Migrating Existing Multimedia Courseware to Moodle

Declan McMullen, Mark Melia, Mark McEvoy, Claus Pahl
Dublin City University
School of Computing
Dublin 9, Ireland
[dmcmullen|mmelia|mmcevoy|cpahl]@computing.dcu.ie

Edward Jennings
Senior College Dún Laoghaire
Information Technology Department
Dún Laoghaire, Co. Dublin, Ireland
ejennings@scd.ie

Abstract: Open source course management systems offer increased flexibility for instructors and instructional designers. Communities can influence the development of these systems and on an individual basis, the possibility to modify the system software exists. Migrating existing courseware to these systems can therefore be beneficial, sometimes even required. We report here about our experience in migrating an existing courseware system consisting of multimedia content and interactive, integrated infrastructure functionality to an open source course management system called Moodle. We will assess the difficulties that we have encountered during this process and, discuss the importance of standards in this context, and we aim to provide other instructors or instructional designers with guidelines and assessment support for other migration projects.

Introduction

Commercial course management systems (CMS) – also called learning management systems (LMS) or virtual learning environments (VLE) – are around for a number of years now. CMS help instructors to organise lesrning resources and to facilitate activies within the environment (Dean, 2002). The discussion of the adoption of CMS and their use has been revived recently with the availability of open source CMS such as Moodle (Moodle, 2004), CourseWork (CourseWork, 2004), or Claroline (Claroline, 2004). Open source systems are freely available software systems that can usually be modified by their users; communities of interested users often drive the further development of these systems.

The introduction of these CMSs in general, whether open source or not, causes problems if content or support systems already exist and these legacy systems need to be integrated with the new environment. A completely new development should be avoided. This scenario, where once-off systems were developed locally in the early days of a new technology before a standardisation and adaptation process starts, is a common one. Consequently, the migration from legacy systems to these new platforms is desirable, often necessary. The reason for migration could be improved quality of the learning experience, interoperability, or sharing needs to reduce costs. Migration can be voluntary or enforced by the organisation in charge.

Our objective here is to look at the migration of a content-oriented courseware system (Pahl, Barrett, Kenny, 2004) that we have developed over a period of several years and to highlight the problems that occurred. We describe the pitfalls that we encountered and classify these problems. We hope that this classification scheme will help other instructors to assess possible problems and the complexity of course migration in other situations. An important aspect in this context that we will address are standards such as SCORM (ADL, 2004).

The benefits of course management systems and similar systems are well covered in the literature. Instructors, however, need to be aware of the difficulties of migrating to these environments as well. We will address problems that arise by migrating to a CMS in general, but also discuss issues specific to open source environments.

Moodle

We introduce the Moodle background briefly, focusing in particular on the open source context of the Moodle development itself and on the perspective on the instructor in creating a new course.

Moodle (Moodle, 2004) is a course management system (CMS). CMSs are software packages designed to help instructors to create quality online courses. Fig. 1 shows the home page of our case study course. Moodle is different from other systems in that it has a strong grounding in social constructionist pedagogy, i.e. it prescribes to some

extent the pedagogical approach. This is, however, not of essential importance for our more technical discussion, as we will see later on.

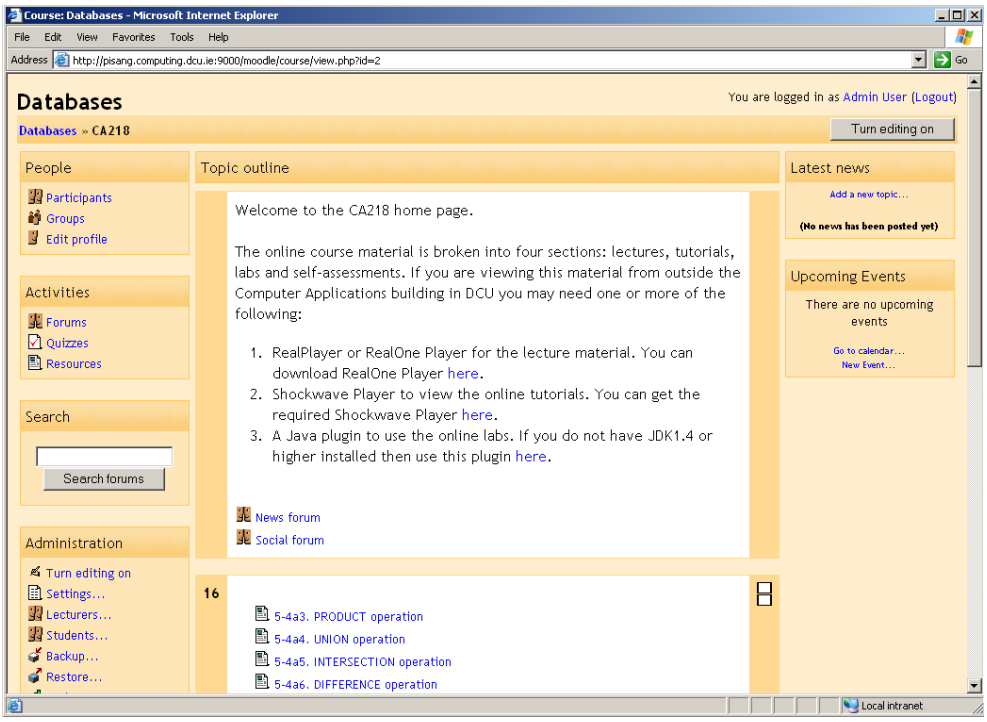

Figure 1. Moodle course environment – course home page after migration.

Moodle is an open source software system. Originally developed by Martin Dougiamas as an academic project, there is now a community driving the development of extensions and improvement of the Moodle platform. The Moodle infrastructure is based on a database (mysql and postgresql are currently supported). The functionality of the system is provided in form of php-scripts. Moodle runs on several operating system platforms including Windows and Linux. Moodle can be downloaded, used, modified, and distributed freely (with some conditions formulated in a licence agreement). The Moodle Web site (Moodle, 2004) provides the discussion forum to decide on further extensions and improvements of the system.

The instructor's perspective is essential for the migration discussion. Moodle offers an instructor a variety of ways to create and administer a course. The following are course creation steps:

- Course format. Moodle supports three basic course formats, which define the basic layout of a course. These are currently weekly, by topic, and social (which is discussion-oriented).
- Upload facility. An upload facility allows an instructor to upload resources that can be made available through the chosen format. A range of standard resource types are possible: Web pages, audio and video files, text documents, etc.
- Activities. A central element of a course are the activities, possibly involving the resources, that a learner can engage in. Examples are assignments, discussion forums, or quizzes.

Course administration involves:

- monitoring and steering the contributions in the discussion forums,
- monitoring the overall learner activities using access logs that are constantly updated by the system.

The Interactive Database Learning Environment IDLE

The objective of this paper is to investigate the migration of existing courseware systems, such as IDLE, to an (open source) course management system, such as Moodle. We discuss the migration of a particular courseware system – the Interactive Database Learning Environment IDLE (Murray, Ryan, Pahl, 2003; Pahl, Barrett, Kenny,

2004) – to illustrate the main issues. We introduce IDLE here first, before describing the migration of IDLE to Moodle in the next section.

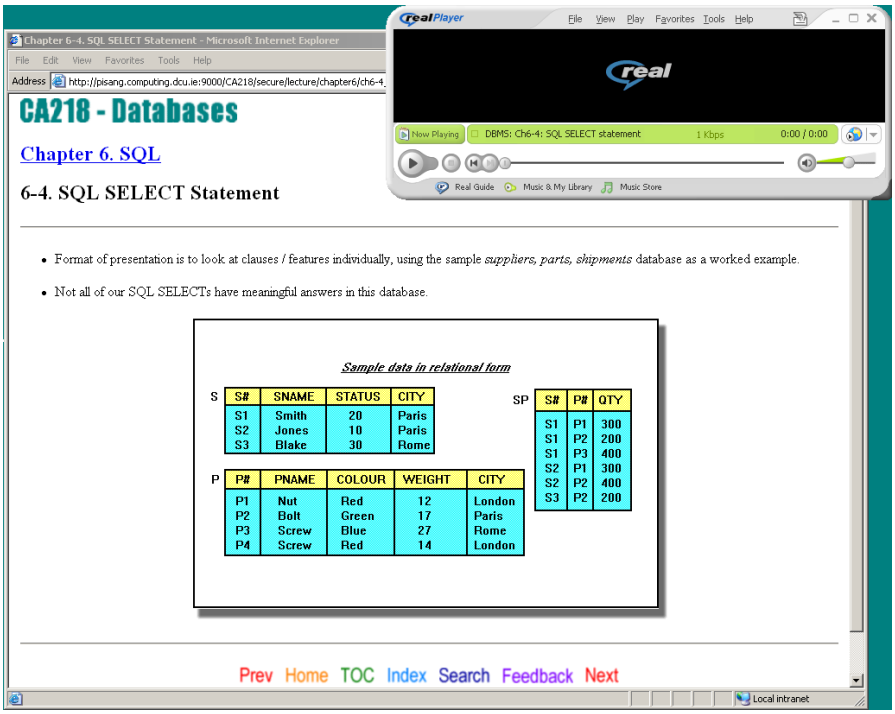

Figure 2. IDLE – navigation and delivery (synchronised lecture system) – before migration.

IDLE is a Web-based courseware system that has been incrementally developed over the past eight years. It is currently in use for a second-year undergraduate course in a computing degree. IDLE can be characterised as an integrated, interactive multimedia environment with learning and training features. Central in the IDLE system are a variety of integrated multimedia applications (Elsom-Cook, 2001). These are supported by an infrastructure that, similar to Moodle, consists of a database to store content, learner records, and also data produced by learners as part of their course, and some software functionality (mainly coded in Java) to integrate the different features and to access the database.

IDLE supports three styles of learning, which are similar to classical forms of instruction: lectures, tutorials, and labs. We will highlight in particular those features here that have caused difficulties in the migration process.

- Lectures consist of audio-visual presentations – see Fig. 2. The audio streaming is synchronised with the display of HTML-pages. The HTML-pages are hierarchically organised into chapters and sections and linked on each level.
- Tutorials use visualisation techniques to explain and illustrate some concepts. Particularly for the computing course in question, the visualisation of algorithms and software is an ideal learning tool. An example is the animation in Fig. 4 – here already incorporated into the Moodle environment.
- Labs are supported by a number of interactive tools: a graphical editor, a database query execution interface (see Fig. 3), and an analysis tool. Important about these tools is that they interact with an underlying database that allow storage and retrieval of results that the students have created.

Underlying these features is an infrastructure consisting of a database to store learner resources, but also work produced by students in their projects (see right-hand side of Fig. 3), and server functionality.

Migration to Moodle

We distinguish three different aspects of courseware elements that were relevant in our context. The three aspects are:

- Content – which includes text, multimedia, and interactive features.
- Infrastructure – which refers to scripting and database support.
- User management – which means access control (accounts/passwords) and student tracking (access logs).

This organisation shall help us to present our migration experience in a structured and accessible manner.

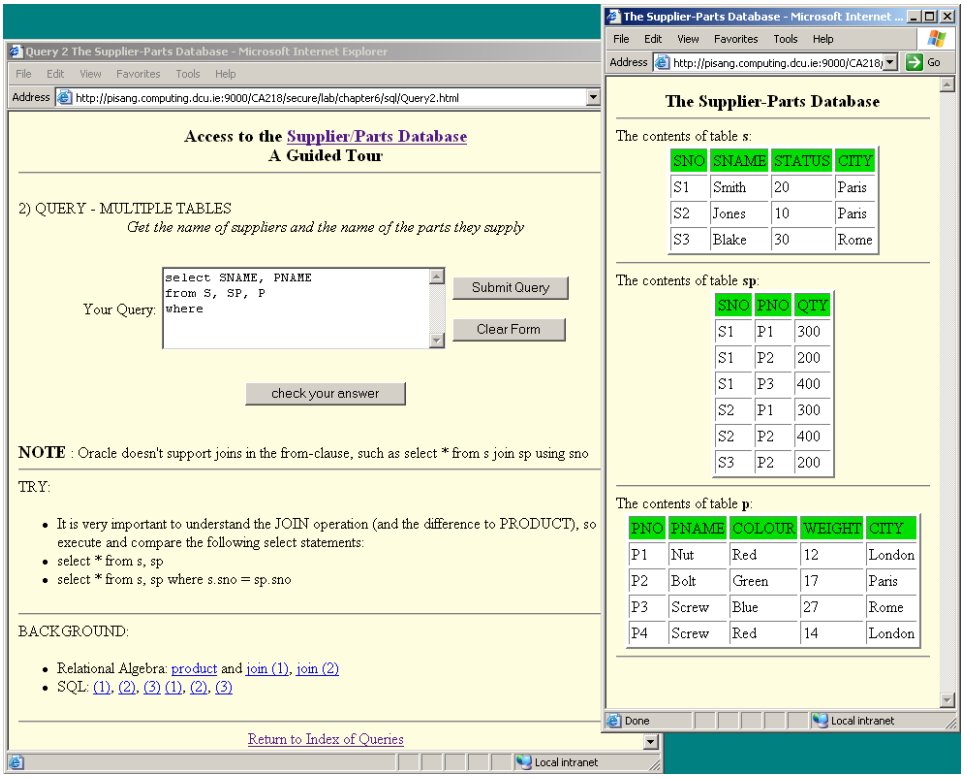

Figure 3. IDLE – interactivity (SQL execution lab) – before migration.

Content

The IDLE content can be described as *multimedia*-based and *interactive*. This content is made available through a *navigation infrastructure* and media-specific *delivery* mechanisms. *Self-assessment* is a specific form of content-based activity.

Multimedia. Most media types (text/HTML, audio (MP3), video, animations) have not caused any problems. We tested audio (MP3), video (Flash, QuickTime, Windows Media Player format), and animations (GIF). These formats can be added as resources using Moodle's file upload and management feature. There is either direct support or Web browser plugins handle their display. Some formats, such as the RealAudio format, could only be incorporated in form of links (which would cause an external player to be launched), but not as resources.

Navigation Infrastructure. Difficulties were encountered, when HTML-pages were uploaded that contained links to other resources within the current system infrastructure.

- The hierarchical structure of the overall collection of pages could not be preserved.
- A navigation tool bar – that includes links such as previous, next, home, or index, see Fig. 2 – could not be preserved.
- Most of the pages contain links to audio material and/or images, which would have to be manually updated.

The current database course system consists of more than 130 HTML pages that are hierarchically organised into chapters and sections. Each HTML-page has a navigation bar with (among others) buttons for *previous*, *next*, and *home*. If these pages were added as resources, then the links in the navigation tool bar would have to be re-done. The reason is that the moodle display function (view.php) is based on identifiers that are assigned to resources on upload.

Since all HTML-pages are similar in structure, we wrote a Java program that parses the files and extracts the necessary part for uploading.

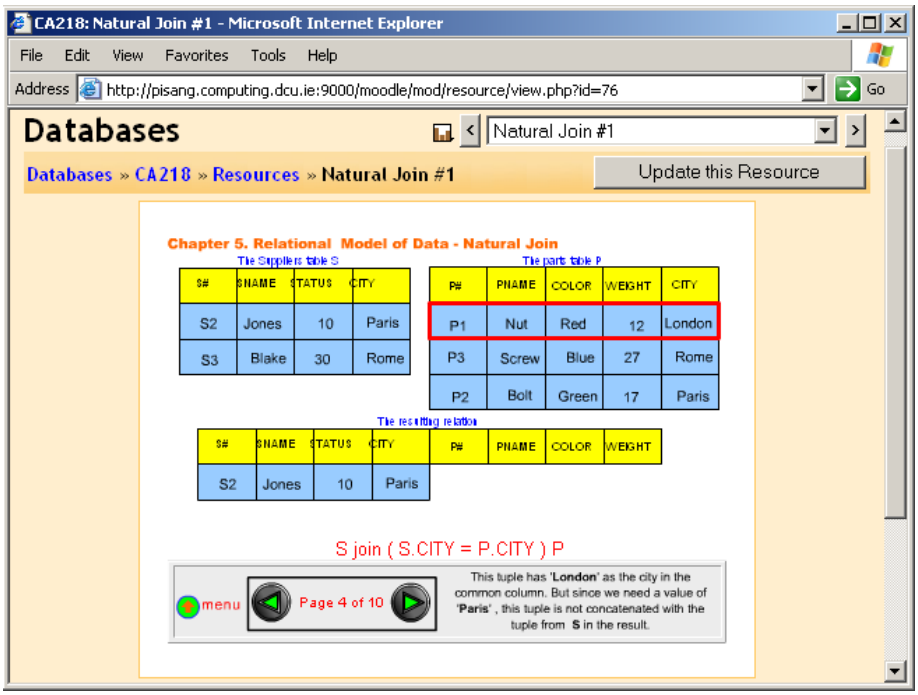

Figure 4. IDLE – multimedia (relational algebra animation) – after migration.

Delivery. Other problems were caused by the synchronised audio-visual lecture presentation. The IDLE lecture part uses a multimedia presentation consisting of synchronised audio (in RealPlayer format) and HTML-pages. The audio server reads the audio stream and an events file that specifies times at which a new HTML-page is to be displayed. The corresponding page update command is send to the Web server. Various problems were encountered:

- RealAudio is not supported by Moodle (as above).
- The events file contains links to the original HTML-pages. If these were to be uploaded as resources into Moodle, all links would have to be updated manually. As these audio features are in use since 1996, we encountered classical software legacy problems, including lack of documentation and non-executable software, that created some problems. We recovered the necessary software, but had to recreate and recompile event files.

Addressing this problem did require a considerable effort.

Interactive Features. Interactive features such as applets – which are in use for a graphical editor and an analysis tool within the lab part of IDLE – also required modifications with respect with internal communication and security issues. An essential aspect of the database courseware system is the workspace and storage functionality. Students can use the learning environment to carry out project work. They can create and store database models and implementations; they can query and analyse/optimise their database implementations. This feature is implemented through applets, servlets, and an underlying database system. Some modifications in the communications infrastructure between applets on one side and the database on the other were necessary due to the inclusion of some resources into a remote Moodle server. The servlets that handle some of the communication were converted into Moodle's scripting language, php.

Infrastructure

The two elements of the infrastructure aspect are the *Web platform* and the *database support*.

Web Platform – Apache and php. We installed Moodle on the same server that hosts the IDLE system. In order to be able to integrate advanced features into Moodle, a number of upgrades of the existing server platform were necessary. In order to support the current Moodle release, alterations needed to be made to the current installations of PHP and Apache on the database course server.

- The database and some other courseware functionality would have to remain on the existing server. This server would therefore also need to host Moodle. As the current configuration did not support the hosting of php-based Web Applications, php and Apache where both upgraded and configured to support the Moodle platform.
- In order to adjust to the Moodle infrastructure, we converted existing Java servlets, which handle the communication between Web browser and additional software (e.g. the database system) to php.

Database Support – Oracle. Since the database course system includes an Oracle database server, it was (unsuccessfully) attempted to replace the Moodle mysql database with Oracle. The Moodle installation documentation claims that Moodle has full support for all database management systems (in addition to the two standard systems mysql and Postgresql) due to its use of ADOdb, a generic databases language. However, installation scripts are database specific. We attempted to alter existing scripts to work with Oracle. After the occurrence of errors, we estimated the number of files/scripts to be altered too high to continue this work. The use of a single database system was therefore not possible, since a conversion of the existing Oracle system to mysql was not feasible.

Learner Management

Two aspects are important for learner management: *accounts and access control* and *student tracking*.

Accounts and Access. Since the current course system requires student authentication for Web resources and the database system, the user account management for these features was integrated with the Moodle login (using the University's LDAP server). We have generated accounts for Web and database access for all usernames returned from LDAP. In order to support the lecturer, an automatic user management system based on php scripts was written. Some criteria had to be considered.

- Additional users can be added (guest login, without write access to the database).
- Security needs to be considered since the database access includes write access. Therefore, passwords need to be stored securely. The account/password generation script should only be available to the administrator.
- Students should be able to change their password (with the usual precautions prevented somebody else's password to be changed).

Student Tracking and Usage Mining. The investigation of an integration of student tracking combining Moodle functions and Web server logs (as used in IDLE) was started. Basic student tracking is available in Moodle. Web servers create access logs for incoming page requests. For Web server logs, open source tools for statistical analysis are available. We also implemented some advanced Web usage mining features for IDLE in the past. Here, we investigated how these features can be integrated into an advanced learning behaviour analysis tool. The first problem for infrastructures like ours, where Moodle and a separate Web-accessed server exist in parallel, is to reconcile the different data structures of the access records. Moodle stores learner activities in form of database tables. Web servers create a standardised text-based log file. Since a range of tools is available for the analysis of log files, exporting Moodle's tables into a text file complying with access log standards would be the best option.

Standards

In the database courseware system, we used multiple-choice based self-assessment features. These were implemented using Java applets. Since the questions (and answers) were not coded as part of the applet software, but were stored in separate text files, their conversion into Moodle quizzes turned out to be easy.

Converting or representing multiple-choice questions in the standardised QTI-format is an example of the benefits of standards. While in particular the SCORM set of standards (ADL, 2004) supports the interoperability of learning content with CMS, and major commercial and open source products support these standards by now, achieving this

interoperability by creating standardised interfaces is not necessarily trivial. Our experience with interactive features, some running on a distributed platform, illustrates this point.

Extending Moodle

Almost certainly an instructor might wish to be able to modify or extend the course management system. If the CMS is an open source product, then this wish can be reality. We report here on an extension of the Moodle Upload and Delivery feature in order to enable XML-based content. XML-based textual content allows separating content from presentation. An XML-file containing content is combined with a transformation script that creates a presentation form for the learner.

A previous system version is based on an XML-based content repository for the database courseware system. While requesting a course page, the corresponding XML-based course content is taken from the repository and converted dynamically into a HTML page using the transformation script. This allows the lecturer the flexible management of course content and it allows the presentation of personalised content for instance based on learner data. We have created an improved Moodle version that allows content in form of XML (based on an educational markup definition we developed) and the necessary XSLT transformation instructions to be uploaded into Moodle.

Packaging content in an XML document could support the migration process. We developed a feature that, based on an XSLT transformation specification, creates a whole course from a large XML content document.

Another feature that we would like to see in Moodle is a more flexible format definition. With more than 130 individual lecture resources on the one hand and different categories of resources (lecture, tutorial, lab, and assessment resources) on the other, a more flexible resource organisation would be necessary. An extension of Moodle could allow us to categorise resources and make this organisation visible for the learner and to organise resources of one category in a hierarchical form.

Discussion

Our experience with migrating an integrated multimedia courseware system to Moodle is both positive and negative. Some multimedia integration was easy to carry out, while other aspects could not be addressed without either reimplementing the feature or changing Moodle itself. The main problems were essentially caused by the fact that IDLE, although content-oriented, is based on an integrated architecture of several components and services (Pahl, 2001). Existing IDLE infrastructure, such as user management and data mining, competes in its functionality with Moodle. In general easy to achieve were the integration of multimedia types, such as text, audio, video, or interactive media, such as Java applets. Stand-alone resources did not cause major problems. The main problems that we have encountered are not specific to either open source or commercial CMS. In some aspects, Moodle allows for an easier integration. For instance, the file upload and management feature, which is missing in some commercial systems (Jennings, 2004), has turned out to be a useful tool.

We can summarise our experience in a structured list of possible aspects. This list can act as guide for assessing the feasibility and complexity of an attempted migration – see Table 1.

Content		Infrastructure	User Management
multimedia	interactive media	Web platform	access control and accounts
navigation	abstract content	database support	student tracking and usage mining
delivery			

Table 1. Migration aspect categorisation.

The list is based on the structure we used earlier on to illustrate our experience. This list is, of course, not meant to be exhaustive – others in similar situations might encounter different problems. Our focus here was on systems supporting learner-content interaction. Problems of a possibly different nature might be encountered, if systems supporting learner-learner or learner-instructor interactions are considered. We feel, however, that due to

the richness of features in our courseware system and the fact that it has been developed over a longer period of time in a number of stages, the problems we encountered are representative of this type of system. Our recommendation is to investigate carefully the complexity of the endeavour if a complex courseware system has to be migrated to Moodle – or any other virtual learning environment. Our problem list and the account of our experience can give constructive guidance for this task.

We have encountered a number of infrastructure-related problems typical for legacy software systems. These are for instance problems resulting from the incompatibility of some versions of software components with others.

In relation to Moodle, we could think of a number of extensions that would simplify the migration. For instance, a navigation infrastructure support, extensive mining features, or an XML-based content could form the core of a content management module for the future.

The problems we encountered in migrating our courseware system to Moodle were more of a technical than pedagogical nature. The fact that the Moodle design is based in constructionist pedagogy is in line with our pedagogical approach supporting active learning and a form of cognitive apprenticeship for the virtual environment.

Another aspect that needs to be discussed in the context of migration is standards. Standards form the basis for interoperability. Adherence to standards improves the integration of for instance content into a CMS and, therefore, supports the migration process (Jennings, 2004). Some of the commercial CMS have adopted the IMS Content Packing standard (IMS, 2004), which allows e.g. e-books and similar resources to be plugged in. Macromedia tools such as Dreamweaver, Director, and Flash, which play an important role in content generation, are SCORM compliant (ADL, 2004). XML as a general Web standard is also a solution for data transfer between CMS and other educational technology systems. Moodle support of these standards would greatly benefit migrations.

References

ADL (2004). Sharable Content Object Reference Model SCORM. http://www.adlnet.org/ (visited 21/10/2004).

Claroline (2004). *Claroline Open Source e-Learning*. Web site http://www.claroline.net/ . (visited 20/10/2004).

CourseWork (2004). *CourseWork Open Source Course Management System*. Web site http://getcoursework.stanford.edu/ . (visited 20/10/2004).

Dean, C. (2002). Technology Based Training & On-line Learning An overview of authoring systems and learning management systems available in the UK. PeakDean Interactive Ltd. http://www.baol.co.uk/PDF/authsys.pdf (visited: 21/10/04).

Elsom-Cook, M. (2001). *Principles of Interactive Multimedia*, McGraw-Hill, London.

IMS (2004). IMS Global Learning Consortium. Web site http://www.imsproject.org/ (visited 21/10/2004).

Jennings, E. (2004). *Moodle: an Open Source Course Management System – an Evaluation from a Multimedia Perspective*. M.Sc. Dissertation. Dublin City University.

Moodle (2004). *Moodle Open Source Course Management System*. Web site http://moodle.org/ . (visited 20/10/2004).

Murray, S., Ryan, J. & Pahl, C. (2003). A tool-mediated cognitive apprenticeship approach for a computer engineering course, IEEE *International Conference on Advanced Learning Technologies* ICALT'03, pp.2-6.

Pahl, C., Barrett, R., & Kenny, C. (2004). Supporting Active Database Learning and Training through Interactive Multimedia. *International Conference on Innovation and Technology in Computer Science Education ITiCSE'04*. Leeds, United Kingdom. ACM

Pahl, C. (2001). Interactivity and Integration in Virtual Courses. *IEEE International Conference of Advanced Learning Technologies ICALT*. pp. 395-395, IEEE Computer Society.

Pahl, C. (2002). An Evaluation of Scaffolding for Virtual Interactive Tutorials. *Proc. 7th E-Learn 2002 Conference*, Montreal, Canada. AACE.

Xu, L., Pahl, C., & Donnellan, D. (2003). An evaluation Technique for Content Interaction in Web-based Teaching and Learning Environments. In *Proceedings International Conference on Advanced Learning Technologies ICALT 2003*. IEEE Press.