Integrating Interactive Learning Content into an Adaptive E-Learning System: Lessons Learned

[1]Maram Meccawy, [2]Peter Brusilovsky, [1]Helen Ashman, [2]Michael Yudelson, [2]Olena Scherbinina
[1]*School of Computer Science and Information Technology, University of Nottingham, UK*
{mzm,hla}@cs.nott.ac.uk
[2]*School of Information Sciences, University of Pittsburgh, USA*
{peterb, mvy3, ols1}@pitt.edu

Abstract: One of the interesting problems that have attracted the attention of both industry and academia in current e-learning environments is the reusability of learning content in Learning Management Systems (LMS). A number of standards have defined a Run-Time environment in order to allow an LMS to deal with learning objects i.e. to launch, track and communicate with it. Such standards achieved high performance when dealing with simple learning objects such as a text file, picture or a simple animation but when dealing with much complicated content such as rich interactive or adaptive content they show many limitations This paper takes a closer look at the issue of reusability and interoperability of interactive learning content and presents a case study that followed one of the non-standardised research efforts where an existing LMS integrates an independently developed interactive learning simulation activity. The lessons that have been learned from this work has also been stated here.

1. INTRODUCTION

Adaptive E-Learning systems provide an alternative to "one-size-fits-all" courseware served by some modern learning management systems (LMS). These adaptive systems maintain 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. A range of studies with various adaptive E-Learning systems demonstrated many benefits that can be provided by systems that are able to adapt to individual user (Brusilovsky and Peylo, 2003). Yet, in one aspect – reusability of external learning content - adaptive systems still lag behind LMS. While any LMS allows a teacher to easily add a piece external learning content, the majority of adaptive E-Learning systems are only able to work with so-called "closed corpus" content provided by the system developers.

This problem was addressed by a new generation of *open corpus* E-Learning systems such as KBS-Hyperbook (Henze and Nejdl, 2001) and SIGUE (Carmona et al., 2002), which allowed adding open corpus educational content to an adaptive E-Learning system. These early attempts demonstrated that adaptive systems can use open corpus content, however they only explored the use of simple content, such as Web pages or simple questions. This is not the kind of content that provides more solid pedagogical impact. As was shown by multiple studies, students benefit most from rich interactive content that allows them to learn by doing and by exploration. For example, educational simulations (frequently augmented with advanced visualization) engage the student in exploration of complicated concepts and help to uncover the dynamics of important processes.

The integration of simple open corpus content into an adaptive E-Learning system is rather straightforward. A piece of presentation content stored in the LMS is invoked by a link and the system immediately records that it was presented. A question is invoked, interacts with the user, and report the answers (success or failure), which the system records. The recorded information is then used to update the student model. The use of rich interactive content is a more complicated procedure that has to include a number of other steps and information exchanges. To start with, the intelligent content needs to have the user identified and authenticated. This is necessary to attribute the result of the user work to this user and to adapt to him/her. Before the very start of the work, intelligent content may need to get access to the history of the user's work and to the user model. The former is required to provide continuity in complex multi-step activities, the latter to adapt to the user (i.e., take into account starting level of user knowledge). The work with rich content can include many steps and each of these steps can bear important information that can be used for deducing the change of user knowledge, goals, or other parameters. To take this information into account for student modeling, it has to be passed to the adaptive system that invoked this content. It can be done in the form of a trace, a student model, or both. With the first option, while working with the user, the rich content passes the trace of student actions to the adaptive system, which uses it for student modeling With the second option, rich content builds its own model of user knowledge and interests. At the end of the work, the student model accumulated by the intelligent content is passed to the host system to be integrated with the main student model. Thus, just one session of work with intelligent content requires a range of information exchanges where different information is communicated between different components.

Therefore, the use of rich content in adaptive open corpus E-Learning system is a serious challenge that has to be addressed on the architectural level. Recently, several research teams suggested architectures for distributed adaptive E-Learning. The list of these systems includes ADAPT² (Brusilovsky, Sosnovsky, and Yudelson, 2005), APeLS (Conlan et al, 2002), MEDEA (Trella, Carmona, and Conejo, 2005) and T-MAESTRO (Rey-López et al, 2006). By their design, all these systems can potentially integrate rich open corpus content. However, so far, these systems experimented mostly with integration of simple interactive content such as questions. Moreover, typically this content was produced by the same team as the distributed system itself. The use of rich external content was not really explored in a practical example.

This paper attempts to fill this gap. It presents a case study, which we performed to explore the problems of integrating rich interactive content in a modern distributed E-Learning system. In this case study we used a simple interactive simulation, which was used in real classes and ADAPT² framework developed by two of the authors. Our case study demonstrates how this integration can be achieved and which typical problems can be encountered on the way. This exercise helped us to learn a few lessons that we are taking into account to develop a new version of ADAPT² framework.

This paper has been structured as follows: the next section presents the context for our study: the simulation system that we attempted to integrate and the ADAPT² framework. Section 3 presents the case study in detail and finally section 4 concludes the research work and the lessons that have been learned. It is the authors' hope that this paper will help in establishing a better framework that supports a broad use of rich interactive content in modern e-learning systems.

2. INTERACTIVE CONTENT & ADAPTIVE LMS

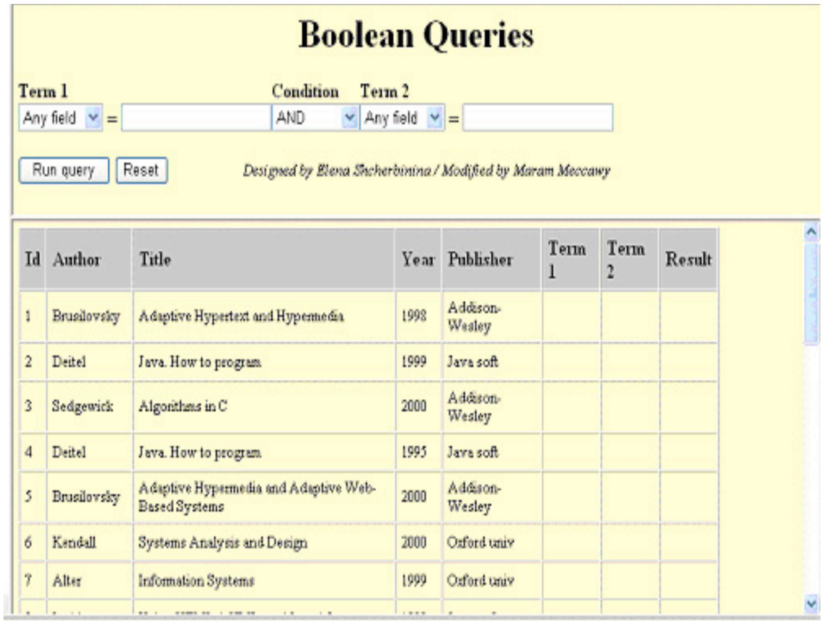
2.1 BOOLEAN QUERY

The Boolean Queriy (BQ) is an example of rich interactive content that support learning-by doing and learning-by-exploration. The goal of this Web-based interactive application is to provide a student with a solid understanding of the logical operators (AND, OR and AND_NOT). It uses a theme of a database of books where a student can search for a specific book using different parameters such as its ID, title, author's name, year of publication or publisher. What he/she will get as a result is a visual representation marking the conditions that were matched or mismatched in a distinguished manner, i.e. using different colours. This activity can be used as a part of an existing computer science or computer engineering course such as: Introduction to Database, Digital Logics or Computer Organization.

From a technical perspective, BQ (Figure 1) was developed using Java Servlets technologies. In addition, the system's database (books'data) is an *XML* file, while the interface is an HTML page which is divided using page frames into two sections: first (upper) section for the query string (user's input) and the second (lower) frame will present the query's results.

The significance of BQ is that a user can learn and explore Boolean operations by changing parameters and trying different options. For example he/she could search for a specific author *AND* title, and view the results, then change the operator to *OR* or *AND_NOT* and check the results which might be different than what has been presented previously. Hence, one session of work with the tool produces a large flow of events that can be used to infer the change of user knowledge. Therefore, recording the user's activity and saving his/her trace is vital for updating the user's profile, which our implementation has aimed to achieve. Teachers could monitor the user's interactivity with the system for every operator and hence know his/her level of knowledge of each one of those operators.

Modification to the original code was essential to be able to "plug" it into the adaptive LMS. The integration process is described in section (3.2).



Boolean Queries

Term 1: Any field =

Condition: AND

Term 2: Any field =

Designed by Elena Shcherbinina / Modified by Maram Meccawy

Id	Author	Title	Year	Publisher	Term 1	Term 2	Result
1	Brusilovsky	Adaptive Hypertext and Hypemedia	1998	Addison-Wesley			
2	Deitel	Java. How to program	1999	Java soft			
3	Sedgewick	Algorithms in C	2000	Addison-Wesley			
4	Deitel	Java. How to program	1995	Java soft			
5	Brusilovsky	Adaptive Hypermedia and Adaptive Web-Based Systems	2000	Addison-Wesley			
6	Kendall	Systems Analysis and Design	2000	Oxford univ			
7	Alter	Information Systems	1999	Oxford univ			

Figure 1: The interface of Boolean Query application

2.2 ADAPT²

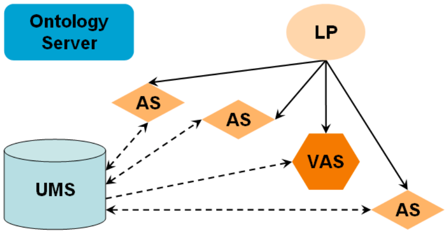


Figure 2: ADAPT^2 distributed architecture for adaptive E-learning

ADAPT^2 (Advanced Distributed Architecture for Personalized Teaching and Training) is an extension of the Knowledge Tree architecture (Brusilovsky and Nijhawan, 2002, Brusilovsky, 2004), a distributed architecture for adaptive E-learning based on reusable intelligent activities. KnowledgeTree has been used since 2002 to support the learning of several courses at the University of Pittsburgh. ADAPT^2 has been created to accommodate a larger variety of adaptive components by applying a higher level mechanism for ontology-based interoperability between self contained adaptive Web-based systems.

Five main components comprise the ADAPT^2 framework. The (1) **Learning Portal** (LP) is similar to modern LMS but the content is not embedded in it. It organizes the learning material and provides student and teachers with the facilities or features necessary for participating in the learning process. The (2) **User Model Server** (UMS) stores information about each user and the user activities to be used to infer the user's level of knowledge and learning characteristics. While the (3) **Ontology Server** stores the ontological structures of the domain models. A Domain Ontology is a network of domain model concepts (topics, knowledge elements) which defines elements and the semantic relationships between them. Its job is to resolve the possible conflicts in the domain models of specific applications. In addition, it provides the platform for exchanging high-level information about students' knowledge, calculated by different user model servers (from different applications which might have their local user modelling servers). The (4) **Application Server or Activity Server** (AS) implements one or more kind of learning activities which might be adaptive or non-adaptive. It typically implements a piece of rich interactive or intelligent content that will engage the user on different series (steps) of activities over a period of time. And finally, the (5) **Value-**

added Service (VAS) that adds some additional interactive value to the raw content provided by Application Servers, for example adaptive navigation support.

A teacher uses the Learning Portal to develop a course using one portal and many activity servers and services. While the student uses this portal to serve through this course and learn, he/she will be interacting with many learning activities served by different activity servers. In this distributed architecture, the UMS will provide the basis for performance monitoring and adaptivity.

ADAPT^2 has been designed in a way that all its components are interchangeable, where the only requirement for this component replacement is the common communication protocol agreement.

3. CASE STUDY

Boolean Query aplication presented in section 2.1 provided an excellent test case to explore the problem of integrating external interactive learning content into an adaptive distributed e-Learning system such as ADAPT^2. This section presents the work that we have done to achieve the integration. Prior to the integration process, some questions were raised and needed to be addressed and agreed upon between both LMS and the activity providers (see section 3.1).

3.1 IMPLEMENTATION

When an attempt is made to launch and integrate an interactive learning activity (such as BQ) with a foreign, independently developed adaptive LMS, some issues needed to be agreed upon in advance. Answers for the following questions needed to be provided.

1. How is the user being authenticated?
2. How is the information about the user passed to intelligent content?
3. How can the trace of the user work be stored?
4. How can the end of the work with the content be registered?
5. How can the student model accumulated by intelligent content be shared with other components?

In ADAPT^2, the portal provides a centralized single login point for enrolled students to be authenticated and permitted to use all the learning tools and content fragments that are provided in the context of their courses. The portal uses an "original" authentication protocol to call the Activity Server (AS), which invoked by a unique URL. During this process, the user parameters (name, group, session ID) are passed to the intelligent server. The AS, in turn, will use another protocol to send the events to the centralized UMS, in which it appends the parameters that were received from the portal in addition to other parameters to the UMS URL to register this user and store the data collected about the student's performance during the learning process. Different components are able to access this centralized UMS to send information to or to extract user's data from, using specified protocols.

When integrating the (Boolean Query) learning with ADAPT^2, It was decided to use the ADAPT^2 protocols. Hence the adaptive LMS (ADAPT^2 portals) had to provide three compulsory parameters when launching the learning activity by calling its URL. The *first step***,** in modifying the BQ, was defining variables that would allow it to store the parameters (name, group, session ID) that are sent by ADAPT^2's portal as soon as the BQ URL is fired by the user's clicking action.

The *next step* was to send those aforementioned parameters in addition to other parameters such as (activity ID) to the UMS to register the user and the activity that he/she has been working with. This is done by invoking the UMS's URL after appending those parameters to it. Those two steps (in addition to recording the user's trace as mentioned in section 3.4) needed to be preformed by the activity provider, while the *the user modeling implementation* of the BQ at the UMS side was the responsibility of the LMS provider (ADAPT^2 developers group).

Later, the system can request the student's profile from the UMS. Figure 3 below, demonstrates the implementation process.

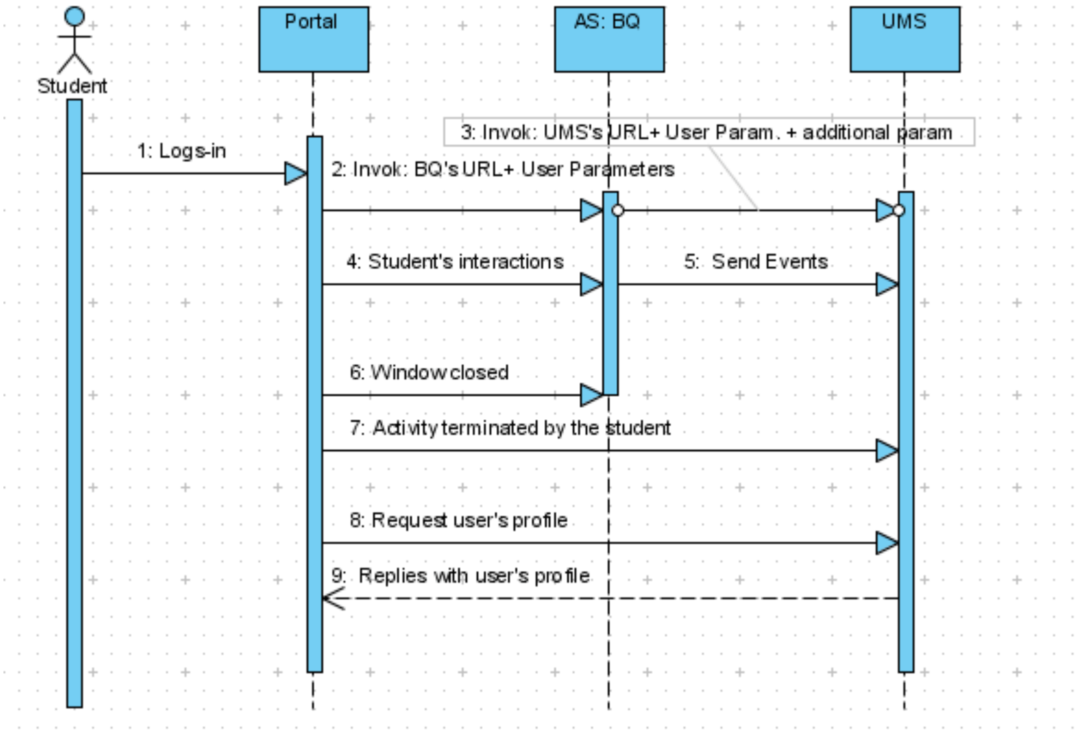

Student
Portal
AS: BQ
UMS
1: Logs-in
2: Invok: BQ's URL+ User Parameters
3: Invok: UMS's URL+ User Param. + additional param
4: Student's interactions
5: Send Events
6: Window closed
7: Activity terminated by the student
8: Request user's profile
9: Replies with user's profile

Figure 3: Implementation process of ADAPT² and the learning activity (Boolean Queries).

3.2 USER TRACE

Being able to access the user trace has many benefits for all the stakeholders in the E-Learning process. The original system that produced this data needs access to it to provide continuity and proper adaptation. The LMS needs it to recognize the completion of educational objectives and to organize sequencing of learning content. Other types of intelligent content (used by the same student) need to use an updated state of the student goals and knowledge observed by an activity in order to provide adequate adaptation. The teacher may want to monitor the student's work with an activity to recognize problems and to assist him/her. The students may want to observe the growth of their knowledge over the use of various learning activities.

Overall, the ability to access and use this information is one of the keys to the success of rich interactive, adaptive or intelligent content in modern LMS. Therefore, the research work provided a teacher or a course administrator with a special *teacher view,* provided only for teachers when they log into the system for reviewing and replaying the user's activities. This will give the teacher a picture of *what* has been done (what books, author, etc has been searched for in our case), *when* did those actions take place (exact time) and *who* performed them (i.e. student's name). This feature can be important and extremely beneficial in cases where students are taking online exams to address students' complaints about their score and performance, since it could be used as a fair proof of their actions.

4. CONCLUSION & LESSON LEARNED

The research work has shown us practically that the interoperability of rich, intelligent or interactive content is still a problem that needs to be resolved. This is done in order to allow different LMS to launch this type of learning content which from pedagogical point of view could have a significant learning impact than the simple content with its "page-turning" style. It is this type of learning content or activities that is most likely to push e-Learning many steps forward in the near future. And while the former *simple* content reusability and interoperability issues has been addressed adequately through standards such as SCORM (`http://www.adlnet.gov/scorm/index.cfm`), the latter rich type of learning content has yet a long way to go in order to achieve a similar stable and satisfactory level.

In addition, the use of “original” protocols such as ADAPT^2 protocols is not a practical or feasible solution for real world settings since a lot of “‘human” communications were involved throughout the integration process between both sides, i.e. between content developer/provider and any LMS provider that wishes to use it. Therefore, this integration should be made as autonomous as possible by using other technologies which provide this capability such as the emerging Web Services technologies which might be the solution for those problems. Some research groups have already started taken this a step forward such in MEDEA (Trella, Carmona, and Conejo, 2005). Other technological options are XML RPC and use of RDF (Henze, 2006).

Finally, it is important for all partners involved in the E-learning process to have the ability to view the student's work trace with a rich interactive learning activity that has been recoded during his interaction with the system. Since it could be used to deduce his/her level of knowledge and update the user’s profile.

ACKNOWLEDGEMENTS

The research work presented in this paper was sponsored by the KESP project and National Science Foundation of Ireland. Many thanks to Marta Rey-López and Scott Bateman, for many useful discussions and for their support during the KESP research project.

REFERENCES

Brusilovsky, P., E. Schwarz, and G. Weber, ELMART: An intelligent tutoring system on World Wide Web, in Third International Conference on Intelligent Tutoring Systems, ITS-96, C. Frasson, G. Gauthier, and A. Lesgold, Editors. 1996, Springer Verlag: Berlin. p. 261-269.

Brusilovsky, P. and H. Nijhavan, A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities, in World Conference on E-Learning (E-Learn 2002). 2002: Montreal, Canada. p. 154-161.

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

Brusilovsky, P. KnowledgeTree: A distributed architecture for adaptive e-learning. in The Thirteenth International World Wide Web Conference, WWW 2004 (Alternate track papers and posters). 2004. New York, NY: ACM Press

Brusilovsky, P., S. Sosnovsky, and M. Yudelson, Ontology-based Framework for User Model Interoperability in Distributed Learning Environments. In *World Conference on E-Learning (E-Learn 2005). 2005 Vancouver, Canada. p. 2851-2855.*

Butler, J.E. and J.B. Brockman, *A Web-based learning tool that simulates a simple computer architecture.* SIGCSE Bulletin - inroads, 2001. 33(2): p. 47-50.

Carmona, C., Bueno, D., Guzmán, E., and Conejo, R. (2002) SIGUE: Making Web Courses Adaptive. In: P. De Bra, P. Brusilovsky and R. Conejo (eds.) Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002), Málaga, Spain, May 29-31, 2002, Springer-Verlag, pp. 376-379.

Conlan, O., Wade, V., Bruen, C., and Gargan, M. (2002). Multi-model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning. In: *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems.* London, UK, 2002,Springer-Verlag, p.100-111.

Henze, N. and Nejdl, W. (2001) Adaptation in open corpus hypermedia. International Journal of Artificial Intelligence in Education 12 (4), 325-350.

Henze, N. and Krause, D.(2006). Personalized Access to Web Services in the Semantic Web. SWUI 2006 - *3rd International Semantic Web User Interaction Workshop*, November 6, 2006, Athens, Georgia, USA, collocated with ISWC 2006.

McKenna, A. and A. Agogino. Engineering for middle schools: A Web-based module for learning and designing with simple machines. in FIE'97, Frontiers in Education Conference. 1997. Pittsburgh, PA: Stipes Publishing L.L.C.

Rey-López, M., A. Fernández-Vilas, and R.P. Díaz-Redondo, *A Model for Personalized Learning Through IDTV*, in *Adaptive Hypermedia, Adaptive Web-Based Systems 2006 (AH2006)*, Springer-Verlag, Editor. 2006: Dublin, Ireland. p. 457-461.

Trella, M., C. Carmona, and R. Conejo, MEDEA: an Open Service-Based Learning Platform for Developing Intelligent Educational Systems for the Web, in Workshop on Adaptive Systems for Web-Based Education: tools and reusability (AIED'05). 2005: Amsterdam, The Netherlands. p. 27-34.

VanLehn, K., et al., *The Andes Physics Tutoring System: Lessons Learned.* International Journal of Artificial Intelligence and Education, 2005. **15**(3): p. 147-204.