A Distributed Architecture for Adaptive and Intelligent Learning Managament Systems

Peter Brusilovsky
School of Information Sciences, University of Pittsburgh
Pittsburgh PA 15260
peterb@mail.sis.pitt.edu

Abstract: This paper presents KnowledgeTree, an architecture for adaptive eLearning based on distributed re-usable intelligent learning activities. The goal of KnowledgeTree is to bridge the gap between the modern approach to Web-based education based on learning management systems and powerful but underused intelligent tutoring and adaptive hypermedia technologies. This architecture attempts to address both the component-based development of adaptive systems and the teacher-level reusability.

1. Introduction

Adaptive Web-based Educational systems (AWBES) have brought to the Web a number of powerful technologies [4] developed originally in the fields of intelligent tutoring systems (ITS) and adaptive hypermedia (AH). A number of researchers in these two fields had a hope that the Web will help their technologies to move finally from labs to real classrooms. Yet, seven years after the appearance of first adaptive Web-based educational systems just a handful of these systems are used for teaching real courses, typically in a class lead by one of the authors of the adaptive system. Instead, the absolute majority of Web-enhanced courses rely on so-called learning management systems (LMS) such as Blackbord [1] or WebCT [24]. LMS are powerful integrated systems that support a number of needs of teachers and students. Teachers can use a LMS to develop Web-based course notes and quizzes, to communicate with students and to monitor their progress. Students can use it for communication and collaboration. Still, the complete dominance of LMS over AWBES in Web-enhanced courses may look surprising. Indeed, for every function that a typical LMS perform we can find an AWBES that can do it much better than the LMS. Adaptive textbooks created with such systems as InterBook [5] or NetCoach [26] can help students learn faster and better. Adaptive quizzes developed with SIETTE [18] allow evaluating student knowledge more precisely with less questions. Adaptive class monitoring systems [16] give the teachers much better chances to notice the students lagging behind. Adaptive collaboration support systems [10] can enforce the power of collaborative learning. In addition to that, most adaptive systems.

It has long been argued that authoring of learning material in adaptive systems is too complicated for an average teacher. However, authoring support in best modern AWBES such as NetCoach [26] or SIETTE [18] is on the same level as in modern LMS. Moreover, a number of existing AWBES come together with a wealth of learning material, while a LMS expects the teachers to provide all learning material themselves. For example, ELM-ART [25] and VC-Prolog-Tutor [17] support comprehensively very solid parts of Lisp and Prolog courses - from concept presentation to program debugging. Yet, these systems are used much less widely than they deserve.

The author argues that the real problem of the current generation of AWBES is that they do not address the needs of both university teachers and administrators. *The first issue is the lack of integration*. AWBES can, indeed, support every aspect of Web-enhanced educational better than LMS, but each particular system can typically support just one function. For example, SIETTE [18] is a great system for serving questionnaires, but it can't do anything else. To cover all needs of Web-enhanced education with AWBES, a teacher will need to use a range number of different AWBES together. This is clearly inappropriate for administration that need to install and maintain all these systems, a teacher who need to master all of them, and a student who need to manipulate several systems at the same time. The university has a clear need in a single integrated system that can support all critical functions in one package. The LMS producers has recognized this need more than seven years ago and just in a few years has grown from systems that can support just one-two functions into Web-based monsters that cover all the needs. *The second issue that many modern AWBES are designed to be used as a whole, not component by component.* A teacher who is interested in re-using some adaptive content from an existing adaptive system (for example, several ELM-ART Lisp problems) has only one choice - to accept the whole system with its specific way of teaching and sacrifice his or her preferred way of teaching the course. Naturally, except from the authors of existing adaptive systems themselves (who built the systems to support *their* way of teaching), a rare teacher is willing to do that. In contrast, LMS have always supported the teacher in developing his course material from various components. Modern courseware-reusability frameworks such as ARIADNE [23] extends this power by providing repositories of re-usable educational objects.

It is clear that ITS and AH systems has to capitalize on the success of LMS in their attempt to reach real classrooms. But what is the way to do it? Should our community just copy the LMS approach by developing "more intelligent" LMS? The author is convinced that instead of developing another generation of "monolithic" systems the efforts have to be centered on developing distributed component-based architectures for building adaptive systems. The problem of component-based architectures (architectures to develop an ITS from multiple components) has long been addressed in the AI-ED community [2; 19; 20]. The Web gave this direction of work a new push adding the works on teacher-level reusability [22] and Web-distributed ITS [7] to the stream. Most visionary proposals were considering frameworks for a teacher to assemble an intelligent course on the fly from multiple distributed components [11; 15; 21]. Our current work continues this stream of research. This paper presents KnowledgeTree, an architecture for adaptive eLearning based on distributed re-usable intelligent learning activities that we are currently developing. The goal of KnowledgeTree is to bridge the gap between the modern approach to Web-based education based on LMS and educational material repositories and powerful but underused ITS and AH technologies. This architecture attempts to address both the component-based development of adaptive systems and the teacher-level reusability. The following sections provide a motivation for the KnowledgeTree framework, review several known problems that it addresses, and describes its most recent version that has already been used in several courses at the University of Pittsburgh.

2. Courseware Reuse vs. Adaptive Web-based Educational Systems

The methods and tools developed by the researches on courseware re-use systems and adaptive Web-based educational systems can contribute to creating better Web-enhanced courses. Each of these approaches has strong and weak sides. The courseware re-use frameworks such as ARIADNE allow a course author to search for the relevant learning objects in repositories of educational material and include them in their courses. This approach reduces course development time and improves the quality of courses by making high-quality educational material available for the learning community. At the same time, current implementations of this approach have at least three serious problems.

First of all, modern reusability frameworks implicitly assume that a learning object is a moveable entity - usually a file that is stored in a repository and can be re-used by *copying* into the course to be created. However, fragments of adaptive educational content in modern Web-based systems are not files but services delivered by a Web server. These activities can't be simply packaged, stored, and copied the same way as an image, a text file, or even an applet - they have to reside on a dedicated server. For example, ELM-ART, an adaptive LISP course [8] includes many LISP programming problems. Problems are more than just textual problem statements. They are fully interactive learning activities backed by ELM-ART unique knowledge-based functionality. In response to student program solution sent to ELM-ART server, the system can check, diagnose, and correct it. ELM-ART problems can't be moved or copied - they have to be served directly from a dedicated ELM-ART server. To benefit from advanced educational content an E-Learning system should be able to re-use these service-based activities. For example, a teacher may want to re-use ELM-ART problems (based on more than 10 man-year of research) in a very different LISP course. Current re-usability frameworks do not support this.

Second, courses developed with modern re-usability approach suffers from "one size fits all" problem. When identifying relevant material and organizing it within a course section, the teacher has to think about the class in general. The students in the class have different interests, knowledge, backgrounds, and learning styles. Some material carefully selected by the teacher can be useless for some students and only distract them. Some material that is important for particular students may not even be selected. An organization of material that benefits one category of learners may create obstacles for other categories. This problem is becoming especially important in Web-based education where the variety of learners taking the same course is much greater.

The situation is essentially different in the case of courses produced using adaptive hypermedia (AH) technologies or intelligent tutoring systems (ITS) technologies. Using individual student models and educational material enhanced with pedagogical metadata (such as required or produced domain knowledge and relevancy to learning styles) AH and ITS technologies are able to dynamically select the most relevant learning material from their knowledge bases and present it at the right time and in the right way for every individual student, thus making the best use of every fragment of educational material. At the same time, none of existing ITS and AH is designed for the modern E-Learning context where every administrator wants an integrated system and every teacher or course provider wants to develop a specifically targeted course re-using existing educational content from multiple repositories. The problems of existing adaptive Web-based educational systems are quite opposite to the problems of existing re-usability frameworks.

We believe that a way to the future starts on the crossroads of courseware re-use and adaptive educational systems. This paper attempts to bridge the gap between the information power of modern educational material repositories and the just-in-time delivery and personalization power of ITS and AH technologies.

3. KnowledgeTree: The Architecture

The KnowledgeTree is a distributed architecture for adaptive E-learning based on re-use of intelligent educational activities. Capitalizing on the success of integrated LMS, KnowledgeTree aims to provide a one-stop comprehensive support of teachers' and students' needs in eLearning. It doing so it attempts to replace the current monolithic Blackbord -style LMS with a community of distributed communicating servers. The architecture anticipates the presence of at least three kinds of servers: activity servers, learning portals, and student model servers (Figure 1).

A *learning portal* in our vision plays a role similar to modern LMS in two aspects. First, it provides a centralized single-login point for the student to work with all learning tools and content fragments that can be distributed over the Web. Second, it allows a teacher to structure the access to various distributed fragments according to the needs of a specific course. A portal is a component of the architecture that is centered on a *course*. Quite similar to a LMS it provides some course-authoring interface for a teacher and maintains a runtime interface with a student. The difference with LMS is the separation of the unique course structure from the reusable course content. Both the learning content and serviced (altogether called *activities*) used by the students reside not in the portal, but in multiple distributed *activity servers*. A portal has an ability to query activity servers for relevant activities and launch remote activities selected by students.

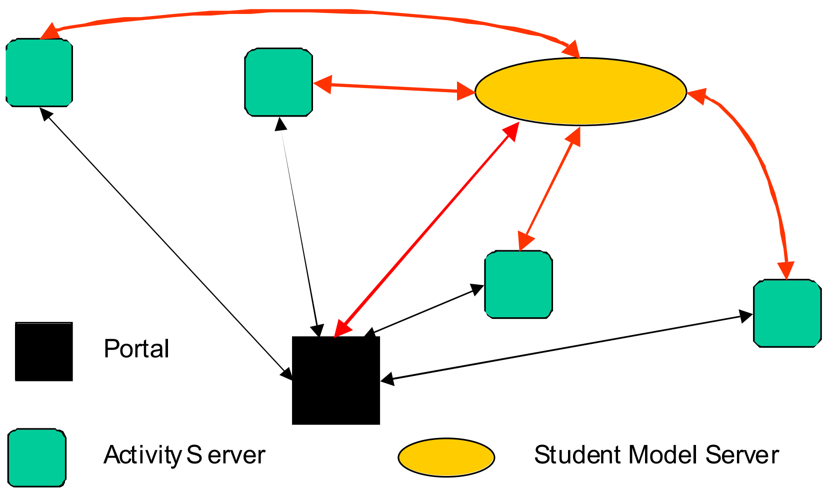

Figure 1: Main components of the KnowledgeTree distributed architecture.

An *activity server* is the component of the architecture that is centered around reusable *content and services*. It plays a role similar to an educational repository in modern courseware re-usability approach in the sense that it hosts reusable learning content. Unlike repositories that are pools for storing simple learning objects that can be copied and inserted into courses, an activity server not only stores, but also delivers the activities. The need for activity servers stems from the nature of adaptive and other advanced learning activities - such as ELM-ART problems. These activities just can't be copied as files, they have to be served by a dedicated Web servers maintained by the content providers. The duty of an activity server is to answer portal's requests for specific activities and to provide a complete support for a student working with each of the activities residing on the server. In particular, by turning an ITS into an activity server it is possible to make intelligent activities, traditionally encapsulated inside an ITS, open to multiple re-use.

Student model server is a component centered on an *individual student*. 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 able to personalize their communication with the student. We think that the presence of multiple adaptive activities requires a centralized user modeling architecture that enables each server to get an access to all information about student progress. The problem of centralized student modeling servers has been investigated in a number of earlier projects dealing with multiple intelligent educational agents [2; 13], but Web context poses new requirements to the student model servers. Our server CUMULATE presented below and Personis server suggested by Kay, Kummerfeld, and Lauder [12] are good examples of student model servers for distributed Web adaptive system.

With the KnowledgeTree architecture, a teacher develops a course using one portal and many activity servers. A 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 or it allows even small research groups or companies to be "players" in the new E-learning market. An activity server that provides some specific innovative learning activities can be immediately used in multiple courses served by different portals. An innovative portal with a good interface can successfully compete with other portals since it has an access to the same set of resources as other portals. A more powerful student model server can successfully replace older servers.

The open nature of the architecture relies on several clearly defined communication protocols between components. First, the architecture needs a protocol for transparent login and authentication. Each adaptive activity should know the identity of the user to use the proper user model, however the student should logs in only once. Second, it requires a standard protocol for a portal to send a query to the activity servers and the standard protocol for the activity servers to respond. Third, it requires a protocol for an activity server to send the information about the student progress to the student model server and a protocol to request information about the student from the student model. Finally, the architecture needs a resource discovery/exchange protocol. A portal can provide an access to a wide variety of learning activities residing on many servers. However, to benefit from this feature, a portal should know about many servers and kinds of activities they can offer.

The current version of KnowledgeTree presented in the next section offers a simple implementation of the first three protocols. Every activity us called directly by a dedicated URL. The transparent authentication is implemented by passing a session and a student identifiers as a part of activity URL. We use a rather simple http-based communication language between components, similar to the one we have developed in our past research on distributed intelligent tutoring [7]. While these protocols offer some solution that enables us to work and explore the distributed architecture, they are clearly "homegrown". More research is requited to develop protocols that can be commonly acceptable. The common acceptance is critical for the success of any distributed architecture. The resource discovery issue has not been addressed in the current version of KnowlegdeTree. Currently, we simply "tell" the portal about all existing activity servers. In the open context none of the portals can know all relevant activity servers and there is no centralized authority to collect this information. This requires a "resource discovery" mechanism for various portals to exchange information (metadata) about known servers and activities. A good example of this mechanism based on peer-to-peer metadata exchange among all kinds of servers is provided by EDUTELLA project (http://edutella.jxta.org/). We may expect other peer-based, centralized, or broker-based solution to appear shortly.

4. Current State of Work

To explore our architecture in the context of real university courses we have developed one portal, one student modeling server and four activity servers all completely protocolcompliant. The whole architecture has been already used for four consecutive semesters in the context of two different programming courses at our department. We have also performed several classroom studies of different servers. More information about the practical side of KnowledgeTree architecture and links to related publications can be found in [6]. The readers are welcome to try the current version of all system components available on our lab home pages: http://www2.sis.pitt.edu/~taler/.

4.1. The KnowledgeTree Portal

The KnowledgeTree architecture allows multiple portals that can support different educational paradigms and approaches while providing the access to the same body of content. The currently implemented portal (also called KnowledgeTree) is targeted to support a lecture-based educational process and is focussed on dynamic and adaptive selection of learning activities. In the lecture-based education, most teachers choose to structure their Web course material as a sequence of lectures. For every lecture they provide their own material (for example, handouts) and assemble a set of links to relevant educational resources. Some teachers provide a deeper structured material in a form of an electronic textbook where a structured chapter corresponds to a lecture. These course-authoring needs are supported by all modern LMS. Following the same needs, the KnowledgeTree portal helps an author to develop a course as a tree of modules. The role of the author here is to structure the set of modules and to select primary educational material for each module. To select the material for each module an author specifies an educational goal for a module. The specification can be done in both natural language and in a formal language that expresses the goal in terms of metadata associated with necessary learning activities. *At the authoring time*, the educational goal is used by the system to select subset of relevant educational activities from multiple learning repositories known to the system. From this pre-selected subset of activities an author can simply manually select most relevant primary and additional learning activities. To complement the set of activities found in the repositories, some activities can be designed by the author. When a particular student accesses the module during educational process, the *learning portal* uses this learning goal as well as the student model to select adaptively most relevant additional material for the given student *at the runtime.* Adaptive runtime selection allows the system to accommodate to the volatile and expanding nature of learning repositories and to student individual differences.

Note that the current version of KnowledgeTree is not student-adaptive and does not query the student model server. We are currently developing a new version based on adaptive hypermedia technologies. Adaptive navigation support (such as adaptive annotation, sorting, and direct guidance) will be used to help the student to select the currently most relevant items in the personalized learning space. In addition, the new version will let the students use the activity search that is currently available for the teachers. In our vision, the students should be able to search for relevant educational material for any module using their own criteria and to add material permanently to the module. The new portal will produce a dynamic and personalized learning space for each course module where personalization is provided by both the system and the student.

4.2. The Activity Servers

Three of four existing activity servers have been developed specifically for the area of teaching programming. Each server supports authoring of a specific kind of activity and supports student's interaction with a selected activity of this kind. The WebEx system serves interactive annotated program examples, the QuizPACK serves parameterized questions, and WADEIn [6] serves adaptive demonstrations and exercises related with expression evaluation. The fourth server KnowledgeSea is domain independent, and currently used to provide an interactive access to open corpus learning material. All activity servers are self-containing Web servers running on different platforms and completely independent from a portal. KnowledgeSea is based on JavaScript functionality and can be delivered by any Web server. Each server can work independently from the KnowledgeTree architecture (though some of the functionality is not available in the standalone context). All these servers implement our simple transparent login protocol, resource delivery protocol, and student modeling protocol. They can work with any portal and user modeling server that supports these protocols. More information and references to publications about the existing activity servers can be found in [6].

4.3. The CUMULATE Student Model Server

The CUMULATE Student Model Server is a new implementation of the event-based centralized student modeling architecture that we have been investigating in the past [2; 3]. The overall idea of the server is to collect evidences (events) about student learning from multiple servers that interact with the student and process these events into shareable name-value pair style student model (Figure 2). The collected time-stamped events are stored in the *event storage* part of the student model. Multiple external and internal inference agents constantly monitor the flow of events and update the values in the *inferenced model* part of the server. The presence of an event taxonomy [2] provides a common ground for interoperability between different event reporting and processing components. In addition to standardized events the new architecture adds flexibility by providing unique events, unique user model variables, and specific inference agents that serves the needs of just one specific activity server or portal. This, the server also can play a role of a student modeling shell solving all student modeling needs of involved components. To test this functionality, WADEIn activity server uses this capabilities of CUMULATE to solve all its user modeling needs.

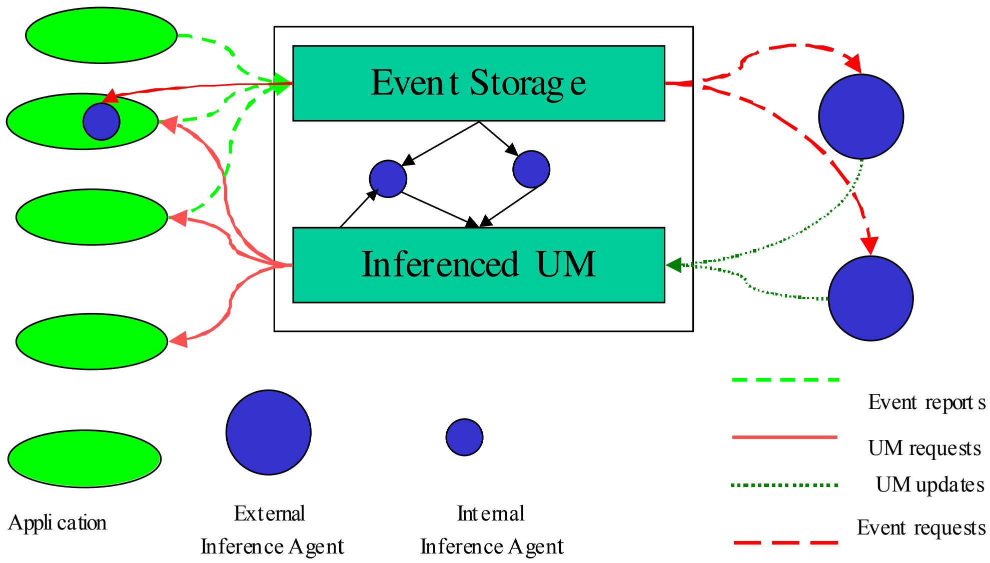

Figure 2: Centralized student modeling in CUMULATE

We anticipate that in the context of pure Web-based education, a student model server can reside on student's own computer and support just one user. In this context, the server can also serve as a tool for the user to monitor his or her own progress with 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. This arrangements can help to solve a number of privacy and security problems associated with student modeling.

5. Conclusion

This paper proposes an architecture for adaptive eLearning based on distributed re-usable intelligent learning activities that integrates the benefits provided by modern LMS and educational material repositories with the power of ITS and AH technologies. This architecture addresses both the component-based development of adaptive systems and the teacher-level reusability. We have started with implementing the core functionality of the system within our local group using some rather simple ways to implement the required protocols. Some other groups driven by similar goals have proposed other architectures that match our vision except the issue of communication standards [9; 12; 14]. Significant amount of work and cooperation of several research groups are required to turn the proposed architectures into the practice of eLearning. Fortunately, our work shares many goals with several other active Web-related research areas enabling us to re-use possible standards, solutions and ideas from these areas. The race for e-commerce, enterprise systems, Web services, personalization, brought to life many technologies that can be used for development of adaptive distributed E-learning. It gives our group together with other groups motivated by similar goals a good chance to succeed in bringing our a new generation of adaptive eLearning systems.

References

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

[2] Brusilovsky, P.: Student model centered architecture for intelligent learning environment. In: Proc. of Fourth International Conference on User Modeling, Hyannis, MA, MITRE (1994) 31-36, available online at http://www2.sis.pitt.edu/~peterb/papers/UM94.html.

[3] Brusilovsky, P.: Intelligent learning environments for programming: The case for integration and adaptation. In: Greer, J. (ed.) Proc. of AI-ED'95, 7th World Conference on Artificial Intelligence in Education, Washington, DC, AACE (1995) 1-8, available online at http://www.contrib.andrew.cmu.edu/~plb/papers/AIED-95.html.

[4] Brusilovsky, P.: Adaptive and Intelligent Technologies for Web-based Education. Künstliche Intelligenz , 4 (1999) 19-25, available online at http://www2.sis.pitt.edu/~peterb/papers/KI-review.html.

[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 Nijhawan, H.: A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities. In: Driscoll, M. and Reeves, T. C. (eds.) Proc. of World Conference on E-Learning, E-Learn 2002, Montreal, Canada, AACE (2002) 154-161.

[7] Brusilovsky, P., Ritter, S., and Schwarz, E.: Distributed intelligent tutoring on the Web. In: du Boulay, B. and Mizoguchi, R. (eds.) Artificial Intelligence in Education: Knowledge and Media in Learning Systems. IOS, Amsterdam (1997) 482-489.

[8] Brusilovsky, P., Schwarz, E., and Weber, G.: ELM-ART: An intelligent tutoring system on World Wide Web. In: Frasson, C., Gauthier, G. and Lesgold, A. (eds.) Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 1086. Springer Verlag, Berlin (1996) 261-269.

[9] 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) Proceedings, Málaga, Spain (2002) 100-111.

[10] Constantino Gonzalez, M. A., Suthers, D., and Escamilla De Los Santos, J. G.: Coaching web-based collaborative learning based on problem solution differences and participation. International Journal of Artificial Intelligence in Education **13**, 1 (2002) To appear.

[11] Eliot, C., Woolf, B., and Lesser, V.: Knowledge extraction for educational planning. In: Proc. of Workshop on Multi-Agent Architectures for Distributed Learning Environments at AIED'2001, San Antonio (2001), available online at http://julita.usask.ca/mable/eliot.doc.

[12] Kay, J., Kummerfeld, B., and Lauder, P.: Personis: A server for user modeling. 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) Proceedings, Málaga, Spain (2002) 201-212.
[13] Machado, I., Martins, A., and Paiva, A.: One for all and all for one: a learner modelling server in a multi-agent platform. In: Kay, J. (ed.) SpringerWienNewYork, Wien (1999) 211-221.
[14] Melis, E., Andrès, E., Büdenbender, J., Frishauf, A., Goguadse, G., Libbrecht, P., Pollet, M., and Ullrich, C.: ActiveMath: A web-based learning environment. International Journal of Artificial Intelligence in Education **12**, 4 (2001) 385-407.
[15] Murray, T.: A Model for Distributed Curriculum on the World Wide Web. Journal of Interactive Media in Education (1998), available online at http://www-jime.open.ac.uk/98/5/.
[16] Oda, T., Satoh, H., and Watanabe, S.: Searching deadlocked Web learners by measuring similarity of learning activities. In: Proc. of Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX (1998), available online at http://www.sw.cas.uec.ac.jp/~watanabe/conference/its98workshop1.ps.
[17] Peylo, C., Thelen, T., Rollinger, C., and Gust, H.: A Web-based intelligent educational system for PROLOG. Institute for Semantic Information Processing, University of Snabrück, Montreal, Canada (2000) 85-96
[18] Rios, A., Millán, E., Trella, M., Pérez, J. L., and Conejo, R.: Internet based evaluation system. In: Lajoie, S. P. and Vivet, M. (eds.) Artificial Intelligence in Education: Open Learning Environments. IOS Press, Amsterdam (1999) 387-394.
[19] Ritter, S. and Koedinger, K. R.: An architecture for plug-in tutor agents. Journal of Artificial Intelligence in Education **7**, 3/4 (1996) 315-347.
[20] Roschelle, J., DiGiano, C., Koutlis, M., Repenning, A., Phillips, J., Jackiw, N., and Suthers, D.: Developing educational software components. IEEE Computer **32**, 9 (1999) 50-58, available online at http://e-slate.cti.gr/Resources/EEEComputerArticle.pdf.
[21] Rowley, K.: The challenge of constructing a Mega-tutor over the Web. In: Brusilovsky, P., Nakabayashi, K. and Ritter, S. (eds.) Proc. of Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education, Kobe, Japan, ISIR (1997) 52-53, available online at http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Rowley/Rowley.html.
[22] Vassileva, J. and Deters, R.: Dynamic courseware generation on the WWW. British Journal of Educational Technology **29**, 1 (1998) 5-14.
[23] Verhoeven, B., Cardinaels, K., Van Durm, R., Duval, E., and Olivié, H.: Experiences with the ARIADNE pedagogical document repository. In: Proc. of ED-MEDIA'2001 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, Tampere, Finland, AACE (2001) 1949-1954.
[24] WebCT: WebCT Course Management System, Lynnfield, MA, WebCT, Inc. (2002) available online at http://www.webct.com
[25] 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.
[26] 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.

Peter Brusilovsky is an assistant professor of Information Science and Intelligent Systems at the University of Pittsburgh. He received his PhD in Computer Science from the Moscow ‘Lomonosov’ State University in 1987. His research interests lie in the areas of adaptive hypermedia, adaptive web-based systems, student and user modelling, intelligent tutoring systems, and web-based education. He has authored over 100 technical papers and edited several books and special journal issues.