Ontology-based Framework for User Model Interoperability in Distributed Learning Environments

Peter Brusilovsky, Sergey Sosnovsky, Michael Yudelson
School of Information Sciences, University of Pittsburgh, USA
{peterb, sas15, mvy3}@pitt.edu

Abstract: This paper presents an update of our work on developing a distributed architecture for adaptive E-Learning. We introduce ADAPT^2 – an extension of KnowledgeTree architecture that was presented at one of the past E-Learn conferences. To accommodate a larger variety of adaptive components, ADAPT^2 employs a higher-level mechanism for ontology-based interoperability of self-contained adaptive Web-based systems within our architecture. We provide an overview of ADAPT^2, argue for the importance of ontology-based interoperability, and describe our implementation of this functionality in ADAPT^2.

1. Introduction

Modern adaptive Web-based educational systems have demonstrated their value for user modeling and personalization in E-Learning context. Yet they failed to win over "one-size-fits-all" learning management systems in the E-Learning classroom. A number of researchers working on adaptive E-Learning technologies argue that the way to E-Learning classroom for these technologies goes through a distributed component-based architecture for adaptive E-Learning (Brusilovsky & Nijhawan, 2002; Conlan, Dagger & Wade, 2002; Henze, 2005; Trella, Carmona & Conejo, 2005). The problem here is to develop an architecture that will allow independent teams develop user-adaptive components that could work in parallel with the same user while exchanging collected information about the user for better adaptation. At one of the past E-Learn conferences we introduced KnowledgeTree (Brusilovsky & Nijhawan, 2002), a distributed architecture for adaptive E-Learning based on reusable intelligent learning activities. Since 2002, KnowledgeTree has been used to support several courses at the University of Pittsburgh. Over the years we have introduced several extensions of the original models such as value-added services (Brusilovsky, 2004) and ontology servers. Most recently, to accommodate a larger variety of adaptive components, we have introduced a higher-level mechanism for ontology-based interoperability of self-contained adaptive Web-based systems within our architecture. This addition caused an essential change of the architecture. We called the extended architecture ADAPT^2. Currently we are developing several adaptive Web-based educational systems with ADAPT^2. This paper focuses on the ontology-based user model interoperability of adaptive Web-based systems, the most important functionality introduced by ADAPT^2. We provide an overview of ADAPT^2, argue for the importance of ontology-based interoperability, and describe our implementation of this functionality in ADAPT^2.

2. ADAPT^2

ADAPT^2 stays for Advanced Distributed Architecture for Personalized Teaching & Training. It provides a general framework for distributed education. The main component types of this framework are:

- Learning Portal (that can be considered as a contentless Learning Management System), which organizes the learning material and provides student and teachers with the facilities necessary for participating in learning process.
- User Model (UM) Server, which stores students' activity and infers the knowledge about their learning characteristics, which are of the basis for personalization (on Learning portal or on side applications compatible with the protocols).
- Ontology Server, which stores the ontological structures of the domains where education is provided and resolves the possible conflicts in the domain models of specific applications. It also provides the platform for exchange of higher-level information about students' knowledge, calculated by different user model servers.

- Application Server (AS, also called Activity Server), which implements one or more kinds pf learning activities (i.e., interactive examples, questions) in either adaptive or non-adaptive manner
- Value-added Service (VAS), which adds some additional interactive value to the raw content provided by Application Servers – such as adaptive navigation support or an ability to annotate.

Figure 1 demonstrates the basic structure of the ADAPT². With this architecture, a teacher develops a course using one portal and many activity servers and services. The student works through the portal serving this course, but interacts with many learning activities, served directly by various activity servers. The student model server provides a basis for performance monitoring and adaptivity in this distributed context. Like its predecessor, KnowledgeTree, ADAPT² architecture is open and flexible. All components of the ADAPT² are interchangeable, i.e. there could be several portals, UM Servers, Ontology Servers and Application Servers. Moreover some systems can combine the functionality of several components (for example, LMS or AS can have its own built-in UM) or act as subcomponents (for a example some adaptivity-adding service can stay between application server providing non-adaptive content and other component). The only requirement is the common interface agreement. The open nature of it allows even small research groups or companies to be "players" in the new E-learning market. It also encourages creative competition between developers of educational systems - i.e., competition based on offering better services, not by monopolizing the market and resisting innovation. For example, an activity server that provides some specific innovative learning activities can be immediately used in multiple courses served by different portals. The architecture reflects the move from a product-based to a service-based Web economy.

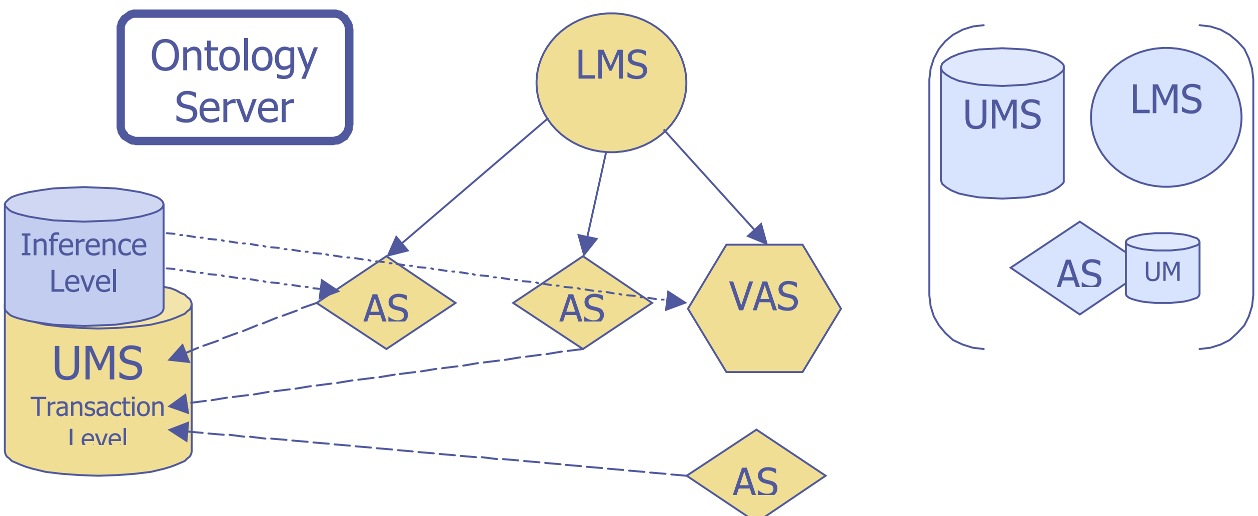

Figure 1 ADAPT² components

As we mentioned above, ADAPT² is an extension of our earlier architecture KnowledgeTree (Brusilovsky & Nijhawan, 2002). With the exception of the Ontology Server, the functionality of major components and the information exchange between them were introduced in several earlier papers (Brusilovsky, 2004; Brusilovsky, Sosnovsky & Shcherbinina, 2004; Brusilovsky, Sosnovsky & Shcherbinina, 2005; Yudelson, Brusilovsky & Sosnovsky, 2004). Due to the lack of space, this section provides additional details for Ontology Server only. We refer the reader to the key paper on Knowledge Tree (Brusilovsky, 2004) for further details. When the Ontology Server was added to our architecture, its main responsibiliy was to provide a centralized point for access to content-level metadata for all educational content indexed with a specific Domain Ontology. Domain Ontology is essentially a network of domain model concepts (topics, knowledge elements) that defines the elements and the semantic relationships between them. For at least two reasons, it is essential for educational content to be indexed with domain concepts. First, it helps course authors to locate appropriate content for their courses. Second, it is necessary for adaptive Web-based systems to select or recommend most relevant content to the individual students. Ontology-based indexing is an important part of learning content metadata (Duval & Hodgins, 2003).

In our earlier KnowledgeTree architecture we have also used ontology-based indexing. KnowledgeTree required each AS to store and provide by request ontology-level metadata for each of its interactive learning activities. In ADAPT² we followed earlier work on peer-to-peer metadata exchange such as EDUTELLA (Nejdl et al., 2002) and LOMster (Ternier, Duval & Vandepitte, 2002) and separated learning content (that still resides on AS) from content level metadata that is now stored on the Ontology Server. Our architecture allows for multiple ontologies and Ontology Servers for the same domain as well as for different domains. Each of these servers is a central “clearing house” for all information related to its ontology: the metadata for all content indexed with a specific ontology resides on its Ontology Server. The most important outcome of this decision was a provision of multiple ontologies for the same domain. Each interactive learning object can now be indexed with multiple ontologies. This arrangement allows effectively separating the content and the ontologies. As a result, any author or design team can use any interactive content with their favorite ontology. Once the required content is already indexed with the target ontology, it can be done instantly. If it is not indexed yet, the team can simply do it using the target Ontology Server and without any intrusion to the AS for which they have no control.

4. The Need and the Place for Ontology-based User Model Exchange

Our original KnowledgeTree architecture assumed that each student works with one User Model Server that collects a trace of educational events from multiple components operating with the student and processes it into a higher-level user model. This approaches worked well for a range of our non-intelligent AS that had no own user modeling functionality. However, we experienced troubles when we attempted to integrate some more intelligent components such as WADEIn (Brusilovsky & Su, 2002) and InterBook-AHA! that used their own user models and elaborated user modeling algorithms to deduce these user models from a stream of user activities. Naturally, our architecture allowed to accept a stream of low-level events from these systems, however, to process these events into the user models of matching quality we had to replicate advanced user modeling processes of each of these adaptive systems. That was not only hard, but also often impossible due to the lack of proper information on the UM Server. The only solution compatible with our architecture was to treat a user modeling component in each self-contained adaptive system in the same way as a User Modeling Server. That has immediately brought the issue of user model exchange.

As long as a student works only with one UM Server, that integrates the user's activity with different AS, there is no need for UM exchange. However, when multiple UM servers can work with the same user, the following situation is possible: a student during a course makes some progress registered in one UM and then (or at the same time) works with different UM Server in another course or even university. In this case to prevent a student and a teacher from loosing the potentially important information about the student's progress, some mechanism for exchanging information in different UM Servers about the same user is necessary. Here is another possible scenario: a teacher might want students to interact with an adaptive system that implements a very beneficial adaptation technique but uses its own UM Server (UMS). Previous activity of the students could be used by such a system to overcome the problem of "cold start", and naturally the interaction with the system itself needs to be used then by the major UM Server where the course lives.

Our new architecture extends KnowledgeTree with a framework for user model exchange. The main idea of our approach to user model exchange is the use of the Ontology Server (OS). As the center of all activities related to a particular ontology, the Ontology Server looks like the most meaningful place to store a user model based on this ontology. Hence among its other functions an ontology server in our new architecture acts as a common central storage of knowledge a student have for different concepts, inferred by different UM Servers. Unlike UM servers, an Ontology Server does not perform any user modeling. It simply stores the level of user knowledge reported by any UM server or any self-contained adaptive system for each concept of the domain ontology. Values reported by different UM Servers are stored separately. Moreover, taking into account that different systems model student knowledge on different level of cognitive activity, ontology server stores student’s knowledge of a concept separately for different cognitive levels according to the Bloom’s taxonomy (knowledge, comprehension, application, analysis, synthesis, evaluation) (Bloom, 1956).

Following our original view to the role of an Ontology Server, our extended architecture allows multiple OS – even for the same domain (because several ontologies for the same domain are possible). Each Ontology Server stores a specific ontology and all information about educational content (metadata) and students (student models) that is expressed in terms of this ontology. Potentially it allows the knowledge of the same student to be modeled by multiple systems along different ontologies and stored on different ontology servers. However, once several adaptive systems decided to collaborate in sharing and exchanging student models, they have to select one specific ontology. It immediately defines which server will be used for user model exchange – the one that hosts the selected ontology. So, As long as an adaptive system selected an ontology to model the student, it immediately knows where to send the data about the student and where to request data about the student reported by others.

4. UMS – OS – UMS Interaction

This sectioin present our current implementation of Ontology-based User Model Interoperability. Two types of OS-UMS communication are possible: PUSH-communication and PULL-communication. In the first case UM Servers "push" updated values for students knowledge to the OS when they decide to and by request they get values currently stored on the OS. In the case of PULL-based protocols UM Servers request the information from the OS with some precision. If the requested values are no up-to-date, OServer first pull current values from UMS of interest and then report them to the requesting UMS.
The nearest plan is to implement both PULL and PUSH protocol using HTTP GET method:
`http://inputURI ? user=userName & domain=domainName & concept=conceptName & clevel=cognitiveLevelName & value=X`
,where:

- userName is the name of the user of interest as it is registered in the reporting UMserver
- domainName is the name of the learning domain, where the knowledge are reported
- conceptName is the name of the concept, for which knowledge are reported
- cognitiveLevelName – is the level of learning activity after Bloom
- X – value from 0 to 1 characterizing the level of student knowledge in probability notation.

  All fields are required.

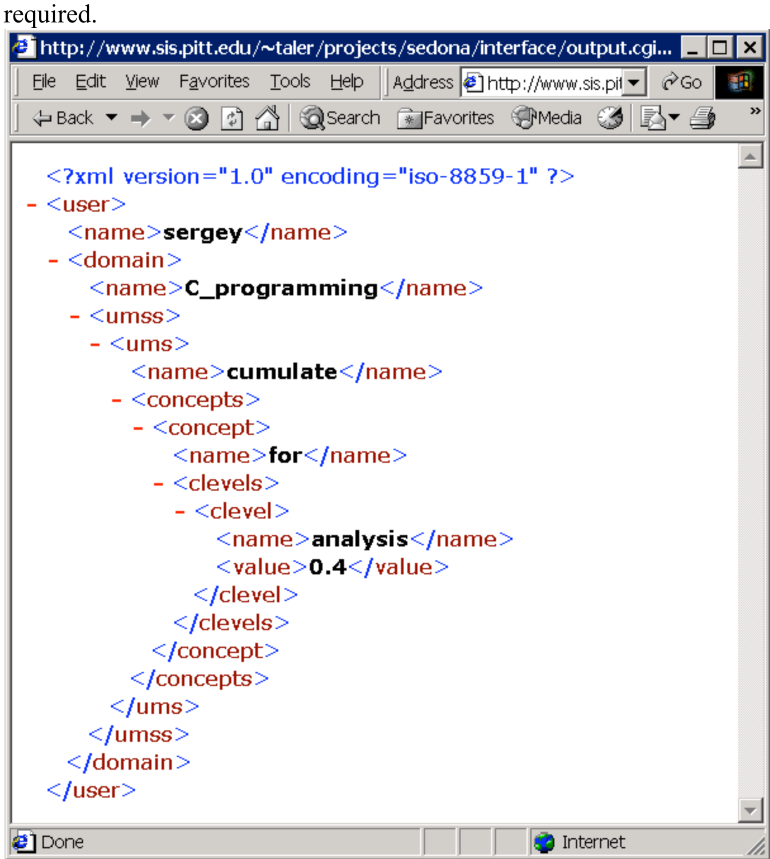

Figure 2. Typical XML-response of OS.

When the UMS needs to get values calculated by another UMS, it requests ontology server using following get http request:
`http://outputURI ? user=username & domain=domainName & ums=umsName & concept=conceptName & clevel=cognitiveLevelName & precision=T`
,where:

- umsName is the name of UMS, which expertise is needed.
- domainName is the name of the learning domain, where the knowledge are requested
- userName is the name of the user of interest as it is registered in the requesting UMserver

- conceptName is the name of the concept, for which knowledge are requested
- cognitiveLevelName – is the level of learning activity after Bloom
- T – time in seconds, defining the precision of the requested data

Fields "ums", "concept", "clevel" and "precision" are optional. If any of these optional filed (except "precision") is missing the report is provided for all possible values of this field. Hence if all are missing, the report will contain Knowledge of the specific student for every concept of the specific domain calculated by all registered UMS for all possible cognitive levels. If field "precision" is missing, OS reports currently stored values.
The report is provided in XML-form (see fig. 2).

Another protocols development direction is the implementation of XML-RPC and SOAP versions. The issues of UM Server automatic registration and user names resolving as well as resolving of ontological conflicts are developed currently.

References

Bloom, B. S. (1956) Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain. New York: David McKay Co Inc.

Brusilovsky, P. (2004) KnowledgeTree: A distributed architecture for adaptive e-learning. In: Proceedings of The Thirteenth International World Wide Web Conference, WWW 2004 (Alternate track papers and posters), New York, NY, 17-22 May, 2004, ACM Press, pp. 104-113.

Brusilovsky, P. and Nijhawan, H. (2002) A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities. In: M. Driscoll and T. C. Reeves (eds.) Proceedings of World Conference on E-Learning, E-Learn 2002, Montreal, Canada, October 15-19, 2002, AACE, pp. 154-161.

Brusilovsky, P., Sosnovsky, S., and Shcherbinina, O. (2004) QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support. In: J. Nall and R. Robson (eds.) Proceedings of World Conference on E-Learning, E-Learn 2004, Washington, DC, USA, November 1-5, 2004, AACE, pp. 1806-1813, also available at http://www2.sis.pitt.edu/~peterb/papers/ELearnQP04.pdf.

Brusilovsky, P., Sosnovsky, S., and Shcherbinina, O. (2005) User Modeling in a Distributed E-Learning Architecture. In: L. Ardissono, P. Brna and A. Mitrovic (eds.) *User Modeling 2005*. Lecture Notes in Artificial Intelligence, (Proceedings of 10th International User Modeling Conference, Edinburgh, UK, July 24-29, 2005) Berlin: Springer Verlag, pp. 387-391.

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.

Conlan, O., Dagger, D., and Wade, V. (2002) Towards a standards-based approach to e-Learning personalization using reusable learning objects. In: M. Driscoll and T. C. Reeves (eds.) Proceedings of World Conference on E-Learning, E-Learn 2002, Montreal, Canada, October 15-19, 2002, AACE, pp. 210-217.

Duval, E. and Hodgins, W. (2003) A LOM research agenda. In: Proceedings of The Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, 20-24 May, 2003, pp. 88-98.

Henze, N. (2005) Personalization Services for e-Learning in the Semantic Web. In: Proceedings of Workshop on Adaptive Systems for Web-based Education at 12th International Conference on Artificial Intelligence in Education, AIED'2005, Amsterdam, July 18, 2005, IOS Press, pp. 55-58.

Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., and Risch, T. (2002) EDUTELLA: A P2P Networking Infrastructure Based on RDF. In: Proceedings of 11th International World Wide Web Conference, Honolulu, Hawaii, USA, 7-11 May 2002, also available at http://www2002.org/CDROM/refereed/597/index.html.

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.

Trella, M., Carmona, C., and Conejo, R. (2005) MEDEA: an Open Service-Based Learning Platform for Developing Intelligent Educational Systems for the Web. In: Proceedings of Workshop on Adaptive Systems for Web-based Education at 12th International Conference on Artificial Intelligence in Education, AIED'2005, Amsterdam, July 18, 2005, IOS Press, pp. 27-34.

Yudelson, M., Brusilovsky, P., and Sosnovsky, S. (2004) Accessing interactive examples with adaptive navigation support. In: Proceedings of IEEE International Conference on Advanced Learning Technologies, ICALT 2004, Joensuu, Finland, August 30 - September 1, 2004, AACE, pp. 842-843.