TACTICAL, DOCUMENT-ORIENTED E-LEARNING COMPONENTS

Michael Piotrowski, Mario Amelung, and Dietmar Rösner
Otto-von-Guericke-Universität Magdeburg
Institut für Wissens- und Sprachverarbeitung
P.O. Box 4120
39016 Magdeburg, Germany
mxp@iws.cs.uni-magdeburg.de

ABSTRACT

Most university e-learning strategies mandate the use of a centralized university-wide learning platform. The learning management systems typically employed in this function are "integrated" platforms, i.e., large-scale systems providing most common e-learning functions in a single application. There are, however, a number of issues with this type of systems: Due to their size and complexity they can be difficult and expensive to operate and administrate; and we feel that they are not flexible enough to allow teachers to make tactical decisions; and, since these systems cannot be used for the management of "normal" Web sites, they separate learning content from other content and duplicate functionality and administration. This paper presents an alternative approach: E-learning components that extend a general-purpose content management system with e-learning functionality, enabling the use of a single platform for learning and non-learning content and the creation of tailor-made e-learning environments.

KEYWORDS

E-learning, learning management systems, content management, component-based architecture

1. INTRODUCTION

Recent e-learning[1] projects in the context of universities[2] are focusing on sustainable integration of e-learning into the organizational structure. To achieve this goal, university-wide e-learning strategies are being developed. Besides defining a strategic vision, most e-learning strategies mandate the use of a specific e-learning platform or learning management system (LMS) (cf. Kleimann and Wannemacher, 2005; PLS RAMBØLL, 2004). We believe, however, that the typical "integrated" learning management systems prescribed by these strategies can be problematic: Since these platforms try to offer a large variety of functions they tend to involve a lot of overhead and a high total cost of ownership. Integrated LMS's separate learning content from other Web content and thus duplicate functionality and administration. We also believe that they do not offer enough flexibility to enable teachers to make the necessary tactical decisions.

As an alternative, we propose a component-based architecture, using a standard content management system as foundation, which integrates e-learning and general Web content on a single platform, enabling teachers and institutions to create tailor-made e-learning environments for their specific needs.

This paper starts with some notes on e-learning strategies and then presents our implementation of this approach, the eduComponents, a suite of e-learning components, and describes our experience with the practical use of these components.

---

[1]We are using the term *e-learning* as a general term covering all forms of teaching and learning which involve the use of information and communication technology.

[2]This paper is concerned with the use of e-learning in typical university settings; in other contexts, where learning is not the main activity, different concerns may apply.

2. E-LEARNING STRATEGIES

2.1 Funding policies

The funding policy for university e-learning projects in Germany can be divided into two phases (this paper describes the funding policy in Germany, however, the development other European countries is similar). During the first phase, from ca. 2000 until 2004, grants were primarily given to single, mostly isolated projects that focused on content creation, especially multimedia-enhanced courseware. The creation of multimedia content is very expensive (the production of videos and animations, for example, is both labor-intensive and requires expensive hardware), and the content is quickly outdated, especially in rapidly evolving subjects. Furthermore, after the end of the project funding, there were typically neither funds nor personnel available to continue maintenance and development of the content. This, in turn, resulted in the main problem: A lack of sustainability (cf. Kleimann and Wannemacher, 2003; Baumgartner, 2003; Kubicek et al., 2004, for evaluations of this first phase of funding).

Since about 2005, a new phase of funding policy has begun, which largely follows the recommendations made in Baumgartner (2003). It was recognized that in order to achieve the sustainable integration of e-learning into universities, it is necessary to consider it as a part of the university development process. The goal is therefore to achieve sustainability by integrating e-learning into the university structures. Universities are trying to realize this goal by developing *e-learning strategies*.

Besides setting strategic goals and defining target numbers, e-learning strategies typically have two main implementational aspects: On the institutional side, the creation of new institutions such as competence centers and coordination offices. On the technical side, they typically mandate the use of a centralized, "strategic" learning management system.

2.2 Issues

While an e-learning strategy is certainly a prerequisite for achieving sustainability, it will not solve all problems. In the context of universities, the first problem is that universities are typically not homogeneous and hierarchical organizations, but rather complex, loosely-coupled systems (cf. Ellström, 1983), so that top-down approaches can be difficult to implement.

Some aspects of these intricate organizational structures and processes typical for universities are certainly historic—after all, some universities have century-long traditions. However, it is important to recognize that subjects differ widely in their content (e.g., philosophy vs. physics), their course types (e.g., discussion-oriented seminars vs. frontal lectures), their students (e.g., liberal-arts students vs. engineering students) and their number of students, and, thus, in their e-learning requirements.

Furthermore, given the organizational structures of universities, the top-down approach can cause acceptance problems: Firstly by the approach itself, but also by opposition against restructuring, by the extra effort necessary for creating teaching materials, by the need to learn a new system and new teaching habits, especially if the return on investment is not (immediately) clear; Newton (2003) discusses some of these factors.

There are also technical problems with the introduction of "strategic" learning management systems. The systems typically mandated by e-learning strategies are "integrated" systems such as Blackboard, ILIAS, Moodle, OLAT, or WebCT. These systems are "integrated" in the sense that they integrate most common e-learning functions and much of the infrastructure—which is not e-learning-specific, such as user management, calendaring, discussion forums, etc.—into a single platform.

While the integration of all commonly used functionality may seem advantageous, due to their complexity, the deployment, administration, and maintenance costs of these platforms can be high; in the case of commercial systems annual licensing fees must also be added to the total cost of ownership.

Since typical learning management systems cannot be used for other, non-learning-related, Web content, they also separate e-learning from other activities on the Web. This results in the duplication of functionality and requires the administration of at least two systems, a content management system (CMS) for general Web content, and a LMS for e-learning. For example, Kleimann and Wannemacher (2005) report that the Freie Universität Berlin is using the NPS content management system alongside Blackboard since the LMS cannot be

used for general Web sites and because of deficiencies of Blackboard in the the management of large content repositories.

Typical learning management systems offer a core set of functionality that is always present; most systems can be extended in some way, however, the flexibility is often limited due to restricted interfaces or interdependencies between various functions. It is thus difficult to add specialized applications needed for certain subjects or certain course types, or to remove unneeded or unwanted functionality. Typical LMS's thus tend to strongly favor—or even force—the use of certain course structures. This in turn raises the question whether this type of systems is flexible enough to adapt to varying and evolving requirements and whether it can provide teachers with the tactical resources, i.e., the means necessary to achieve concrete educational goals in a certain environment. Grob et al. (2004) note: "Many organisations have introduced commercial LMS and gained the experience that monolithic solutions do not fulfill the dynamic requirements of complex educational institutions and are very cost-intensive."

3.   A COMPONENT-BASED APPROACH

Given the problems outlined above, we posed two questions:

a) Is it possible to use a component-based architecture—enabling the tactical combination of components according to concrete requirements—instead of the typical "integrated" systems, and

b) is it possible to use a shared platform for both e-learning and other Web content?

We have made the observation that a large percentage of e-learning is actually document management: The production, presentation, and review of written material is one of the most important aspects of higher education. Instead of re-implementing CMS functionality, we therefore decided to rely on a general-purpose CMS to provide a reliable implementation of basic document management functions.

We are using the Plone open-source CMS as the basis for our components (cf. http://plone.org/). After evaluating a number of other systems, we chose Plone for its portability, its conformance to Web standards, its workflow capabilities, and, most importantly, for its architecture, which enables the fast development and deployment of components, so-called *products*. Plone products can leverage the basic CMS functionality and interact with other products; a large number of products for a wide variety of applications is readily available. We therefore consider Plone to be a near-perfect basis for our approach, even though the basic ideas could be implemented on top of other CMS's as well.

We then proceeded to design, implement and deploy a number of Plone components, collectively called *eduComponents*. Currently the following components are available (see also Amelung et al. (2006); Rösner et al. (2006)):

ECLecture: Course information, registration, and resource management.

ECQuiz: Electronic multiple-choice tests.

ECAssignmentBox: Electronic submissions for assignments and support for assessment and grading.

ECAutoAssessmentBox: A version of ECAssignmentBox with automatic checking and assessment of assignments with immediate feedback.

These components can be used separately or in combination, and they can also be combined with other Plone products to create tailor-made learning environments. For example, if a bibliography, a glossary, a wiki, or a discussion forum is needed, third-party Plone products can simply be added. Also, all components use a uniform content representation. All objects in Plone are documents (or folders containing documents) and can be manipulated in the same ways, regardless of whether the document is a multiple-choice test or an image. This ensures a consistent and easy-to-learn user interface.

Figure 1. Example course homepage realized with ECLecture. To ensure readability, the site-wide navigation provided by Plone is not shown in this and the other screenshots

3.1 ECLecture

ECLecture is a Plone product for managing lectures, seminars and other courses. ECLecture objects group all course-related information—including course metadata (such as title, instructor, time, location, credits, etc.)—and resources. ECLecture objects can thus serve as a "portal" to all course-related materials like slides, exercises, tests, or reading lists. These materials are managed using the appropriate content types (e.g., ECAssignmentBox for assignments) and appear as resources to the course. Since an ECLecture object is a folder-like object, these resources can be stored inside of it, but they can also be stored somewhere else, even on another server. ECLecture also handles the registration to courses. Figure 1 shows a typical view of a course homepage realized with ECLecture.

3.2 ECQuiz

ECQuiz supports the creation and delivery of multiple-choice tests (see also Piotrowski and Rösner (2005)). The product supports single-answer, multiple-answer, and free-text questions. Related questions can be grouped into question groups, which are then treated as a unit. Questions and answers can be displayed in fixed or randomized order. It is also possible to present different randomly selected subsets of questions and answers to each student. Figure 2 shows a typical test view.

Multiple-choice tests are especially useful as formative tests to quickly assess the performance of *all* students of a class without the need for extra grading work, or for self-assessment. For self-assessment tests, ECQuiz can provide students with instant results and students can be allowed to take a test multiple times. Answers can also be annotated with feedback, which is shown to the candidate in self-assessment mode.

Teachers can access detailed reports, providing an overview of the performance of all students. The reports can also be exported for further processing in a spreadsheet or statistics program. Tests and individual questions and answers can be imported and exported using the IMS QTI v2.0 standard (IMS Global Learning Consortium, 2005).

Figure 2. Example multiple choice test in ECQuiz

3.3 ECAssignmentBox and ECAutoAssessmentBox

ECAssignmentBox implements the creation, submission, and grading of essay-like assignments. The assessment process is semi-automated, i.e., the assessment is done by the teacher, who is aided by the tool during the entire process of grading students' work and giving feedback. ECAssignmentBox defines a specialized workflow for student submissions: During the grading process, assignments are put through a number of *workflow states*. The workflow is designed to accommodate different ways of handling submissions, but typically the states are "submitted," "accepted," and "graded."

In computer science, programming is an essential topic in the curriculum. Practice is essential for the acquisition of programming skills, so students should be given frequent programming assignments. However, assessing large number of programming assignments is a very time-intensive task—consequently the first systems for automatically testing student programs exercises were developed and used as early as 1965 (cf. Forsythe and Wirth, 1965).

ECAutoAssessmentBox is a Plone product derived from ECAssignmentBox, i.e., it has the same basic functionality as described above. In addition, ECAutoAssessmentBox provides special support for programming assignments and automatic evaluation of student submissions in conjunction with a Web service (ECSpooler) which manages a submission queue and several backends. A backend provides syntax checking and testing for a specific programming language. Currently implemented are backends for Haskell, Scheme, Erlang, Prolog, Python, and Java. However, with the appropriate backends, the system can also be used to check submission in other formal notations or to analyze natural-language assignments (we have already experimented with style checking and keyword spotting, (cf. Feustel, 2006)) to help with the assessment of large numbers of essay-like submissions.

ECAutoAssessmentBox allows students to submit their solutions for programming assignments via the Web at any time during the submission period. Submitted programs are automatically checked and students get immediate feedback on whether their programs are syntactically correct and yield the expected results, as shown in figure 3.

4. EVALUATION

We have evaluated the eduComponents and the approach they realize by using them in our own courses. Since winter semester 2003/2004 we have been gathering experiences with eduComponents for online multiple-choice

Figure 3. ECAutoAssessmentBox automatically checks submissions to programming exercises and immediately offers feedback (student view).

tests, electronic submission of assignments and automatic testing of programs in our exercise courses. During winter semester 2006/2007 our learning environment was used by over 200 students at our institution.

Comments from teachers are very positive. We have also been actively gathering feedback from students. Since winter semester 2005/2006 we have been conducting surveys among our students: At the end of each semester we ask them to fill out a questionnaire on their experience with the learning environment. The questions cover three areas: The use of electronic submissions in general, their effect on the students' working habits, and the usability of eduComponents. The results in all three areas have consistently been very positive. For example, 87.1% of our students would like to see the system used in their other courses as well; 61.3% of them said that they worked more diligently on their assignments (probably because all assignments can now be easily reviewed by the teacher), and 92.6% of the students found that the statistics features are helpful for getting an overview of their coursework. Due to space constraints we cannot give a detailed report of the results in this paper.

The most important aspect is perhaps the use of the eduComponents by others. The eduComponents modules are open-source software and can be freely downloaded and used. The use of eduComponents by numerous institutions world-wide is, in our opinion, a good indicator that our approach provides a viable alternative to typical learning management systems.

5. CONCLUSION

We have presented a component-based e-learning system architecture, realized as software components which extend a general-purpose CMS with educational content types for courses, tests, essay-like assignments, and automatically tested programming assignments.

While computer-aided assistance in the areas of instruction, feedback, and student tracking is not new, the eduComponents are distinguished by their component-based and document-oriented approach based on a general-purpose CMS. The eduComponents are implemented as extension modules for the Plone CMS. The individual modules can be used on their own, together, and in conjunction with other Plone products. Since Plone is a general-purpose (i.e., not e-learning-specific) CMS, it can be used to manage *any* Web content, thus offering uniform usage and administration—unlike typical LMS's, which can only be used for learning content.

Also, since the e-learning objects are documents just like other objects in Plone, they all offer the same basic functionality and user interface. The eduComponents enable a low-threshold entry to e-learning since one can start with just a single module. However, as needs become more elaborate, customized e-learning environments can be created that closely fit the actual requirements. The CMS also serves as *item bank*, a central repository of tests and assignment, enabling the reuse of teaching and learning materials.

In our experience, the component-based, document-oriented approach on the basis of a CMS has proven to be a good decision, resulting in robust and easy-to-manage products, and we can use a single system for e-learning and other Web content. Even though the eduComponents certainly do not yet implement all features of larger (and older) systems, we have also noticed that by using the CMS infrastructure a small number of components is sufficient to cover the most frequently used e-learning functions.

We are currently developing further components, for example, a peer-review component, which will offer the distribution of anonymized submissions for peer review by other students. In computer science education, Zeller (2000) and Sitthiworachart and Joy (2004) report noticeable benefits for the learning of programming, and the seamless integration into the eduComponents will enable teachers to use this teaching device more often.

The eduComponents modules are freely available as open-source software licensed under the terms of the GNU Public License from `http://wwwai.cs.uni-magdeburg.de/software/`.

References

Amelung, M., et al., 2006. EduComponents: Experiences in E-Assessment in Computer Science Education. *In ITiCSE '06: Proceedings of the 11th annual conference on Innovation and technology in computer science education*. ACM Press, New York, NY, USA, pp. 88–92.

Baumgartner, P., 2003. *Förderprogramm Neue Medien in der Bildung: Audit-Bericht des Experten/innen-Teams unter Vorsitz von Prof. Dr. Peter Baumgartner*. Tech. rep., DLR Projektträger Neue Medien in der Bildung und Fachinformation, St. Augustin. URL `http://www.dlr.de/pt_nmb/Foerderung/Bekanntmachungen/Audit_Bericht_2003.pdf`.

Ellström, P.-E., 1983. Four faces of educational organizations. *In Higher Education*, vol. 12, no. 2, pp. 231–241.

Feustel, T., 2006. *Analyse von Texteingaben in einem CAA-Werkzeug zur elektronischen Einreichung und Auswertung von Aufgaben*. Master's thesis, Fakultät für Informatik, Otto-von-Guericke-Universität, Magdeburg.

Forsythe, G. E. and Wirth, N., 1965. Automatic grading programs. *In Commun. ACM*, vol. 8, no. 5, pp. 275–278.

Grob, H. L., et al., 2004. Developing, Deploying, Using and Evaluating an Open Source Learning Management System. *In Journal of Computing and Information Technology*, vol. 12, no. 2, pp. 127–134. URL `http://cit.zesoi.fer.hr/downloadPaper.php?paper=555`.

IMS Global Learning Consortium, 2005. *IMS Question and Test Interoperability Version 2.0 Final Specification*. IMS Global Learning Consortium. URL `http://www.imsglobal.org/question/`.

Kleimann, B. and Wannemacher, K., 2003. *Nachhaltigkeitsstrategien für E-Learning im Hochschulbereich*. HIS-Kurzinformation B3/2003, Hochschul-Informations-System, Hannover.

Kleimann, B. and Wannemacher, K., 2005. *E-Learning-Strategien deutscher Universitäten: Fallbeispiele aus der Hochschulpraxis*. HIS-Kurzinformation B4/2005, Hochschul-Informations-System, Hannover.

Kubicek, H., et al., 2004. *Organisatorische Einbettung von E-Learning an deutschen Hochschulen*. Tech. rep., Institut für Informationsmanagement, Bremen. URL http://www.ifib.de/publikationsdateien/MMKH_Endbericht_2004-05-26.pdf.

Newton, R., 2003. Staff attitudes to the development and delivery of e-learning. *In New Library World*, vol. 104, no. 10, pp. 412–425.

Piotrowski, M. and Rösner, D., 2005. Integration von E-Assessment und Content-Management. *In* Haake, J. M., et al., editors, *DeLFI2005: 3. Deutsche e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.*, vol. P-66 of *Lecture Notes in Informatics (LNI)*. GI-Verlag, Bonn, pp. 129–140.

PLS RAMBØLL, 2004. *Studies in the Context of the E-learning Initiative: Virtual Models of European Universities (Lot 1)*. Draft final report to the EU Commission, DG Education & Culture, PLS RAMBØLL Management A/S, Århus. URL http://www.elearningeuropa.info/extras/pdf/virtual_models.pdf.

Rösner, D., et al., 2006. E-Learning-Komponenten zur Intensivierung der Übungen in der Informatik-Lehre – ein Erfahrungsbericht. *In* Forbrig, P., et al., editors, *2. GI-Fachtagung Hochschuldidaktik der Informatik*, vol. P-100 of *Lecture Notes in Informatics (LNI)*. GI-Verlag, Bonn, pp. 89–102.

Sitthiworachart, J. and Joy, M., 2004. Effective peer assessment for learning computer programming. *In ITiCSE '04: Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education*. ACM Press, New York, NY, USA, pp. 122–126.

Zeller, A., 2000. Making Students Read and Review Code. *In ITiCSE '00: Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education*. ACM Press, New York, NY, USA, pp. 89–92.