GRAPPLE: Integrating Adaptive Learning into Learning Management Systems

Paul De Bra, Mykola Poberezhnyk, Kees van der Sloot, David Smiths
GRAPPLE Project, Eindhoven University of Technology (UTE)
Eindhoven, The Netherlands
grapple@win.tue.nl

Abstract: GRAPPLE is an EU funded IST FP7 project that brings together a group of researchers into adaptive learning technology and environments and developers of learning management systems (LMSs), in order to offer adaptive learning as a standard feature of future LMSs. This paper presents the overall architecture of GRAPPLE, and explains some of the main challenges in creating a truly generic-purpose adaptive learning environment (ALE) that can be used with different LMSs and a distributed architecture of user modeling services. In particular the paper describes the conceptual adaptation model, its translation into adaptation rules, and some technological challenges that result from separating the concerns of adaptation and user modeling.

Introduction

After some initial isolated research efforts the field of adaptive technology-enhanced learning (or adaptive TEL) really took off with the publication in 1996 of the seminal paper on adaptive hypermedia by Peter Brusilovsky (Brusilovsky, 1996), which was updated in 2001 (Brusilovsky, 2001). A large number of research papers appeared in journals and at conferences such as WebNet/ELEARN, Intelligent Tutoring Systems, ED-MEDIA, User Modeling, and later also the dedicated Adaptive Hypermedia conference series. A number of research prototypes of adaptive TEL systems were built, and mostly used only for demonstrative purposes, or sometimes also for adaptive course delivery in the authors’ institutes. One noteworthy exception is the AHA! system (De Bra et al., 2006), a general-purpose open-source adaptive hypermedia platform that has been used by researchers and educators from all over the world. However, due to the use of adaptive technology in learning applications remains very limited. The GRAPPLE project is aimed at changing that by bringing adaptive TEL to the masses. This is done by integrating an adaptive TEL environment (henceforth abbreviated as ALE, for adaptive learning environment) with major learning management systems (or LMSs) using a service-oriented (web) framework approach.

GRAPPLE bundles the expertise of researchers from 14 universities, research institutes and companies, including the creators of the adaptive systems AHA! (De Bra et al., 2006), KIBS-Hyperbook (Heuvel et al., 1999), RATH (Hockenmyer et al., 1998), APELS (Cosalan et al., 2002) and WINDS (Kravkav et al., 2004), of user modeling and languages and services, including User/ML (Heckmann et al., 2003), (Heckmann et al., 2005) and the work of (Van der Sloot et al., 2006), of experts in learning standards (e.g. the Open University Nederland and Atos Origin Spain), of developers of and distributors to LMSs including Moodle, Croleline and Sakai, and of developers of industrial TEL applications (Atos, Quinti Labs and IMC Information Multimedia Communication AG). The goal of GRAPPLE is to have the ALE become a “standalone” component of the LMS so that the thousands of institutions (world-wide) using these LMSs automatically have access to the ALE.

In this paper we first describe why combining an ALE with an LMS is a natural combination of two tools with complementary functionality. We then give an overview of the GRAPPLE architecture and briefly describe the different components. We emphasize the adaptation engine and user modeling infrastructure and discuss the separation of these two components.

1 Some noteworthy examples are the AlcoZone alcohol tutorial from Virginia Tech (Bhouns, 2006), an automatic theory course from Keera University (Lee et al., 2005) and a programming course from the Slovak University of Technology (Bieliková et al., 2005). We are also aware of on-going work in Brazil, Colombia, and South Africa.

The “marriage” between an ALE and LMS

Many institutions in higher education, but also (large) knowledge-intensive companies, use a learning management system to manage the learning process. This management consists of both administration and of facilitating the learning itself by means of course selection, delivery and evaluation tools. The functions of an LMS include (but there are many more):

- registering (and later perhaps deleting) users, and authorization (login, access restrictions)
- enrollment in courses (or other types of learning modules)
- workflow (task management, notifying learners of assignments that are due, assignment of new tasks after assessment of completed tasks, notifying tutors of completed assignments to be graded, etc.)
- distribution (or delivery) of learning material
- assessment (including multiple-choice tests, but also upload of assignment work for off-line assessment by a tutor)
- portfolio management (certifications, registration of completed courses or course programs)

One would expect that the distribution of learning material would be very well supported by all LMSs and widely used. However, in a lot of cases this part of the LMS is only used (in practice) to serve documents (complete course texts as a single Word or pdf file, Powerpoint slides, etc.) that are not well integrated with the functionality of the LMS and do not enable fine-grained tracking of the learner’s progress. This is where the ALE comes in. It performs the following functions:

- presenting a course text as a website (with links, allowing fine-grained tracking of the learner’s progress)
- adaptive guidance through link generation, sorting, hiding or annotation
- adaptive page content to automatically compensate for missing pre-requisite knowledge
- possibly other adaptive tools like adaptive tests, collaborative filtering (of links), etc.

In order to successfully combine an LMS and an ALE the following types of integration facilities are needed:

- single-sign on: when a learner logs in on the LMS, to go to a course sub-site, and then to a course page (the page must be automatically be logged on in the ALE (and registered if this was not yet the case)
- user model/profile exchange and communication: the ALE must have access to the information the LMS stores about the user (e.g. results of multiple-choice tests, but also previously entered courses or skills and knowledge obtained elsewhere but registered in the LMS); potentially the LMS may need user information from the ALE as well, to record what the user has studied, at a high level of abstraction (whereas the ALE keeps a fine-grained user model).

When an LMS and ALE are properly integrated the learner should not be aware that some of the used services are offered by the LMS and others by the ALE. This can be achieved by having the ALE operate as one of the LMS’s tools, presenting its output in one or more frames within the total presentation form offered by the LMS. GRAPPLE aims at realizing such seamless integration between its ALE and different LMSs. It remains to be seen how seamless the integration will be in the end. In an early experiment we have realized (in a collaboration with the University of Linz, Austria) a truly seamless integration between the AHA! system and Sakai. This experiment has shown that such ALE-LMS integration is possible.

The GRAPPLE architecture

Figure 1 shows the overall architecture of the GRAPPLE ALE part. Although inspired by past development on the AHA! system a significant architectural difference is the separation of the adaptation engine from the user modeling service. In GRAPPLE a learning application is represented at the conceptual level through a conceptual adaptation model (DM) and the Adaptation Model (AM), still called the “Teaching Model” in (De Bra et al., 1999). In the CAM concepts are “connected” with each other at different (semantic) levels. Unlike in the AHAM-based LADS model
(Cristea et al., 2003) which has 5 fixed levels, in the GRAPPLE CAM there may be arbitrarily many levels, and they may be different in each learning application. Typically the DM will correspond closely to a subject ontology (the subject domain of the learning application or course). There can be several levels related to adaptation. A task or goal level may relate tasks or learning goals to sets of concepts that need to be studied in order to achieve a goal or to be able to perform a task. A prerequisite level may connect pairs of (sets of) concepts to indicate that the prerequisite concepts should be studied before the dependent concepts. This may be used to perform linking, hinting or annotation, but also to automatically include prerequisite explanations. A process or sequencing level may indicate desired sequences of concepts to be studied in a fixed order, or a lattice offering limited freedom in navigating through the concepts. This may be used to generate a menu-like structure of links to the concepts, or a "next" button for sequential navigation. Several conceptual adaptation structures in the CAM will use and possibly define rules how to update the user model (UM). The language used to describe a CAM is still being designed. It will be inspired by elements from IMS-LD (IMS, 2003) and from the LAG language used in the LAOS research. Even though this is a high-level language, independent of actual adaptation engines to be used, authors cannot be expected to define the desired adaptation in any "technical" language. Therefore a graphical authoring tool for CAMs will be built to make the creation of CAMs possible without any technical knowledge of the CAM language. The use of the Graph Author in AHA! has already shown that with such a graphical tool authors can define reasonably complex adaptation. (We have been using AHA!’s Graph Author for a number of years already in a course on adaptive hypertext.)

![Diagram of the GRAPPLE ALE architecture](image)

The separation between UM and the adaptation engine is necessary in GRAPPLE because UM is no longer an integral part of the ALE but is shared between the ALE and the LMS. What looks like a simple (black) box in the ALE architecture may be a distributed infrastructure of services that all know something about how the user is learning (St.etc. the user’s learning state). The adaptation engine may then consist of: a distributed infrastructure (services) that know something about the user’s learning state and are able to provide personalization services, or a change in presentation order of material, examples, explanation, theory and activities (activist/reflective style). The adaptation engine can query the UM infrastructure or broker once, and cache the values obtained.

For adaptation to current knowledge values and for instance some associated prerequisites the situation is more complicated: each action of the user may cause one or more knowledge updates, which in turn may result in one or more prerequisite changes. In a tightly integrated system like AHA! a single engine, with an in-memory prerequisite model can be used. But in a more complex system there may be several engines, each responsible for a different part of the adaptation. In the distributed GRAPPLE scenario page is sent to the learner's browser, without causing any noticeable delay. In the distributed GRAPPLE scenario there is no longer possibility:

- When event is signaled (by the engine) that UM is in error when all the UM updates will be performed. This requires coordination of the actions of the different engines, which may be difficult due to the need for knowledge updates to be performed in the right order. This is a no-longer possibility:

- When the generation of a page or a navigation menu involves checking the suitability of the destination of each link this involves too many queries to UM to wait for the answers before generating the (adapted) content.
Conclusions and Future Work

The GRAPPLE project promises to bring adaptive TEL to the masses by incorporating an adaptive learning environment (ALE) in popular learning management systems (LMSs). This integration however is not as easy as it looks: the aspects of user modeling and adaptation are necessarily split between different components, resulting in foreseeable performance problems. Because GRAPPLE is not just a research project but aims to deliver a production-quality ALE in a lot of attention must be paid to performance issues right from the start.

GRAPPLE also emphasizes the usability of the ALE for authors (or course designers). Therefore authoring is done using comfortable graphical authoring tools (that will be designed with existing tools such as the Graph Author of AHA in mind). Actual low level adaptation rules will be generated through templates (in a Translation Model), thus making the high level Conceptual Adaptation Model (CAM) completely system-independent and thus a candidate for going through a standardization process.

Further developments in the GRAPPLE project, before, during and after ED-MEDIA 2008, can be followed at www.grapple-project.org.

Acknowledgement

This work has been performed in the framework of the IST project IST-2001-15434 GRAPPLE which is partly funded by the European Union. The authors would also like to acknowledge the contributions of their numerous colleagues from all 14 GRAPPLE project partners.

References


