# Integration of a Course Management System with Interactive Educational Multimedia

Mark Melia, Declan McMullen, Claus Pahl
*School of Computing*
*Dublin City University*
*Dublin 9*
*Ireland*

## Abstract

With the recent emergence of open source course management systems, learning support infrastructures have received increased attention. More and more organisations adopt these systems to provide their educational services. Often, existing, locally developed learning support software – ranging from delivery support to multimedia content – have to be integrated into these new systems. We will report here on approaches to integrate educational multimedia into a (open source) course management system. We will use our own courseware environment – IDLE, the Interactive Database Learning Environment – and its integration into Moodle – an open source course management systems – to illustrate our experience.

*Keywords: course management systems, educational multimedia, legacy systems, software and content migration.*

## 1 Introduction

Course Management Systems (CMS) – also known as learning management systems or virtual learning environments – are software systems that allow an instructor to upload and organise educational resources. A learner can access these resources through a structured, graphical interface. CMS are increasingly introduced by third-level institutions to streamline the provision of education.

Since electronic courseware and content might already exist, often the need to migrate to or integrate these with the new system arises. We can distinguish two

aspects: the integration of pre-existing content resources into a CMS and the migration of existing courseware systems to a CMS. Integrating a CMS with educational multimedia or migrating to a CMS is an engineering problem that requires an adequate methodology. We focus on the integration of educational multimedia content integration here, using a case study to illustrate our ideas.

## 2 Integration and Migration – the Need

We report on the integration of existing educational multimedia technology [2] into a CMS. Our case study is based on the adoption of Moodle [4], an open source CMS, at our University. Our own courseware support, an interactive educational multimedia system to support active learning for a computing course [5], see Fig.1 where an interactive Web-based programming lab with feedback is shown, had to be integrated with Moodle as the delivery/management platform. A number of other multimedia features are part of the system: streamed audio synchronised with textual features, animations, and graphical editors. Problems were caused in particular since

- some features were part of a project workspace area, where students could create and store project results,
- some media were integrated, e.g. audio and HTML-based text and images,
- the organisation of resources was hierarchical (HTML-based navigation) with interspersed interactive multimedia-based access to some features.

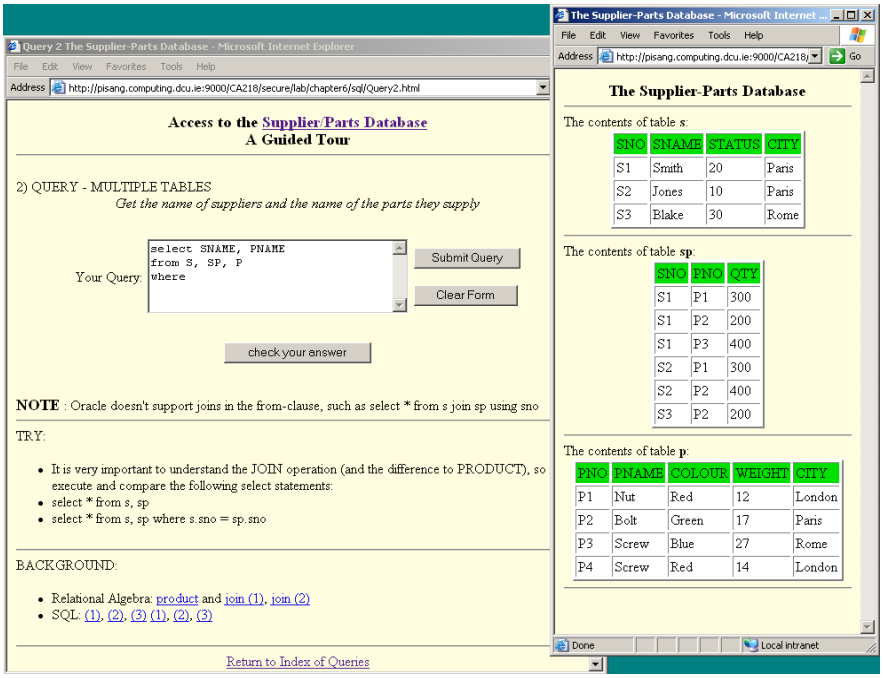


Figure 1. The original course system – an interactive SQL training feature.

## 3 Integration and Migration – the Solution

We focus on the integration of multimedia into a CMS. Central to this focus is the relationship between a CMS and educational content, which in our case is represented through educational multimedia.

- Most multimedia types (text, images, audio, video) have not caused any problems – they can be uploaded using the CMS upload features and displayed using either Web browser plug-ins or additional CMS support. In Fig. 2, we have presented the integration of a flash animation into the Moodle environment as a stand-alone resource.
- The navigation infrastructure that links individual content elements together is more difficult to migrate. Links embedded in Web pages can often not be kept unchanged, since for example Moodle assigns internal identifiers for resources. Navigation tool bars within the page could not be preserved.

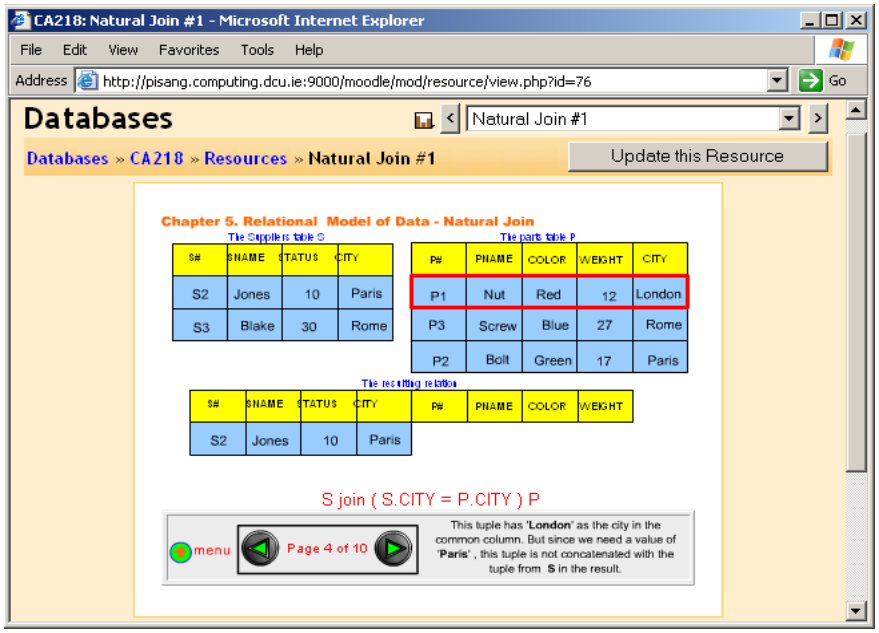


Figure 2. Multimedia object (Flash animation) integrated into Moodle.

While the integration of some media content types into a CMS is easy, advanced interactive and/or multimedia content creates problems:

- Content often comes with its own packaging, i.e. containers that also handle the delivery. Standards-based packaging, for instance based on the SCORM CAM standard [1], simplifies the integration of these content types.
- Interactive educational multimedia is often based on non-trivial software architectures with interface, server, and database, since an intelligent server component with database access is required to handle interactions.

This problem has been recognised by the community, and standards such as the SCORM standards [1] address this relationship between content and its technical environment. In particular the SCORM Content Packaging in the Content Aggregation Model (CAM) is a standard that a number of learning technology providers are now supporting. Adopting this standard avoids problems associated with static navigation links embedded in the resources by abstracting the hierarchical navigation infrastructure to an XML document. However, some technical and principal problems still remain.

Often, standards-based integration or interoperability is not sufficient. Technical solutions for these advanced integration problem encompass wrappers and bridges, sometimes requiring re-engineering of software components (of both the CMS and the educational multimedia). An open source CMS in our case has added some flexibility to the integration of some content types. Some examples of costly changes required include:

- Java applets, implementing graphical design editors, that had to be re-engineering with respect to their communications and security aspects,
- Java servlets, handling communication between the interface and the database backend that stores resources and workspace artefacts, had to be converted into the scripting language php in order to be interoperable with the Moodle environment.

A number of other software changes were required – including the database system we used to support our original infrastructure which was not compatible with the Moodle database support, the access control and authentication features, and student tracking and behaviour mining features. These software changes require an understanding of the architecture of the original and the new system [3]. Some general re-engineering and software change methods and techniques apply here [6]. For instance, interfaces of software components have to be preserved, if possible, in order to reduce the change impact. These problems are typical for the re-engineering of legacy systems.

Classical re-engineering of software legacy systems – a common software engineering problem – needs to be combined here with

- a focus on multimedia technology [2], since educational content is usually multimedia-based,
- the adoption of education-specific standards such as the SCORM content packaging and infrastructure standards [1].

An even more advanced solution, which we have attempted on an experimental basis, is the extension of the target platform. A feature of our original systems allows XML-based textual resources, which are stored in a database, to be converted into HTML on the fly at the time of the resource request (which allows for instance personalisation and adaptivity). This allowed for the abstraction of learning content from the rest of the system. Learning content in this XML format could then be loaded into the Moodle system by re-

implementing Moodle's delivery function to include the dynamic XML-to-HTML conversion process.

## 4 Discussion

Integrating or migrating multimedia courseware to a new platform including content and infrastructure is a frequent requirement. The educational technology development reaches maturity at the moment, resulting in numerous experimental, locally developed systems to be converted or integrated into stable commercial or open-source platforms. Our experience with this problem shows that three aspects are of major importance in the integration or migration process:

- Content: multimedia content objects, interactive media, navigation, delivery,
- Infrastructure: Web and server platform, database support,
- User management: access control and authentication, student tracking and usage mining.

This structured list of aspects summarises the experiences we made with our case study system. It can act as a guide for others to assess the feasibility and complexity of possible integrations and migrations.

An understanding of the architecture and the data formats of the CMS and the multimedia content is essential to facilitate integration and migration. Principles of an engineering solution to a migration or integration problem are based on

- standards for content types and delivery (cf. SCORM, [1]) and architecture (IEEE LTSA, [3]),
- interoperability questions (related to the standards issue).

To some extent standards can achieve a solution; however, additional technical and methodological support in particular for software changes is needed in some situations. We have illustrated some of these aspects using our own system as a case study.

## References


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

[2] Elsom-Cook, M., *Principles of Interactive Multimedia*, McGraw-Hill: London. 2001.

[3] IEEE Learning Technology Standards Committee. *Learning Technology Standard Architecture LTSA*. http://ltsc.ieee.org/wg1/index.html 2005. (visited 11/01/2005).

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

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

[6] Sommerville, I., *Software Engineering*. Addison-Wesley. 2004.