the course and the “raw content”. The availability of this service will allow a teacher to simply to add a whole set of activities to the course. Instead of providing navigation support manually for each activity, the teacher can simply rely on automatic adaptive navigation support to advise the students about what is the right activity to try at any given time.

To explore this model we have created two adaptive hypermedia services. NavEx provides an adaptive access to interactive examples developed with WebEx [4] and QuizGuide provides an adaptive access to QuizPACK quizzes [17].

### 3 The NavEx Service

NavEx (Navigation to Examples) was designed to explore the idea of providing adaptive navigation support for accessing programming examples. NavEx is an adaptive interface to our older system WebEx [4]. 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 analyzes the context of each example, identifies programming concepts behind it, and uses the traditional “traffic light” metaphor for adaptive navigation support [5] in order to guide students to the most relevant examples.

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. The red bullet means that the student has not mastered enough prerequisite concepts to view the example. Thus the link annotated with the red bullet is disabled. The green bullet means that the student has enough knowledge to view the example. A green check mark denotes that the example has already been seen by the student. The green “play” bullet means that the example is currently being viewed. The order of links to examples is fixed, so students can find them at the same place no matter how far they progress through the course.

The upper frame displays the name of the current example. Underneath it are two links: one loads the source code of the example to the central frame (to be copied, compiled, and explored), the other – loads the interactive *dissection* of the example (served directly by the WebEx system that is now a component of NavEx). Dissection uses the same source code but adds comments. These comments address the meaning

and purpose of each line of code, thus helping the student to understand the example better. The existence of extended comments is shown by green bullets on the left side of the code and can be activated by clicking on the bullet next to each line of the code. If the comment is available the bullet is green, white otherwise.

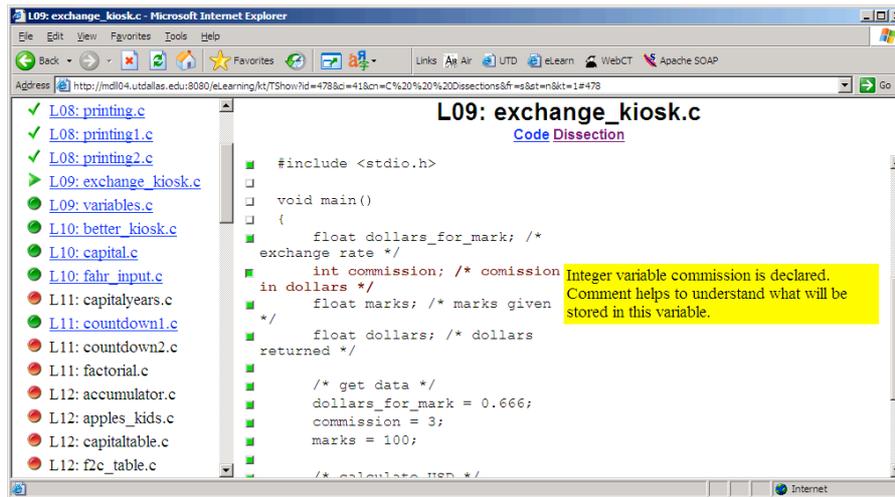


Figure 2: The interface of NavEx.

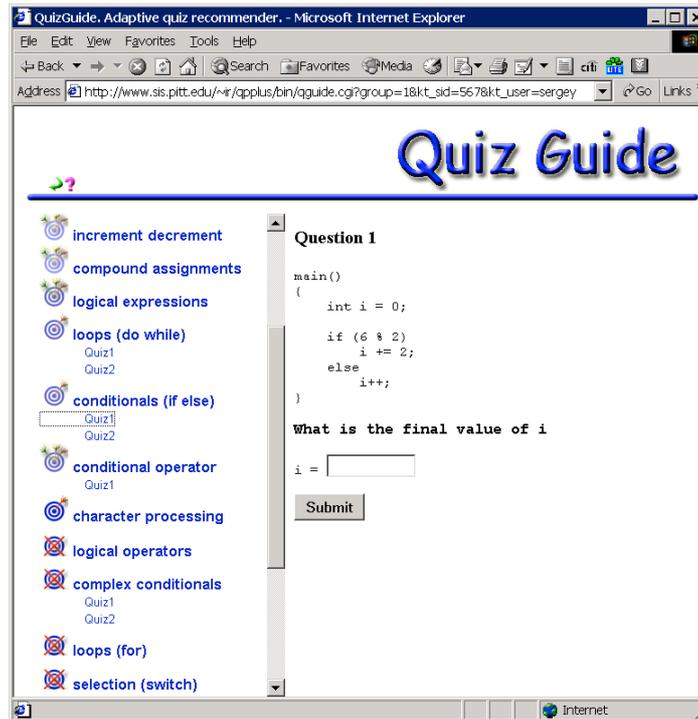
## 4 The QuizGuide Service

QuizGuide is an adaptive service that helps students select the most relevant quizzes that can be offered by the QuizPACK system [17]. Similar to WebEx, QuizPACK can deliver an activity, but has no knowledge of when this activity is appropriate to a student. QuizGuide adds a layer of navigation support between a learning portal and raw content. It uses another innovative kind of adaptive annotation in order to show every student which topics are currently most important and which ones 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 also presented. After answering a question, students are provided with two options to continue. They can move to the next question in the quiz or repeat the previous one (with a different value parameter randomly generated). The student work with the quiz area of QuizGuide is supported by the original QuizPACK [17].

The quiz navigation area (left in Figure 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 can open one or several topics. Clicking on an opened

topic collapses the list back. Clicking on a quiz link loads the first question of this quiz in the quiz presentation area. Altogether, QuizGuide provides access to more than 40 quizzes organized in 20 topics. In this context, our use of stretchtext technology decreases the information load on the student by removing links to the unrelated quizzes while allowing students to see “the whole picture”.



**Figure 3:** Student interface of QuizGuide.

Adaptive navigation support is provided in the navigation area by adaptive icons shown to the left of each topic. QuizGuide adapts to two of the most crucial characteristics of the user: knowledge level and learning goal. To reflect both the goal and knowledge relevance of each topic in one icon, QuizGuide uses the “target-arrow” abstraction (Figure 3). The number of arrows in the target reflects the level of knowledge the student has for that 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 saturated the color is, the more relevant the topic is. Topics which the student is not yet prepared to study are annotated with a crossed-out 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).

QuizGuide system is implemented as a value-added service that resides between the original QuizPACK system and a 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 adaptive quiz navigation, QuizGuide sends a request to the central User Modeling server CUMULATE [6] and gets the current values for the knowledge level of all course topics.

## 5 Summary

We have developed two adaptive hypermedia services QuizGuide and NavEx that reside 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 causing overhead for the teacher. With our services, teachers can bypass the time-consuming process of selecting reusable content for each course lecture by determining which ones are relevant to goals and prerequisite restrictions. Instead, they can add a whole set of content as one package and rely on the navigation support provided by the AH Services. At the same time, the students receive better guidance in selecting content than when they are given the traditional list of resources for a specific lecture because the list is now limited and adapted to their learning goals and knowledge.

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

An AH Service like QuizGuide and NavEx is a new entity in e-Learning technology. So far, almost all researchers and practitioners in e-Learning have only considered two major kind of entities – LMS and reusable content (residing in content repositories). Some projects have also considered non-adaptive services. Can we fit these adaptive services into modern e-Learning? If we recognize that reusable content can also be an interactive and adaptive content delivered from a content server, then an adaptive service is a natural add-on to e-Learning. For an LMS, it looks like any other adaptive content (as we have explained, the teacher can connect the whole service just like a single fragment of interactive content). At the same time, from the content side, the service looks like an LMS that can call and serve the content. Thus, AH Services can simply use established protocols that exist between LMS and content. This is exactly the method we have used to integrate our services into the KnowledgeTree architecture. Any portal and any content server that understands KnowledgeTree protocols can also be used with adaptive services. AH Services can be developed for any other distributed architecture that has a communication protocol between LMS and its adaptive interactive content. In particular, SCORM [1; 2] and CMI [16] standards come very close to this requirement. They allow an LMS to communicate with advanced interactive content. However, due to the rather primitive treatment of student modeling in these standards [6], the development of an adaptive service based on CMI and SCORM is difficult, as has been shown by [10]. We hope

that our work with KnowledgeTree and adaptive services will contribute to developing more powerful standards for the future.

## References

- [1] ADLI: Sharable Content Object Reference Model (SCORM), Advanced Distributed Learning Initiative (2003) available online at <http://www.adlnet.org/>
- [2] ADLI: Overview of SCORM 2004, Advanced Distributed Learning Initiative (2004) available online at [http://www.adlnet.org/screens/shares/dsp\\_displayfile.cfm?fileid=992](http://www.adlnet.org/screens/shares/dsp_displayfile.cfm?fileid=992)
- [3] Blackboard Inc.: Blackboard Course Management System, Blackboard Inc. (2002) available online at <http://www.blackboard.com/>
- [4] 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
- [5] 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
- [6] 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
- [7] 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
- [8] 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.
- [9] 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
- [10] 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
- [11] 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
- [12] 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

- [13] De Bra, P.: Teaching Through Adaptive Hypertext on the WWW. *International Journal of Educational Telecommunications* **3**, 2/3 (1997) 163-180
- [14] De Bra, P. and Calvi, L.: AHA! An open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia* **4** (1998) 115-139
- [15] Henze, N. and Nejdil, 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
- [16] IEEE LTCS WG11: The Semantic Document v3.4, Computer Managed Instruction Working Group of the IEEE Learning Technology Standards Committee (2001) available online at <http://ltsc.ieee.org/wg11/index.html>
- [17] 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
- [18] 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
- [19] Wade, V.: Dynamic adaptive composition of adaptable Web services. Eindhoven University of Technology, Budapest, Hungary (2003) 53-62
- [20] WebCT: WebCT Course Management System, Lynnfield, MA, WebCT, Inc. (2002) available online at <http://www.webct.com>
- [21] 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](http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/weber.html)
- [22] 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://www.wis.win.tue.nl/ah2001/papers/GWeber-UM01.pdf>