The Learn@WU Learning Environment[1]

Georg Alberer, Peter Alberer, Thomas Enzi, Günther Ernst, Karin Mayrhofer, Gustaf Neumann, Robert Rieder, Bernd Simon

Vienna University of Economics and Business Administration

Department of Information Systems, New Media

Augasse 2-6, A-1090 Vienna, Austria

Abstract: This paper is an experience report describing the Learn@WU learning environment, which to our knowledge is one of the largest operational learning environments in higher education. The platform, which is entirely based on open source components, enjoys heavy usage by about 4,000 WU freshmen each year. At present, the learning environment supports 18 high-enrollment courses and contains more than 20,000 – mainly interactive – learning resources. The learning platform has reached a high level of user adoption and is among the top 20 web sites in Austria. This paper describes the overall project as well as the design principles and basic components of the platform. Finally, user adoption figures are presented and discussed.

Keywords: E-Learning, Learning Environment, Learning Management, Experience Report

1 Introduction

The Vienna University of Economics and Business Administration (German: *Wirtschaftsuniversität Wien*, abbreviated WU) is one of the largest business schools worldwide. It has a total of more than 25,000 students, and more than 4,000 freshmen enrolled for the academic year starting in October 2002. Each semester, approximately 2,000 courses are offered at the WU.

In the fall of 2002/2003, six new degree programs (Business Administration, International Business Administration, Economics, Business and Economics, Business Education and Information Systems) were introduced in parallel to replace the degree programs taught to date. The new degree programs share 80%

---

[1] To be presented at "Wirtschaftsinformatik 2003", 6th International Conference on Business Informatics, September 17th to 19th, Dresden, Germany 2003.

of their courses in the first year, in order to allow students to switch between the different degree programs more easily. 20% of the courses are specific to each degree program, meaning that beginners are confronted with subjects typical of these programs (e.g., programming and modeling courses in the Information Systems (IS) program).

Access to universities is practically free in Austria. While there is a tuition fee of approximately 360 euros per semester, no general entrance examination or similar selection mechanisms are in place. As a consequence, the university has traditionally seen high initial enrollment, with many students failing to pass the courses of the first year. The degree programs are designed in such a way that the first year is taught mainly in "mass courses" (18 courses with a maximum of 600 students per class), while the second and later years are taught in group sizes which allow regular interaction between faculty and students (target size: approximately 30 students per class). The idea was to concentrate drop-outs in the first year in order to focus teaching resources on students who are actually likely to finish a degree program.

The Learn@WU project was launched in order to ease the learning situation for freshmen. The main objective of this project was to provide an electronic learning environment for students and faculty in order to support the mass courses. However, it was not our goal to introduce a new distance education program (and to attract even more students). The government-funded project started in the fall of 2001, when 40 full-time project staff were recruited (36 content developers, 2 people for didactic support, 2 for technical support) for a period of two years. The content developers are primarily experts from the fields of marketing, public law, mathematics, etc., who either developed completely new materials or transformed pre-existing materials into learning resources suitable for the electronic learning environment.

This paper is structured as follows: Section 2 describes the design of the learning environment, provides an introduction to the open source components used and presents the content model implemented, which is referred to as the concept space. In addition, the functionality supporting learning communities is sketched and Learn@WU's authoring tools are described. Section 3 presents usage statistics for the learning environment and draws a comparison between these statistics and pass/fail rates. Section 4 lists related work, and Section 5 addresses future research and development activities.

2 Designing the Learning Environment

As in any other learning environment, learning content is a key element of Learn@WU. *Learning content* refers to any kind of information provided by the content providers which can be used to facilitate knowledge acquisition on the part of the learners. Another key component is the platform itself. The *platform* – also referred to as learning management system – hosts, organizes and controls access to the content. I also provides communication facilities (in our case mostly learner-to-leamer interaction). *Content development tools* – also referred to as authoring tools – interface with the platform and allow content developers to publish educational content in the learning environment.

In Section 2.1, we present the processes of eliciting requirements and selecting the platform, after which we describe the platform itself and its components (Section 2.2). In Section 3, we proceed with an overview of the central structuring components of the content model (Section 2.3.1), the supported learning content types and the content development tool (Section 2.3.3). The chapter ends with a description of our personalization and community services in Section 2.4.

2.1 Eliciting Requirements and Selecting the Platform

It was clear from the beginning that the didactic design of learning materials had to be rather open, since learning content differs substantially in fields such as Mathematics, Statistics, Information Systems, English, Marketing, Human Resources, Law and Economics. Some courses focus on hard facts to be learned, some try to cover complex concepts in context, and others focus more on skills. Moreover, some professors prefer to keep the traditional classes and provide additional materials through the platform, while others place emphasis on diminishing the need for student presence in the classroom. Others have changed the nature of their classes from full coverage of all relevant facts to a more detailed discussion of complex issues, leaving the simpler issues for independent study.

For the purpose of eliciting requirements, the professors and content developers were asked to respond to a questionnaire-based survey consisting mainly of open questions. The questionnaire yielded somewhat surprising results: The professors responsible for the courses showed great interest in features such as self-assessment exercises (94%), downloads of course materials (88%), sample exams online (71%) and electronic text books (47%), but little interest in community and collaboration support (24%). Practically no one was interested in the functionality offered by most off-the-shelf learning management systems, such as learner registration, curriculum management, shared workspaces, online learner-instructor interaction, etc. Furthermore, the expectations of the self-assessment functionality were highly detailed and yielded six different types of quizzes.

It soon became apparent that none of the standard solutions available would provide a perfect fit for the situation at our university, since we required the following:

- Kerberos support for authentication (every student enrolling at the university automatically receives a Kerberos account for access to all of the university's computing resources);
- Integration with a heterogeneous legacy system for the university's administration;
- Focus on a content model rather than a course management model;
- Complex rights management: multiple departments developing materials for a single course held in several class sessions by internal and external lecturers;
- Large transaction volumes: We expected sustained transaction volumes in the approximate range of 15 web server requests per second.

We evaluated WebCT, Blackboard, Hyperwave eLearning Suite, Orbis Net Coach, Ingenio Learning Platform (which recently filed for bankruptcy) and the OpenACS open source community framework.

We decided for the OpenACS open source framework mainly for the following reasons:

- The project was not equipped with a large budget for software acquisition and could not commit itself to long-term payments to software vendors.
- At the time the decision was made, some LMS vendors had already started to disappear from the market. On the one hand, it was feared that a relationship with one vendor would not prove to be of a long-term nature because of the vendors' inability to stay in the market. On the other hand, the active open source development community of OpenACS was very appealing.
- Another influencing factor was the possibility of development sharing with MIT's Sloan School of Management, which had decided to develop its next generation e-learning platform .LRN [LRN03a]. In the context of the SloanSpace project, MIT is introducing a course management and community-building system on the basis of OpenACS.
- Scalability was a major issue from the very outset, and OpenACS promised to fulfill this requirement while keeping hardware costs low.
- A number of interfaces to legacy systems needed to be implemented and maintained. Such tasks can be performed more easily with white-box components for which the source code is available. At the same time, many vendors were unable to provide a convincing solution for our demands.

- The team had proven expertise in managing, adapting and developing open source solutions. Last but not least, we had had positive experiences with OpenACS in the past.

2.2 Open Source Components of the Platform

The OpenACS platform is based on the Ars Digita Community System (ACS), which was developed in 1996 by a team led by Phil Greenspun at MIT [Gree99a]. ACS originally started as a university project. After some early success in commercial projects, the company Ars Digita was founded, which was acquired by Redhat Inc. in 2002. In parallel to the commercial product, a completely open source based system was developed under the name OpenACS. This software is currently available in Version 4.6 and can be used with both Oracle and Postgres relational database systems. The main components of OpenACS are a relational database, a web server (the AOLserver), a generic web programming framework and several application components.

The Learn@WU system uses the Postgres relational database system (Version 7.2.3) as its content repository. Postgres (which has been called PostgreSQL since 1996, when SQL support was added) has gained a reputation for being the most advanced open-source DBMS. It supports the core SQL99 specification and most of the SQL92 standard, including transactions, sub-selects, triggers, views, foreign keys, stored procedures, etc. Postgres can be used with many popular operating systems, including most Unix derivatives and the Microsoft Windows family.

The AOLserver was developed by NaviSoft (acquired by America Online in 1994) to run America Online's busiest sites, such as DigitalCity.com or aol.com. It has been reported that as early as mid-1999 multiple AOLserver instances were serving more than 28,000 requests per second for America Online [Gree99b]. The AOLserver is an application server which is written in C and contains an interpreter for the Tcl scripting language. Tcl is used as an extension language – sometimes called a "glue language" [Oust98] – to provide a flexible means of composing systems from predefined components (typically written in a "system language"). Several Tcl components (called OpenACS packages) can be plugged into the AOLserver, ranging from cryptographic libraries to object-oriented systems. Most of the components we have added were actually written in the object-oriented Tcl extension named XOTcl [NeZu00].

The AOLserver achieves its high performance through the extensive use of resource pooling: It maintains a thread pool for connection threads as well as a database connection pool to manage database connections. Typically, a multitude of database connections are kept open between the web server and the relational database. These connections are assigned to connection threads on demand. This pool of multiple database connections has the advantage that a single, slow and complex SQL query cannot block other requests.

Figure 1 shows the relationship between the main components of Learn@WU, including the AOLserver and the higher-level components. OpenACS consists of a core layer (the OpenACS Core Services) which provides functions such as user management, group management, permission checking and several components (called OpenACS packages) which typically handle various kinds of end-user services. The OpenACS distribution contains packages for discussion forums, personal file storage, a portal system, full text search, workflow, an e-commerce-shop and a variety of other functions.

The current version of Learn@WU uses the following OpenACS components: user management, content repository, news (a web log component called "lars blogger"), simple surveys and forums. All learning resources are kept in the content repository, which stores them in the relational database and handles access control and versioning. The versioning feature allows concurrent updates. For example, a content developer might upload a newer version of an examination, but students can continue to work on the version they have started. We added an authentication module to OpenACS to allow users (students, faculty, administrative personnel) to use the Kerberos authentication system deployed throughout the university (see Figure 1).

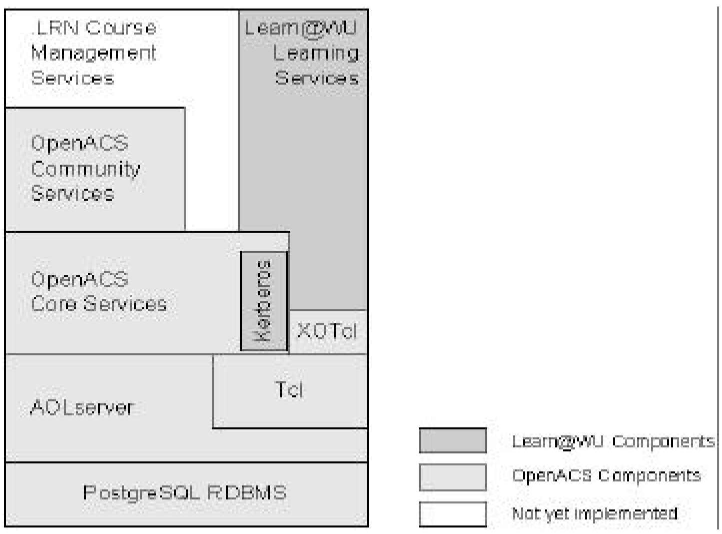

Figure 1: Main System Components of Learn@WU

Learn@WU's Learning Services comprise modules for managing the various kinds of content (see Sections 2.3 and 2.4) and for personalization and community purposes (see Section 2.5). The components described in the sections below were developed as OpenACS packages so that they can also be used in other OpenACS-based systems such as .LRN. Finally, it is worth noting that as most of the generic functionality was covered by standard OpenACS components, all technical support and development was provided by two full-time employees only.

2.3 The Content Model

The focus of the Learn@WU project is on content development. In order to achieve full coverage of the university courses included in the project, the system had to cover a great diversity of domain knowledge and support a large number of content developers (36 full-time equivalents), which required a certain amount of coordination. It was necessary to provide (a) uniform means of organizing and structuring the content, and (b) a rich set of learning content types to fit the diversity of the materials.

2.3.1 Content Structuring Components

Prior to the new degree programs, a number of different introductory courses were in existence that emphasized related topics in different ways. Since it is more feasible to support a common body of knowledge through an e-learning platform, several departments teamed up (e.g., three marketing departments) to develop common learning resources, with due attention to their core competences. In order to foster the idea of distributed content development, Learn@WU supports a *concept space,* which is in some respects similar to the catalog of the DMOZ open directory project [DMOZ03]. All learning materials provided on the platform are associated with hierarchically organized categories, where each course has its own top node and the concepts taught in this course form the lower nodes.

The concept space serves the following purposes:

1. For faculty, the concept space provides a means of organizing the content of their courses, ensuring transparency between different courses, and providing links between related or overlapping content in parallel or prerequisite courses.

2. For students, the concept space provides a means of locating the relevant resources for a lesson and its learning objectives easily. Students can find related resources for further reading, they can see their own learning progress in a learning map which shows their success rate and coverage at various aggregation levels in the concept space, and they can use it to find peers currently working on the same topics (see also Section 2.2).

One driving idea behind the concept space was to organize all full covered knowledge in a transparent manner, so that students would enjoy a uniform navigation structure, faculty could organize and cross-reference their materials (e.g., a marketing course can link to the relevant statistics materials), and courses in the higher semesters could depend on clear deliverables from earlier courses. In this context, it is important to note that the concept space does not aim to provide an ontology, since the depth of the concepts are course-specific.

For example, the concept *"Databases"* is covered in the *"Introduction to IS"* in far less detail than in the *"Database Systems"* course. The learning resources associated with nodes in the concept space are always suited to both the domain and depth of knowledge. This property has a positive side, as a content developer for a marketing course can link to the learning materials *"regression analysis"* taught in *"Statistics 1,"* for example. Figure 2 illustrates the organization of the common body of knowledge into a transparent learning space.

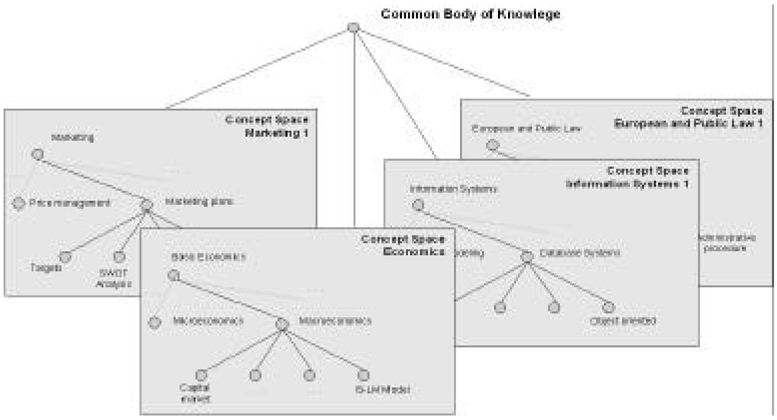

Figure 2: Organization of the Common Body of Knowledge into a transparent Learning Space

By assigning learning resources to the categories of the concept space (see Figure 3), all of the related resources (covering the same concept) can be presented automatically and without manual maintenance of such navigation structures. This is especially important when different content developers in other departments continue to extend the learning corpus.

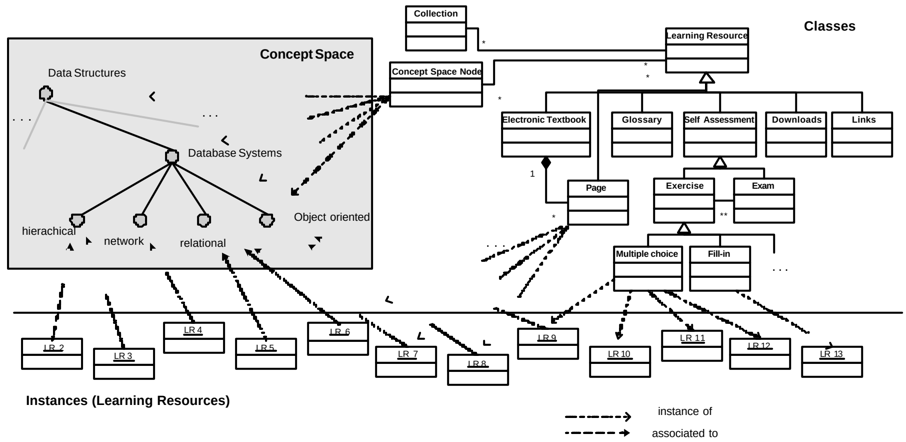

Figure 3: Concept Space and Learning Resources (excerpt)

The concept space is also used as the basis for the recommendation system, which is able to recommend learning resources independently of their learning resource type. For example, the same mechanism allows us to suggest related multiple-choice questions for a textbook page, or further exercises or glossary entries on the same topic for a multiple-choice question (see Figure 4).

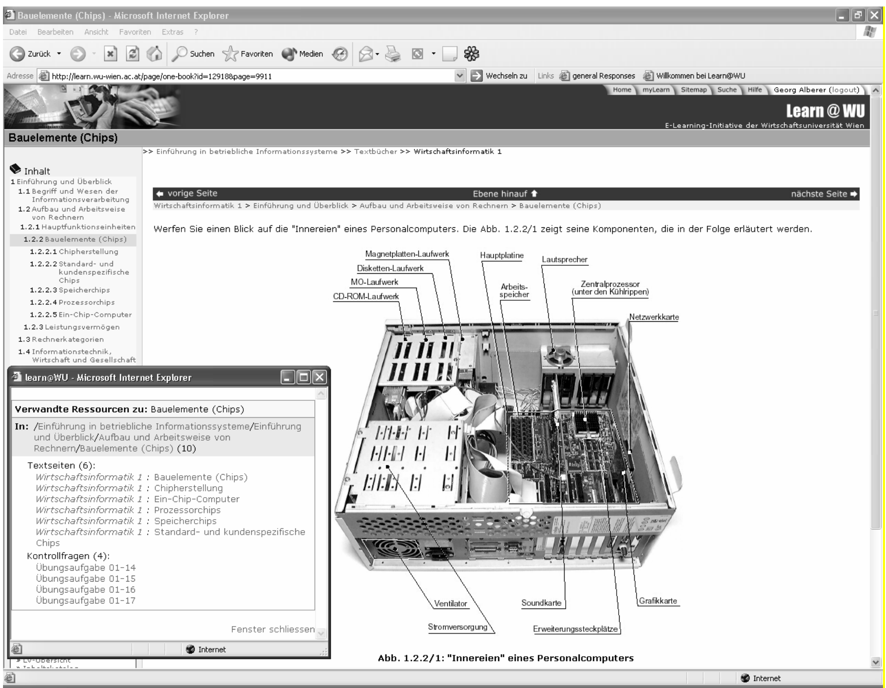

Figure 4: A Textbook Page with related Learning Resources

While the concept space is a good means of organizing learning resources in general, there are many situations in which resources have to be grouped according to arbitrary criteria as well. For example, for the same concept space there might be different slide sets used by different lecturers in various semesters, or only a subset of resources might be relevant to one lecture, etc. For such purposes, Learn@WU supports *resource collections*. In many respects, resource collections are similar to folders in a file system. They can contain arbitrary learning resources, nodes in the concept space as well as other resource collections.

2.3.2 Types of Learning Content

At present, Learn@WU supports the following types of content:

- **Self-assessment exercises:** These allow students to check their current level of knowledge. One exercise can contain several questions, enabling content developers to build small case-study-like exercises. The following question types were implemented in Learn@WU:
  - True/False questions
  - Multiple-choice questions
  - Fill-in questions
  - Matching questions
  - Ordering questions
  - Open questions (Open or essay questions cannot be graded automatically, but representative answers are provided).

  The option of specifying a problem text to explain the context and solution is available for all question types. The content developer can provide feedback at both the exercise and the individual question level. They can also provide a variety of optional meta-information, such as the degree of difficulty, allowed time frame, score, etc.
- **Sample exams** are used to simulate an exam situation. A sample exam contains an arbitrary set of exercises covering all topics in a course. A student can use a sample exam to test his/her readiness for the actual classroom exam. In order to simulate the exam situation, the system includes a timer and uses the same grading scheme and pass marks of the real-world exams. Online exams can also be printed as scrambled assignments for use in real exam situations.
- **Electronic textbooks** are the counterparts of printed textbooks and contain detailed information on each topic. An electronic textbook consists of chapters

and pages (see Figure 4 for an example of a textbook page). An electronic textbook can be used to provide the primary reading materials for a course and to insert links to internal and external resources.

- The **glossary** provides definitions for the most important terms used in a course. A definition can cover one or more terms. Each term can include an English translation, synonyms, antonyms, type information and abbreviations. The keywords contained in the definitions are linked to other glossary entries.
- **Links:** Learn@WU supports both internal and external links. Content developers can create a library of stand-alone links, which can be provided with various meta-information such as title, description and author. Links can also be associated to concept space nodes, a feature which supports the creation of link catalogs similar to Yahoo or DMOZ.
- **Downloads** refer to the various kinds of files available to the student. Typical examples include reading materials such as papers, sample program files or slide presentations.

2.3.3 Content Development Tools

XML was chosen as the primary exchange format for the platform. For all types of learning resources, XML schemata were developed in order to formally define the representation of XML document types. The XML documents contain the learning resources with all their metadata and can be imported into and exported out of the Learn@WU platform.

XML is thus also used as the output format for Learn@WU's authoring tools. The primary authoring tool in Learn@WU is Microsoft Word with a custom extension which converts special style sheets into XML schema instances. Word was chosen because most content developers were familiar with it, and because a large quantity of learning materials had already been created in that format.

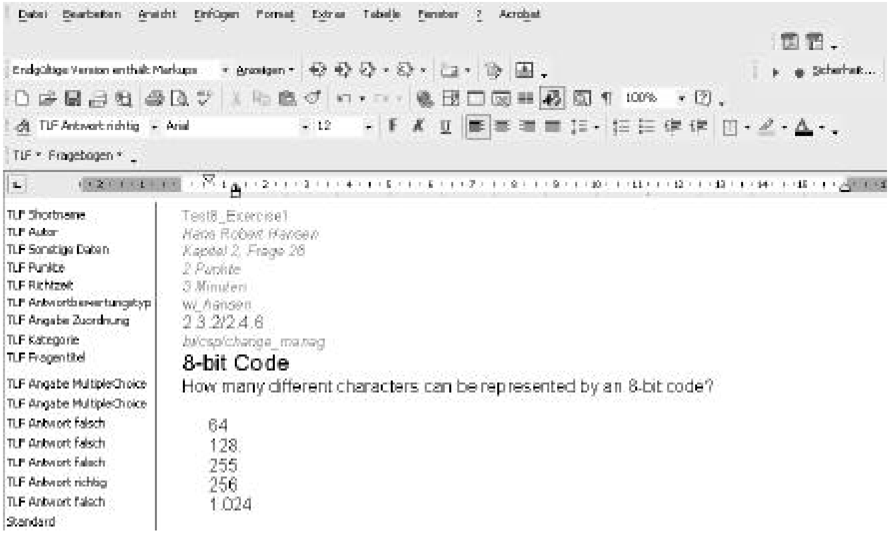

Figure 5: Microsoft Word interface for content developers

Figure 5 shows the Microsoft Word interface as made available to Learn@WU content developers. This example shows an exercise containing a multiple-choice question. The exercise is uniquely identified by a "short name," and it consists – like all other learning resources – of a block of metadata and a content block. The metadata block can contain (among other information) the author and title of the exercise, the suggested time for completing the exercise, the number of points, and the nodes of the concept space to which it is assigned. The content block can consist of a sequence of questions of the types mentioned in the previous section. In our example, the exercise contains a single multiple-choice question.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<learning_resources xmlns="http://lore.wu-wien.ac.at/TLFSchema">
  <exercise area="bi" restype="excs" shortname="Test8_Exercise1">
    <metadata>
      <author>Hans Robert Hansen</author>
      <time unit="minute">1</time>
      <points>2</points>
      <category area="Bi" restype="csp" shortname="change_manag" />
      <title>8-bit code</title>
    </metadata>
    <question>
      <multiplechoice presentation="scramble" assessment="wi_hansen">
        <problem_text>
          <p>How many different characters can be represented by an
          8-bit code?</p>
        </problem_text>
          <answer value="false"><answer_text><p>64</p></answer_text>
          </answer>
          <answer value="false"><answer_text><p>128</p></answer_text>
          </answer>
          <answer value="true"><answer_text><p>256</p></answer_text>
```

```
        </answer>
        <answer value="false"><answer_text><p>255</p></answer_text>
        </answer>
        <answer value="false"><answer_text><p>1.024</p></answer_text>
        </answer>
      </multiplechoice>
    </question >
  </exercise>
</learning_resources>
```

Figure 6: XML representation of a multiple-choice question

The Word style sheets for the various content types contain code for verifying the Word content, for help and support (e.g., in linking to the concept space or to other resources), and for generating the XML files. A template file which maps Word's text properties to XML elements controls the conversion of the Word documents. Once the Word document has been processed, the resulting XML instance and all included graphics are packed into a ZIP file, which can then be uploaded to the platform. The server-side application unpacks the ZIP file and processes it by updating various tables in the database.

Figure 7 shows the XML document from Figure 6 (which was generated from the Microsoft Word document in Figure 5) as it is displayed on a client's browser.

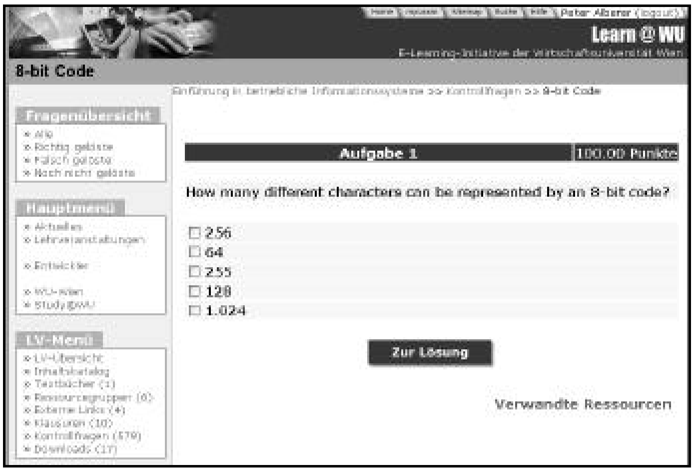

Figure 7: HTML representation of the multiple-choice question in Figure 5

2.4 Personalization and Community Services

In its current version, Learn@WU is used to a great degree as an interactive environment for self-assessment. Each learner can access his/her personal course coverage statistics, which allows them to monitor their learning progress in a particular course. Figure 8 shows the course coverage statistics of a learner studying "*Einführung in betriebliche Informationssysteme*" (Introduction to Information Systems), which displays the topics currently covered in the course along with the individual's success rate.

Figure 8: Personal Course Coverage Statistics of Learn@WU

In order to encourage collaboration among students, traditional forums and a peer-to-peer communication tool called "Buddy Finder" were also included in Learn@WU's learning services. Students working on certain topics can check whether one of their peers is currently working on the same topic. This information can be derived from the content activated by the students within a certain time frame, for example 10 minutes. The students can get in touch with each other in two ways:

1. Via e-mail: Right next to each topic, the "Buddy Finder" lists all the e-mail addresses of active students.

2. Via instant messaging systems: In their profiles, students can define the various instant messaging systems they use (Learn@WU supports ICQ, Windows Messenger, AOL Instant Messenger and Yahoo Messenger). An interface to these messaging services checks the availability of all active students in "their" instant messaging systems and shows this information to fellow students who wish to know who else is currently online.

3 Experience

Although Learn@WU is still in its development phase, it is regarded as a great success both from the student and faculty perspective. The usage statistics below demonstrate its high acceptance rates. Learn@WU is probably one of the most active learning environments in operation at universities worldwide. So far, we only know of one site at the University of Alberta which comprises four WebCT servers, supports a total of 1,500 courses and reaches approximately four times the weekly hit rate of Learn@WU [UA03]. However, WebCT is mostly a course management tool rather than a complete learning environment like the one presented here.

Another possible measure for the intensity of use is the number of visits per student and course per day. While Learn@WU sees about 4,000 visits/day for 18 courses (ratio: 222), SloanSpace (see Section 2.1) reported about 1050 visits/day for 87 courses at MIT's business school (ratio: 12) [LRN03b]. This shows that the number of visits per course and day is nearly 20 times higher for Learn@WU compared to SloanSpace.

3.1 Usage Statistics

As of May 2003, Learn@WU had more than 8,500 registered users, of whom about 4,000 use the system regularly (all first-year students). Usage patterns on the platform change substantially during the semester. At the beginning, students use Learn@WU to download educational material accompanying the courses they wish to take. This generates traffic of up to about 10 GB/day. During the semester, usage is rather low (300,000 requests per day). Toward the end of the semester, however, students prefer to use the self-assessment functionality of the platform to test their knowledge, creating intense interaction between the application and the web clients. During the two preparation weeks before the exams in April 2003, for example, 4,076 students completed more than 1.4 million exercises online (averaging over 340 per student). The server received more than 1.2 million requests (hits) per day at the peak, with a maximum of about 50 requests per second. Figure 9 shows the log-file statistics for the Learn@WU server in April 2003. The exams were taken between April 28 and May 2, which explains the usage peak immediately prior to those days.

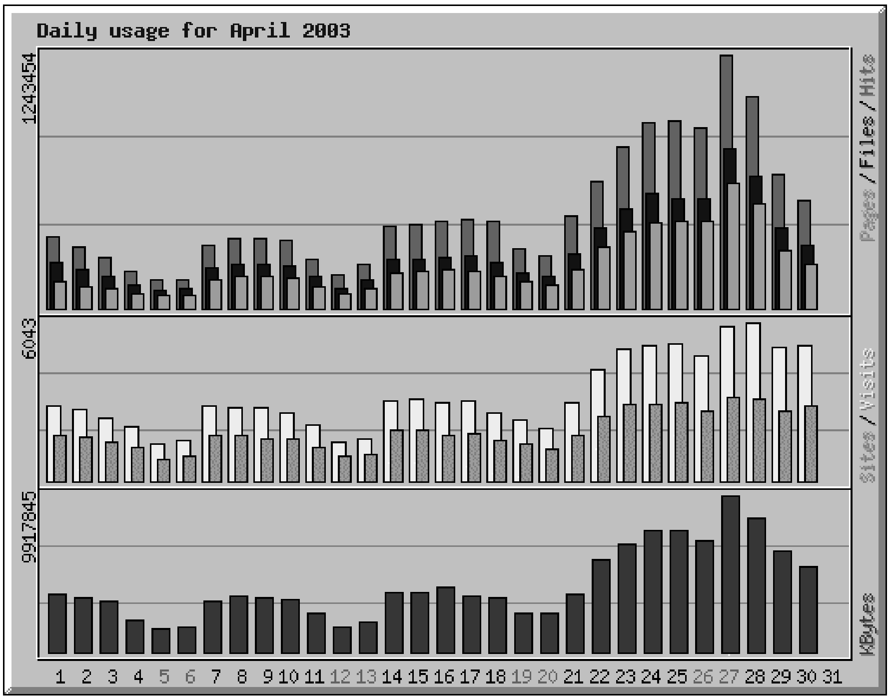

Figure 9: Log-file statistics for the Learn@WU server for April 2003

3.2 Learning Behavior

The question of how the use of the learning environment influences the students' learning results is more important than simple usage analysis. Naturally, this question is also far more difficult to answer, especially as not all exam results are available in the system. The best indicator of intense platform use is the number of self-assessment exercises completed by the students. It has to be noted that the number of online examples available differs significantly among the various courses. As the following table shows, the average number of self-assessment exercises completed is considerably higher among students with passing grades than among students with failing grades. One might conclude from this that intense use is an indicator for success. However, the same students might also have passed the exams without using the platform.

		Students with *passing* grades		Students with *failing* grades		
Subject	Number of exercises available	Avg. number of exercises completed	Percentage of exercises completed	Avg. number of exercises completed	Percentage of exercises completed	Ratio of exercise completion between passing/failing grades
Introduction to IS	579	813	140%	308	53%	2.64
Finance	136	265	195%	175	129%	1.51
Marketing	827	1795	217%	733	89%	2.45
Public Law	58	160	276%	119	205%	1.35
Human Resources	630	1272	202%	564	89%	2.26
Private Law	252	319	127%	242	96%	1.32
Sum / Average	2482	4625	186%	2141	86%	1.32

Table 1: Usage levels among students with passing and failing grades

Since most of the self-assessment exercises can be evaluated automatically, it is possible to compare the students' online success rate with their success rate in the final exams. We observed several indications that some students tried to train for the online exams by solving the same exercise multiple times and to memorizing the results. Our analysis shows that some students failed the exam despite a large number of trained exercises and an overall success rate of over 70% in the online exercises.

The current learning design of Learn@WU courses can still be improved further, as indicated by one of the last final exams in "Introduction to IS". The exam consisted of approximately 30 exercises. Some of the exercises were taken straight from the repository (which contains about 600 exercises on this subject) and some were modified slightly, but most exercises were new. About 30% of the students solved the exercises from the repository correctly, while only 15% solved the slightly modified exercises correctly. A comparable percentage of examinees solved the new exercises correctly.

These figures support the hypothesis that the current design of the learning environment rather supports training for the exam questions rather than studying the subject and understanding the underlying problems. This observation leaves much room for improvement to address this learning (mis-)behavior. One possible countermeasure is to prohibit frequent repetitions of the same exercise, or to formulate explicit learning objectives that have to be reached before additional exercises are presented to the students (see Section 5). The students' performance in the self-assessment tool could also be used to give them advice as to which areas of the learning materials should be recapitulated.

3.3 Forums

In March 2003, discussion forums were added to Learn@WU. Instructors can create discussion forums at the course, class session or topic level. The instructors are also responsible for content and for providing timely responses. In the first two months, 18 forums were created in 10 different courses, which gives us an initial idea of usage patterns: Student interest in posting questions or replies in forums seems to depend heavily on the kind of content taught in the different courses. Forums involving frequent calculations and similar topics (which make it easy to ask very precise questions) tend to attract many students, whereas areas which mainly deal with theoretical matters tend to be avoided. For example, finance has been the most popular forum with a total of 177 threads and 434 replies to date, while Public Law has only seen 2 threads and 1 reply.

Another experience was that when multiple forums were created for one course (e.g., one for each class session), only one of the forums was finally used by students. The preferred forum has always been the one with the first contributions, which suggests that students are more interested in contributing messages to an ongoing discussion than posting to a forum for a particular class session. Timely responses are a critical success factor for running these forums [Pic+01].

Previous studies have shown that organizational matters can sometimes take up most of the communication in distance learning forums [Simo01]. In Learn@WU, the forums are mostly concerned with content-related issues, but other unrelated issues were discussed as well (small talk between students). Some of the forums provide valuable feedback on the educational material, for example when students identify a flaw in an example or typographical errors in a text. The types of topics primarily discussed, as well as the predominant communication culture, are strongly influenced by the forum's moderators and vary considerably among the forums in Learn@WU.

4 Related Work

Learn@WU can be compared with similar projects, both on a learning environment level (for an introduction to different types of electronic learning environments in higher education see Simo02) and on a platform level. The OpenACS-based platform competes with other *open source-based learning management systems* such as ILIAS and EDUZOPE, and with commercial learning management systems such as WebCT, Blackboard, Clix and many others. This section focuses on open source learning management systems.

The open source learning management system **ILIAS** was developed by the VIRTUS project at the University of Cologne in 1999 [ILIA03]. Since October 2000, the source code has been freely available under a General Public License (GPL). In ILIAS, which stands for "*Integriertes Lern-, Informations- und Arbeits-kooperationsystem*," learning with the help of new internet-based technologies such as the web and e-mail is seen mainly as a self-controlled process. ILIAS thus focuses on an independent learner model, while Learn@WU also aims to support traditional modes of learner-instructor relationships. One important idea behind ILIAS is to help students in the learning process by providing guidance from a tutor. As with Learn@WU, learners should also be able to support each other by interacting in forums and chat rooms.

ILIAS distinguishes four different types of users: the simple user, who can view course materials; the tutor, who is mainly responsible for supporting the students in the learning process; the author, who provides content; and the administrator, who manages the entire system.

The content provided by the content developers forms what is referred to as the *backbone*, which is enriched by the system with automatic navigation elements, menus, search functions, etc. This automatic multimedia environment is called the *body*. Some parts of this *body* are created by default for every course, while other parts can be requested by the authors. The most important part of the learning content is a kind of textbook to which the other features of the system are linked.

Besides this textbook, the main features of the system are annotations and bookmarks which students can add to the content, as well as multiple-choice questions and search functions. Cooperative learning is supported by discussion forums and chat rooms.

The aim of the **EDUZOPE** project is to produce an open source authoring environment for learning materials [EDUZ03]. The learning management system component of EDUZOPE will be based on the ZOPE web application framework. ZOPE comprises an open source web application server specializing in content management, portals, and custom applications. While ZOPE is a widely deployed framework, EDUZOPE is still in its initial phases. Thus far, no prototype of education-specific ZOPE extensions has been released.

5 Conclusion and Future Work

This paper describes how an open source content management system, combined with powerful authoring, can provide a platform which forms the basis for a frequently used learning environment. Among the top 20 web sites in Austria in terms of hits and page views, the learning environment has proven itself as a successful tool for supporting learners as they prepare for exams. However, further research and development work has to be carried out in order to move away from a drill-and-practice learning design toward a content model that places stronger emphasis on conveying concepts. As we have noticed that students tend to learn "exercises" rather than "content," we will investigate means of sequencing the learning materials to a greater degree. This will involve a clear definition of learning milestones. which have to be achieved before other materials become accessible. Standards such as the sequencing definition model proposed by SCORM [ADL03] will also play an important role in this development.

In future development, we plan to integrate more functionality from .LRN into our system. At present, Learn@WU is mainly used to distribute digital learning resources for a rather small number (about 250) of mostly large class sessions in a few courses. Therefore, the functions of a course/class management system have not yet become essential. In the future, we will try to provide a knowledge corpus for courses in the higher semesters similar to the one developed for the first year. Furthermore, we plan to offer course/class management facilities to all instructors at our university. This will require a more structured course catalog with richer course management facilities.

Another important area of development will be the community aspect. The basic metaphor of OpenACS is a community. OpenACS can be seen as a platform for multiple communities that can organize their "sub-sites" and create their own sub-communities. Each community has members, manages resources and provides internal communication tools. A community administrator can decide to "mount" instances of a rich set of packages and tailor the community's portal. We will investigate the use of this metaphor to model (part of) the university's organizational structure, which can also be seen as a community of communities (e.g., academic and organizational departments, projects, classes, alumni). The goal is to provide an environment where the department's portals can be administered easily and with uniform navigation structures, at the same time providing some room for individualism and personal taste.

We also plan to release an open source version of our development.

References

[ADL03] Sharable Content Object Reference Model (SCORM) Version 1.2 - The Scorm Overview. Available at http://www.adlnet.org/index.cfm?fuseaction=scormabt.

[DMOZ03] The DMOZ Open Directory Project. Available at http://dmoz.org.

[EDUZ03] EDUZOPE. Available at http://www.eduzope.org/.

[Gree99a] Greenspun, P.: Philip and Alex's Guide to Web Publishing. Boston 1999: Morgan Kaufmann Publishers, also available at http://photo.net/wtr/thebook/.

[Gree99b] Greenspun, P.: Introduction to AOLserver. LinuxWorld, available at http://www.linuxworld.com/linuxworld/lw-1999-09/lw-09-aolserver_1.html.

[ILIA03] ILIAS. Available at http://www.ilias.uni-koeln.de/.

[LRN03a] .LRN. Available at http://dotlrn.mit.edu/.

[LRN03b] .LRN Usage Statistics. Available at http://dotlrn.mit.edu/features/powerpoints/prof-over.htm.

[NeZu00] Neumann, G.; Zdun, U.: XOTcl, an Object-Oriented Scripting Language. In: Proceedings of the 7th USENIX Tcl/Tk Conference. Austin, Texas, USA 2000:

[Oust98] Ousterhout, J. K.: Scripting: Higher Level Programming for the 21st Century. In: IEEE Computer. 31 (1998) 3, 23-30.

[Pic+01] Piccoli, G.; Ahmad, R.; Ives, B.: Web-based Virtual Learning Environments: A Research Framework and A Preliminary Assessment of Effectiveness in Basic IT Skills Training. In: MIS Quarterly. 25 (2001) 4, 401-426.

[Simo01] Simon, B.: E-Learning an Hochschulen: Erfolgsfaktoren und Gestaltungsräume von Wissensmedien. Lohmar, Köln 2001: Eul.

[Simo02] Simon, B.: Medien und Hochschulentwicklung. Wien, Duisburg 2002: Ed-Media Studienbrief, Gerhard-Mercator Universität Duisburg.

[UA03] University of Alberta - E-learning Services. Available at http://www.ualberta.ca/IDEAL/.