

Creating, Handling and Implementing E-learning Courses Using the Open Source Tools OLAT and eLML at the University of Zurich

Joël Fisler and Franziska Schneider¹

This paper presents two open source e-learning tools developed at the University of Zurich: OLAT (Online Learning And Training), a learning management system (LMS) used by universities worldwide to present and manage e-learning content and scenarios, to provide collaborative tools to student groups and to run assessments. And eLML (eLesson Markup Language), a self-contained tool used to create structured e-content which can be imported into OLAT. eLML was originally started in 2002 by the Swiss GIS-project "GITTA" and is now used by many universities in Switzerland, Germany and Austria. OLAT was originally started as a student project in 1999 and became the strategic LMS of the University of Zurich in 2003. One year later – after a complete redesign based on the programming language Java – OLAT was released under the Apache Open Source license and made available to a worldwide user community. Today OLAT is used not only by over 40'000 students at the University of Zurich, but OLAT servers are also running in nearly 40 countries and it has been translated to 30 languages. After an introduction, the authors present both eLML and OLAT, its features but also its limitations. The paper concludes with a discussion about open source software development and a short outlook.

Keywords: *Internet, Education, E-Learning, Teaching, Learning Management System, LMS, Java, XML*

I. INTRODUCTION

The Swiss University of Zurich has a worldwide reputation for its e-learning courses covering topics such as Geographic Information Systems GIS [1], Cartography [2], Urbanism [3], Psychopathology [4], Finance [5] and other fields of research. Less known are the tools used to create content, to manage, and to implement the e-learning courses. The strategic tools at the University of Zurich are eLML [6], an XML framework to create e-learning lessons, and OLAT (Online Learning And Training), a learning management system (LMS), used to set up and manage e-learning courses, provide collaborative tools to student groups and run e-assessments [7]. Both eLML and OLAT are open source projects used and developed by a worldwide community. Their development started at the University of Zurich in 1999 (OLAT) and 2002 (eLML).

Today e-learning content and e-learning course management are of high importance in higher education. The Bologna declaration demands that large student groups are assessed more frequently, that maximum mobility is offered to students, so that they can spend exchange semesters studying abroad and furthermore that university teaching

should focus more on electronic media, assessments and online courses. The University of Zurich developed the presented tools partly to satisfy the demands of the Bologna declaration. This paper will give a detailed record of other reasons why these projects were initiated, share the experiences made while developing and organizing the worldwide open source community, describe how e-assessments can be carried out, and offer an outlook on the planned functionality enhancements.

II. eLML: CREATING E-LEARNING CONTENT

The Canadian e-learning specialist Stephen Downes calls eLML "A first-class job, something some other organizations could learn from." and asks, "Why didn't IMS or ADL or anyone else come up with something like this?" [8]. The following pages should provide an insight into eLML and underline the statements with background information and facts about its usage.

eLML was developed by the Swiss e-learning project GITTA [1], a modular online course in Geographic Information Science and Technology. Within the GITTA project, nearly forty authors from ten partner universities created about fifty lessons and ten case studies. The heterogeneous and multilingual consortium needed strict pedagogical and technical guidelines to create consistent lessons with the same look and feel. After an extensive evaluation of existing tools, the consortium agreed in 2001 to use XML for the implementation and to base the XML structure on a pedagogical model. Thus, the lessons can be checked and validated for certain rules and restrictions by an XML schema and therefore all authors must create identically structured lessons.

The launch of eLML as a standalone open source project followed in 2004 [6]. Since then a constantly growing number of projects and authors in Switzerland, Germany and Austria have used eLML as their tool for creating e-learning lessons. At the University of Zurich eLML has become the main XML framework for creating and maintaining e-learning content and until 2009 over 500 lessons have been created. This guarantees further funding and development maintaining and enhancing eLML.

A. Structuring content based on pedagogical concepts

The aim of eLML was to offer authors a tool that ensured conformity to pedagogical guidelines and thus consistency throughout all the lessons of a project. These guidelines were adapted from the ECLASS model developed by Gerson [9]. ECLASS is an acronym for the terms entry, clarify, look, act, self-assessment and summary. Together with additional

¹ University of Zurich, IT-Services, Winterthurerstr. 190, 8057 Zurich, Switzerland (joel.fisler, franziska.schneider)@id.uzh.ch

important elements like glossary, bibliography or metadata, the ECLASS elements build the main structure of the XML framework. The different elements allow the creation of a pattern of learning experiences helping people to learn effectively and efficiently [10]. Although eLML is based on a strict schema, the structure is flexible enough to allow the creation of content for different e-learning scenarios. The GITTA project created both standard e-lessons [1] and case studies [11] but various projects have used eLML to create structured reports or implement other learning scenarios.

The described pedagogical model ECLASS is mapped onto an XML structure using XML Schema:

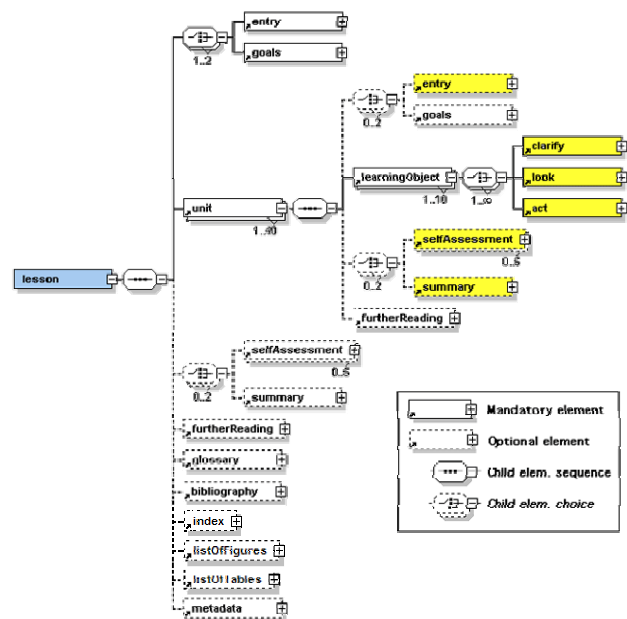


Figure 1: The top-level elements of eLML (yellow=ECLASS)

An eLML lesson always starts with either the mandatory introduction (element “entry”) or a concise listing of the lessons learning objectives (element “goals”) followed by units. The unit elements contain the actual content of a lesson. They are built up using the ECLASS model. A unit starts again with an introduction (entry) and the unit’s goals followed by various learning objects. Each learning object describes a certain concept, model, equation, term, or process using the three elements clarify (theory), look (example) and act (become active) in arbitrary order. A learning object typically fits on one or two screen pages and takes the student about five to ten minutes to go through. A unit ends with a self-assessment, to check if the students understood the goals of the unit and a summary plus an optional further reading list with relevant literature. Each lesson in eLML can have a glossary, index, bibliography, and automatically generated list of figures or list of table.

B. Presenting content in different output formats

There are two main reasons for using an XML framework like eLML: consistency – described in the last chapter – and flexible output possibilities, described in this chapter. The basic concept behind XML and thus also eLML is a strict separation between content and layout. Once a lesson is

available in the eLML format it can be transformed using standard technologies (XSLT) into different formats. eLML provides transformation files for the output formats listed in Table 1. Examples can be found either in Fisler [6] or on the eLML website.

TABLE 1: VARIOUS OUTPUT FORMATS AVAILABLE FOR eLML

<i>Output format</i>	<i>Technology used by eLML:</i>
Web/Online	(X)HTML
Learning Management System	SCORM or IMS CP
Print (PDF)	XSL-FO
Print (LaTeX)	LaTeX
Office Document	Open Doc. Format (ODF)
Reports	DocBook Format

The format itself can be adapted using eLML templates based on the XSLT technology. A designer can create a layout for a project, which then is used to transform the project’s lessons. Larger projects can create different eLML templates to ensure that each partner can present a lesson using its own corporate identity.

In 2008 the University of Zurich mentored a Google Summer of Code project [12] with the aim to release an “eLML template generator”. With this web based tool the creation of templates is feasible even for designers without any XSLT knowledge. The tool is available on the eLML website www.eLML.org for download.

C. Authoring tools to create and manage content

Currently most authors write eLML lessons using a standard XML editor. Having to deal with XML files as an author is a clear drawback of eLML. Following approaches have tried to eliminate this deficit:

1. Firedocs, a WYSIWYG eLML editor developed as a Firefox plugin and released in late 2008.
2. OpenOffice plugin to allow writing of lessons in a Word-like environment. This student project was realized in 2006.
3. An eLML extension for the Apache Lenya-based Content Management System (CMS) called “UniCMS” (www.unicms.uzh.ch) to import, export, manage and edit (using Firedocs) eLML lessons.

All three tools were originally developed at the University of Zurich and are now released as open source projects. The standalone eLML editor Firedocs has become a very popular alternative for authors to create and update their eLML lessons. The OpenOffice plugin is in its current version outdated and will be rewritten and re-released by a German University. Finally, the Lenya-based eLML extension is currently running on a test server at the University of Zurich and will be incorporated into the main UniCMS server soon.

D. Comparing eLML to other markup languages

There are a great number of HTML-based content creation tools available that allow authors to create IMS or SCORM compatible content packages. These tools are not comparable to the XML-based approach of eLML since they do not offer the separation of content and layout and thus also do not offer

the possibility to export content in various formats. But during the last years other markup languages for eLearning content have appeared. Most of them were the result of a masters thesis or PhD thesis and thus do not have a community and/or institution behind them that supports them. Therefore we state that eLML is today to most well-known and widely used open source XML markup language available for creating e-learning content.

In 1999 the German University of Passau started a framework called LMML, the Learning Material Markup Language [13]. They had a very strict approach for using semantic objects like motivation, definition, remark, example etc. for assuring a very strict separation between content and layout. Theoretically this is an excellent approach. eLML also tried to follow this strategy but soon had to find out that authors would not use a markup language if there are not at least the most basic structural objects like column, table, box or list available. Furthermore eLML introduced the "class" attribute allowing authors to attach CSS-classes to objects and thus allowing them to format elements according to their needs. LMML has been discontinued for many years now, the last available publication is the PhD-Thesis of Christian Süß [14] published in 2005 and giving a roundup of LMML.

An even more interesting approach was ML3, the Multidimensional Learning Objects and Modular Lectures Markup Language developed at the University of Rostock together with 12 partner institutes in Germany [15]. On the content level it works with similar educational objects like LMML (e.g. description, remark or example) therefore it seems that there has been cooperation between these two Universities. But ML3 is far more developed and it offers an authoring tool based on FrameMaker. In ML3 a lesson can be described in three axes: Intensity (basic, advanced, expert), target (teacher or learner) and device (online, print or slide). An author can define if a paragraph or illustration is used in the basic and/or advanced version, only on slides, visible for teachers only etc. eLML offers the same possibility with the attribute role (author and student version of a lesson) and visible (online, print, latex, odf etc.). We did try out a possibility to define the intensity but the authors were confused which paragraph appears in which version of a lesson. Furthermore, it is important that authors do not lose track of what is shown in which version and what is not, considering the 18 possibilities to present a lesson. ML3 offers more interesting ideas concerning reusability of objects and the creation of self-assessment. However, it looks like ML3 has been put on hold as the last update was published only in 2005.

The most popular tool falling into the category of XML content creations tools is Giunti Labs very popular eXact Packer [16]. Its main features are the support for both SCORM 1.2 and 2004, templates, wizards and a WYSIWYG authoring tool for rapid content development. A very nice feature is the neat integration into commercial content repositories like Hive and Lobster (both owned by Giunti Labs). In comparison to eLML with its fix ECLASS-based schema, the eXact Packager allows loading different kinds of models depending on the use case. The drawback of this approach is that there is no output format available for these different models except for the basic default models. Authors

have to program the transformation files themselves if they want to use some of the more sophisticated models. Higher education users (was ist das?) complained that there is no model available that fits their needs (e.g. including powerful bibliography, glossary or list of tables etc. as in eLML) since most models tend to support rapid content creation. Moreover, as the tool is not open source users have to pay high license fee and they are restricted to create their own models within the limits that Giunti Labs sets up. Authors cannot use XML tools of their choice like the very powerful open source XSLT parsers "Saxon" but are bound to the Microsoft XML engine. Because of these compromises the University of Applied Sciences Lübeck "oncampus" is thinking about switching from Giunti Labs commercial tool to the open source eLML framework in 2008 if the eLML Firedocs editor should become more sophisticated. Andreas Wittke, the manager of the oncampus system development, presented a comparison of these two tools at the Online Educa conference 2008 in Berlin, Germany [17].

To summarize, eLML is one of the only open source markup languages for creating e-learning lessons that survived and evolved through the last decade with only commercial tools as alternatives (apart from using HTML-based tools) ..

III. OLAT: LMS TO MANAGE ONLINE COURSES

OLAT started originally as a student project in 1999. The PHP-based prototype convinced both the users and the responsible professor, which led into the development of a more stable second OLAT 2.x release. After winning the "Medida Prix" in 2000, one of the most prestigious European e-learning prizes, the University recognized its potential and decided that starting from release 3.x, OLAT would be the strategic learning management system at the University of Zurich. This decision brought up the need for new features. Due to security issues and technical limitations of PHP, OLAT was completely redesigned using the Java programming language. The resulting OLAT 4.x version was released as an open source project in 2004. Other universities started to use OLAT and with the Bildungsportal Sachsen in Germany and the two Swiss spin-off companies Frentix GmbH and JLS Goodsolutions GmbH new partners contributing to the project were found. The range of functionality OLAT offers constantly grew. New requirements led to new features added to OLAT and released as version 5.x in 2006. The latest major release was OLAT 6.x (summer 2008), including a completely redesigned layout based on a CSS framework called YAML, fully scalable clusterability offering the possibility to distribute OLAT onto many different servers (true load-balancing) and many new features.

The following chapters will present OLAT considering the three main roles used in OLAT:

- *Author*: The person creating and managing a course. Usually a professor/lecturer, assistant or an e-learning professional responsible for creating the course.
- *Tutor*: Persons (usually students in higher semesters) assisting the students online and/or offline. Tutors coach students in discussion forums, check the tasks they

handed in, grade them or issue certificates if a student has passed or failed a course.

- *Participant*: Usually a student participating at a course.

A. Creating courses (author's view)

Today's learning management systems (LMS) usually cover a wide range of scenarios from enriching a lecture with course supporting materials available in several electronic formats such as PDF or PowerPoint slides to e-learning courses taught online without any classroom attendance. The most common use of a LMS lies in between the "enriched" and the "pure e-learning" scenario, usually referred to as "blended learning": a lecture held at the university enhanced with online education and collaboration elements [18]. OLAT and the courses held at the University of Zurich focus both on the enriched and the blended learning scenario. The so-called "building blocks" – inspired by IMS Learning Design – allow authors to create courses in a flexible way without being restrained by a rigid didactical concept. This didactical freedom is one of the most appreciated features in OLAT. The following building blocks are available:

- *Content*: SCORM and IMS Content Packages, single or multiple pages, external content (tunneling) or download folders.
- *Assessment*: Tests/Quizzes and self-tests (both based on the IMS QTI Standard), tasks and assessments.
- *Collaboration*: Forum, wiki, file dialog and chat. OLAT supports a sophisticated, extensive group management tool to facilitate collaboration functionalities.
- *Organization*: Enrolment, contact form and questionnaire.

Figure 2 shows the OLAT course editor available to authors for easily creating an OLAT course:

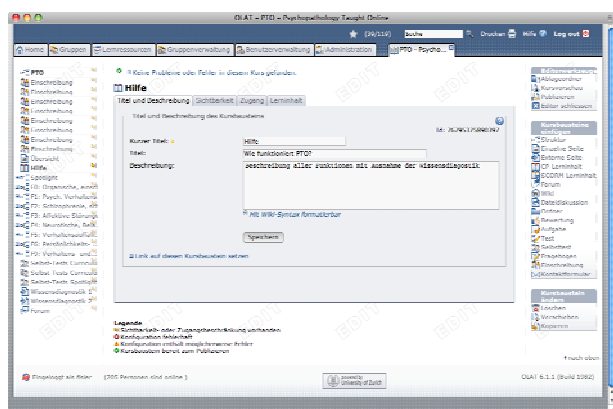


Figure 2: Screenshot of the OLAT course editor

A typical OLAT course starts with a syllabus, a page with a description of the course and an enrolment form for a course. A large majority of authors do not allow access to a course unless a participant/student is enrolled and is a member of the respective student group. An enrolment can be set up in a flexible way in OLAT: With or without a limited number of participants, with or without the possibility to cancel enrollment, defining a waiting list if a group is full, with or without automatic moving up if a space gets available, etc.

The actual content of a course is usually managed with

eLML – see chapter II for information about eLML – and included into OLAT via the IMS Content Package or SCORM format. Some authors prefer to include PDFs or PowerPoint slides using an OLAT download folder or just include course modules like HTML pages, Wiki, forum, tests, etc. using the included web based WYSIWYG course editor.

Online assessments and quizzes are also created and managed in OLAT. Both regular tests and self-tests can be created in OLAT or imported from any tool supporting the IMS Question & Test Interoperability (QTI) Specification. Tasks can be offered as PDF documents in OLAT and students have the possibility to both draw a task randomly or to first read the task and then pick one (according to the author's decision). The task building block offers a drop box where students can upload their work and a return box where tutors can comment and return the commented tasks. All the presented task and testing building blocks allow tutors to assess the student with grades, assigned either manually or automatically (in the case of single/multiple-choice, KPrim-questions or fill-in-the-blank quizzes) and to export excel lists with test results of all the students. The assessment possibilities of OLAT are an important factor when it comes to the implementation of the Bologna reform and thus are widely used at the University of Zurich [19, 20].

The collaboration tools OLAT offers are: Forum, wiki, chat (based on the Jabber protocol and therefore also usable with compatible chat clients like iChat, Adium or PSI) and the OLAT-specific "file dialog" building block that can be used within any course as often as needed. For an extensive overview of all course elements and options of the OLAT course editor, check the documentation available from the OLAT website www.OLAT.org.

The author of an OLAT course defines who can view and access the course through a role based access control mechanism. Authors can either make their courses accessible to all OLAT users, just to students from certain universities or students studying a specific topic or even just to the members of a particular group. Within a course the access/visibility settings are even more granular: Each building block allows fine tuned visibility and access settings based on user attributes, based on the date, scores achieved or on group membership. These settings allow authors to setup a target/ achievement based learning paths for their students.

B. Managing courses (tutor's view)

Tutors must evaluate and coach/support students. The evaluation can either be done automatically based on points earned within tests or manually by grading student work within the task course element. These functionalities have already been briefly discussed in the chapter above. It is important to understand that the whole scoring process in OLAT is kept transparent: Each step (adding or subtracting points or evidences of achievement) is logged and both the student and the author/teacher can see how a tutor rated or altered a participant's rating. Since 2006 many exams at the University of Zurich – often with hundreds of students – are being carried out using OLAT.

Tutors do not only decide whether a student passed or failed a class, they also coach students and help them benefit

as much as possible from the courses. As using collaborative tools is only efficient if students get feedback or a peer-review by tutors. Coaching also means answering questions in forums, reviewing submitted tasks or documents created within wikis, discussing results using the chat function, etc.

C. Working through courses (student's view)

After login OLAT users find themselves on the starting page, their "home" page. The home tab contains various so-called portlets containing bookmarks to courses, group memberships, notes, event notifications from the calendar or important messages published by the system administrator, comics and much more. Every portlet can be enabled or disabled, moved up and down and configured according to each user's preference.

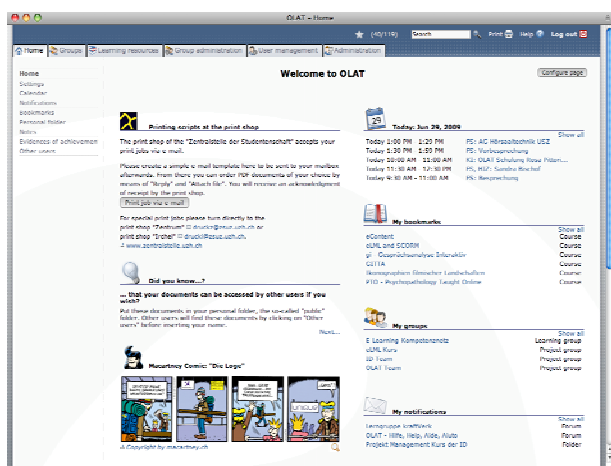


Figure 3: The OLAT "home" tab showing different portlets

The home tab also contains the navigation to important functionality like your own settings (language, personal profile etc.), the calendar with your personal, course or group events, your notes, news and RSS feeds, your evidences of achievement and the personal folder that allows storing important documents either in your private folder or available for download to all users in your public folder.

D. Collaborative activities

Collaborative activities are key features of OLAT. The group concept of OLAT anticipates two types of groups:

1. *Learning groups* belong to a course and can only be created by either the authors of a course or the tutors responsible for a course
2. *Project or buddy groups* can be created by each user. Group creators are free to invite any other OLAT user to collaborative work using the group tools.

Both group types offer the same tools:

- Calendar for managing group events
- Folder to exchange documents (can be mounted on a user's computer desktop using WebDAV)
- Wiki to collaboratively write and edit content
- Forum to discuss issues
- Chat/Instant Messaging (either using the integrated web client or using a Jabber-compatible tool like iChat)

- Mail form to send e-mails to all group members

These tools are only available to the members of a particular group and cannot be accessed by outsiders – not even authors or tutors – unless they are invited to the group. This assures that students can work without interference or "fear" of being controlled. The use of these collaborative tools depends on the author and his/her guidelines and on the motivation or skills of the group members. There are different didactical scenarios that can be realized using these tools: group work, expert chats or group puzzles [21] are just some of the possibilities that help students get acquainted with important skills they will develop throughout their study career.

IV. COMPARISON WITH OTHER LMS PLATFORMS

OLAT falls into the category of Java-based LMS platforms. From a functionality point of view, all today available LMS only differ slightly, like SCORM standard support (1.2 or 2004). However, they noticeably differ when it comes to modularity and flexibility, scalability like clustering support, usability and options for personalization. OLAT is primarily used in the university environment, where adaptation and personalization are important factors. Characteristics of OLAT are the completeness in its native mode, didactical freedom without limitations and the simplicity of personalization for professionals with Java skills. There are two other famous Java-based LMS available today: the open source Sakai project and the commercial software Blackboard which will be compared to OLAT in the following chapter.

A. Sakai

The Sakai project started 2004, five years later than OLAT. Today both tools are – with a few hundred servers running worldwide – minor players compared to Blackboard or Moodle and they do not differ very much when comparing the feature list. Sakai is mainly dominant in the US and UK and does not offer as many translations as OLAT. Currently in OLAT there are over 30 languages under development, whereas Sakai only has about 10 ongoing translations. Another difference is how the organization and the development community are being structured. Sakai has a steering committee responsible for the core development and for strategic decisions. Getting into this committee requires universities to pay membership fees. As shown in the last chapter, OLAT has a different organizational structure and so called "core community" members do not have to pay. Last but not least many users say that they prefer the OLAT 6 layout as compared to the Sakai 2 user interface.

B. Blackboard and WebCT

Blackboard is the main commercial player in the LMS market. They acquired their two competitors WebCT and Angel [22] during the last years and with a patent lawsuit tried to fight their third competitor Desire4Learn [23, 24]. Within the e-learning community these actions were not received very warmly and Blackboard is accused of trying to get "market dominance" for any prize. Thus more and more educational institutions are switching from one of the

Blackboard products (WebCT, Angel or Blackboard) to an open source solution [25]. In Europe for instance the University of Hamburg evaluated OLAT as an alternative. The University of Hamburg chose OLAT and is currently switching from Blackboard to OLAT [26]. The following statements were taken from these evaluations.

Blackboard and WebCT are commercial tools and unlike open source tools universities have to pay high license fees. Former WebCT users told us that it seems that both products do have a lot more bugs than OLAT. This results in expensive phone calls with the Blackboard support which was described by many users as being “incompetent, unfriendly or totally overpriced”. Input by highly qualified university IT staff is either ignored by Blackboard or incorporated into the next release without any reward for the institution that originally submitted the input. Many universities are therefore switching to open source tools where their valuable input is honored and rewarded by a large user and development community. Angela Peetz, the e-learning responsible at the University of Hamburg, told that one of the main reasons for switching to OLAT was that the support is great, the developer community is very active and acts quickly when posting questions on the mailing lists. Furthermore, the cooperation with the University of Zurich has been very satisfying in her eyes.

C. Moodle

Moodle is probably the most widely used open source LMS mainly because of its simple installation procedure. When comparing Java to PHP it is obvious that PHP server tools are installed much quicker than Java-based tool.

Further comparisons of OLAT and Moodle mainly refer to a chapter written by Cerstin Mahlow in “Learning Management Systems Technologies and Software Solutions for Online Teaching: Tools and Applications” [27]. Mahlow has been responsible for running both a Moodle and an OLAT server and is thus a competent expert when it comes to comparing the two tools. She states as main differences between the two:

- OLAT offers more integrated editors than Moodle: OLAT course editor, HTML editor, IMS QTI editor, IMS Content Package editor
- The access rules in OLAT are much more fine grained than Moodle and can be applied to every building block in a course and not just to the whole course.
- The OLAT core offers a lot more tools than the Moodle core (e.g. tunneling, file dialog, theme exchange). Moodle on the other hand offers hundreds of plugins developed by users. But these plugins are mostly untested, unsupported and usually only run with a specific Moodle version. OLAT, however, integrates new features developed by the community only after a strict review by the core developer team but once a feature/plugin is part of the core it is also tested and supported continuously by the core developers.
- OLAT offers special “rooms” not affiliated to a course allowing any group of people to collaborate using tools like forum, file sharing, wiki, calendar or e-mail. Giving students the possibility to choose themselves with whom they want to collaborate is a crucial feature in Higher

Education.

- In Moodle an instructor has to add and remove students to groups manually. OLAT additionally offers the possibility that instructors can allow students to enroll themselves into groups.
- Moodle only allows implementing courses according to three didactical scenarios: by topic, by date or by an activity. The course editor in OLAT on the other hand permits the instructor to construct courses tailored to his particular needs using so called building blocks. Each building block is a tool or function that can be used as often as needed.
- Moodle only runs on MySQL, OLAT can be used with any database (MySQL, Oracle, Postgres) through the object-relational mapper Hibernate.

At the School of Social Work of the University of Applied Sciences Northwestern Switzerland the results of this study led to a switch from Moodle to OLAT in early 2009 and according to the responsible staff the authors and instructors seem very happy with the new system.

D. Usability and marketing studies

The study by Mahlow [27] contains a chapter where usability and the user-front-end of both Moodle and OLAT are compared. She found that in OLAT the user-front-end is much more convincing than in Moodle. There is no hidden structure of activities or documents used within a course; the structure or hierarchy presented corresponds to the hierarchy the instructor chose when setting up the course. Moodle does not fulfill the main principles for usability; its navigation is misleading and confuses the user. The way OLAT presents courses fulfills the demands of usability of websites. Users always know where they are, where they came from, and where they can go. Elements for navigation are always visible at a certain space; users get familiar with the layout very quickly. Her final conclusion: “*Generally speaking, we come to the conclusion that Moodle may be suitable for supporting teaching and learning in primary schools, but it is not suitable for hierarchically structured institutions of higher education offering a wide range of courses, seminars, and lectures with different didactical scenarios for educating adults, i.e., universities. We argue that universities should thus use systems offering both instructors and students a wide range of features, which can be used individually or in combination to meet educational objectives and intentions. OLAT can be regarded as a prototypical e-learning system that meets these requirements.*”

A more extensive usability study was done in 2007 by the University of Berne [28]. In this study both students and tutors who never worked with OLAT had to solve tasks like solving an assessment or creating an OLAT course. Both their mouse and eye movements were tracked at the usability lab at the Institute of Medical Education Berne together with an audio and video recording of their session. After evaluating these recordings the authors of the usability-study wrote a report with suggestions – ranging from crucial to “nice to have” – for improving OLAT 5. Some findings of the report were:

- For authors it was not always clear where they are: course editor, course preview or course run view?

- There is no back button
- The tab metaphor in OLAT was not always used correctly
- More affirmative user feedback was needed
- Icons in OLAT were not clear or descriptive
- Learning resources like wiki or tests used in a course had to be created outside of the course editor
- Author had to leave the course editor when creating a group to e.g. define access rules
- Some workflows needed too many mouse clicks
- The use of AJAX technology could bring great improvements for authors and students

The complete results can be found in the extensive report written by Prof. Sissel Gutthomsen and Dr. Stefan Minder [28]. For the OLAT team this usability study was an “eye opener” helping the team to gain a deeper understanding into the way inexperienced users are working with OLAT and how potential usability issues could be prevented. The OLAT team included the findings of this study in their roadmap. In OLAT 6.0 release the whole layout was redesigned according to the suggested improvements..

In 2007 I. Wille published a study focusing on the perception of OLAT within the e-learning community and how the awareness of OLAT could be improved [29]. The author concluded that OLAT is a mature product but he located room for improvement in its marketing activities and international publicity. The extensive report contained many suggestions for improvements. The main reason why many of them have not yet been realized is that the University of Zurich mainly focuses on the development and improvement of OLAT code and does not put much effort in promoting its tool to the open source community

E. User Feedback

OLAT also considers qualitative user feedback to be an important indicator of the quality of a LMS. For this paper the authors asked OLAT server administrators throughout the world why they switched from another LMS to OLAT. Here are some of the replies:

- The technical level of management in OLAT seemed more like an enterprise solution when compared other LMS.
- We chose OLAT because it is open source, it is based on the right technology for our business, and it seemed to have all the features we were looking for. We have performed quite an extensive evaluation in order to confirm what additional development may be necessary and have concluded that very little is necessary, since OLAT already has the features we are looking for.
- Compared to Moodle, OLAT had more functions that were crucial to my company's requirements
- The long-time development since 1999, the support by the University of Zurich, the Portuguese-BR language option (new at the time), and many prizes that OLAT has won, all these things contributed to my decision.
- OLAT is Java-based application and thus more secure than PHP based LMS. In Angel and Moodle for example it is very easy to manipulate the URL and become another student thus gaining access to their records and

even delete their submissions.

But the administrators or authors did not only bring up positive feedback. Some of the negative statements when comparing OLAT to other LMS were:

- Impossibility to use "back" button from the browser.
- Tab navigation is a little confusing for the newbies.
- Search not very efficient to find text inside documents
- The Home page could be more dynamic, with useful information like: Last additions, New content, Pictures/Images, etc.

V. CONCLUSIONS

It was a risky step taken by the University of Zurich to start the development of a new LMS in 1999 and a new XML-based content creation tool in 2002. But both projects were realized in a “bottom-up” approach: OLAT started as a student project and eLML was created out of an XML structure of an e-learning project. The decision to use these tools as strategic e-learning tools at the University of Zurich was made by the steering committee when the tools were already created, proven and in use. The decision by the lead of the IT Services is one of the responsible factors for the longevity of both eLML and OLAT. But probably the main reason why these tools still exist today, is because both tools were made available to the e-learning community by releasing them under an open source license in 2005.

As a result, OLAT has today a very active developer community supporting own, adapted OLAT versions like OPAL or FELIX 2 in Germany and others in countries like Italy, UK, US and some in Asia just to name a few. This developer community also implements features according to their own priorities and roadmaps. For example, features like personalized “Student Homepage”, extended “collaborative functionalities” like access to WEB 2.0 features or AI based “Tutoring and Learning path adaptation” [30] facilities ePortfolio functionality, etc. are projects implemented by this developer community.

Moreover, the development team at the University of Zurich has put a lot of effort and resources into their tools. The result of these efforts is that OLAT has reached a high maturity level in terms of functionality and stability. The hardware requirements for running an OLAT server prove this fact: The Moodle documentation [31] recommends 1GB of RAM for the server per 50 concurrent users, meaning that for 1000 and more concurrent users a Moodle server would require at least 20GB of RAM. In contrast, the OLAT server works with only 2GB RAM assigned [32], both due to the technology used (Java instead of PHP) and the optimized code for high loads.

Thousand concurrent users is the maximum amount of users one OLAT server can handle today. Until release 6.1 one OLAT installation could only run on one server and the number of users being able to access the system was limited by the hardware. Starting with the 6.1.x branch OLAT offers full scalability: one OLAT installation can run on different servers, the load can be distributed and thus the limiting factor will be the budget available for buying new servers. Optimizing scalability will be one of the most important tasks

for OLAT in the near future.

Not only the Java developers are very active. The open source community is contributing actively to the translation of OLAT. Today a total of 30 translations are available or under development. In addition to the main European languages German, French, English, Italian and Spanish OLAT is also available in international languages like Chinese, Farsi, Russian or Arabic. A full list including a world map of OLAT users worldwide can be found on the OLAT website OLAT.org.

A German marketing study about OLAT [29] published in 2007 found that both the acceptance of OLAT and the satisfaction when using OLAT are very high compared to other learning management systems. The main problem of OLAT is, however, that throughout the world the software is not very well-known. The community behind OLAT should therefore allocate more resources into conferences and marketing activities. This has been addressed by participating at conferences not only in Europe but also in China in 2008 and in the United States and Canada in 2009. But the OLAT team does not only participate in conferences, in March 2008 it organized its first own OLAT conference at the University of Zurich. With nearly 200 registrations the interest was higher than expected [33]. The two-day conference was a big success and a next edition is the OLAT User Day 2009 taking place on September 11th 2009 in order to celebrate the 10th anniversary of OLAT.

VI. OUTLOOK

Today we can clearly identify a much stronger pedagogical / didactical influence requesting a freedom of choice in selecting a pedagogical / didactical model. Furthermore, we think course authors put growing focus on collaboration activities integrated into a course. Collaborative functions should be used in a transparent way and support extensions to use other WEB 2.0 facilities. The modularity of OLAT will allow to implement functionalities regarding the above mentioned trend. First steps in this direction are taken by advanced projects initiated by the international OLAT community.

REFERENCES

- [1] Bleisch, S., and S. Nebiker, "The Swiss Virtual Campus Project GITTA - A multi-disciplinary, multi-lingual learning platform for Geographic Information Technology", *XXth ISPRS Congress*, Istanbul, Turkey: ISPRS, 2004.
- [2] Schnabel, O., R. Stopper, and L. Hurni, "New modular approach for knowledge-transfer in multimedia cartography: The e-learning project CartouChe", *International Cartographic Conference (ICC)*, Moscow, Russia, 2007.
- [3] Müller-Böker, U., "Globalization and Livelihood Options of People Living in Poverty (GLOPP)", *Department of Geography, University of Zurich*, 2007. <http://www.forschungsportal.ch/unizh/p7562.htm>
- [4] Streule, R., Oberholzer, René, Läge, Damian, "Der E-Learning Kurs "Psychopathology Taught Online" - Flexibilität und Adaptivität dank OLAT und eLML", *OLAT Conference*, Zurich: University of Zurich, 2008.
- [5] Lautenschlager, P., "'eCF - Get involved in Corporate Finance" Development of an E-Learning Course within the Swiss Virtual Campus Framework", *5th International Conference on New Educational Environments*, Bern, 2003, pp. 71-75.
- [6] Fisler, J., and S. Bleisch, "eLML, the eLesson Markup Language: Developing sustainable e-Learning Content Using an Open Source XML Framework", *International Conference on Web Information Systems and Technologies (WEBIST)*, Setubal, Portugal, 2006.
- [7] Schneider, F., M. Weber, and H.-J. Zuberbühler, "The open source learning management system OLAT (Online Learning And Training)", *EUNIS Conference*, Tartu, Estonia: University of Tartu, 2006.
- [8] Downes, S., "eLML", *Personal Blog*, 2007. <http://www.downes.ca/cgi-bin/page.cgi?post=42034>.
- [9] Gerson, S., "ECLASS: Creating a Guide to Online Course Development For Distance Learning Faculty", *Online Journal of Distance Learning Administration*, 2000. <http://www.westga.edu/~distance/ojdl/winter34/gerson34.html>
- [10] Horton, W., *Designing Web-Based Training*, New York: Wiley, 2000.
- [11] Niederhuber, M., Heinimann, H. R., Hebel, B., "e-Learning basierte Fallstudien zur akademischen Ausbildung in der Geoinformatik: Methodisches Konzept, Umsetzung und Erfahrungen", Zurich, Switzerland: Eidgenössisch-Technische Hochschule (ETH), 2005.
- [12] Linowsky, T., "eLML template generator", *Google Summer of Code*, 2008. <http://code.google.com/soc/2008/uzh/appinfo.html?csaid=8BE887D7E47F3CB5>
- [13] Süß, C., Freitag, Burkhard, "LMML Learning Material Markup Language", *IFIS Report*, No. 03, 2001.
- [14] Süß, C., "Eine Architektur für die Wiederverwendung und Adaptation von e-Learning-Inhalten", *PhD Thesis*: University of Passau, 2005.
- [15] Anonymous, "ML3 - Multidimensional Learning Objects and Modular Lectures Markup Language", 2005. <http://www.ml-3.org>
- [16] Anonymous, "Giunti Labs eXact Packager", 2009. <http://www.giuntilabs.com>
- [17] Wittke, A., Fisler, Joël, "Mega Content Transformation with Open Source Educational Content Project eLML (eLesson Markup Language)", *Online Educa*, Berlin, 2008.
- [18] Bachmann, G., and M. Dittler, "Integration von E-Learning in die Hochschullehre": Universität Basel, 2004. http://www.sws.uni-dortmund.de/docu/Bachmann_Dittler.pdf
- [19] Pospischil, A., "E-Prüfungen praktisch", *E-Learning Podcast des ELC der Universität Zürich*, 2007. <http://www.elc.uzh.ch/static/elearningpodcast/>
- [20] Schneider, F., "OLAT & E-Assessment", *E-Learning Podcast des ELC der Universität Zürich*, 2007. <http://www.elc.uzh.ch/static/elearningpodcast/>
- [21] Kohl, K., "Didaktische Modelle für die computerunterstützte und virtuelle Lehre", *Konferenz: NMB-Projekte in den Ingenieurwissenschaften*, Hochschule Anhalt Dessau: Pinkau, S. & Gerke, T., 2003.
- [22] Severance, C., "Blackboard Acquires Angel Learning", *dr. chuck's web log*, 2009. <http://www.dr-chuck.com/csev-blog/000620.html>
- [23] Feldstein, M., "Blackboard Wastes Money on Worthless Patent", *e-Literate*, 2009. <http://mfeldstein.com/blackboard-wastes-money-on-worthless-patent>
- [24] Korcuska, M., "Sakai statement on 2nd Bb-D2L patent suit", *Michael Korcuska Sakai Blog*, 2009. <http://sakaiblog.korcuska.net/2009/04/10/bbd2lpatent2/>
- [25] Feldstein, M., "Blackboard's Market Share Erosion", *e-Literate*, 2009. <http://mfeldstein.com/blackboards-market-share-erosion/>
- [26] Payome, T., "Universität Hamburg: Umstieg auf OLAT", *Checkpoint.eLearning*, 2009. <http://www.checkpoint-elearning.de/article/6450.html>
- [27] Mahlow, C., "Choosing the Appropriate E-Learning System for a University". Y. Kats (Ed.): *Learning Management Systems Technologies and Software Solutions for Online Teaching: Tools and Applications*: IGI-Global, 2009.
- [28] Guttormsen, S., Minder, Stefan, "Usability Testing and Evaluation, and Redesign Recommendations for the LMS OLAT", Bern: Institute of Medical Education (IML) University of Bern, 2007.
- [29] Wille, I., "OLAT - Ein Kommunikationskonzept für das Open Source Learning Management System", *PR Plus Kurs 31* Heidelberg: Universität, 2007.
- [30] Pedrazzoli, A., "OPUS One - OLAT: An Artificial Intelligence - Multi Agent based Adaptive Learning Environment", *World Congress on Engineering and Computer Science*, San Francisco: IAENG, 2009.
- [31] Dougiamas, M., "Installing Moodle: Hardware Requirements", 2009. http://docs.moodle.org/en/Installing_Moodle#Hardware
- [32] Brunner, P., "OLAT Server Hardware Requirements", 2009. http://www.olat.org/website/en/html/download_requirements.html
- [33] Fuchs, M., "UZH-Software für alle", *unipublic*, 2008. <http://www.unipublic.uzh.ch>