CONTENT MANAGEMENT SYSTEMS VERSUS LEARNING ENVIRONMENTS

Towards Architectural Content and Learning Management Systems

STEFAN BOEYKENS, HERMAN NEUCKERMANS
K.U. Leuven, Department of Architecture, Urbanism and Planning,
Kasteelpark Arenberg 1/2431, B-3001 Heverlee (Belgium)
stefan.boeykens@asro.kuleuven.be
herman.neuckermans@asro.kuleuven.be

Abstract. Schools and teachers increasingly apply Online Learning Environments for teaching and course management. In many cases, an existing platform is used to support the curriculum. At the K.U. Leuven in Belgium a campus-wide platform is provided, using a commercial Content Management System. At the same time, the Design and Building Methodology research group at the Department of Architecture, Urbanism and Planning developed a custom portal to organize Computer Aided Architectural Design courses and seminars. Integration of this portal into the university system was rather disappointing, since much of the flexibility and customizability was lost, without any chance of filling in the gaps. This article discusses the possibilities and limitations of existing web-based systems to support Computer Aided Architectural Design teaching and reports on our own experiences from the last decade. A comparison of selected systems is juxtaposed with the requirements derived from these experiences.

Keywords. Education; CMS; LMS; CAAD.

1. Introduction

The application of Online Learning Environments varies from basic, custom websites to large, campus-wide educational portals. The experience level of the teachers also varies, from people with sometimes limited Information Technology (IT) knowledge to certified trained technical staff.

At the K.U. Leuven, in Belgium, the commercial Blackboard Academic Suite (www.blackboard.com) was chosen as the default platform. The system was deployed by the main support group at the university services level and is currently being used for the management of courses and the related communication towards students, including calendars and announcements. Additionally, Blackboard was extended into a full support environment, including additional self-assessment possibilities using Questionmark-Perception (www.questionmark.com) and custom-developed extensions. The whole system, under the name TOLEDO, also connects to the university student, staff and curriculum administration, which is managed in SAP, to allow for a correct representation of users with their current credentials (toledo.kuleuven.be). Figure 1 displays the Toledo portal alongside the CAAD portal.

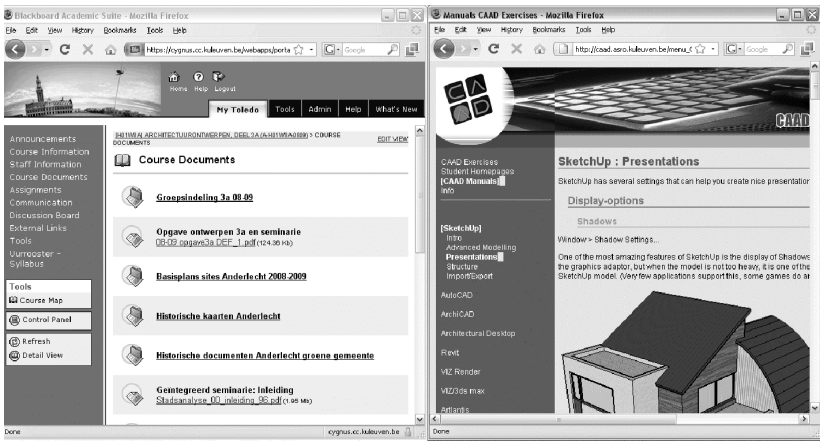

Figure 1. Toledo Portal (left) versus CAAD (right) Exercises at K.U. Leuven

At the same time, on a smaller scale, we developed a custom portal to organize Computer Aided Architectural Design (CAAD) courses and seminars, both for the Bachelor-Master program and the Post-Graduate schools (caad.asro.kuleuven.be). As shown in Figure 1, this comprised a combination of static pages with curriculum content and a database-driven portal to manage calendars, student registrations and assignments.

The integration of this CAAD portal into TOLEDO was unsuccessful. Much of the flexibility and customizability was lost, without any chance of filling in the gaps ourselves. After discussion with the university helpdesk, it was confirmed that several aspects of our own, home-grown system were simply unavailable and the system did not allow the use of a server-side scripting language, such as ASP or PHP. As a result, our own portal stayed in use, rather than migrating to the university portal, albeit loosing some of the features that

the global portal could have offered, such as the direct connection to the student accounts.

2. CMS vs. LMS - Towards ACLMS

The failed integration experience led us to investigate the possibility of applying alternative systems, considering the fact that it was not trivial to maintain a custom system with changing staff at different competence levels. A brief survey informed us about the major categories of platforms.

Content Management Systems (CMS) focus on structuring content, with dedicated support to manage users. Typical examples are Joomla (www.joomla.org) and Drupal (www.drupal.org). While they are not targeted at providing online learning environments, they are extendable and could be used as the "engine" for such a system.

On the other hand, Learning Management Systems (LMS) and, in extension, Learning Environments are dedicated to manage and maintain an online curriculum, where courses and assignments are natively provided. Examples include Blackboard and Moodle (www.moodle.org) as discussed in (Martens and Achten, 2008), also in the context of an architectural curriculum. Other notable examples are ATutor (www.atutor.ca) and Claroline (www.claroline.net), as discussed in Docq et al. (2007). Usually LMS systems lack some of the features which are common in a CMS and which appear to be essential for CAAD teaching.

A third category is Virtual (Learning) Environments (VLE), with characteristics of both systems. Gül and Maher (2006) looked at design collaboration using augmented 3D worlds, where additional content is related to the design environment, making the virtual world the actual CMS. Dalla Vecchia et al (2008) described a VLE which starts to function as a LMS incorporating interactive 3D Worlds. Both examples witness the application of LMS/CMS into architectural teaching. This article will not go into detail on VLEs, however.

The following section will discuss more specific requirements and it will be argued that the ideal system needs functionality from both LMS and CMS systems, to be usable as an "**Architectural Content and Learning Management System**", which we will note as **ACLMS**.

3. Requirement Formulation

The long experience with our in-house solution, which evolved throughout the last decade, led us to formulate a list of specific requirements for an ACLMS.

Our experience lies within CAAD seminars, but it is assumed that many of these requirements are generic and could be mostly applied to other educational contexts, especially where design and the creation of graphical content is paramount. We will not focus on trivial requirements, such as user management, content authoring or news posting, which are supported in all systems. Additionally, weblogs (blogs) and RSS feeds, which fit very well in a teaching context, are fairly well supported in contemporary systems, so they will not be further discussed.

3.1. ARBITRARY AND TEMPORARY GROUPING OF USERS/STUDENTS

During Design Studio and CAAD assignments, students are often allowed or even required to collaborate. They form groups of typically three or four students to work on a particular task. However, these collaborations change throughout the term and even over different assignments, so the ACLMS should provide means to define temporary groups of students. This implies that grading and evaluation should be possible on both the individual and the group level. While most LMS and CMS systems allow the creation of groups, they are sometimes inflexible with regard to this requirement or do not allow students to be a member of multiple groups.

3.2. VISUAL REPRESENTATION OF CONTENT

In Design Studios and CAAD courses, students of architecture create to a large extent graphical documents, rather than textual data. It is therefore imperative that there are means to display content visually on the assignment and evaluation pages. In HTML, the `<IMG>` tag allows the display of image types such as JPEG, GIF or PNG. Similarly, `<OBJECT>` and `<EMBED>` tags assist with the display of Flash, video and Java-applets. For the end user, however, this commonly requires the installation of a browser plug-in, which is not always available for the used platform and in many cases not allowed by the system administration for security reasons.

Some Java applications, such as GeoGebra (www.geogebra.org) as discussed in Hohenwarter et al. (2008) or Demicron Wirefusion (www.demicron.com), allow the publishing of a project as an interactive applet, which can be viewed directly in the browser window, provided the Java runtime is installed on the client system.

An alternative is to automatically convert CAD and 3D models into regular images, presenting them to the user as if they were looking at a full model. Project Freewheel uses this approach (freewheel.autodesk.com).

Our in-house system displays most submitted files inline on the portal pages with dynamically generated code on the server. However, this is not trivial and sometimes impossible to integrate into existing CMS or LMS systems. In the particular case of our university, the development of custom modules is only available to the global services group, who focus on supporting the whole community, rather than individual groups or departments. It is possible to edit a particular page and insert the necessary <EMBED> and <OBJECT> tags, but this has to be repeated for each individual page and it can not display dynamically generated content, using a server-side scripting language. With the submission of large amounts of reference documents and photographs by teachers or students, tens or hundreds of files are uploaded, which are only available as links to be downloaded one by one. In some cases, teachers had to fall back on using an ftp server, outside of the university platform.

3.3. ONLINE EVALUATION

Closely related to visual representation is online evaluation and feedback. A teacher should be able to browse through submitted files and directly mark them using a simple form. Moreover, instead of assigning a global score, grading should reflect required accomplishments as defined for each assignment. Unfortunately, with closed systems, teachers often fall back on separate, unconnected spreadsheets to maintain scores for students.

While it is possible to download assignments to open them in a dedicated application, this is a real hindrance for evaluation and comparison purposes. The time required to load, display and inspect CAD or 3D documents is quite extensive. In many cases, student results can be adequately evaluated using an interactive preview. In case of doubt, teachers can fall back on the original document. While simple online evaluation, such as commenting or voting, is available in most of the surveyed systems, this is often not combined with visual display.

3.4. MODULARITY OF COURSES

To be able to inform students about the learning path, references to other course modules are necessary, to define prerequisites and relations. Most LMS systems provide means to define at least the order of particular learning units and can guide students on a particular learning path. However, the choice of alternative paths or routes is not commonly supported. A modular course system can be used to support different tracks or alternative routes to reach a certain goal. By properly formulating requirements and skills learned for each course and expressing them into generic terms, it is easier to structure this.

This is especially important for the organization of our CAAD seminars, as they are organized throughout different semesters and embedded in different courses. The tutorial resources are also shared between Bachelor-Master and Post-Graduate courses, so they should not be maintained at different web-pages. Since many of the skills taught in CAAD modules rely on concepts from prior modules, it is imperative that students are informed about them, so they can easily refer to them when required. For our system, this aspect is currently in development.

3.5. INTEGRATE STATIC CONTENT

As part of the courses, teachers provide syllabi, presentations and manuals. These can be a combination of texts, images and downloadable files. Many CMS and LMS systems assume the teacher/user to have limited technical skills for the creation of online documents, so they use WYSIWYG applets to edit or create content. When importing HTML files, embedded scripts or applets are usually ignored or not transferred correctly. Moreover, all further editing and updating has to be done on the imported text. To our understanding, it is desirable to embed external documents at page-generation time; to be integrated into the main navigation, using the same formatting. Alterations and improvements to the text will be reflected automatically on the main site.

One approach to incorporate external pages is the usage of HTML-frames. While they still display the regular navigation interface and menu surrounding the page, there is no real interaction and navigation inside the frame is not reflected towards the surrounding navigation frames, unless complex scripting is applied. Inline frames (`<iFrame>`) present similar disadvantages. An alternative approach is the use of the "include" functionality on the server level, but this simply copies the full text verbatim into another context, which turns relative links to local files or images unusable.

In our in-house system, we wrote a parsing script, which reads the original HTML document, strips out the headers and maintains the section between `<BODY>` tags. The script also adjusts all local, relative URLs, since they have been written to be used from another location on the server. While there are still some minor issues, the system allows using the embedded document within the main portal structure page, including the main style sheet and navigation structure.

4. Overview of Existing CMS and LMS systems

This section will give a qualitative overview of a small selection of systems from both categories, which allows us to form a more motivated evaluation of

the current state of CMS and LMS systems and on which levels they can be improved to support our vision of an environment supportive of teaching CAAD or usable in Design Studio Teaching. A more comprehensive overview of CMS systems can be found on the CMS Matrix (www.cmsmatrix.org), which includes advanced searching and comparison. To a lesser extent, a similar comparison can be found at the (unrelated) LMSMatrix (www.lmsmatrix.org). While being much smaller in scale and seemingly unfinished, it provides insight into their functionalities. Another comparison source is the WP2 deliverable from the Information and Communication technologies in Lifelong Learning project (ICOTEL, 2006; icotel.ktl.elf.stuba.sk, unavailable at time of writing; and www.adam-europe.eu/adam/project/view.htm?prj=415).

In this overview, we will only focus on two popular CMS systems (Joomla and Drupal) and three LMS systems (ATutor, Moodle and Blackboard), mainly to point out where they fail to meet our requirements.

4.1. JOOMLA (WWW.JOOMLA.ORG)

This is an Open Source CMS, which is widely used and described in several books and courses. The modularity of the system and its free availability generated a large amount of user-contributed modules, which can extend the original system in many directions. Some companies develop commercial extensions, themes or training. However, despite its popularity, it is worth to notice the rather inflexible category system. By definition, sections contain categories, which contain articles. This enforces users to structure all content according to these depth levels and creating a deeper, hierarchic structure, which is not trivial. The default navigation menu is also manually maintained and does not fully follow the site structure. There are some attempts to extend Joomla into a LMS, as discussed in the Edugeek community (www.edugeek.net).

4.2. DRUPAL (WWW.DRUPAL.ORG)

The Open Source Drupal CMS is another very popular system, with a thriving community of module developers, who provide additional themes, content types and new ways to organize content. The system is said to be somehow more programmer-oriented and its interface is largely text-based. There have been some attempts to extend Drupal into a LMS (groups.drupal.org/taxonomy/term/1898; and www.phpedu.org.uk).

For our requirements, Drupal lacks the integration of external pages. There was an older (and discontinued) module to load an external HTML site, but all actual content was embedded inside Drupal nodes, which form the core of the system. We experimented with a custom node type where only the URL to the

external page on our local server was stored. We then added an additional page component to render the content within the rest of the page, but this did not integrate well with the rest of the Drupal functionality, e.g. in summaries or views. Additionally, the main focus is clearly on the Apache web server and the MySQL database system. When we tested with the Microsoft Internet Information Server (IIS), certain limitations and compatibility problems arose.

4.3. MOODLE (WWW.MOODLE.ORG)

It is a very popular Open Source LMS system to manage curriculum material. It is possible to define different user roles, such as students and teachers and to set up courses and assignments. Due to its popularity, numerous extensions are available, usually for free. The Moodle system has good support by its community, but many of our requirements can only be (partially) solved using external add-ons.

In our internal tests, we had little success with a Windows-based IIS server, but on an Apache system, it worked really well. There is also some support to integrate external pages, without leaving the Moodle interface. The visual display of uploaded files, however, requires custom development.

4.4. ATUTOR (WWW.ATUTOR.CA)

This is also an Open Source LMS. There are several extensions available and the system is freely available, but it does not have the same large community as Moodle. While we evaluated this system as an interesting solution, time limitations did prevent us to experiment with the development of the required functionality to extend it into an ACLMS. Having a smaller end user community and less documentation than the more popular systems also raises the threshold to engage with such as system.

4.5. BLACKBOARD (WWW.BLACKBOARD.COM)

This is a suite of commercial solutions, revolving around the BlackBoard CMS. There is a wide range of features and modules, which influence its total license cost. Despite its possibilities, much of the editing requires a fair amount of wading through dialog-like pages, with submission or cancel buttons. Above all, the page content is mostly static and controlled by textual forms. It is possible to attach files, define assignments and allow submission of documents by student users, but without visualization of the documents. While it is possible to insert HTML code into the page, this remains a static copy, so subsequent changes to the course text have to be reinserted or be edited directly in the Blackboard

page. The system provides working groups within a single course or community and content can have visibility rules, to only enable it for a particular group. However, when defining assignments for a particular course or community, every assignment was added to every user, rather than by group.

5. Applying Requirements to Existing Systems

The systems described in the previous section are compared in a Table 1, listing the main characteristics and adding our own requirements.

TABLE 1. Comparison of LMS/CMS systems.

	Drupal	**Joomla**	**Moodle**	**ATutor**	**Blackboard**
License	Open Source	Open Source	Open Source	Open Source	Proprietary
Platform	Any	Any	Any	Some	Some
CMS/LMS	CMS	CMS	LMS	CMS/LMS	CMS/LMS
Web Server	Apache, IIS*	Apache, IIS*	Apache, IIS…	Apache, IIS…	IIS, Apache
Scripting	php	php	php	php	free SDK
Database	MySQL, PostgreSQL*	MySQL	many	MySQL	MS SQL, Oracle
Cost	free	free	free	free	commercial
Arbitrary Grouping	Organic Groups (A)	Community Builder	Groups Workshop	Work Groups	Working groups*
Online Evaluation	(A) fivestar, custom review*	(A) rating/voting, comments*	YES	Partially	Assessment Gradebook*
Integrate Static Content	NO	Wrapper (iFrame)	Resource Module	NO	Frame-only
Visual Display of Content	Add-ons and scripting	Add-ons (e.g. JoomVRML)	Resource Module*	Images only	Images only
Modularity of Courses	NO	NO	YES	NO	YES

(A) = Add-on
* certain limitations apply considering the requirements

6. Conclusion

The article elaborates the desired functionality for a system supportive of CAAD and Design Studio Teaching and illustrates this with examples from our own experiences and custom developments. At the same time, a critical reflection

was made regarding the application of closed systems, preventing the required customizations. It is hoped that this could open the debate to provide both the tools and the flexibility to create valuable and performing learning environments, usable at different staff knowledge levels.

The article concludes with a tabular comparison of different existing systems, Content Management or Learning Environment. This is not meant to be as complete as possible, but rather to learn about some of the limitations or shortcomings of both systems.

Acknowledgements

Part of the work in this article was funded by the K.U. Leuven Research Fund (Reference OI/2006/07).

References

Dalla Vecchia, L., da Silva, A., Pereira, A.: 2008, Teaching/Learning Architectural Design Based on a Virtual Learning Environment. In: *Architecture in Computro*, 26th eCAADe Conference Proceedings, Antwerpen (Belgium), Sep. 2008, pp. 801-808.

Docq, F., Lebrun, M. and Smidts, D.: 2007, Claroline, une plate-forme d'enseignement/ apprentissage sur Internet. Pour propulser la pédagogie active et l'innovation ? *In*: Frenay, M., Raucent, B. and Wouters. P. (Eds.), *Actes du quatrième colloque "Questions de pédagogies dans l'enseignement supérieur"*, Louvain-la-Neuve : Presses universitaires de Louvain. 2007, pp. 99-109.

Gül, L.F. and Maher, M.L: 2006, Studying Design Collaboration in DesignWorld: An Augmented 3D Virtual World, *In* E Banissi, M Sarfraz, ML Huang and Q Wu (eds), *Proceedings of 3rd International Conference on Computer Graphics, Imagining and Visualization Techniques and Applications (CGIV06)*, IEEE Computer Society, pp. 471-476.

Hohenwarter, M., Hohenwarter, J., Kreis, Y., and Lavicza, Z.: 2008, Teaching and learning calculus with free dynamic mathematics software GeoGebra. In: *11th International Congress on Mathematical Education*. Monterrey, Nuevo Leon, Mexico

ICOTEL: 2006, Deliverable WP2-E-Learning Platforms, present state, In: *ICOTEL (Information and Communication technologies in Lifelong Learning)*, from the Leonardo Da Vinci programme (SK/02/B/F/PP/-142261) – project homepage unavailable at the time of writing

Martens, B. and Achten, H.: 2008, Do you Moodle? In: *Architecture in Computro*, 26th eCAADe Conference Proceedings, Antwerpen (Belgium), Sep. 2008, pp. 153-160.