From Learning Objects to Adaptive Content Services for E-Learning

Peter Brusilovsky

School of Information Sciences
University of Pittsburgh
Pittsburgh PA 15260, USA
+1 412 6249404

peterb@mail.sis.pitt.edu

Vincent P. Wade and Owen Conlan

Knowledge & Data Engineering Research Group

Department of Computer Science
Trinity College Dublin
Ireland
+353 1 6082091

Vincent.Wade@cs.tcd.ie, Owen.Conlan@cs.tcd.ie

Abstract

This paper argues that a new generation of powerful E-learning systems could start on the crossroads of two emerging fields: courseware re-use and adaptive educational systems. We argue for a new distributed architecture for E-learning systems based on the idea of adaptive reusable content services. This paper discusses problems that have to be solved on the way to the new organization of E-learning and reviews existing approaches and tools that are paving the way to next generation E-learning systems. It also presents two pioneer systems - APeLS and KnowledgeTree that have attempted to develop a new service-based architecture for adaptive E-learning.

Keywords

Web-Based Education, Electronic Learning (E-Learning) Comparative Study, Learning Resources, System Architecture

INTRODUCTION

Adaptive Web-based educational systems and standard-based courseware re-use systems constitute two significant streams of research and development in the field of E-Learning. *Courseware re-use* systems have emerged as a reaction to the standard practice of "hardwiring" high-quality educational materials within course content. This practice made it impossible to reuse educational material and resulted in the wasted efforts of the educational community as a whole 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, Persico, Sarti, & Tavella, 1990). The courseware reuse ideas have 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 such a courseware reuse approach. ARIADNE provides a very good example of a courseware re-use 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 projects driven by the same motivation are PROMETEUS (http://www.prometeus.org/) and GESTALT (Wade & Doherty, 2000). 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 & Peylo, 2003) 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, 1996b; De Bra, 1996; Nakabayashi et al., 1995; Okazaki, Watanabe, & Kondo, 1996). Since that time, a number of systems have been created all around the world. The majority of adaptive Web-based educational systems are based on technologies developed in the areas of Adaptive Hypermedia (AH) (Brusilovsky, 1996) and Intelligent Tutoring Systems (ITS) (Polson & Richardson, 1988).

The methods and tools developed by both researchers of courseware re-use systems and adaptive Web-based educational systems can contribute to creating better Web-enhanced courses. 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 retrieval abilities of modern educational material repositories and the just-in-time delivery and personalization power of ITS and AH technologies. We start with a brief analysis of these approaches comparing their strong and weak points (illustrated later in Table 1).

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 "paste" them into 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 two serious problems.

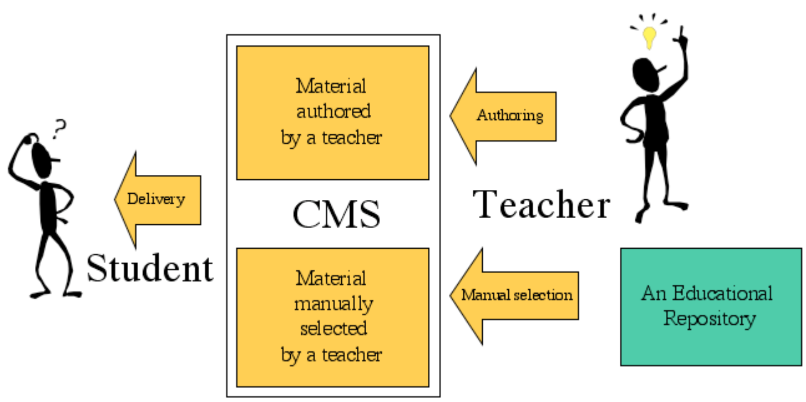

Figure 1: Courseware re-use approach to course design and delivery. Authoring tools allow the teachers to find and include resources into their course material. The student accesses static course material.

Firstly, courses developed with this re-usability approach suffer from "one size fits all" problem. When identifying relevant material and organizing it within a course section, teachers have to think about the class in general. The students in any 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 might 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 constantly increasing.

Secondly, 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* the learning object into the course to be created. However, fragments of adaptive educational content in modern Web-based systems are often not files but services delivered by dedicated Web servers. These activities can not 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 and launched from it on demand. An inability of modern reusability frameworks to handle interactive services makes it impossible for their users to work with more highly interactive learning content that is highly interactive or adaptive by its nature. In turn, without this kind of advanced content, courses developed with existing frameworks are most often simple “page turners”.

The situation in the world of adaptive Web-based systems is quite different. These systems offer a different set of benefits to their users. Adaptive E-Learning systems solve the “one size fits all” problem by attempting to provide the best support for every student. Using individual student models and educational material enhanced with pedagogical metadata (such as required competencies, or relevancy to learning styles) Adaptive Hypermedia Systems (AHS) and Intelligent Tutoring Systems (ITS) 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 (Figure 2). Many of these systems are able to offer advanced interactive and adaptive learning activities to their users. Observing students’ work with these activities is the best way for these systems to keep their student models up-to-date.

The problems of existing adaptive e-Learning systems are also quite different from the problems of existing re-usability frameworks. The source of these problems is simple: adaptive educational systems are not designed for the modern E-Learning context, where a teacher or course provider is typically interested in developing a specifically targeted course by re-using existing educational content from multiple repositories. Currently, existing adaptive e-Learning systems have to be used as a whole, not component by component. This creates a significant obstacle for their practical application. While adaptive technologies themselves can be applied to achieve any specified learning goals, in most existing systems, educational goals are pre-defined by the authors (of the adaptive content). The adaptivity in the course is frequently embedded across the content itself or is in the actual adaptive system, which executes the content. A teacher who is interested in re-using some adaptive content from an existing adaptive hypermedia system has only one choice - to accept the whole system with its goals and sacrifice his or her specific teaching goals. Naturally, except from the authors of existing adaptive systems themselves, such acceptance is difficult to achieve. Secondly, most known adaptive hypermedia systems are built around “closed corpus” learning material and are not able to take into account any external content. Those who adopt such adaptive systems have to accept the inability to benefit from any new content coming from existing repositories of learning material.

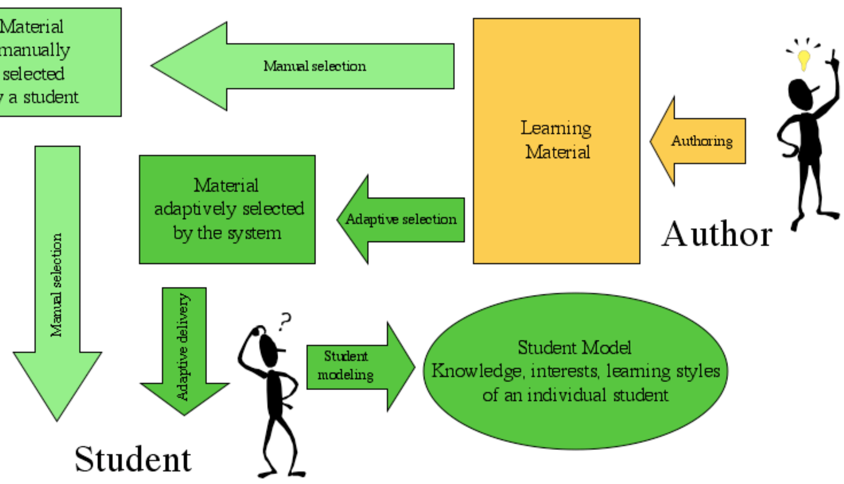

Figure 2: An adaptive educational hypermedia system can adaptively serve relevant educational material to every student. However, it is not designed for the modern E-Learning context.

As we can see from the analysis above, both leading approaches have strengths and weaknesses (Table 1). While their strong points allow them to support a reasonable number of E-Learning scenarios, their weaknesses prevent them from supporting the most exciting and promising model of E-Learning – allowing teachers to re-use adaptive and interactive learning activities in the context of their courses. Adaptive E-Learning systems can include "hardwired" adaptive and interactive activities, but they provide little support for the teacher in re-using external activities organized along his or her preferred course structure. Re-usability frameworks allow the teacher to develop a course re-using some external activities, but they can't handle interactive and adaptive content.

	Reusability frameworks	Adaptive E-Learning systems	Adaptive Content Services
Support teacher in developing a custom E-Learning course	Yes	No	Yes
Allow the use of external repositories of learning material	Yes	No	Yes
Can adapt to individual students	No	Yes	Yes
Can include advanced learning activities	No	Yes	Yes
Allow re-use of external advanced learning activities	No	No	Yes

Table 1: The features of the E-Learning paradigms compared

A good illustration of this unfortunate situation is the case of ELM-ART LISP problems. ELM-ART is an adaptive LISP course (Brusilovsky, Schwarz, & Weber, 1996b) that includes many LISP programming problems. ELM-ART problems are more than just textual problem statements. These problems are fully interactive learning activities backed by ELM-ARTs unique knowledge-based functionality. In response to student program solutions sent to an ELM-ART server, the system can check, diagnose, and correct them. The results of this diagnosis are used to update the student model that ELM-ART uses to adapt to that student. ELM-ART problems can't be moved or copied - they have to be served directly from a dedicated ELM-ART server. At the moment, in order to work with ELM-ART problems a student has to log in to ELM-ART and to navigate to the parts of the ELM-ART adaptive hypermedia course where these problems are located. Imagine that a computer science teacher wants to re-use ELM-ART problems to enrich her own LISP course that is different from the one supported by ELM-ART. This would greatly enrich the course allowing it to benefit from more than 10 years of research behind ELM-ART adaptive diagnosis technology. Currently it is simply not possible – these problems can't be extracted from ELM-ART and re-used elsewhere. If a teacher wants to use these problems, she has to use the whole ELM-ART course giving up her "own" course structure. Alternatively, if a teacher insists on her own way to teach LISP, she will be unable to use the ELM-ART problems and settle for simpler "reusable" learning content such as static Web pages or pictures. ELM-ART problem are only some of the hundreds of advanced adaptive and interactive content items, which currently are available on the Web. They can be used in their original context (as long as the original system is running which is often not the case), but can't be re-used in the context of new courses. Moreover, this situation prevents content developers from creating new advanced (and expensive) interactive content. Without clear prospects for broad re-use, the cost of developing advanced content is rarely justified.

To address this problem we argue that E-learning needs a new re-usability framework that shifts the focus from re-usable "static" learning objects to re-usable "dynamic" adaptive content services. This new framework should allow course developers to develop their courses in their preferred way, while facilitating the re-use of powerful, adaptive and interactive content. It should also allow content developers to focus on developing more advanced content. An adaptive content services framework will combine strong features of the two analyzed approaches (Table 1). The authors of this paper have been exploring the ideas of adaptive content services in their projects for several years and developed a new vision of a service-based distributed architecture for adaptive E-Learning. In this paper we present our vision of this architecture, analyze several known problems, and describe our attempts to implement these architectures in our Adaptive Personalized eLearning Service (APeLS) and KnowledgeTree frameworks. It is important to stress that both frameworks, while implementing innovative ideas, attempt to stay close to the needs of practical Web-based education. The authors integrate a vision of academic researchers with a solid practical experience in commercial E-Learning. To prove our ideas we have used our frameworks in several real courses at the University of Pittsburgh and Trinity College Dublin.

ADAPTIVE SERVICES FOR E-LEARNING

The research challenge discussed in this paper was to develop a framework for E-Learning that combines the attractive features of the modern re-usability approach to E-learning with the power of adaptive Web-based systems. It means that the target framework should keep the winning features of the re-usability approach allowing teachers to structure a course according to their specific needs while also helping them to re-use existing relevant learning content instead of creating everything from scratch. At the same time, the framework should allow teachers to create adaptive courses and re-use not just files, but any interactive learning activity. This challenge was addressed by several research groups, which suggested and explored a range of innovative ideas (see *Related Work* section below).

This paper presents and compares the frameworks developed by research groups Pittsburgh and Dublin. Both groups attempted to address the problem as a whole and it resulted in two solutions, which are very similar in a number of critical features. The first key feature of our solutions was to separate the course management system (also referred to as a learning management system) from the content. In our vision, the course management system is a portal that provides structured access to educational content without storing it. The content itself comes directly from different content services that are independent from any portal and generally reside on different servers distributed over multiple locations. Portals are maintained by *course providers* while content services are maintained by *content providers*. Many portals can use the same content service in different contexts.

The second key feature was to separate content specification from the real content. In the traditional (re-use) model the search for the relevant content starts with some kind of content specification in terms of duration, pedagogical type, topics covered, etc. The teachers then attempt to find the desired content in a repository by issuing a formal search query in terms of content metadata. Finally, the relevant content is manually selected, copied, and integrated into the course. In our model, the teacher is able to stop at the stage of specification of the desired material (a narrative in APeLS, a sequence of lectures with objectives in KnowledgeTree). The portal at runtime, resolves this specification by automatically finding or generating relevant content.

The key features of our frameworks listed above allow us to achieve the target functionality. The portal-service separation makes it easy for the system to re-use any interactive or adaptive content that can't be just embedded into the traditional course management system. The separation of content specification and resolution in time opens the way to adaptivity. While staying within the objectives specified by a teacher in a content specification, the framework can take into account knowledge, goals, and other features of an individual student and adaptively select, generate, and arrange relevant learning content. In addition, this architecture solves the problem of outdated content. The resource repositories are being constantly updated. Some better resources could be added to the existing repositories, some completely new repositories could become available. When content specification is resolved dynamically the overall system can benefit from the best content available at the time of using the system.

It is important to note that the proposed architecture is not a radical replacement, but an extension of the existing re-usability approach. The new architecture allows teachers to be flexible by specifying content requirements as well as allowing them to re-use customizable, distributed and dynamic services. At the same time new functionality does not replace the old one, but safely co-exists with it. A teacher who wants to use a specific piece of content (either a file or a service) can still use the existing approach based on manual selection of content and its static allocation to a specific location in the courseware. Figure 3 presents a combination of automatic runtime selection of material based on the concepts in the course structure specified by the teacher, as well as allowing teacher intervention in manually selecting material for the course. It indicates how the teacher can provide the overall course structure i.e. define the scope, concepts and kinds of learning activities to be included in the course, and indicate the kinds of adaptivity that the teacher would like to be offered to the student e.g. adaptivity based on student's prior knowledge/competencies, goals, learning styles etc. The adaptive service then reconciles this structure (course requirements) with appropriate composed learning material (learning objects) and includes appropriate adaptive techniques to support kinds of adaptivity indicated by the teacher. The teacher is also able to include manually selected content if she so wishes. The adaptive (personalized) course is then offered to the student. It should be noted that in better adaptive E-Learning systems, the selection of actual content is made while the student is using the course, so that the very latest student and content information is used to satisfy the students learning needs. On first logging on to the adaptive course, the student is asked about various aspects to assist in populating that student's model. Instruments to illicit a student's model could include questionnaires, pre-tests, or monitoring instruments which augment the student model during the usage of the course. A more comprehensive description of the challenges and methodology for adaptive course composition and the tools that support it, are presented in (Dagger, Wade, & Conlan, 2004).

Since our groups were driven by similar goals, our solutions are very similar - they advocate the distributed architecture based on re-usable and adaptive content services and runtime adaptive resolution of content specifications. At the same time, technically our solutions are quite different and demonstrate two possible approaches to the incremental move from the current approach based on courseware management systems and learning objects to the more powerful frameworks based on open portals and distributed adaptive content services. These solutions are presented in the two following sections.

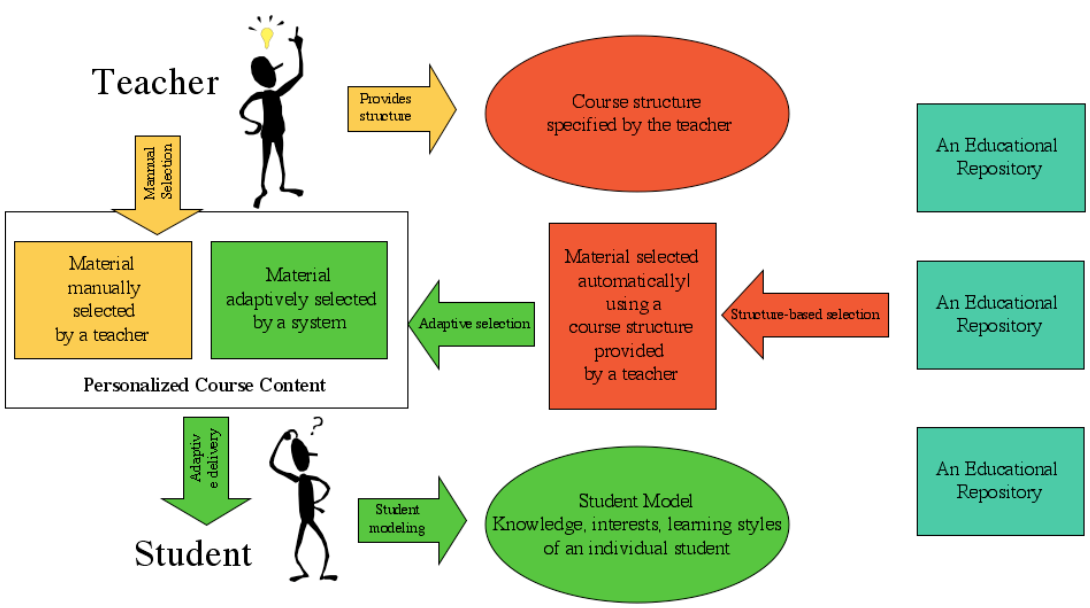

Figure 3: The target system should combine the benefits of courseware re-use systems and adaptive Web-based educational systems.

ADAPTIVE PERSONALIZED E-LEARNING SERVICE

The Adaptive Personalized eLearning Service (APeLS) was developed as a service to deliver personalized educational courses based on a multi-model, metadata driven approach (ADLI, 2003). Two important features of the architecture address the twin goals of Adaptive Content Services:

- The adaptive courseware is NOT offered as content but rather as a service which can be delivered through a portal (or a conventional learning management system). APeLS offers a service interface via which the adaptive course can be delivered. The interface also offers a separate API to pass administrative and learner performance information to a portal, LMS or another management system.
- The adaptive service is driven by separate models of learning content, narrative (i.e. a concept traversal including learning strategy), and learner model. An adaptive engine component of the service, reconciles the three models at runtime to dynamically generate the personalized course for the learner. By allowing the learner access to the learner model, he/she is free to re-personalize and adapt of the course during runtime.

The Content Model contains metadata descriptions of the actual small size learning objects (we term the learning objects pagelets to indicate their typical size as occupying less than a screen area). The Narrative Model only refers to concepts which may be selected as part of a course. There is no direct reference between the narrative model and actual content. The mapping between narrative and content is performed at run-time by the service engine, which reconciles the metadata imperatives of the narrative model with the metadata of the content model. Thus the narrative and content models are linked via a shared (or mapped) metadata vocabulary. The metadata used in APeLS is an extension of the IEEE LOM (IEEE LTCS WG12, 2002) and ADL SCORM (ADLI, 2003). The third model upon which APeLS provides adaptivity is the learner model. These models can be populated from the learning portal, Learning Management System (LMS), or (more typically) captured via a learner instrument under the control of the learner. The APeLS architecture is extensible in that other models can be developed by the content service provider and can be easily integrated into the service. One such model would be a ‘terminal’ or ‘device’ model. This allows adaptivity based on the presentation power of the learners access device (e.g. PDA, laptop, eBook etc.). In this way the Adaptive Learning Service can be provide even more flexibility of content delivery.

APeLS is currently being used to deliver personalized online courses in SQL (Structured Query Language) to final year undergraduate students in Trinity College, Dublin. It has also been used within the EASEL (Wade & Doherty, 2000) IST project and iClass Project (2004-2008) to demonstrate the discovery and integration of Adaptive Hypermedia Services with traditional (static) online learning content.

APeLS Models and Architecture

APeLS utilizes a number of metadata and information repositories (Figure 4) –

- *Learner Metadata Repository* – This repository stores all of the metadata representing the individual learners in the system. This metadata conforms to the Learner Model

(Owen Conlan, Wade, Bruen, & Gargan, 2002; O'Keeffe, Brady, Conlan, & Wade, 2006).

- *Content Metadata Repository* – Stored in this repository are the metadata records, conforming to the Content Model corresponding to each piece of learning content (or learning object metadata)
- *Narrative Metadata Repository* – Stored in this repository are the metadata records that describe the learning objectives and pedagogical approach for each narrative in the narrative repository (i.e. description of the available narratives).
- *Content Repository* – All of the pagelets referred to by the Content Model Repository are contained in this repository

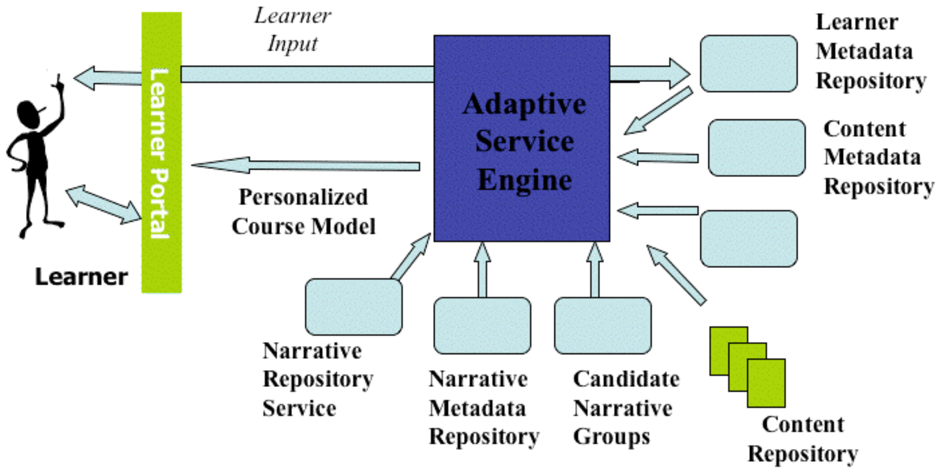

Figure 4. Adaptive Hypermedia Service Architecture

- *Narrative Repository* – The narrative repository stores all of the narratives used to construct personalized courses.
- There are also two candidate group repositories:
- *Candidate Content Groups* – The groups in this repository reference metadata in the Content Metadata Repository that fulfill the same learning goal. The content model metadata determines how the content differs, i.e. technically or in educational approach.
- *Candidate Narrative Groups* – This repository determines groups of narratives that encapsulate the same knowledge, but employ different pedagogical approaches to structuring the content.

At the core of APeLS is the Adaptive Engine (AE), which uses the Java Expert System Shell (JESS) with customized functions as the basis of its rules engine. The role of the rules engine is to produce a model for a personalized course based on a narrative and the learner model. The XML-based personalized course model encapsulates the structure of the learner’s course and contains the candidate content groups that fulfill the learner’s learning requirements in accordance with the narrative. The candidate content groups can be thought of as the abstraction layer between the narrative (which defines various dynamic courses in terms of concepts) and the actual content (fine grained learning objects).

The AE also utilizes a candidate selector for choosing the appropriate narrative by reconciling information in the learner model with the candidate narrative groups. The candidate selector is also used to choose the appropriate piece of content to deliver from a candidate content group when the personalized course content is being generated from the personalized course model.

The AE has a learner modeler component that enables input from the course to be translated into changes in the learner's information. This learner modeler component is used to populate the learner's model when the learner initially enters the Adaptive Hypermedia Service. It can also be used directly during runtime to modify the learner's model – these modifications may either be initiated by the learner or by the engine itself and can be initiated directly from the JSP.

The APeLS service is based on the notion that an adaptive content provider should be a service provider rather than a repository for extraction of content. Communication between APeLS and a learning portal (or LMS) is achieved by enhancing the SCORM Runtime Communication API as used in SCORM v1.3.

This requires a modification to the HTML frame layout for the APeLS to enable calls to API functions residing on the LMS from APeLS content. The actual API calls used are the same as those used in SCORM v1.3 as the API is designed to get and set values that are separately defined by an external data model. The remote APeLS calls the Content Interworking API to access the data model on the Learning Portal (or LMS).

The learning content (visible on the learners screen) and JavaScript API (via a hidden browser frame) are delivered to the learner's browser. An API function, (which is in the hidden frame) is called from the content frame e.g. LMSGetValue("cmi.core.lesson_status"). The hidden API frame then communicates the request to the Learning Portal (or LMS). The Learning Portal returns the value (in this case of cmi.core.leason_status) to the API Frame. The API function returns the value to content frame from which it may be passed back to the Adaptive Hypermedia Service (Figure 5).

Using these services, the deep complexity of the various metadata models (content, narrative, user) is simplified. The exported information model of the learner (and her performance) is made available via the API. The modified SCORM v1.3 interface facilitates integration with IMS and SCORM Compliant LMSs with only very minor adjustment of the information model passed between the Learning Portal (LMS) and the Adaptive Content Service. There is no change to the actual API function signatures (Owen Conlan, Wade, Gargan, Hockemeyer, & Albert, 2002).

APeLS offers teachers the ability to scope the overall course (or part of a course) they wish to use from the Adaptive Service Provider, while still allowing a considerable degree of learner adaptivity within that scope. The service-oriented approach empowers the teacher to construct courseware or educational experiences from different content service providers without the necessity of importing or extracting content. For the Content Provider, the architecture also allows the graceful growth of content and axes of adaptivity. Using the power of the adaptive engine new models of adaptivity can be created to address new markets or changing learner requirements (e.g. mobility and wireless access).

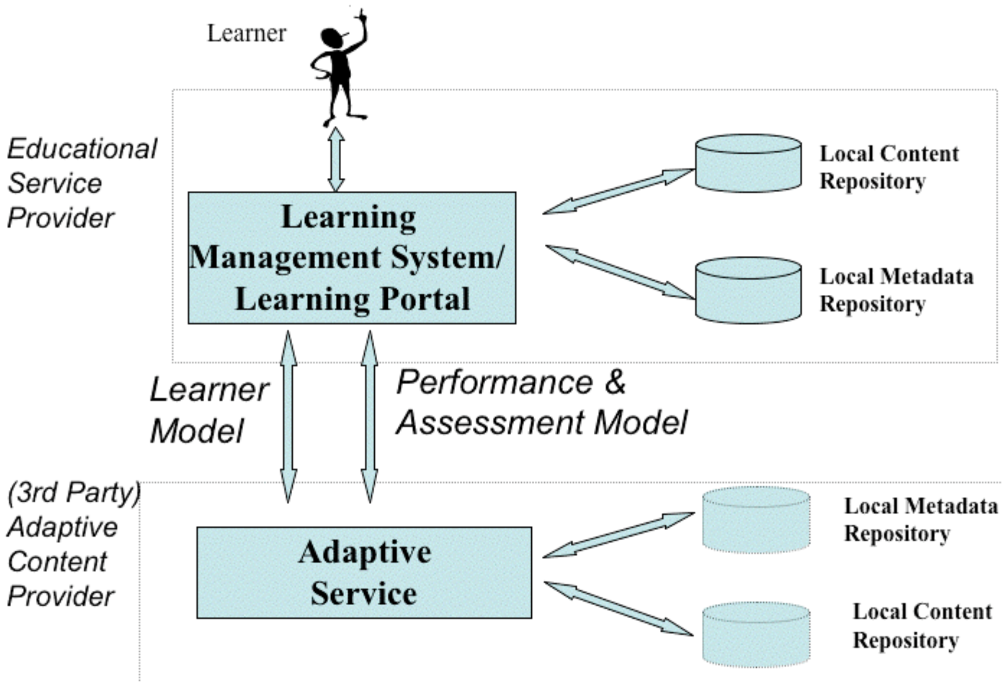

Figure 5. Learning Portal and Adaptive Service Interface

KNOWLEDGE TREE ARCHITECTURE

KnowledgeTree attempts to replace the current monolithic course management systems (CMS) such as Blackboard (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 6). A *learning portal* plays a role similar to a modern CMS. It allows a teacher to design a course and manages the student interaction with the course. The difference between KnowledgeTree and a LMS/CMS is that the learning content (activities) in KnowledgeTree resides 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 is responsible for both storing, and delivering *learning activities*. A portal has an ability to query activity servers for relevant activities and launch remote activities selected by students. An activity server is able to inform portals about available activities and provide complete support for a student working with one of its activities. The *Student model server* collects data about student performance from each portal and each activity server that work with a student. It also 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 student modeling architecture.

With the KnowledgeTree architecture, a teacher develops a course using one of the existing portals 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 supported by a student model server that collects student performance data from the portal and the activity servers. A student model server can reside on the student's own computer and support just one user. It also can reside on a university computer and support the whole class of students.

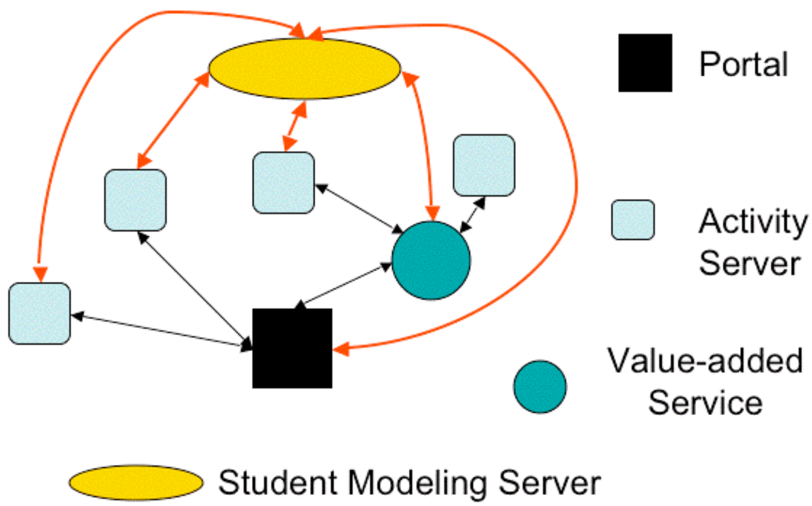

Figure 6: Main components of the KnowledgeTree distributed architecture.

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. 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 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 has a protocol for transparent login and authentication. Each adaptive activity should know the identity of the student to use the correct student model, however the student logs in only once. Second, it has 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 has 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. In addition to that, our architecture needs a resource discovery/exchange protocol. To benefit from rich distributed learning content, a portal should know about many servers and kinds of activities they can offer. However, the resource discovery issue has not been addressed in the current version of KnowledgeTree. Currently, we manually register existing activity servers with the portal.

The current version of KnowledgeTree provides very simple implementation of the first three protocols. Every activity is called directly by a dedicated URL. The transparent authentication is implemented by passing a session and a student identifier as a part of activity URL. The session identifier is required for security reasons and is issued by the student model server at the beginning of each session. Every activity server is able to extract this data from the activity URL and check the validity of the request with the student model server. We use a simple http-based communication language between components that we have developed in our past research on distributed intelligent tutoring (Brusilovsky, Ritter, & Schwarz, 1997).

The KnowledgeTree architecture allows multiple portals that can support different educational paradigms and approaches. Moreover, while the APeLS architecture suggests adaptive services as the main source of adaptivity, in KnowledgeTree adaptive functionality can reside on several components including activity servers, value-adding services, and portals. At the moment, we have implemented two versions of portal also called KnowledgeTree that is targeted to support a lecture-based educational process and is focused on dynamic and adaptive selection of learning activities. The KnowledgeTree model allows an author to develop a course as a tree of modules (Figure 7) and to specify educational material for each module. We distinguish primary material that comprises a minimal set of activities necessary for an average student to learn the module and additional material that enhances the learning experience and provides relevant activities for the students with different learning styles and levels of knowledge.

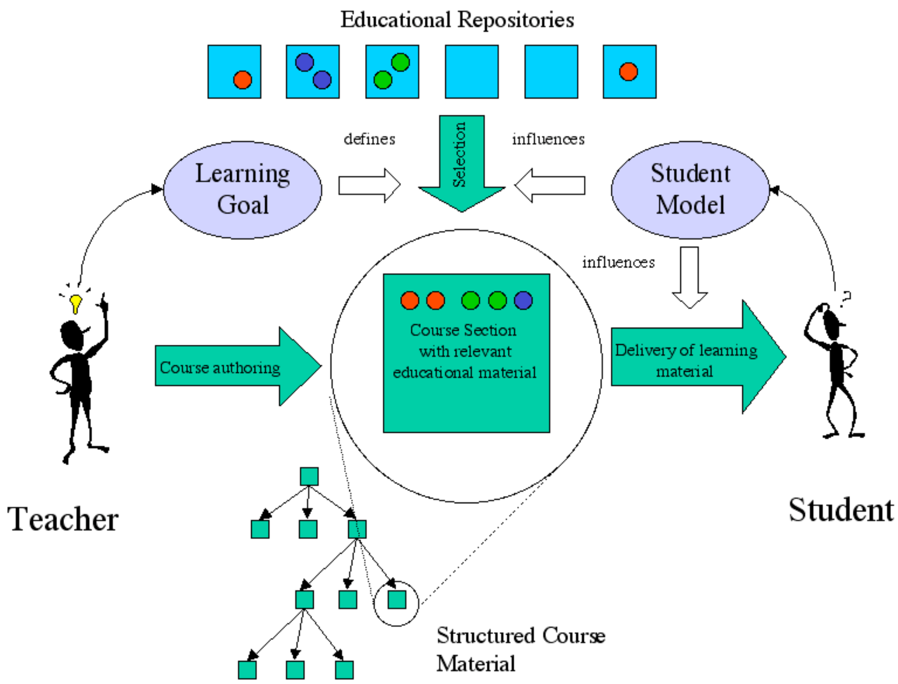

Figure 7: KnowledgeTree portal enables a teacher to structure a course as a tree of modules. Teaching material from multiple repositories can be statically or dynamically attached to any node of the course.

To select the material for each section an author specifies an educational goal for a section in terms of metadata associated with necessary learning activities. During the course design process, the educational goal is used by the system to select a subset of relevant educational activities from multiple learning repositories known to the system. From this pre-selected subset of activities an author can manually select the 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 learning goal specified by the author is retained and stored with the module. When a particular student accesses the module during the educational process, the portal uses this learning goal as well as the student model to select and present adaptively the most relevant additional material for the given student *at runtime.* Adaptive runtime selection and adaptive navigation support techniques allow the system to accommodate the volatile and expanding nature of learning repositories and student individual differences (Figure 7).

In addition to the overall architecture, a set of protocols, and the KnowledgeTree portal, the list of components developed so far includes several protocol-compliant activity servers, three value-adding services (Brusilovsky, Sosnovsky, & Yudelson, 2004), and two student modeling servers. All of the activity servers were developed for the area of teaching programming. The WebEx system (Brusilovsky, 2001) serves interactive annotated program examples, the QuizPACK (Brusilovsky & Sosnovsky, 2005) serves parameterized questions, and WADEIn II (Brusilovsky & Loboda, 2006) delivers demonstrations and exercises related with expression evaluation. The fourth server Knowledge Sea (Brusilovsky, Chavan, & Farzan, 2004) 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. Each server can work independently from the KnowledgeTree architecture, but will require a student to login in this mode of work. 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 student modeling server. Figure 8 shows the interface of our most recent KnowledgeTree server with a QuizPACK question opened.

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. Since that we have used progressively more powerful versions of the portal and all along with the four activity servers as a primary course support tool in the context of several courses. Many activities have been re-used in more than one course. Through this re-use we have appreciated how easy it was to assemble a new course from re-used interactive activities. The students have been using the system and its components on an everyday basis. All components of the system were formally evaluated and got very positive feedback from the students. The KnowledgeTree system itself was also highly praised by our students for providing a clear single-point interface to many interactive activities organized by lectures. Currently we are completing the second version of the architecture. We have also just completed the second version of the student model server CUMULATE that implements, in full, the centralized user modeling approach developed in our earlier research. (Brusilovsky, 1994).

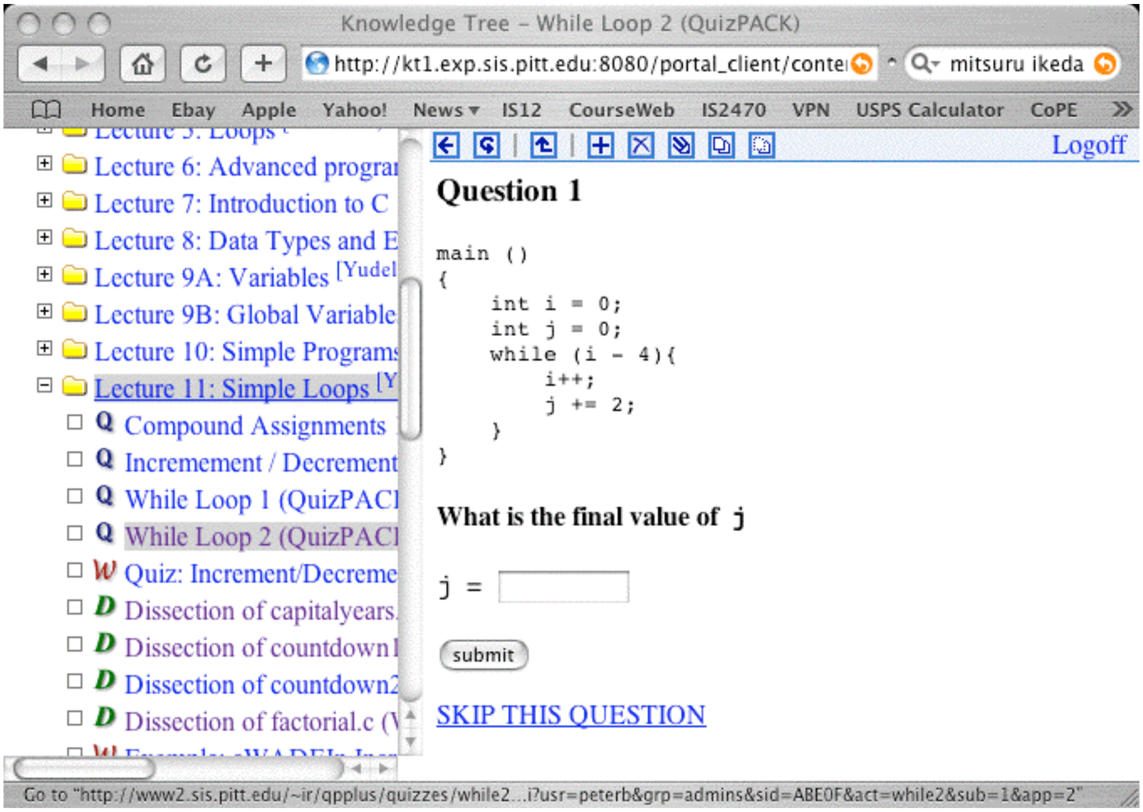

Figure 8: KnowledgeTree portal showing a range of interactive activities assembled for an introductory programming course. The left side shows the structure of the course. The right side shows an opened question delivered from QuizPACK activity server.

RELATED WORK

The components of the distributed service-based architecture that we are proposing have been investigated in a number of past research and development efforts.

The problem of searching for relevant educational activities in learning repositories is well explored by courseware re-use movement (Verhoeven, Cardinaels, Van Durm, Duval, & Olivié, 2001; Wade & Doherty, 2000). Solutions developed within this field can be directly adopted by our framework. One of the European re-use oriented projects, MTS (Graf & Schnaider, 1997) has explored the issue of runtime resolution of content requests. More recently, Learning Object Metadata groups (such as LTSC http://ltsc.ieee.org) contributed to the development of metadata standards that can be used to develop a universal resource search mechanism. Unfortunately, the existing standard does not include several metadata elements that are essential for adaptive selection and generation of learning content. This problem is discussed in more detail in (Owen Conlan, Wade, Gargan, Hockemeyer, & Albert, 2002).

Several consortia and organizations such as uPortal (http://www.uportal.org), AICC (http://www.aicc.org), and ADL (ADLI, 2003) 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. Unfortunately, the current solutions can't be used "as is" to support distributed adaptive E-Learning services. APeLS attempts to stay faithful to the SCORM and AICC approach by providing a necessary extension to the content API (O. Conlan, Hockemeyer, Lefrere, Wade, & Albert, 2001).

The problem of gathering and sharing metadata of distributed resources has been carefully investigated in the field of Web information retrieval. Some interesting centralized and decentralized architectures were suggested. 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.

A wide 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 content services. In particular, adaptive generation of educational content in response to educational objectives was explored in DCG (Vassileva & Deters, 1998) and ActiveMath (Melis et al., 2001) systems.

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 (Kay, Kummerfeld, & Lauder, 2002; Kobsa, 2001). These works can certainly contribute to the development of the user model component of KnowledgeTree framework.

DISCUSSION

This paper advocates the benefits of a service-based architecture for distributed adaptive E-Learning that integrates the most powerful features of courseware re-use frameworks and adaptive educational systems. This architecture was designed independently by two research groups in University of Pittsburgh and Trinity College Dublin in an attempt to overcome several problems of modern E-Learning architectures. Both groups combine extensive expertise in E-learning research, E-Learning standardization activities and industrial E-Learning projects. It is certainly remarkable that our groups come with very similar solutions to the problems identified in this paper. While the common parts of our approaches have been stressed in a previous section, here we would like to list a few key differences.

The main difference between KnowledgeTree and APeLS stems from their approach to using existing standards and frameworks for E-Learning. The APeLS architecture facilitates the migration to service based adaptivity by enhancing existing E-Learning frameworks and standards, yet achieving an very flexible and low complexity solution. While critiquing current metadata and component-oriented standards, the APeLS team chooses to extend the standards whenever it is necessary instead of completely rejecting them. Thus the APeLS approach provides a means by which service oriented distributed paradigm can be leveraged to provide the necessary freedom and open adaptive E-Learning service framework. In contrast, KnowledgeTree has started as a research project with a goal to develop a new architecture for E-Learning that takes into account current standards, but does not commit to use them.

One outcome of this difference is that APeLS attempts to stay very close to and evolve current LMS paradigm initiated by AICC and supported by ADL SCORM that stresses a two-component model (LMS – content). KnowledgeTree project considers this model as not appropriate for adaptive distributed content and argues for a 3-4-component model: portal – (value-adding service) – content – student model server.

Similarly, APeLS is structured to work with (i) existing courseware management systems (CMS), (ii) via a generic web portal or (iii) in a 'chained' approach where one APeLS service uses other instances of APeLS services. In (i) or (ii) APeLS does not require the CMS or the portal to be adaptive. It is envisaged that adaptive selection and structuring of content (from a particular adaptive service) is delivered via an adaptive service. The APeLS architecture is designed to be recursive if required, i.e. an adaptive service can invoke another adaptive service for a (subset) of the course it is trying to deliver. In comparing KnowledgeTree and APeLS, APeLS does not distinguish between an "adaptive portal" or an "adaptive service". In contrast, KnowledgeTree, supports the separate concepts of a portal, a value-adding service, and an activity. Thus KnowledgeTree allows portals, services, and content to be adaptive i.e. portals are used to provide adaptivity across services. In this vision, an adaptive portal provides different adaptive support such as, for example, as adaptively selecting the best of existing static or adaptive content and adaptively arranging it for the student.

Analysis of existing differences between our approaches is very helpful to understand the spectrum of opportunities in implementing the new architecture. It is important to note, however, the differences between the approaches are not critical. In fact, both approaches can easily co-exist within the same distributed architecture. It is the current challenge for our research groups to develop a practical architecture where our approaches can co-exist and benefit from each other. We are attempting to gather a community of researchers who are also interested to work on service-based architectures for adaptive E-Learning. Having such a community working together is an essential precondition to producing a solution that will be acceptable for stakeholders with different prospects. Some of the future development of APeLS is being conducted within the EU 6th Framework project, iClass, which is developing an open framework for adaptive and non adaptive E-Learning targeted at secondary level education (http://www.iclass.info). The authors invite any research or development team interested to work in this direction to get in contact with them.

REFERENCES

ADLI. (2003). Sharable Content Object Reference Model (SCORM) (Version 1.3): Advanced Distributed Learning Initiative.

Blackboard Inc. (2002). Blackboard Course Management System (Version 5.1): Blackboard Inc.

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

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction, 6*(2-3), 87-129.

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

Brusilovsky, P., Chavan, G., & Farzan, R. (2004). *Social adaptive navigation support for open corpus electronic textbooks*. In P. De Bra & W. Nejdl (Eds.), *Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004)*: *Lecture Notes in Computer Science,Vol. 3137* (pp. 24-33). Eindhoven, the Netherlands: Springer-Verlag.

Brusilovsky, P., & Loboda, T. D. (2006). *WADEIn II: A case for adaptive explanatory visualization*. In M. Goldweber & P. Salomoni (Eds.), *11th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE'2006* (pp. 48-52). Bologna, Italy: ACM Press.

Brusilovsky, P., & Peylo, C. (2003). Adaptive and intelligent Web-based educational systems. *International Journal of Artificial Intelligence in Education, 13*(2-4), 159-172.

Brusilovsky, P., Ritter, S., & Schwarz, E. (1997). Distributed intelligent tutoring on the Web. In B. du Boulay & R. Mizoguchi (Eds.), *AI-ED'97, 8th World Conference on Artificial Intelligence in Education* (pp. 482-489). Amsterdam: IOS.

Brusilovsky, P., Schwarz, E., & Weber, G. (1996a). *A tool for developing adaptive electronic textbooks on WWW*. In H. Maurer (Ed.), *WebNet'96, World Conference of the Web Society* (pp. 64-69). San Francisco, CA: AACE.

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

Brusilovsky, P., & Sosnovsky, S. (2005). Individualized Exercises for Self-Assessment of Programming Knowledge: An Evaluation of QuizPACK. *ACM Journal on Educational Resources in Computing, 5*(3), Article No. 6.

Brusilovsky, P., Sosnovsky, S., & Yudelson, M. (2004). Adaptive Hypermedia Services for E-Learning. On *Workshop on Applying Adaptive Hypermedia Techniques to Service Oriented Environments at the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004)*. Eindhoven, the Netherlands: Technische University Eindhoven.

Conlan, O., Hockemeyer, C., Lefrere, P., Wade, V., & Albert, D. (2001). *Extending educational metadata schemas to describe adaptive learning resources*. In *Twelfth ACM Conference on Hypertext and Hypermedia (Hypertext 2001)* (pp. 161-162). Aarhus, Denmark: ACM Press.

Conlan, O., Wade, V., Bruen, C., & Gargan, M. (2002). *Multi-model, metadata-driven approach to adaptive hypermedia services for personalized eLearning*. In P. De Bra, P. Brusilovsky & R. Conejo (Eds.), *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002)*: *Lecture Notes in Computer Science,Vol. 2347* (pp. 100-111). Málaga, Spain.

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

Dagger, D., Wade, V., & Conlan, O. (2004). *A Framework for Developing Adaptive Personalized eLearning*. In J. Nall & R. Robson (Eds.), *World Conference on E-Learning, E-Learn 2004* (pp. 2579-2587). Washington, DC, USA: AACE.

De Bra, P. M. E. (1996). Teaching Hypertext and Hypermedia through the Web. *Journal of Universal Computer Science, 2*(12), 797-804.

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

Graf, F., & Schnaider, M. (1997). IDEALS MTS - EIN modulares Training System für die Zukunft. On *8. Arbeitstreffen der GI-Fachgruppe 1.1.5/7.0.1 "Intelligent Lehr-/Lernsysteme*. München: Technische Universität München.

IEEE LTCS WG12 (Artist). (2002). *IEEE Standard for Learning Object Metadata*

Kay, J., Kummerfeld, B., & Lauder, P. (2002). *Personis: A server for user modeling*. In P. De Bra, P. Brusilovsky & R. Conejo (Eds.), *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002)*: *Lecture Notes in Computer Science,Vol. 2347* (pp. 203-212). Málaga, Spain.

Kobsa, A. (2001). Generic user modeling systems. *User Modeling and User Adapted Interaction, 11*(1-2), 49-63.

Melis, E., Andrès, E., Büdenbender, J., Frishauf, A., Goguadse, G., Libbrecht, P., et al. (2001). ActiveMath: A web-based learning environment. *International Journal of Artificial Intelligence in Education, 12*(4), 385-407.

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

O'Keeffe, I., Brady, A., Conlan, O., & Wade, V. (2006). Just-in-time Generation of Pedagogically Sound, Context Sensitive Personalized Learning Experiences. *International Journal on E-Learning, 5*(1), 113 - 127.

Okazaki, Y., Watanabe, K., & 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., & Tavella, M. (1990). *On the concept of database of multimedia learning material*. In *World Conference on Computers and Education* (pp. 431-436). Amsterdam: North Holland.

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

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

Vassileva, J., & Deters, R. (1998). Dynamic courseware generation on the WWW. *British Journal of Educational Technology, 29*(1), 5-14.

Verhoeven, B., Cardinaels, K., Van Durm, R., Duval, E., & Olivié, H. (2001). *Experiences with the ARIADNE pedagogical document repository*. In *ED-MEDIA'2001 - World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 1949-1954). Tampere, Finland: AACE.

Wade, V. P., & Doherty, P. (2000). *A Meta-data Driven Approach to Searching for Educational Resources in a Global Context*. In G. Davies & C. Owen (Eds.), *WebNet'2000, World Conference of the WWW and Internet* (pp. 136-145). San Antonio, TX: AACE.

WebCT. (2002). WebCT Course Management System (Version 3.8). Lynnfield, MA: WebCT, Inc.