

An E-Learning Platform with a Deliberately Simple Design

Eveline Riedling¹ and Karl Riedling²

¹ TELAB GesbR, Vienna, Austria
eriedling@telab.at

² Institute of Sensor and Actuator Systems,
TU Vienna, Austria
karl.riedling@tuwien.ac.at

E-Learning systems have become increasingly complex; too complex for some applications and user groups. This paper demonstrates an intentionally simple platform for content presentation, which, in contrast to competing much more powerful E-Learning systems, was happily accepted by the teachers and students at an Austrian academy for continuing education of nurses, and has been in use very successfully over a period of four years by now.

Introduction

The complexity of E-Learning systems has greatly increased over time; systems now usually offer a wealth of features and functionality. However, the increased functionality of modern E-Learning platforms inevitably makes them more complex to handle, and requires more and more specific knowledge on part of the users. Some user groups flatly refuse to work with such systems: Particularly in areas that are far away from IT, e.g., in nursing, many teachers and their students consider a full-blown E-Learning system a burden rather than support to their teaching and studying, respectively. In organizations where there is no staff specifically assigned to preparing contents for use in an E-Learning system, the teachers themselves would have to do it. However, time constraints frequently make it impossible for a teacher to learn how to operate complex E-Learning platforms, and to enter contents there. Likewise, too complex E-Learning systems often impose barriers to students, who may use them, in the best case, only reluctantly because they find them too complicated.

An E-Learning Platform – Developers' and Users' View

The E-Learning platform described here dates back to 2001, when Doris Gross, a teacher at the *Akademie für Fortbildungen und Sonderausbildungen* (Academy for Continuing Education for Nurses) in Vienna, Austria, contacted the authors at the VIEWDET conferenceⁱ, expressing her need for easy to handle E-Learning software. She identified the two user groups, teachers and their students (in her case, nurses), and their respective reasons for their resistance against E-Learning:

Teachers at the Academy wanted to distribute course contents conveniently, also from at home; mainly text and pictures. They wanted to concentrate on the course

ⁱ VIEWDET – Vienna International Conference on Distance Education and Training, organized by the authors from 1996 to 2003.

contents rather than on technological features. Some teachers already knew state-of-the-art E-Learning platforms, but were not willing to use them. They claimed there were too many features and confusing options for handling E-Learning situations. They simply wanted a facility to supply documents to their students in electronic form, and with a minimum overhead.

Students (i.e., nurses) should get the chance of barrier-free access to learning material, whenever and wherever they wanted. Since the students were already working as nurses, they needed access to the documents at any time of the day, e.g., from at home or during night shifts while there was no urgent work to do. This implied a Web-based E-Learning platform design allowing access via common web browsers. Retrieving the course material should be fast and reliable in order to keep connection costs as low as possible.

Both groups opposed advanced E-Learning systems. They were not even interested in the use of advanced features recommended by E-Learning experts, e.g., course maintenance, collecting students' work, grading students' papers, and publishing students' progress.

We then developed, together with Doris Gross, a concept for gently bringing users to E-Learning, designing the most user-friendly and easy platform we could think of. At this time, the first author was working as an internationally recognized E-Learning expert. To her, the concept of E-Learning involved advanced programming of advanced features, courses following cognitive concepts, and many varieties of course presentation, e.g., simulations. It was an adventure to go "back to the roots" of E-Learning, anticipating the negative reactions of colleagues who rather tend to educate the users than to change or omit features of their platforms. However, our users obviously did not want technological education, and did not want to learn much about using a platform; we had to accept those limitations. It was, however, not too pleasant to foresee unfriendly reactions of the scientific E-Learning community, which promoted advanced systems and had been developing more and more advanced systems for years. Therefore, we concentrated on our users and on offering them the simplest features possible:

- Fast, easy, and therefore low cost access to course material;
- Fast contact to a discussion forum; and of course
- Very convenient uploading features for teachers.

With this concept, we intended to supply an easily acceptable entrance to E-Learning. We wanted our users to think about E-Learning as a quite "normal" way of learning – and, indeed, we have won! Due to its inherent simplicity, teachers and students accepted the system very well; it has been in continuous use since 2002.

The following chapters describe the platform in its current state, a few years after the first small version. Although it now presents a few additional features, it is definitely still very easy to use.

System Concept

Free accessibility to the E-Learning platform from any arbitrary computer dictated a Web-based solution. We chose a LAMP (Linux – Apache – MySQL – PHP) concept for the realization of the system: All components in this concept are, on principle, free software, and the performance possible with such a system is by far sufficient for the

purpose in mind. (In fact, the system was running in full operation on a retired desktop PC for years, essentially without performance problems.)

Like all systems of this kind, the E-Learning platform consists of an administration module and an output module. Access to the administration module is strictly authenticated with individual user name / password combinations. The master administrators may assign to users, generally teachers, rights to specific administration operations on an individual base. Some output pages are freely accessible; these pages hold general information on the platform and on courses offered. Although the documents presented to the students are more or less published work that is essentially in the public domain, protection of the intellectual property of the teachers and some unpublished documents demanded that access to the documents should require authentication with a password. The pages that hold the course contents proper – links to uploaded documents and to related web sites, and user forums pertaining to the course – are therefore restricted to authenticated users. For ease of maintenance, the Academy demanded a rather weak authentication scheme with a course-specific password known to all students in a course. The weakness of this authentication method does not constitute a problem here: There are no real secrets to protect. The initial versions of the platform used JavaScript for authentication within the output pages; for various reasons, we later changed this to a slightly more secure cookie-based approach, which also allows a time-out after prolonged periods of inactivity.

Since the Academy offers a number of different courses to its students, the platform presents contents on a course basis. Some of the courses are not repeated regularly; hence, the number of courses held in the platform tends to grow. To preserve clearness, we introduced the concept of course groups that hold related courses. The administration module allows the definition of course groups and the assignment of courses to course groups.

Courses may comprise the following items of information, which can be defined and modified in the administration module:

- General information on the course (optional; freely accessible);
- Uploaded documents;
- Links to related Web sites; and
- An arbitrary number of forums for teacher-student interaction.

There is no limit to the number of uploaded documents and links in a course.

The administration module also allows the management of non-course specific information in a news board and an FAQ list. The platform features a mail form for user feedback; the recipients of such mails can be dynamically defined in a function of the administration module. Actually, there are two mailing lists for technical and administrative inquiries, respectively; mail addresses may belong to either or both lists.

A design requirement for the platform was that access to uploaded documents should only be possible for course participants. This not only requires user authentication, it also prohibits the use of a directory for uploaded files somewhere in the document tree of the web server. The first version of the platform therefore stored the uploaded documents as BLOBs (Binary Large Objects) in a database table, and explicitly sent them to the client browser upon an authenticated request. Since the average size of the uploaded documents is close to 1 megabyte, the size of a table holding 1000 documents plus additional information, and hence the size of the MySQL table file, is about 1 gigabyte. (Currently, the number of documents uploaded to the platform al-

ready exceeds 1000.) The ext2 Linux file system commonly in use a few years ago limited the size of files to 2 gigabytes, which imposed a hard limit to the possible number of documents that the platform could hold. Furthermore, database tables of that size are difficult to maintain and even more difficult to backup: In the default table type of MySQL (MyISAM), changes to the table contents are not flushed to disk immediately but kept in memory for relatively long periods. However, the table files are marked as “dirty” during this time. Restoring table files that were written to a backup in a potentially “unclean” state may render them useless. Of course, this can be overcome by executing a MySQL “flush” command prior to a backup, which, however, system administrators frequently forget, and which may not be effective if there is database activity between the “flush” command and the actual writing of the tables. On the other hand, an ASCII table dump, which properly works regardless of the state of the table files, of one gigabyte of binary data, which would have to be represented in hexadecimal notation, is not very practical. Therefore, we decided to move the document files into a directory outside the Web server’s document path, and to copy them to the HTML output upon receipt of a properly authenticated request. In this case, the number and size of the documents the platform can hold is only restricted by the disk size and by file system limitations, which are far beyond the limitations the database table solution would have imposed. The database now holds only administrative information and references to the uploaded documents. During upload and download, documents are strictly treated as binary data; hence, their actual file type does not matter.

The preparation of binary contents for HTML output is very similar, regardless of whether they are stored in a database table or taken from a disk file: The PHP code required for this purpose must create the proper HTML headers, particularly, specify the proper MIME file type, and then copy the document contents to the web server output. The original version of the platform derived the MIME type from the file name extension; later, the MIME type specified by the client browser during file upload was stored in the database table and used preferably. (Relying on the file name extension proved to be dangerous: From some platforms where, in contrast to *Microsoft Windows*, file name extensions are not mandatory, teachers uploaded documents without file name extensions that simply were useless in a *Windows* environment.) Aside from the fact that some browsers ignore the original name of an uploaded file, which the platform also provides in the PHP generated HTML headers, files presented this way appear as they would if directly accessed via an HTML link.

In addition to uploading documents, and deleting uploaded documents, the administration module allows shifting uploaded documents between courses. To preserve clearness, courses must reside in the same course group to allow the exchange of documents. This facilitates, e.g., dividing courses with a very large number of uploaded documents into sub-sections with shorter and therefore clearer lists of documents.

Several counters built into the platform register the accesses to courses and documents, respectively. Teachers can inspect the number of logins to their courses, and the download counts for each document. This provides some kind of feedback for the teachers who easily can recognize what their students consider interesting and what not. Regular users of the administration module see only their own courses and their access counts, and can only modify their own courses and their contents. A special administrator right permits access to all courses, regardless of their owner.

The platform also features a forum for the communication between teachers and their students. Teachers may define one or several forums for each course. As usual, the

administration program allows deleting inappropriate postings; there are intentionally no provisions for editing the text of the postings.

The output module features a two-level selection of course groups and courses, respectively; optionally, users may display a list of all courses regardless of their association to a course group. When a user selects a course the first time, the platform requests the course password for authentication. Upon correct entry of the password, the client browser receives a cookie with session lifetime that identifies the selected course and holds an expiration timestamp and an md5 hash of the timestamp plus a secret “magic number”. This approach makes forging an authentication cookie close to impossible. Each time the user requests a protected course page or a document associated with the course, the cookie is renewed with an updated expiration timestamp. The platform displays a login dialog again if either the cookie has expired according to its timestamp (after one hour of inactivity), or the client browser could not present the proper cookie. This approach is necessary to make allowance for over- rigorously configured browsers that do not accept cookies. Particularly in a hospital environment, where a high level of security is required, the scenario of browsers set not to accept cookies needs consideration. Since the names of the authentication cookies relate to the chosen courses, several course cookies can co-exist at a given time. It is therefore possible to look up documents belonging to several courses in any order, without the need to re-enter the password upon return to a previously visited course, unless the cookie has expired meanwhile.

The users enthusiastically welcomed even seemingly unimportant modifications: Sorting the list of documents for a course in inverse chronological order rather than chronologically renders newly added documents at the most visible place on top of the list. In the administration module, the individual course passwords, which allow access to the course information in the output module, are displayed as tool tips when the mouse cursor hovers over a course title; this spares the teachers maintaining separate lists of their course passwords.

Practical Experiences

We introduced the first version of the platform in 2002, and teachers and students accepted it immediately. The main reason for its acceptance was that it was very straightforward to use even for persons who are only occasional computer users.

With one exception, the only technical problems encountered to date resulted from configuration problems of the proxy and firewall servers that separate the network of Viennese hospitals from the internet. The only exception was due to an error in the configuration of the MySQL tables while the document contents still resided in a database table. This configuration problem caused an almost total blockage of the server when a large number of database requests came in and the server had to re-read the database file repeatedly, which then already had a size of 140 megabytes.

Some design limitations resulted from the facts that many of the computers from which users access the platform are located in hospitals and other health-care institutions, where security settings for browsers more rigorous than usual are likely, and that the intended users hardly can be expected to perform complex configuration adjustments of their browsers. The software has to work properly and browser-independently on out-of-the-box configured systems, and it has to handle possibly increased security settings gracefully. Neither the original JavaScript authentication

scheme for the output pages nor the later cookie-based approach posed problems worth mentioning, though.

The platform does not impose restrictions on the file types of uploaded documents, and teachers take full advantage of this fact, which allows them to upload files simply in the format at hand. Hence, the majority of uploaded files (36%) are in *Microsoft Word* format, followed by 28% PDF and 18% *Microsoft PowerPoint* files.

Recent statistics show close to 600 course logins and more than 2100 document downloads per month, for a monthly average of 15 courses with a total of 150 to 200 participants. Although older courses and about 1000 pertinent documents have meanwhile been removed, more than 1000 uploaded documents for 35 courses in 11 groups are currently available.

Conclusion

For the E-Learning platform presented, there was never the intention to provide the entire course contents in electronic form, only documents supporting these contents. Likewise, no additional services such as testing or grading of students were required. By deliberately designing a very simple, easy-to-use system for providing documents as a support for courses for a specific user group, we could obtain a very high level of acceptance by the users.

Acknowledgement

The authors want to thank Doris Gross of the *Akademie für Fortbildungen und Sonderausbildungen* (Academy for Continuing Education for Nurses) in Vienna, who initiated the project, for her valuable input concerning the expectations of the users at her institution in an E-Learning environment. We further thank Veronika Maier-Wagrandl, head of the Academy, for making this project possible.