A Generic Solution Approach for Integrating Adaptivity into Web-based E-Learning Platforms

Alexander Nussbaumer[1], Christian Gütl[2], Cord Hockemeyer[1]

[1] Department of Psychology, University of Graz, Austria
[2] Institute for Information Systems and New Media, Graz University of Technology, Austria

Key words: *Adaptive E-Learning, E-Learning Platform, Knowledge Space Theory, Web Service, Assessment, Personalised Learning Path, AdeLE, iClass*

Abstract:

Adaptation is a well known concept in the field of e-learning and is increasingly applied in modern learning systems. In order to gain more flexibility and to enhance existing e-learning platforms, we have developed a generic solution approach which enables to add adaptation functionality to existing Web-based e-learning systems. In this paper we present a distributed architecture on the basis of Web services, which is based on experiences gained in the AdeLE and iClass research projects. This flexible solution allows for adapting towards various devices, such as PCs and handhelds. The first running prototype implements the formal Competence-based Knowledge Space Theory for effective assessment of pre-knowledge and knowledge acquisition as well as for determining the sequence of learning content.

1 Introduction

The concept of adaptation has been increasingly addressed in the literature [1], [4] over the last two decades. In an early paper about adaptive hypermedia Brusilovsky [1] gives an overview about methods and possible application fields. One of the most important and interesting one is the educational application domain. Since there are great differences with respect to the learners' knowledge states and preferred learning styles, adaptive learning systems can support the learning process by adapting to such parameters and characteristics. This might be one of the most important system behaviours of an adaptive e-learning system. There are several methods published in literature and implemented in existing systems which can be applied to adapt the content towards the users' needs, known as "personalisation".

Educational adaptive systems usually work with user models, which contain personal data about the learner, and a domain model, which provides meta-information about the content. Furthermore, adaptation procedures fulfil the task of adapting the presented content to the user. Traditionally, these components are integrated monolithically in a learning system together with all other features. This paper provides a detailed description about implementing adaptation concepts in learning systems.

Two problems arise from a seamless integration into one system. First, an absence of encapsulation makes it hard to change the models and procedures without changing the other components. Second, applying adaptation functionality of one system to another system needs much re-implementation because of the lack of reusability. To overcome these problems a system architecture is needed, which separates models and procedures in a way that specific adaptation functionality is independent from the rest of the system. Therefore, a more modular system design is needed.

This paper presents an approach which postulates a system design with an encapsulated adaptation system in an own web service. This service can be connected to arbitrary e-learning platforms through a generic adaptation control interface. This solution has several advantages. Existing e-learning platforms can be enriched with adaptation functionality without changing its implementation. The separation of the adaptation functionality creates flexibility and reusability; different learning platforms can use the same adaptation methods. The web service approach enables the creation of the adaptation functionality as a business model, which can be offered to learning platform operators.

Since the described solution approach refers to the server-side part of a learning system, it can also be applied to mobile learning frameworks. Reusable adaptation functionality is also a relevant feature for distant mobile learning.

2 State of the art

There are already systems which encapsulate adaptation by using a distributed and service-oriented system architecture. This section discusses the two research projects AdeLE [5] and iClass [7], whereby in both projects one of the authors is involved. They build the basis for a more generalised design of adaptation control, which will be described in the next section.

The AdeLE software design builds on a service-based component model with a particular component for adaptation functionality - the adaptive system - and a LMS. To connect the adaptive system with the LMS, an adaptation control interface has been developed, which is based on a product factory design pattern of adaptors for sequencing, aggregation, and representing. In AdeLE a generalisation of the adaptation control interface could be realised by an abstract design pattern, which must be implemented and applied on concrete systems. However, this approach has its limitations, because the adaptors have to be implemented for and inside each new system.

The iClass design [10] builds on a service-oriented architecture (SOA), where the various personalisation functionalities are realised as Web services. For example, Monitor and Profiler services model the learner information and Selector and LO Generator services are responsible for delivering personalised learning path and content. Adaptation is achieved though a coordinated system behaviour of the several services. The communication between these services is realised by a tailored interface design.

There are further systems which have chosen a more or less modular system design in order to encapsulate adaptation functionality. For example, APeLS [2][3][9] is a modular system which builds upon a multi-modal approach. It comprises three models, the learner model, the content model, and the narrative model.

3 Flexible Approach

The combination of both generalisation approaches leads to a flexible approach, which is the central part of this paper. The main idea of this approach is to encapsulate the adaptation functionality in a separate Web service. Then it can be connected to an arbitrary LMS with little or no adaptation ability in order to enrich it with adaptation functionality. The user connects with the browser to the LMS and needs no extra information about the Web service.

3.1 Adaptation and Personalisation

Before specifying the system architecture and interface, it must be specified which tasks and requirements should be accomplished with this approach. The main goal is to form a framework for adapting the knowledge and competence state and the preferences of a learner to the course and assessment sequence. Since the system behaviour adapts to a person, this kind of adaptation is also called personalisation.

Input of the adaptation process is the knowledge and competence state of a learner as well as the learner's preferences and profile. Knowledge and competence state must be determined by the system using assessment or monitoring the learner's behaviour. Usually adaptation models include procedures therefore. Preferences and profiles of learners must be manually input.

Output of the adaptation is a sequence of content objects personalised to a learner. It depends on the particular adaptation model, which adaptation input data are processed and how they are used for the personalisation. In any case the content is sequenced and structured by the adaptation system, for example the order of learning or assessment objects is created dynamically.

3.2 System Architecture

In order to accomplish this flexible approach, we propose a distributed system architecture which consists of three components (see Figure 1): A browser, a learning management system (LMS), and an adaptation Web service (AWS). The browser - Web browser on a computer or a WAP browser on a mobile device - is the user interface which displays content and interacts with the user. It is connected over the internet (cable or wireless) to the learning management system.

The learning management system features the typical e-learning functionality, such as storing learning content and user data, as well as providing administration and authoring tools. There is a variety of different learning management systems, also depending on the browser type (desktop computer or mobile device). Though they can be integrated in the system and connected to the AWS in the same way, in order to be enriched with adaptation functionality. Therefore an extension of the LMS is necessary which acts as a bridge between the LMS and AWS. This extension has a defined interface to the AWS, but it must be tailored to the individual LMS.

The Web service is the component which has implemented the adaptation model. In order to accomplish adaptation this service needs to get information from the LMS and to control the LMS. Since the connection is established to the LMS extension different types of LMS can be connected to the Web service. Furthermore multiple LMS can be connected at once. The communication between LMS and AWS is done over SOAP (originally Simple Object Access Protocol) which provides a basic layer for exchanging messages over computer networks. In this way browser, LMS, and AWS can be implemented in different programming languages and running on different hosts with different operating systems.

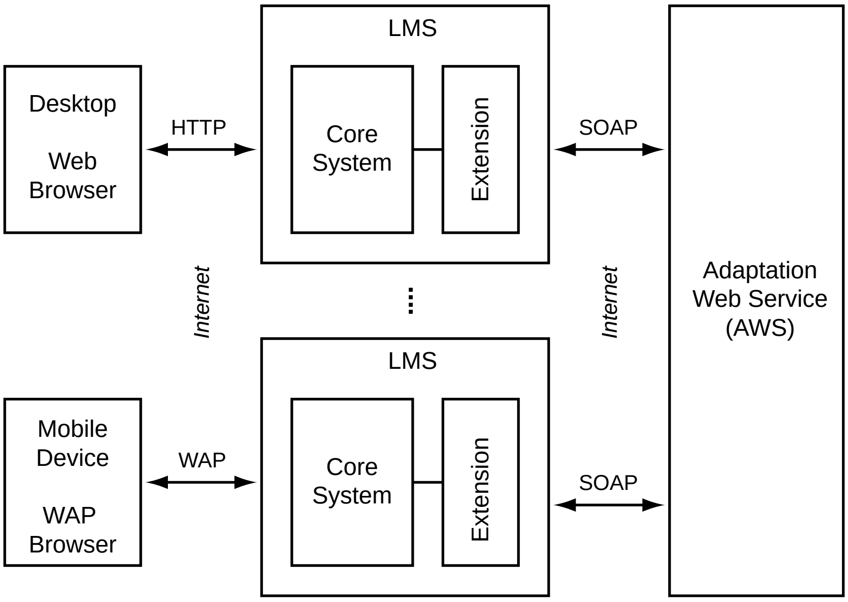

Figure 1: System architecture. The diagram shows the main components of the system architecture, which includes browser, the learning management system (LMS), and the adaptation Web service (AWS).

Before describing the interface between LMS and AWS, it is necessary to outline which modules and models are comprised by which of the two components (see Figure 2). Some of the modules and models needed for an adaptive e-learning system are described in the last section. In this approach the responsibility of the LMS is to deal with content and user data with respect to authoring, editing, and displaying. They are accessible to learners and administrators (content creator and teacher). This is also the typical functionality of existing learning management systems. Therefore the LMS needs to have repositories to store learning objects (for example HTML pages or SCORM packages), learning information (for example IMS LIP or PAPI), and content meta-data (for example LOM).

The adaptation system is responsible for the adaptation process. Its implementation depends on the particular adaptation model, however it typically consists of an adaptation engine and data models describing content, learners, and narratives. The data models refer to data in the LMS and contain additional information, such as relations between learning objects, relations between learning objects, learners, and competences, or knowledge and competence states of learners. The adaptation engine is supposed to process these data in order to generate personalised learning paths. Though this is a typical principle adaptation system, this approach is also open for different adaptation models.

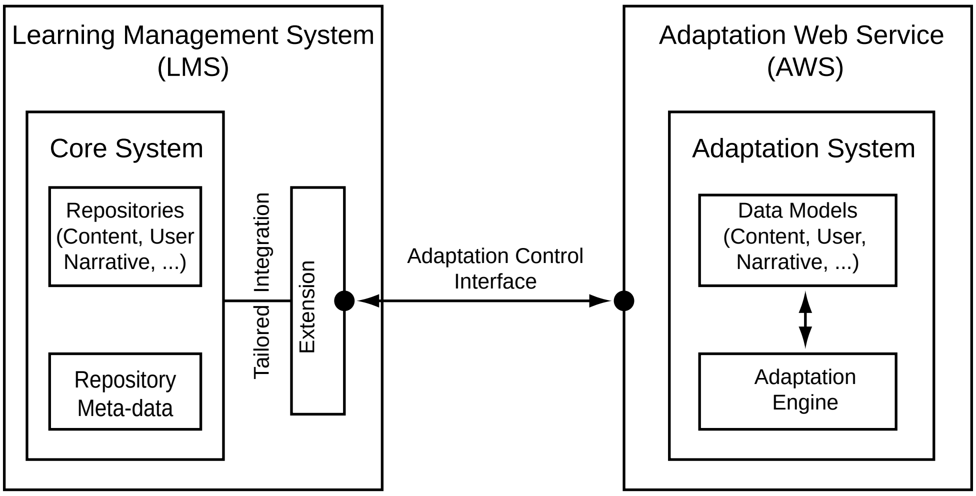

Figure 2: This diagram outlines the models and modules, as well as the interface between the learning management system (LMS) and the adaptation Web service (AWS).

3.3 Adaptation Control Interface

A main concern of this approach is the interface design between the LMS and AWS. Since in this flexible design these two components are envisaged to be arbitrary, the design of this interface is important, because it is the persistent part. However, at the technical level it cannot be designed fully precisely, in order to keep it open for a wide range of learning management and adaptation systems.

There are several requirements for the interface design to ensure that adaptation can be applied on the LMS. Exchanging information about users and content between the components must be guaranteed, as well as initiating system behaviour by sending control commands. In detail the interface design must comprise the following parts:

1. *Transferring learning and content information to AWS:*
   In particular existing information about registered users (learners) and about content (learning objects) must be transferred to the adaptation system, because content should be adapted to the learner. There are e-learning standards for these data which are suggested to be used. For example, the Learner Information Package (LIP) and the Public and Private Information for Learners (PAPI Learner) standard can be used to describe the learner and the Learning Object Meta-data (LOM) standard can be used to describe learning objects. Authoring of these data is usually provided by learning management systems.
2. *Transferring specific adaptation information to AWS:*
   Usually adaptation systems need additional information which is not available in ordinary learning management systems, for example relations between learning objects, skills and skill assignments to learning objects, and didactical information. Information of this kind can be modelled as ontologies and, for example, described with the Web Ontology Language (OWL). An authoring tool is needed on the side of the LMS, which provides input facility for these data. Since the LMS should not be modified, this authoring tool should be implemented in the LMS extension module.
3. *Transferring lesson and course information to AWS:*
   A lesson or course is a set of learning objects (or assessment objects). The information

which objects are part of the lesson has to be sent to the AWS. This can be done by using the Resource Description Language (RDF) or the Web Ontology Language (OWL). The sequence of the objects within a lesson is determined by the adaptation system.

4. *Transferring control commands to AWS:*
   Since the user interface is under the responsibility of the LMS, the user control commands (for example start course, stop course, get next learning object) are input in the LMS. Therefore they have to be sent to the AWS in order to initiate adaptation procedures.
5. *Transferring sequence information to LMS:*
   In order to adapt content to learner (controlling the adaptation of the learning experience), the adaptation system decides the sequence of learning objects (including assessment objects for adaptive testing). The information which object is next has to be transferred to the LMS after a respective user control command was received from the LMS.

3.4 Use Cases

There are four use cases which are typical for e-learning systems: system administration, content authoring, creation of courses and lessons, and learning experience. System administration is related to the particular LMS and adaptation system and includes tasks such as software maintenance, user and user role administration, and setting user permissions.

Content authoring consists of two parts. First, creating and editing learning content stored in the LMS is performed by using the genuine tools of the LMS. Second, authoring specific adaptation information (see Section 3.3) in order to enrich learning content with additional information needed for the adaptation process is performed by using an authoring feature which must be provided by the LMS extension.

Creating lessons is accomplished by selecting learning objects and putting them together to a lesson or course. Since the sequence of a lesson is determined by the adaptation system, this is not the task of the lesson author. This functionality has to be provided by the LMS extension, because it is different from ordinary LMS.

The learning experience of the learner is done by connecting to the LMS and choosing a lesson. Before starting a lesson, personal profile and preferences information must be input by the learner. The system can personalise the content to the learner and offers learning objects fitting to the learners knowledge and competence state as well as to the learners profile and preferences.

4 Application Prototype

An application prototype has implemented the flexible approach described above (see Figure 3). The adaptation system is founded on the Competence-based Knowledge Space Theory (CbKST) [6], a formal psychological theory which models a learner’s knowledge and competence state. This theory provides methods to assess the learner's competence and knowledge state and to create a personalised learning path adapted to the learners' knowledge level.

For this prototype Moodle [8] has been chosen to act as learning management system. Moodle is a popular open source Web application which provides an open architecture for integrating new modules. Therefore the extension which is the bridge to the Web service can easily be plugged to the core system. Moodle has a proven and mature user interface, which is a benefit for learners, content creators, and lesson authors.

The adaptation system integrated into the Web service is built on the formal framework CbKST. Algorithms and methods built on CbKST are implemented to perform content structuring, to assess learners' competence and knowledge levels, and to create learning paths. Utilising competence levels of learners personalised learning paths can be created by the adaptation system.

Goal of this prototype is to demonstrate the research work done for the EC-funded iClass [7] project. Within this project theoretical foundation of structuring content, assessing learners' competence and knowledge levels, and creating learning paths are researched. Though they will be implemented in the iClass system by project partners, in this prototype their applicability can demonstrated and evaluated independent from the iClass system. The work for this prototype is still in progress and it is planned to be published by end of 2007.

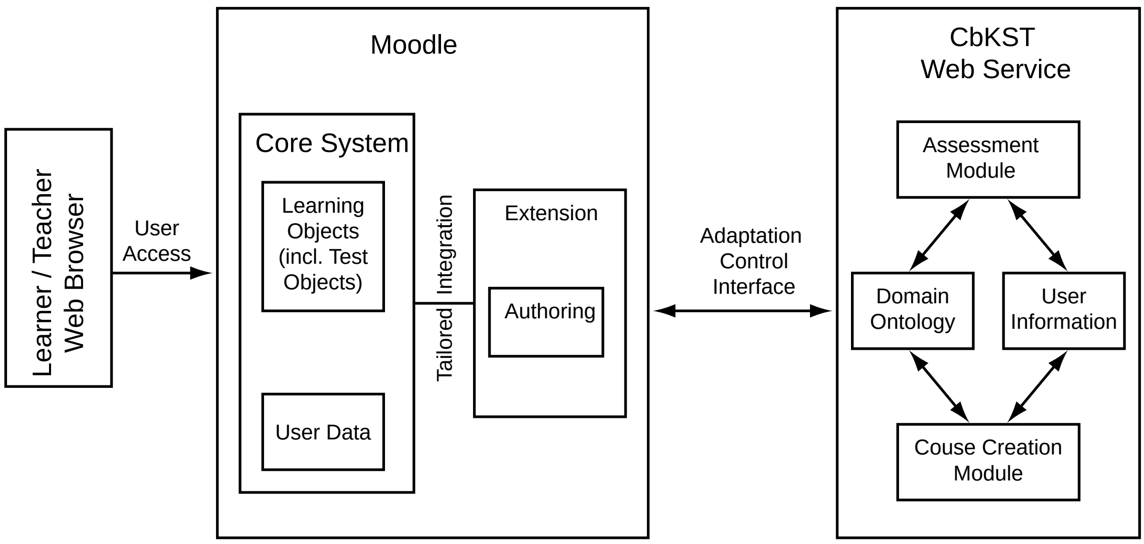

Figure 3: Prototype Application. This diagram outlines an application which has implemented the flexible adaptation approach.

5 Conclusion and Outlook

In this paper an attempt were made to find an approach to integrate adaptivity to arbitrary learning management systems which usually have little or no adaptivity functionality. Adaptivity is encapsulated in a Web service which is connected to the LMS and which controls the adaptation of the course sequence to learners. An application prototype which is currently developed was presented to demonstrate the applicability of this approach.

Didactical models are not included in this approach at present. The next step in further development of this approach could be to integrate adaptation to didactical models. If a teacher chooses or creates a didactical model for a lesson, then the adaptation system could include this information when creating the sequence of the learning objects.

Further considerations can be made about monitoring the learner's behaviour and actions in order to exploit these data for the adaptation process. However, this might be difficult, because it could be needed to modify the core LMS to achieve the behavioural data. On the other hand, if an adaptation system can process these data, the degree of adaptation would be increased.

Acknowledgements:

The work presented in this paper is partially supported by European Community under the Information Society Technologies (IST) program of the 6th FP for RTD - project iClass contract IST-507922. The authors are solely responsible for the content of this paper. It does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

Parts of this paper are based on experiences gained in the AdeLE project. This project is partially funded by the Austrian ministries BMVIT and BMBWK, through the FHplus impulse program. The support of the Department of Information Design, Graz University of Applied Sciences (FH JOANNEUM) and Institute for Information Systems and Computer Media (IICM), Faculty of Computer Science at Graz University of Technology as well as individuals involved in the AdeLE project are gratefully acknowledged.

References:

[1] Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *Journal on User Modeling and User-Adapted Interaction, 6,* 87-129.
[2] Conlan, O., Hockemeyer, C., Wade, V., & Albert, D. (2002). Metadata driven approaches to facilitate adaptivity in personalized eLearning systems. *Journal of Information and Systems in Education, 1*, 38-44.
[3] Conlan, O., Hockemeyer, C., Wade, V., Albert, D., Gargan, M. (2002). An Architecture for Integrating Adaptive Hypermedia Service with Open Learning Environments. *World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA'02).*
[4] De Bra, P., Brusilovsky, P., and Houben, G. (1999). Adaptive hypermedia: from systems to framework. *ACM Computing Surveys, 31*.
[5] Gütl, C., Barrios, V., Mödritscher, F., (2004). Adaptation in e-learning environments through the service-based framework and its application for AdeLE. *In Proceedings of the ELEARN 2004, Washington, USA* (pp. 1891–1898).
[6] Hockemeyer, C., Conlan, O., Wade, V., & Albert, D. (2003). Applying competence prerequisite structures for eLearning and skill management. *Journal of Universal Computer Science, 9,* 1428-1436.
[7] iClass. http://www.iclass.info/
[8] Moodle. http://www.moodle.org/
[9] O'Connor, A., Wade, V., Conlan, O. (2004). Context-Informed Adaptive Hypermedia. In: Proceedings of the Sixth International Conference on Ubiquitous Computing (Ubicomp'04).
[10] Türker, A., Görgün, I., Conlan, O. (2006). The Challenge of Content Creation to Facilitate Personalized E-Learning Experiences. *International Journal on E-Learning (IJeL), Special Issue: Learning Objects in Context, Volume 5, Issue 1, pp. 11-17*

Authors

Alexander Nussbaumer, Dipl.-Ing.
Cognitive Science Section, Department of Psychology, University of Graz, Austria
Universitätsplatz 2 / III, A-8010 Graz, Austria
alexander.nussbaumer@uni-graz.at

Christian Gütl, Dipl.-Ing. Dr. techn.
Institute for Information Systems and New Media, Graz University of Technology, Austria
Inffeldgasse 16c, A-8010 Graz, Austria, Europe
cguetl@iicm.edu

Cord Hockemeyer, Dipl.-Inform.
Cognitive Science Section, Department of Psychology, University of Graz, Austria
Universitätsplatz 2 / III, A-8010 Graz, Austria, Europe
cord.hockemeyer@uni-graz.at