Integrating Adaptive Functionality in a LMS

doi:10.3991/ijet.v4i4.958

Kees van der Sluijs[1] and Kai Michael Höver[2]

[1] Technische Universiteit Eindhoven, Eindhoven, The Netherlands

[2] IMC Information Multimedia Communication AG, Saarbruecken, Germany

***Abstract*—Learning management systems are becoming more and more important in the learning process in both educational and corporate settings. They can nowadays even be used to server actual courses to the learner. However, one important feature is lacking in learning management systems: personalization. In this paper we look into this issue of personalization that enables courses to be adapted to the knowledge level and learning preferences of the user. We shortly review the state of the art in adaptive systems that allow creating adaptive courses. Then, exemplified in the popular LMS called CLIX we look at the authoring of an adaptive Business English course. We demonstrate how such a static course can be made adaptive by using the GALE adaptive engine. We then show that GALE can be integrated into CLIX, and in other LMSs as well, so that personalization and adaptation can become widely established technology.**

***Index Terms*—learning management system, adaptivity, personalization, adaptive learning engine, GALE, CLIX, SSO, real integration, user model exchange, GRAPPLE, middleware**

I. INTRODUCTION

A Learning Management System (LMS) is a system that allows the tracking and managements of learners as well as actually providing a learning environment for those learners. Many LMSs nowadays support providing complete courses to their learners, which allows learners to learn course material from any location at anytime. However, all LMSs thus far that provide courses only support static courses which are not tailored towards theirs specific learners.

The key to creating a *responsive* learning environment that motivates, engages and inspires learners, and through this leads to better learning results, is *personalization.* Personalization enables the environment to present the most appropriate, interesting and challenging learning activities, while it avoids learning resources that only present knowledge the learner already has, and also avoids learning material that is beyond the reach of the learner at the time (but it may present that material later, after the learner has acquired the necessary prerequisite knowledge). Organizations that use learning environments like universities and corporations realize more and more that personalization is a key to creating acceptance of the technology for technology enhanced learning by the end users.

Personalization in courses is achieved through *adaptive learning* technology. Learning technology uses *designed intelligence* to perform the adaptation. Research has brought forward several adaptive learning frameworks that facilitates. If we look at the wider field of adaptive hypermedia adaptive engines, which also include engines that are not only tailored for adaptive courses but other adaptive applications as well, the most well known systems include AHA! [1], InterBook [2], and APeLs [3]). AHA! is an Open Source adaptive hypermedia system mainly used in education domain. It supports several adaptation techniques like adaptive guiding, link annotation, link hiding, and adaptive presentation support. AHA! is currently the most widely used adaptive hypermedia system. Interbook provides an environment for authoring and serving adaptive online textbooks. It supports adaptive navigation that guides the users in their hyperspace exploration. The guidance is visualized by using annotations (e.g. icons, fonts, and colors). Adaptive Personalized eLearning Service (APeLS) is a multi-model, metadata driven adaptive hypermedia system that separates the narrative, content, and learner into different models. The adaptive engine in APeLS is a rule-based engine that produces a model for personalized courses based on a narrative and the learner model.

In the GRAPPLE Project[1] we are currently building a new and improved adaptive engine, based on our experience with AHA!. This new engine, named GALE (which stands for GRAPPLE Adaptive Learning Environment), intends to be more flexible and powerful than the old AHA!, while also providing an authoring environment which allows teachers to easily create adaptive courses. GALE is built to excel and distinguish itself from previous work in the following selection of points:

- It (learns and then) knows about the learner's *preferences*, e.g. in terms of preferred types of activities or learning material and the order in which they appear, etc.
- It keeps track of the *learner's knowledge and skills*, in a *fine-grained user model*. This model may be distributed over different courses and even different engines.
- It allows the use and reuse of data on the Web, so that existing non-adaptive courses and other Web resources can be easily used and transformed into adaptive courses.
- The Grapple framework facilitates exchange of user model information between courses and engines, so that moving between companies or universities does not need to mean loss of data and force learners to start from scratch elsewhere. This exchange of data is based on semantic technologies.
- The presentation style is now also adaptive, e.g. the engine can present a different presentation style based on the preferences and skills of the user. The

[1] http://www.grapple-project.org/

presentation style can also be adapted to external styles, for example to match a corporate presentation style

- It adapts the learning experience through *adaptive presentation* techniques and through *adaptive navigation support* [4]. Adaptive presentation means that either the information presented to the learner, the presentation form, and/or the activity type, (or all of these,) are adapted.
- GALE offers explicit support for adaptive simulation (e.g. adaptation based on workflows) and support for learning environments in a Virtual Reality setting

However, the main goal of the GRAPPLE Project is to integrate adaptive functionality in existing LMSs, so that the many users of these systems world-wide can start profit from the now mature technology of adaptive learning (i.e. most large LMSs are represented in the project). In this paper we discuss the integration of adaptive technology in one of those LMSs named CLIX. In section 2 we introduce CLIX and we show the existing non-adaptive course on Business English as it can currently be served in CLIX directly, i.e. non-adaptive. Than we show in section 3 how we transform the business English course in an adaptive course by using the Graphical Author tool for AHA! and the GALE adaptive engine. Finally we show how we currently (as a first prototype) integrated GALE within the CLIX LMS.

II. A NON-ADAPTIVE COURSE DISPLAYED IN A LMS

In this section the structure of the widely used Learning Management System CLIX[2] is described as an example Learning Management Systems. First, we briefly describe the CLIX system architecture and as such identify typical common features of Learning Management Systems. Subsequently, we explain how a course (in this case a Business English course) and its content are typically added to a LMS and how learners can access it. We further explain what a tutor has to do, when he/she wants to adapt the content of a course to the diverse characteristics of different learners.

A. *The CLIX system architecture*

As many Learning Management Systems the CLIX learning platform is a classic web application for which the user needs merely a web browser (client). The client is connected using TCP/IP to communicate with a web server.

CLIX follows a 3-tier architecture [7]: the provision of web pages, the CLIX program logic and the data maintenance are implemented in three separate layers. As a J2EE application, CLIX is also platform-independent.

B. *Features of a LMS*

Using the web browser interface, different users with different roles can fulfill their tasks: administrators can change system settings, or register, edit, and delete both users and courses; tutors can enroll learners to courses and monitor their learning progress; and learners can view available courses and access their training material. The basic features of CLIX (the same goes for other standard LMSs) can be classified under three core groups of features: Learning Organisation Management, Learning Content Management, and Learning Process Management (c.f. Figure 2).

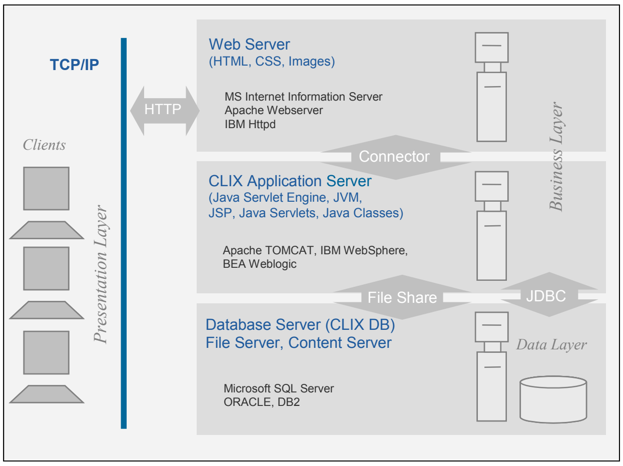


Figure 1. The CLIX architecture

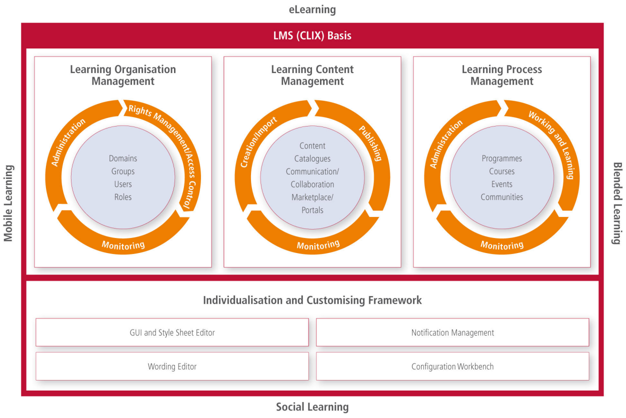


Figure 2. Common LMS features
(CLIX serves as an example)

The Learning Organisation Management allows the administrator to manage the users, to define roles and to assign access rights to learning objects, and user groups or individual users.

The Learning Content Management enables a tutor to model different learning scenarios. This includes content management functionalities and compliance with standards such as AICC, SCORM, QTI and IMS Learning Design. Furthermore, a tutor can manage processes for the creation, release and delivery of learning content. The Learning Process Management includes the functionality for creating educational programs, courses, syllabi and curricula. Using these functionalities a tutor can control learning processes and overview the status of current training activities.

The above-mentioned functionalities of a LMS allow a tutor to create the content of a course, organize it in a syllabus and deliver the content to the learners. We will describe this common procedure in the next section as it is typically performed in a LMS.

C. *A non-adaptive course*

In this section we describe how a course is typically created in a LMS. For this example we use a business English course entitled "Ten days in Manchester" which is

[2] http://www.im-c.com/

based on a course offered by the BBC world service[3]. The course consists of ten learning units with six lessons each.

When authoring a new application in an LMS, every learning object which should be part of a course has to be created and added to the LMS repository or an external one which is accessible by the LMS. Afterwards these learning objects are arranged in the syllabus of the course. Subsequently, the learners are enrolled for the created course. Every enrolled learner will see the same content ("one size fits all approach"). Unfortunately, this does not take into account the diversity of learners. Learners differ in many respects. In business settings, learners have for example different previous knowledge, skills, or job and learning goals, respectively. They may have even different learning styles. This diversity of learners is not taken into account by current LMSs when offering course material.

Consider for instance a learner that already has basic knowledge about business English and a learner that has no business English experience at all. In current LMS this means that you can either force the advanced learner to go through the whole course, bothering him with beginner concepts her already knows, or you can create separate beginner and advanced courses even though you cannot tailor advanced courses to the exact knowledge of the user as this can vary on a per user basis. Now also consider that some users are slow learners (e.g. need many examples), while other learn much more quickly and only need one example to learn about a concept. It is clear that creating separate specific courses for groups of users is not feasible if you consider even several user characteristics, while it is very laborious and very inflexible, i.e. it is impossible to adapt to specific learning characteristics for hundreds of learners. This problem can be addressed by an adaptive system which allows a personalized delivery of a course to learners. In the following section it is described how this business English course can be made adaptive by considering different learner properties.

D. An adaptive course displayed in a LMS

In this section we describe how the previously described business English course (or any other web content) can be processed in such a way to become adaptive.

The authoring process mainly involves two steps. First, the existing content is stored in single XHTML files, one for every single lesson (concept). This can be done straightforwardly, because the static content of the web-based course already exists in several HTML pages. Second, rules have to be created expressing the adaptive behavior. For this purpose we used the Graphical Author tool for AHA! [5]. It allows creating adaptive web-based systems by using a graphical interface.

In this example, the adaptive Business English course takes into account the following learner characteristics and properties: previous knowledge, job role, test results, preferred learning activities, and the user platform. The menu structure of the course is influenced by a learner's previous knowledge, her job role, and her preferred learning activities. If a lesson fits these properties it is annotated in the course menu with an icon signalizing if this recommended or not. This adaptation technology is known as *adaptive navigation support* [4,6]. In other words, these three properties can be seen as preconditions: if a lesson matches both the learner's job role and the preferred learning activity, and the learner has not already visited this lesson, then this lesson is recommended for this learner. Such a condition can be created with the Graphical Author tool by dragging lines from the learner characteristics to the lesson. These lines are translated into rules in the adaptation process later on. The Graphical Author tool contains some predefined rules, e.g. the precondition that we introduced earlier, but also allows the creation of new ones. Figure 3 displays a screenshot of some created rules. With these rules it is defined which lessons are recommended for a certain preferred learning activity. This is done in the same way for job roles.

The example uses a very simple rule:

```
           <setdefault location
     ="destination.suitability" combina-
tion="AND">___source.interest </setdefault>
```

Put into words, this rule says that if the “interest” attribute of the source (beginning of the arrow) is true, then the “suitable” attribute of the destination (end of the arrow) is also true.

It is further possible not only to adapt the menu presentation but also the content itself by taking into account the test results. This is explained in more detail in another paper [7]. In this paper an adaptive web based training multimedia course on business English was created with a learning time of about 5 hours. It was used to demonstrate adaptive learning concepts to learners, authors and training providers. The authored course can be visited on http://www.learning-demo.eu/.

In the next section it is outlined how an adaptive course can be integrated in a LMS in a simple way.

III. INTEGRATION BETWEEN CLIX AND GALE

In this section we introduce a first prototype which delivers an adaptive business English course to learners. In this solution both presentation and adaptation process are done by GALE, while the GALE engine is seamlessly integrated in the LMS system. The login name of the LMS user is send to GALE in order to identify the correct user and to adapt the content presentation according the user's user model and the defined adaptation rules (section II.II.D). In case there is no existing record for this learner in the GALE user model (UM), a new one will be created in order to a learner’s properties and to keep track of the learner’s domain knowledge. The LMS can communicate with GALE either through the GALE event-based Web service or through HTTP messages (Figure 4). Subsequently, we describe a first prototype using the second approach.

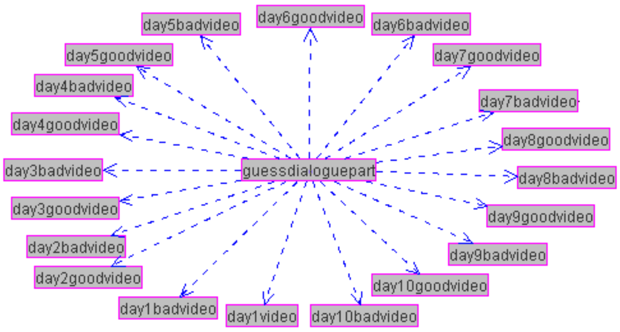


Figure 3. Assigning a learning activity to lessons with the Graphical Author tool

[3] http://www.bbc.co.uk/worldservice/learningenglish/business/tendays/index.shtml

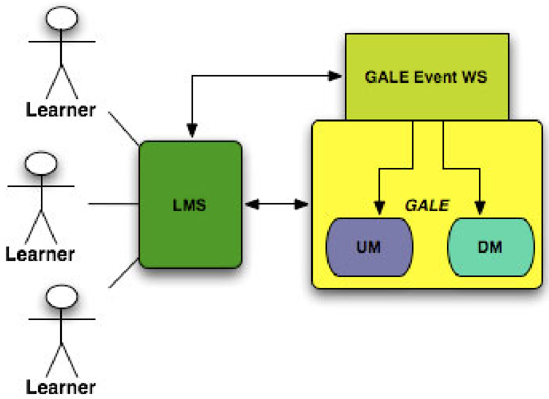
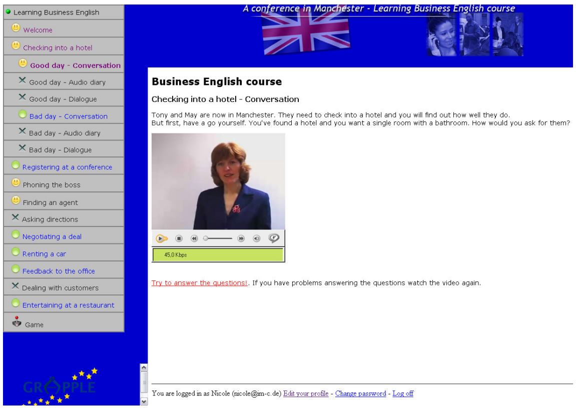


Figure 4. First prototype architecture



Business English course

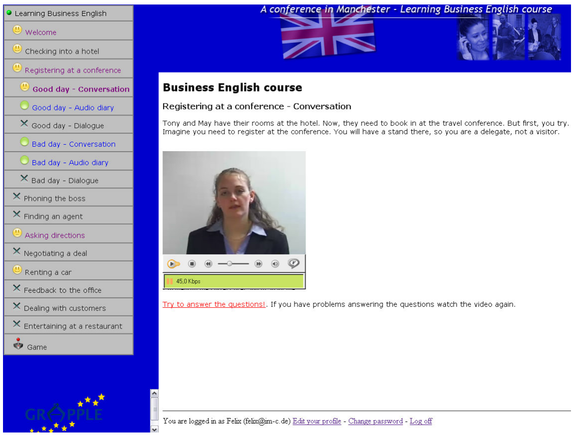
Checking into a hotel - Conversation

Tony and May are now in Manchester. They need to check into a hotel and you will find out how well they do.
But first, have a go yourself. You've found a hotel and you want a single room with a bathroom. How would you ask for them?

Try to answer the questions!. If you have problems answering the questions watch the video again.

You are logged in as Nicole (nicole@im-c.de) Edit your profile - Change password - Log off

Figure 5. The business English course adaptively presented for Nicole



Business English course

Registering at a conference - Conversation

Tony and May have their rooms at the hotel. Now, they need to book in at the travel conference. But first, you try.
Imagine you need to register at the conference. You will have a stand there, so you are a delegate, not a visitor.

Try to answer the questions!. If you have problems answering the questions watch the video again.

You are logged in as Felix (felix@im-c.de) Edit your profile - Change password - Log off

Figure 6. The business English course adaptively presented for Felix

In a first step the administrator or tutor needs to add the course to the LMS. In CLIX this is done by adding a WBT (Web Based Training) resource by using the CLIX Content Manager. This WBT resource representing the adaptive course has several properties (figure 23). This includes

- the title (in this case "Business English course"),
- a short course description,
- the source describing the course's URL which contains a shared key in order to authenticate the LMS and a link to a concept of the course (in this case the URL points to the "welcome" concept of the course),
- and some screen properties of the screen which displays the course.

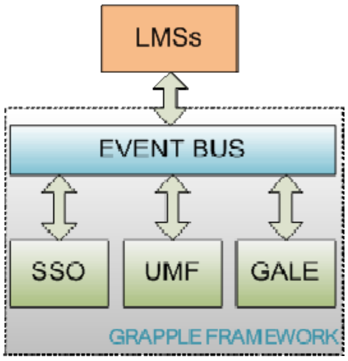


Figure 7. GRAPPLE Framework Architecture

In a second step the tutor enrolls the learners which use the LMS. As an example the tutor enrolls two learners named Nicole and Felix. Both learners can access the adaptive business English course after they have logged in the CLIX LMS. As they differ in the job roles and knowledge levels the content presentation is adapted to their user profile, so that the course is displayed in different ways for each learner although the tutor only needed to create this course and its adaptation rules once. The different content presentations for Nicole and Felix are displayed in Figure 5 and Figure 6 respectively. Both screenshots show that the course menu differs for the two learners due to different learning goals and previous knowledge.

IV. EXTENDED ARCHITECTURE AND CURRENT IMPLEMENTATION

The current implemented integration, as described in the previous section, is only the first simple step in the project. We are currently working on the extension of this simple approach to a more sophisticated setup. Figure 7 shows a high level (simplified) architecture of the GRAPPLE framework that is currently developed. In this framework we discern the components GALE (the adaptive engine), the User Model Framework (UMF), the Single Sign-On (SSO) facility and the Event Bus communication layer.

Communication within the GRAPPLE framework is facilitated by the Event Bus Web Service. Applications (LMS, but also the GRAPPLE components) that connect with the Grapple Framework subscribe via the Event Bus. Every application can set a list of event types in which it is interested and every application can post events of a specified type to the Event Bus. If an application posts an event to the Event Bus it will send this to all listening applications that specified interest in that type of event. The listening application can then process the event and if needed send an event-reply. The GRAPPLE event bus abstracts from the need that every GRAPPLE component needs to be exactly aware about which LMS (or other application) accesses it. The Event Bus also simplifies the process of adding new components to the Framework and facilitates distribution of components (as the components might have instances running at several locations).

The UMF facilitates exchange of user data between LMS (or other application) and the GALE adaptive engine, but also between different LMSs or different applications in possible different instances of the GALE adaptive engine. Examples of what might be exchanged between GALE and LMS are for instance a measure of progress with the adaptive course from GALE to LMS and the results for a specific test in the LMS to GALE (which allows blending LMS capabilities like taking tests) with the GALE's adaptive course. Exchange between GALE applications allows reuse of data between similar domains, e.g. we could estimate from a regular English course what a users level is when starting the business English course. The UMF is based on the use of a standardized syntax (based on the W3C RDF recommendation[4]) and mappings to bridge semantic differences, while at the same time providing a core ontology that is useful for most LMSs and the GALE adaptive engine.

The Single Sign-On facility is needed to safely and securely integrate user authentication between systems. This is vital when seamlessly integrating LMS and GALE, but also necessary within the GRAPPLE system if the user uses different adaptive applications in GALE under different accounts. Determining the user identity in this way enables reuse of user data and therefore takes away the burden of the user to (explicitly or implicitly) provide this data to the system time and again. As several Single Sign-On frameworks are already openly available, we choose to reuse such a system named Shibboleth[5].

Even though the GRAPPLE Framework Architecture is much more complex than our first prototype, its integration with LMSs is not much more complex at all as the LMS simply only has to implement some Event Bus calls to start using the framework. This pays off enormously with some very useful features like secure SSO and user model exchange functionality.

V. CONCLUSION

As LMSs get used more and more in learning environments in both academia and corporate settings, the lack of personalization in these systems starts to become an issue. In this paper we showed that a static course, exemplified by a Business English course in the CLIX LMS, can be easily transformed into an adaptive course using the GALE engine. We also described our efforts to integrate the GALE engine in LMSs like CLIX. While we mainly highlighted a first prototype implementation that is finished and available, we also outlined our current work in extending this integration work by creating the GRAPPLE Framework which provides a powerful adaptive learning environments with features like Single Sign-On, user model data exchange, and a flexible GALE engine that allows reuse of course material.

ACKNOWLEDGMENT

This paper and demonstration is part of the ongoing research and development in the EU FP7 project GRAPPLE (215434) and could not be realized without the close collaboration between all 15 GRAPPLE partners, not listed as authors, but nonetheless contributing to the ideas described here.

REFERENCES

[1] P. De Bra, D. Smits, and N. Stash. "The Design of AHA!," *Proceedings of the 17th ACM Conference on Hypertext and Hypermedia (Hypertext 2006), Odense, Denmark*, pp. 133-134, August 2006.

[2] P. Brusilovsky, J. Eklund, and E. W. Schwarz. "Web-based Education for All: A Tool for Development Adaptive Courseware," *Computer Networks and ISDN Systems*, 30(1-7), pp. 291-300, 1998.

[3] O. Conlan, V. P. Wade, C. Bruen, and M. Gargan. "Multi-model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning." *Proceedings of the Second Adaptive Hypermedia and Adaptive Web-Based Systems Conference (AH 2002), Malaga, Spain*, pp. 100-111, May, 2002.

[4] P. Brusilovsky. "Adaptive Hypermedia," *User Modeling and User-Adapted Interaction*, 11 (1-2), pp. 87-110, 2001. (doi:10.1023/A:1011143116306)

[5] P. De Bra, N. Stash and D. Smits. "Creating adaptive web-based applications". *Tutorial at the 10th International Conference on User Modeling, Edinburgh, Scotland*, July, 2005.

[6] P. Brusilovsky. "Methods and techniques of adaptive hypermedia," *User Modeling and User-Adapted Interaction*, 6(2-3), pp. 87-129, 1996 (doi:10.1007/BF00143964)

[7] K. Höver, N. Faltin. "Authoring of an adaptive group-oriented business English course with AHA!," *Proceedings of the First International Workshop on Individual and Group Adaptation in Collaborative Learning Environments, Maastricht, The Netherlands*, September, 2008.

AUTHORS

Kees van der Sluijs is with the Eindhoven University of Technology, 5600 MB Eindhoven, Netherlands (e-mail: k.a.m.sluijs@tue.nl).

Kai M. Höver is with the New Business department at IMC AG, Altenkesseler Str. 17/D3, 66115 Saarbruecken, Germany (e-mail: kai.hoever@im-c.de).

This work was supported in part by the European Commission under the seventh framework programme in the IST research priority, contract number 215918 (GRAPPLE, www.grapple-project.org).

Submitted 15 April 2009. Published as resubmitted by the authors on 18 November 2009.

[4] http://www.w3.org/RDF/

[5] http://shibboleth.internet2.edu/