A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities

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

Abstract: This paper argues that a way to the new generation of powerful E-learning systems starts on the crossroads of two emerging fields: courseware re-use and adaptive educational systems. This paper presents the KnowledgeTree, a framework for adaptive E-learning based on distributed re-usable learning activities that we are currently developing. The goal of KnowledgeTree is 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.

Introduction

Adaptive Web-based educational systems and standard-based courseware re-use systems constitute two large research and development streams in the field of E-Learning. *Courseware re-use* systems emerged as a reaction to the standard practice of "hardwiring" high-quality educational material items in the course content. This practice made it impossible to reuse educational material and resulted in wasted efforts of the educational community as a whole due to the need to re-develop the same material again and again. The early answer to this problem was a database of educational resources and a courseware-reuse approach to authoring new courses (Olimpo et al., 1990). The courseware reuse ideas has found a fertile ground in Web-enhanced education. Some early large projects in the field of Web-based education like ARIADNE (Forte, Forte & Duval, 1996) and MTS (Graf & Schnaider, 1997) funded by the European Community were centered on courseware reuse. ARIADNE provides a very good example of courseware reuse architecture. It includes multiple *pools* (repositories) of educational material indexed with metadata and an open set of tools to produce, index, and reuse this material. Other well-known European project driven by the same motivation are PROMETEUS (http://www.prometeus.org/) and GESTALT (http://www.fdgroup.com/gestalt/). In the USA the reusability approach has been promoted by EOE Foundation (http://www.eoe.org/) and GEM Consortium (http://www.geminfo.org/).

Adaptive Web-based educational systems (Brusilovsky, 1999) emerged as an alternative to the traditional "one-size-fits-all" approach in the development of educational courseware. These systems build a model of the goals, preferences and knowledge of each individual student, and use this model throughout the interaction with the student in order to adapt to the needs of that student. The first pioneer adaptive Web-based educational systems were developed in 1995-1996 (Brusilovsky, Schwarz & Weber, 1996a; Brusilovsky, Schwarz & Weber, 1996b; De Bra, 1996; Nakabayashi et al., 1995; Okazaki, Watanabe & Kondo, 1996). Since that time, a good number of systems were created all around the world. The majority of adaptive Web-based educational systems are based on technologies developed in the areas of Adaptive Hypermedia (Brusilovsky, 1996) and Intelligent Tutoring Systems (Polson & Richardson, 1988).

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 (Figure 1). 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, advanced re-usable educational objects in modern Web-based education are not files, but activities (services) delivered by a Web server. These activities that can't be simply packaged, stored, and copies as an image, a text file, or even an applet - they have to reside on a dedicated server. This kind of activities is quite typical for adaptive Web-based systems. For example, ELM-ART, an adaptive LISP course (Brusilovsky et al., 1996a) 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. There is a clear need 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.

The second problem is related to the very idea of finding and attaching resources to online course material at the time of course development. The resource repositories are being constantly updated. Some better resources could be added to the existing repositories, some completely new repositories could become available. However, the students can't benefit from these resources due to the static nature of the approach.

The third problem is related to the "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.

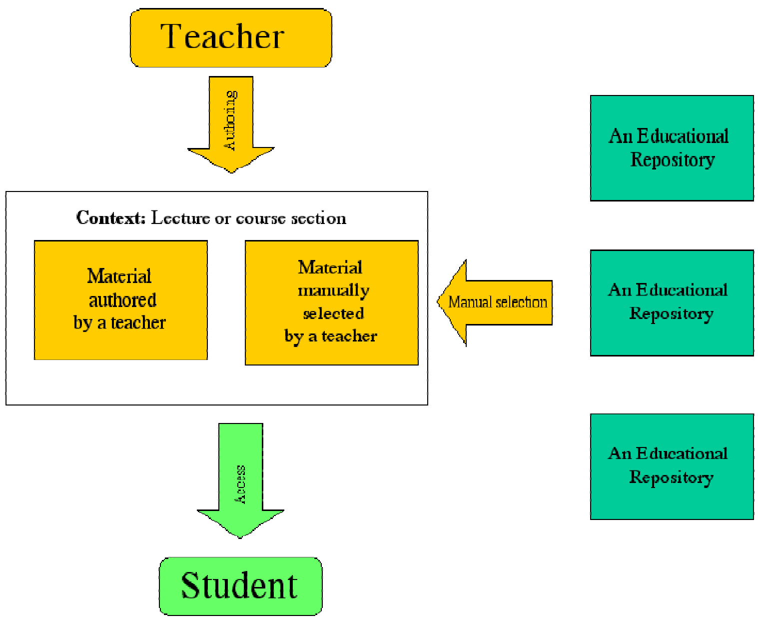

Figure 1: Courseware re-use approach to course design and delivery. Authoring tools allow the teachers to find and include resources into their course material. The student accesses static course material.

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 domain knowledge, 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. Educational material in many systems includes "service-style" activities backed by the intelligent capabilities of the system. At the same time, all known ITS and AH systems are build around a "close corpus" material. Collecting and preparing this material for use in adaptive systems is an expensive process. Thus these systems can't directly benefit from existing repositories of learning material.

We believe that a way to the future starts on the crossroads of courseware re-use and adaptive educational systems. This paper presents the KnowledgeTree, a framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities that we are currently developing. The goal of KnowledgeTree is 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. The following sections present our vision of 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.

KnowledgeTree: The Architecture

The KnowledgeTree is a distributed architecture for adaptive E-learning based on re-use of educational activities. It replaces the current monolithic course management systems (CMS) such as Blackbord (Blackboard Inc., 2002) or WebCT (WebCT, 2002) with a community of communicating servers. The architecture anticipates the presence of at least three kinds of servers: activity servers, learning portals, and student model servers (Figure 2). A *learning portal* plays a role similar to modern CMS. It allows a teacher to design a course and manages the student interaction with the course. The difference with CMS is that the learning content (activities) used by the students reside not in the portal, but in multiple distributed *activity servers*. An activity server plays a role similar to an educational repository in the sense that it hosts some (usually specialized) learning content. Unlike repositories that are essentially pools for storing learning materials that can be copied and inserted into courses, an activity server not only stores, but also delivers the activities. A portal has an ability to query activity servers for relevant activities and launch remote activities selected by students. An activity server should be able to inform portals about available activities and provide a complete support for a student working with one of its activities. *Student model server* collects data about student performance from each portal and each activity server that work with a student. In exchange, it provides information about the student that can be used by adaptive activity servers to personalize their communication with the student. The presence of multiple adaptive activities requires a centralized user modeling architecture.

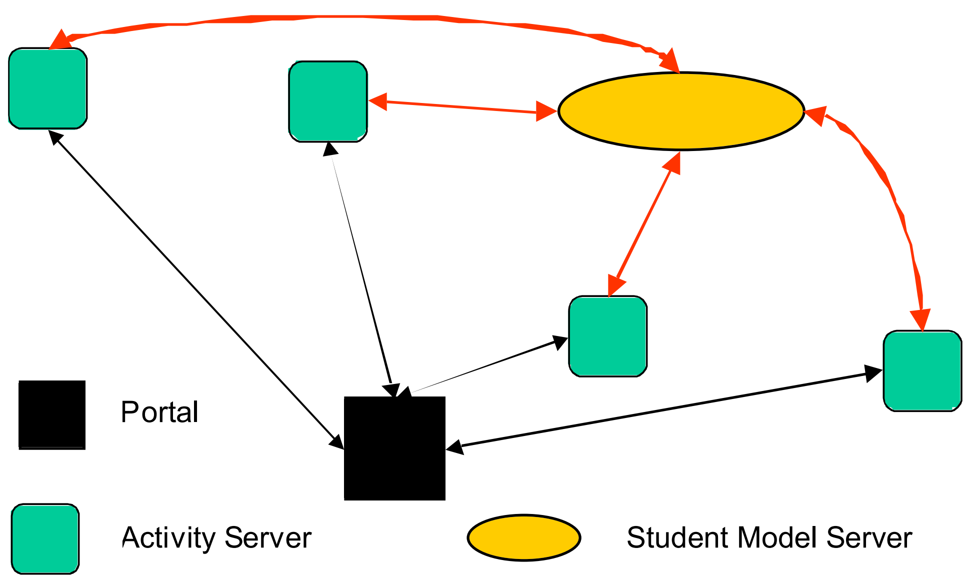

Figure 2: Main components of the KnowledgeTree distributed architecture.

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 adaptivity is provided by a student model server that collects student performance data from the portal and the activity servers and provides them with integrated information about the student. In particular, a student model server can reside on student's own computer and support just one user. It also can reside on a university computer and support the whole class of students.

The KnowledgeTree architecture is open and flexible. It allows the presence of multiple portals, activity servers, and user 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 is based on several clearly defined communication protocols between components. To start with, 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 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 provides offers very 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 (Brusilovsky, Ritter & Schwarz, 1997). 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 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 propagation" mechanism for various portals to exchange information (metadata) about known servers and activities.

KnowledgeTree: The Portal

The KnowledgeTree architecture allows multiple portals that can support different educational paradigms and approaches. At the moment, we have implemented one portal also called KnowledgeTree that is targeted to support a lecture-based educational process and is focussed on dynamic and adaptive selection of learning activities.

Main users of any portal are course authors (teachers) and learners. Course authors are responsible for shaping a course as a structured container of educational activities. The KnowledgeTree model allows an author to develop a course as a tree of modules (note that a sequence is also supported since it is a a one-level tree). While for most authors a module will correspond to a lecture, an author has a freedom to define larger modules that comprise several lectures as well as smaller-size modules. The course can also be structured independently from a sequence of lectures – as an interactive electronic book. In any case, the role of the author here is to structure the set of modules and to select primary educational material for each module. We distinguish primary material that comprises minimal set of activities necessary for an average student to learn the module and additional material that enhances learning experience and provides relevant activities for the students with different learning styles and levels of knowledge.

To select the material for each section an author specifies an educational goal for a section. 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. During the course design process, the educational goal is used by the system to select subset of relevant educational activities from multiple learning repositories known to the system. The selection can be done using a formal query to metadata-backed repositories or using a fuzzy text-based matching for repositories that have no metadata. 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.

The above process is quite similar to the process supported by advanced courseware re-use tools. New feature of the KnowledgeTree model is that the learning goal specified by the author is retained and stored with the module. 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 (Figure 3).

It is easy to anticipate that in the future, when learning repositories will be quite rich, the runtime selection will return a relatively large number of relevant learning activities. In this context, adaptive hypermedia technologies will provide further adaptation for an individual student. 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. Here the system will adapt to student knowledge and individual learning styles. Adaptive presentation will be used to deliver selected items adaptively. Here the system will adapt to the student level of knowledge and educational goal. In addition, the system also allows the student to search for relevant educational material using her own criteria and to add material permanently to the module. It produces a dynamic and personalized learning space for each course module where personalization is provided by both the system and the student (Figure 3).

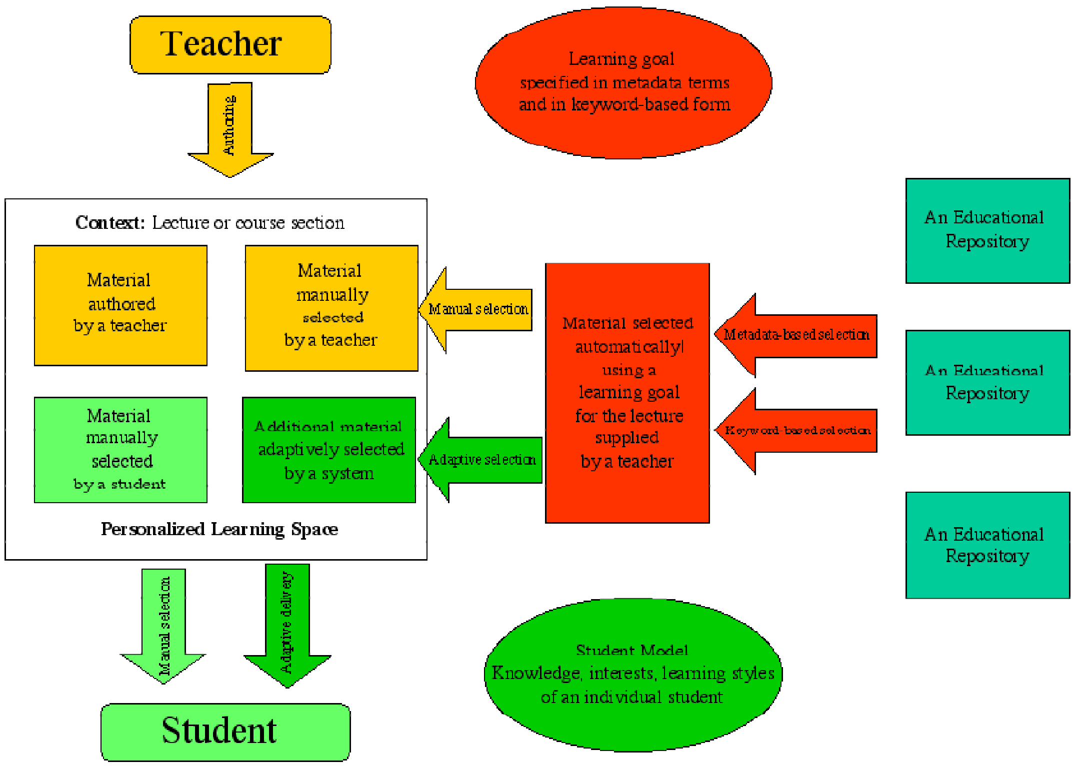

Figure 3: KnowledgeTree portal combines the benefits of courseware re-use systems and adaptive Web-based educational systems. It solves the problem of statically built courses, and provides personalized support that maximizes an educational opportunity for every learner.

The above scheme presents the most generic case of the system use. The system naturally supports any subset of the underlined functions. For example, the author may specify a partial educational goal and choose not to select any primary educational material. In that case, the system still will be able to use a partial goal to select and organized relevant educational material for every section. Thus, with little efforts from the author the system will be able to provide an impressive level of adaptive support.

Current State of Work

In addition to the overall architecture, a set of protocols, and the KnowledgeTree portal, the list of components developed so far includes four protocol-compliant activity servers and a simple user modeling server. Three of these activity servers have been developed 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 (Brusilovsky, 2001) serves interactive annotated program examples, the QuizPACK (Pathak & Brusilovsky, 2002) serves parameterized questions, and WADEIn (Brusilovsky & Su, 2002) serves demonstrations and exercises related with expression evaluation. The fourth server KnowledgeSea (Brusilovsky & Rizzo, 2002) 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. WebEx is implemented using Microsoft ASP technology and served by Internet Information Server (IIS) working on a Windows PC. QuizPACK is developed as a set of C++ CGI scripts and is served by Apache server working on a SUN Solaris platform. WADEIn is implemented as a configurable Java applet embedded into a page generated by a Java servlet running on a Tomcat server. KnowledgeSea is based on JavaScript functionality and can be delivered by any Web server. Each server can work independently from the KnowledgeTree architecture, but will require a student to login in this mode of work. Only one of the activity servers (WADEIn) serves adaptive activities. It use the information about students to adapt to their knowledge. Other servers use information about the student simply to trace the student performance. All these servers implement our simple transparent login protocol, resource delivery protocol, and student modeling protocol. They can work (with transparent login) with any compliant portal and user modeling server.

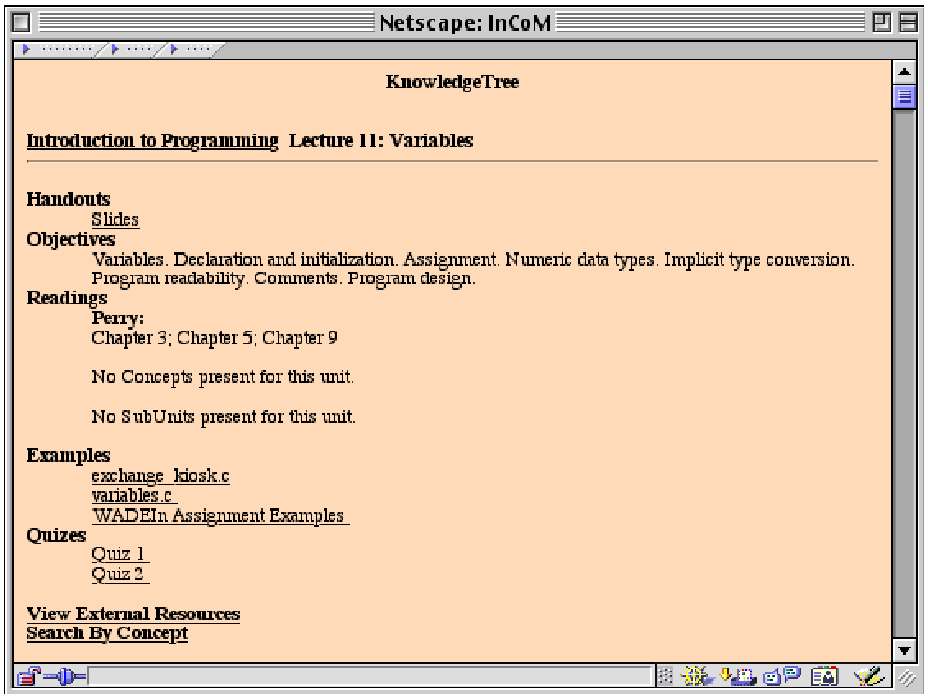

Figure 4: A module view in a second version of KnowledgeTree portal. The learning activities in Examples and Quiz categories are delivered by external activity servers WADEIn, WebEx, and QuizPACK.

The first version of the KnowledgeTree portal together with WebEx and QuizPACK servers and a primitive student model server was piloted in the Fall 2001 semester in the context of a programming and data structure course. In Spring 2002 semester the version 1.5 of the portal (Figure 4) and all four activity servers were used as a primary course support tool in the context of another programming course. Many activities created for the first course were re-used in the second course and we have appreciated ourselves how easy was it to assemble a course from re-used activities. The students were using the system and its components on everyday basis. All components of the system were formally evaluated and got very positive feedback from the students.

Currently we are completing the second version of the portal. Following the earlier versions, it is implemented using Java Servlet and JDBC technologies and delivered by Tomcat server. It supports hierarchical course design (first version supported sequences only), specification of educational objectives of a module in terms of course concepts/topics. It is able to retrieve external learning activities that matches the educational objectives of the module. The second version also supports multiple course authors and groups. We are also developing the second version of the student model server. The second version implements in full the centralized user modeling approach developed in our earlier research. (Brusilovsky, 1994; Brusilovsky, 1995). This version is based on Java Servlet and JDBC and will replace the current simple user model server that is based on Microsoft ASP technology.

We expect that in the Fall 2002 semester several faculty will be using the KnowledgeTree in the context of their courses. The readers are welcome to try the current version of the system available from our lab home pages: http://www2.sis.pitt.edu/~peterb/taler.html.

Other Relevant Research and the Prospects for the Distributed Framework

Significant amount of work and cooperation of several research groups is required to turn the proposed framework into the practice of e-learning. 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. However, any open architecture has to be based on a number of standards. Fortunately, our work shares some core goals with several other active research areas. Instead of inventing the solution we intend to re-use as much as possible standards, solutions and ideas from these area.

- The problem of searching for relevant educational activities in learning repositories is now well explored by courseware re-use movement and Learning Object Metadata groups (such as LTSC http://ltsc.ieee.org) in particular. Solutions developed within this field can be directly adopted by our framework
- A good variety of powerful adaptation methods and techniques have been explored in the field of adaptive Web-based educational systems. This field is our primary source of ideas for developing both the portals and the adaptive activities.
- Several consortia such as uPortal (http://www.uportal.org) and AICC (http://www.aicc.org) explore the issues of distributed component-based architectures for E-learning as an alternative to monolithic courseware management systems. These groups have already produced some solutions for transparent authentication and communication standards between a portal and an "intelligent" learning activity.
- The problem of gathering and sharing metadata of distributed resources has been carefully investigated in the field of Web information retrieval. Some interesting centralized architectures as FAB (Balabanovic & Shoham, 1997) and decentralized architectures as Harvest were suggested. These ideas can be certainly re-used for the needs of E-learning. In the context of e-learning EDUTELLA (http://edutella.jxta.org/) and LOMster (Ternier, Duval & Vandepitte, 2002) projects develop frameworks for peer-to-peer metadata exchange.
- The issue of user and student modeling for multi-component adaptive systems has been well-researched in the fields of ITS and User Modeling. A number of user and student model servers have been already reported (Kobsa, 2001). These works can certainly contribute to the development of the user model component of KnowledgeTree framework. The AICC CMI standard can also contribute to this aspect of our framework since it suggest a way for an "activity" to report the results of student work with it.

We should conclude by mentioning that the problems of developing distributed adaptive and intelligent educational system based on shared educational resources has also been explored in the field of ITS (Brusilovsky, 1995; Brusilovsky et al., 1997; Eliot, Woolf & Lesser, 2001; Murray, 1998; Ritter, Brusilovsky & Medvedeva, 1998; Ritter & Koedinger, 1996). However, the lack of matching works in other fields and the proper technology in general didn't allow these pioneer works to go beyond the level of ideas and simple lab systems. The current situation is quite different. Not only we can now use some matching solutions from the areas listed above, the Web in general now provide a powerful platform for implementation of the long discussed ideas. 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. We hope that our group together with other groups motivated by similar goals will be able will now succeed in bringing our a new generation of E-learning systems that integrates best features of several emerging systems.

References

Balabanovic, M. and Shoham, Y. (1997) Fab: content-based collaborative recommendation. *Communications of the ACM* 40 (3), 66-72.

Blackboard Inc. (2002) Blackboard Course Management System 5.1, Blackboard Inc. http://www.blackboard.com/ (Accessed 21 January, 2002)

Brusilovsky, P. (1994) Student model centered architecture for intelligent learning environment. In: Proceedings of Fourth International Conference on User Modeling, Hyannis, MA, 15-19 August 1994, MITRE, pp. 31-36.

Brusilovsky, P. (1995) Intelligent learning environments for programming: The case for integration and adaptation. In: J. Greer (ed.) Proceedings of AI-ED'95, 7th World Conference on Artificial Intelligence in Education, Washington, DC, 16-19 August 1995, AACE, pp. 1-8, also available at http://www.contrib.andrew.cmu.edu/~plb/papers/AIED-95.html.

Brusilovsky, P. (1996) Methods and techniques of adaptive hypermedia. In P. Brusilovsky and J. Vassileva (eds.), *User Modeling and User-Adapted Interaction* 6 (2-3), Special Issue on Adaptive Hypertext and Hypermedia, 87-129.

Brusilovsky, P. (1999) Adaptive and Intelligent Technologies for Web-based Education. In C. Rollinger and C. Peylo (eds.), *Künstliche Intelligenz* (4), Special Issue on Intelligent Systems and Teleteaching, 19-25, http://www2.sis.pitt.edu/~peterb/papers/KI-review.html.

Brusilovsky, P. (2001) WebEx: Learning from examples in a programming course. In: W. Fowler and J. Hasebrook (eds.) Proceedings of WebNet'2001, World Conference of the WWW and Internet, Orlando, FL, October 23-27, 2001, AACE, pp. 124-129.

Brusilovsky, P., Ritter, S., and Schwarz, E. (1997) Distributed intelligent tutoring on the Web. In: B. du Boulay and R. Mizoguchi (eds.) *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*. (Proceedings of AI-ED'97, 8th World Conference on Artificial Intelligence in Education, 18-22 August 1997) Amsterdam: IOS, pp. 482-489.

Brusilovsky, P. and Rizzo, R. (2002) Map-Based Horizontal Navigation in Educational Hypertext. In: K. M. Anderson, S. Moulthrop and J. Blustein (eds.) Proceedings of 13th ACM Conference on Hypertext and Hypermedia (Hypertext 2002), College Park, MD, June 11-15, 2002, ACM, pp. 1-10.

Brusilovsky, P., Schwarz, E., and Weber, G. (1996a) ELM-ART: An intelligent tutoring system on World Wide Web. In: C. Frasson, G. Gauthier and A. Lesgold (eds.) *Intelligent Tutoring Systems*. Lecture Notes in Computer Science, Vol. 1086, (Proceedings of Third International Conference on Intelligent Tutoring Systems, ITS-96, Montreal, June 12-14, 1996) Berlin: Springer Verlag, pp. 261-269.

Brusilovsky, P., Schwarz, E., and Weber, G. (1996b) A tool for developing adaptive electronic textbooks on WWW. In: H. Maurer (ed.) Proceedings of WebNet'96, World Conference of the Web Society, San Francisco, CA, October 15-19, 1996, AACE, pp. 64-69, also available at http://www.contrib.andrew.cmu.edu/~plb/WebNet96.html.

Brusilovsky, P. and Su, H.-D. (2002) Adaptive Visualization Component of a Distributed Web-based Adaptive Educational System. In: *Intelligent Tutoring Systems*. Vol. 2363, (Proceedings of 6th International Conference on Intelligent Tutoring Systems (ITS'2002), Biarritz, France, June 2-7, 2002) Berlin: Springer-Verlag, pp. 229-238.

De Bra, P. M. E. (1996) Teaching Hypertext and Hypermedia through the Web. *Journal of Universal Computer Science* 2 (12), 797-804, http://www.iicm.edu/jucs_2_12/teaching_hypertext_and_hypermedia.

Eliot, C., Woolf, B., and Lesser, V. (2001) Knowledge extraction for educational planning. In: Proceedings of Workshop on Multi-Agent Architectures for Distributed Learning Environments at AIED'2001, San Antonio, May 19, 2001, also available at http://julita.usask.ca/mable/eliot.doc.

Forte, E., Forte, M. W., and Duval, E. (1996) ARIADNE: A supporting framework for technology-based open and distance lifelong education. In: F. Maffioli, M. Horvat and F. Reichl (eds.) Proceedings of Educating the engineer for lofelong learning. SEFI Annual Conference '96, Vienna, Austria, September 11-13, 1996, pp. 137-142.

Graf, F. and Schnaider, M. (1997) IDEALS MTS - EIN modulares Training System für die Zukunft. In: C. Herzog (ed.) Proceedings of 8. Arbeitstreffen der GI-Fachgruppe 1.1.5/7.0.1 "Intelligent Lehr-/Lernsysteme, Duisburg, September 18-19, 1997 Published as No. TUM-19736, Technische Universität München, München. pp. 1-12.

Kobsa, A. (2001) Generic user modeling systems. *User Modeling and User Adapted Interaction* 11 (1-2), Special Issue on Ten Year Anniversary Issue, 49-63, http://www.ics.uci.edu/~kobsa/papers/2001-UMUAI-kobsa.pdf.

Murray, T. (1998) A Model for Distributed Curriculum on the World Wide Web. In J. Spohrer, T. Sumner and S. B. Shum (eds.), *Journal of Interactive Media in Education* , Special Issue on Educational Authoring Tools and the Educational Object Economy, http://www-jime.open.ac.uk/98/5/.

Nakabayashi, K., Koike, Y., Maruyama, M., Touhei, H., Ishiuchi, S., and Fukuhara, Y. (1995) An intelligent tutoring system on World-Wide Web: Towards an integrated learning environment on a distributed hypermedia. In: H. Maurer (ed.) Proceedings of ED-MEDIA'95 - World conference on educational multimedia and hypermedia, Graz, Austria, June 17-21, 1995, AACE, pp. 488-493.

Okazaki, Y., Watanabe, K., and Kondo, H. (1996) An Implementation of an intelligent tutoring system (ITS) on the World-Wide Web (WWW). *Educational Technology Research* 19 (1), 35-44.

Olimpo, G., Persico, D., Sarti, L., and Tavella, M. (1990) On the concept of database of multimedia learning material. In: Proceedings of World Conference on Computers and Education, Amsterdam, Australia, 1990, North Holland, pp. 431-436.

Pathak, S. and Brusilovsky, P. (2002) Assessing Student Programming Knowledge with Web-based Dynamic Parameterized Quizzes. In: P. Barker and S. Rebelsky (eds.) Proceedings of ED-MEDIA'2002 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, Denver, CO, June 24-29, 2002, AACE, pp. 1548-1553.

Polson, M. C. and Richardson, J. J. (eds.) (1988) Foundations of intelligent tutoring systems. Hillsdale: Lawrence Erlbaum Associates.

Ritter, S., Brusilovsky, P., and Medvedeva, O. (1998) Creating more versatile intelligent learning environments with a component-based architecture. In: B. P. Goettl, H. M. Halff, C. L. Redfield and V. J. Shute (eds.) Lecture Notes in Computer Science, Vol. 1452, (Proceedings of 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX, August 16-19, 1998) Berlin: Springer Verlag, pp. 554-563.

Ritter, S. and Koedinger, K. R. (1996) An architecture for plug-in tutor agents. *Journal of Artificial Intelligence in Education* 7 (3/4), 315-347.

Ternier, S., Duval, E., and Vandepitte, P. (2002) LOMster: Peer-to-peer Learning Object Metadata. In: P. Barker and S. Rebelsky (eds.) Proceedings of ED-MEDIA'2002 - World Conference on Educational Multimedia, Hypermedia and Telecommunications, Denver, CO, June 24-29, 2002, AACE, pp. 1942-1943.

WebCT (2002) WebCT Course Management System 3.8, WebCT, Inc., Lynnfield, MA. http://www.webct.com (Accessed 2 July, 2002)