A Content Management System for the TILE Managed Learning Environment

Chris Jesshope and Zhenzi Zhang
Hull University and Massey University
C.R.Jesshope@dcs.hull.ac.uk & j.zhang@massey.ac.nz

ABSTRACT:
This paper will describe the TILE MLE, which comprises a learning content management system and an educational delivery system; it will focus on the former as the delivery framework has been described elsewhere. From a pedagogical point of view, content creation and delivery are the two key factors in web-based learning systems. When supported by the Internet, content delivery are relatively cheap. But creating high quality online learning content is very expensive. Currently different organizations are creating similar content for their own need, with little reuse of content. This is a waste of valuable resources and is the reason why there is an imperative need for standards-compliant content management systems to facilitate content development and deployment. E-learning content management systems face the challenge of collecting, organizing, managing, maintaining, re-using, delivering and targeting the content. We must differentiate between a content-management system and authoring tools for content creation. The latter are used to create content that is organised into a course. This may be animations, graphics, text, audio, video or other multimedia segments. These learning objects are organised and catalogued by the learning content management system and this creates units of study or on-line courses that can be navigated and perhaps monitored. That unit of study usually has structure, which may include hierarchy and precedence. An LCMS should support the management of any content level, which includes multimedia segments, learning objects or units of study. The design of such a system is described her, together with an analysis of the issues considered.

INTRODUCTION
There is currently a rapid uptake in is e-leaning. E-Learning enables the learner to learn without the restriction of time and distance. It can increase the accessibility of traditional education, increase the learning efficiency and facilitate collaboration. There are already many existing systems in this market, such as Blackboard and WebCT, but there is still a large body of research being undertaken in developing next generation systems. Much of this research is in the arena of defining common standards[refs], aimed primarily at cataloguing and the reuse of educational material. There are also developments, such as the TILE managed learning environment (MLE), Gehne et. al. (2001), being undertaken, which have enhanced the traditional, thin-client web architecture, to provide flexibility and scalability in the delivery of educational material, no matter where the student is or what internet connection they have (or not!). This paper looks at the design requirements for an authoring tool for TILE.

We will investigate both the technological constraints and the various users needs in designing such a tool. We have considered a variety of standards for the description of both on-line course packages as well as student models. We also consider issues such as interfaces to other standard authoring tools, e.g. web editors, multimedia packages etc. The output of these third-party tools must also be integrated into the course structure as learning objects and so we must develop open standards for that integration. This is not so easy, as TILE is designed to support student learning models and hence must be able to track the student s progress. The TILE delivery architecture includes a server on the student s computer to achieve this even when the student is browsing bandwidth unfriendly material off-line, from CD say. The problem we face in authoring, is that TILE may not be aware of student actions if the media produced by third-party tools contains choice in the form of hyperlinks. We propose a solution to this situation in this paper. We also describe a version control system that enables multiple authors to collaborate on course development. Also using a hierarchy of capabilities, we enable material, once placed in a repository and if licenses and

permissions allow, to be reused and re-annotated within other courses. This paper describes the schema and techniques to maintain a coherent, structured corpus of educational material.

E-Learning Systems Requirements

Generally, an e-learning system comprises three key components, they are infrastructure, services and content Lennox (2001). *Infrastructure* is the software that allows learning to be created, managed, delivered and measured. It can be divided into a Learning Management System (LMS) and a Learning Content Management System (LCMS). *Services* involve the planning, customization, integration and management of the e-learning application. *Content* can be categorized according to subject, preferred format, student s progress and language requirements. The content s origin might be off-the shelf, customized or custom designed by a lecturer. E-learning system s infrastructure, services and content are complementary to each other. Content is the core of LMS and LCMS. And most services are delivered through the LMS and authored in the LCMS.

First we look at the general requirements for an e-learning system and identify how the TILE system meets these needs, a good starting point is as proposed by Singh (2000):

- *Accessibility*. Knowledge can be access at anytime and anywhere.
- *Flexibility*. The e-learning environment can be customized to an organization's needs.
- *Extensibility*. The system must allow for additional components to be integrated easily.
- *Reusability*. Content can be reused by creators or consumers
- *Interoperability*. The system should allow content and other data to be exchanged and shared by separate tools and.
- *Scalability*. The system should permit access to potentially hundreds of thousands of users and large content repositories.
- *Security*. The security of data, information, or knowledge should be promise in the system.
- *Standards compliance.*

TILE meets the first of these requirements, though the use of web-based technology, and the second and third through the use of a data-base application and servelets that communicate with it. Points 4 and 5 are achieved by dividing the course into structural components that are held in the data-base and learning objects that are created in an open environment and linked using the authoring tool described here. Of course the latter also requires the us satisfy 8, by using meta-data standards, (Rover, 2002) for describing the course and possibly other standards for packaging the material so that it can be migrated between different systems, (IMS 2001). Finally, if a student learning model is used, then there must be standards to describe this model and how the course material relates to it (EML 2001). TILE provides a unique solution to point number 6 by distributing a part of the server s functionality onto the student s computer. Thus the more students that use the system, the more computing power is brought into play. Although this sounds simple, it actually introduces a non-trivial, distributed database synchronisation problem and many security issues, as described in Zhang (2001).

In addition to the above features we believe there is also a requirement for strict version control, to enable many users to concurrently author a single course unit or parts thereof. This should allow for multiple versions, change audits, rollback to earlier versions and purging of old versions. This is similar to good software engineering practices in source code control systems..

The division of labour in e-learning

An LCMS focuses on content creation, reuse and management, and can says Lennox (2001), compress the lifecycle of capturing, delivering, managing and measuring knowledge and learning reuse content in many different ways. Content can be selected and used at very granular level. The general goal of course is to provide a lower content maintenance cost. An LMS, on the other hand focuses on delivery, learning activities and student competency management. They have three common areas: content, user and management (Rengarajan, 2001). Integration of LMS and LCMS together has the advantage of sharing a common content repository and unified schema, and this has always been the aim of the TILE system.

THE TILE LCMS

TILE is an integrated system for the management, authoring, delivery and monitoring of education at a distance. TILE is different from current web-based e-learning systems, as TILE aims to provide more accessibility, flexibility and scalability to the various users (institution administrators, lecturers and students). It has an unique architecture which allows users to access the LMS no matter they have internet connection or not (Gehne 2001).

Before we introduce the TILE LCMS, several concepts need to be defined. First we define a *Learning Object* to be a self-contained piece of educational material that contains content and/or assessment based on specific learning objective. We also define the *Learning content* as the material that is used to convey subject matter; it may include raw media elements such as text, graphics, audio, some form of interaction etc. We define *Metadata* is the data which is used to describe a learning objects, an example is the LOM model (IEEE 2002). An LCMS uses metadata to organize, search, reuse or to protect learning objects and contents. A *Course unit* is defined as a component of a course, possibly the whole course that comprises a set of structured learning objects and where both the learning objects or the nodes in the structure may have metadata associated with them. This structure may provide hierarchy, for example as an index to the material, or it may provide precedence, such as a set of prerequisites. TILE supports both of these.

The TILE LCMS does or will implement the following features:

- Provide one or more repositories to store learning content.
- Maintain the structure, metadata and learning objects as separate entities. The former two are held as relations in a database and the latter are treated as standard html objects and referenced by URLs.
- Provide three level of re-usability, e.g. course unit, learning object and raw media material can and should all be re-usable.
- The system should allow the author to create learning content and raw media elements in an open environment and not constrain the choice of tools used.
- A version control system and protocols to allow collaboration of multiple authors on a single course unit. The system will keep track of all versions and who was responsible for any changes until old versions are purged.
- Authorization is imposed by course unit, at any level. Only authorized users can update the learning objects or metadata.
- Intellectual property protection features will be provided to allow the author to decide who can re-use the content and there is support for communication between the authors and the re-users.
- In addition to metadata, personalized labels can be applied to course units and learning objects to facilitate searching.
- The repository of LMS is a subset of the LCMS repository. Only one version of an LCMS course unit and learning objects will be visible to the learners, the published version. The TILE delivery system already supports versioning and automatic synchronisation with newly published versions of course units and learning objects.
- We will support a range of standards for importing and exporting compliant learning content, e.g. SCORM (Rover 2002), IMS (IMS 2001) and possibly EML (EML 2001)

The problem with standards is that there is not just one. There are many organizations developing learning standards, these include Institute of Electrical and Electronics Engineering (IEEE 2002), the Advanced Distributed Learning Network (ADLNet), the Aviation Industry CBT Committee(AICC) and the Instructional Management System Global Learning Consortium (IMS). The Sharable Content Object Reference Model (SCORM) is a set of interrelated technical specifications built upon the work of the AICC, IMS and IEEE to create one unified content model (Rover 2001). The US federal government has already announced that any e-learning product it uses must be SCORM-compliant. That is just one factor that is driving the acceptance of SCORM as a de facto standard for e-learning (Brennan 2001). This new standard however, does not fully support student learning models, course unit personalisation and how the learning models interact with the content. Figure 1. Shows a conceptual model, taken from a presentation

on EML by Jocelyn Manderveld of the Open University of the Netherlands. EML is one of the developing standards for modeling this kind of interaction in courseware.

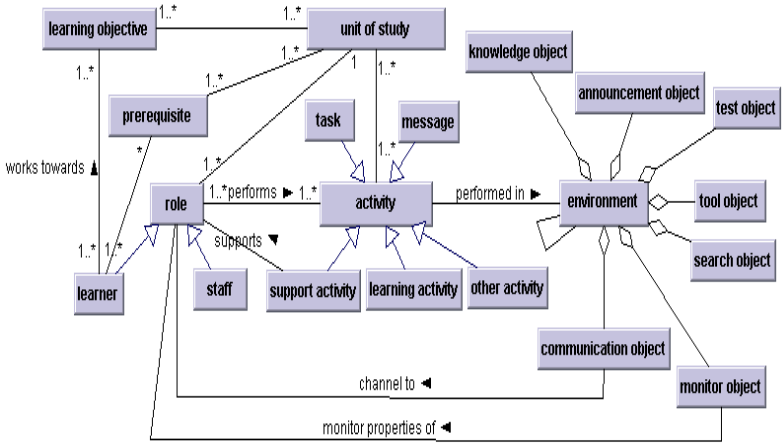

Figure 1. The EML conceptual learning model.

EML is not the only development in this area but it is probably the most complete. Its XML binding, a technical manual and further details can all be found at the EML web site (EML 2001). Others standards in this area include the Tutorial Markup Language (TML), which was developed by the University of Bristol, UK, and is described in http://www.ilrt.bris.ac.uk/mru/netquest/tml/. TML is an interchange format designed to separate the semantic content of a question from its screen layout or formatting. The language is designed to support several different types of question within the same content model and is essentially a super-set of HTML. Another is the Learning Material Markup Language (LMML). LMML is an implementation of the XML binding of the teachware-specific meta-model described in (S ss 2000). LMML is extensible and it therefore represents a family of various languages. There is a short introduction in the LMML tutorial at http://daisy.fmi.uni-passau.de/pakmas/lmml/11/doc/en/html/. These are just a few of the ongoing developments. It should be noted that because SCORM is a set of interrelated standards, developments such as EML may yet be incorporated into it.

TILE LCMS Architecture

TILE LCMS architecture is illustrated in Figure 2. It shows that on the education providers computer(s) the TILE LCMS comprises a LCMS server and an LMS server and labeled versions of courses are published from one to the other. Each has its own repository for storing course units and learning objects. The TILE LCMS client is the authoring tool described in this paper and it is a Java application on the lecturer or teacher s computer. It is used to edit and bind raw media segments, learning contents, learning objects and course units locally, even while off-line. If the LCMS user logs into the server, they can browse all the course information held in the repository and retrieve information to be held locally, either for updating, or for inclusion into other course units, either as is or in modified form. The user can browse any version of course unit structure, check out course units or learning objects (or both), check in new course unit, check in edited course unit that the user previously checked out, import or export a course unit that is compatible to SCORM standard.

Security is clearly important and the TILE LCMS server is in charge of the end user s authentication and authorization, managing and delivering learning contents. For example, all material in the development database has capabilities. Controllership can be allocated by administrators to course units and can then be delegated to others, so that at any point in the tree there will be various permissions on who can access and modify that material.

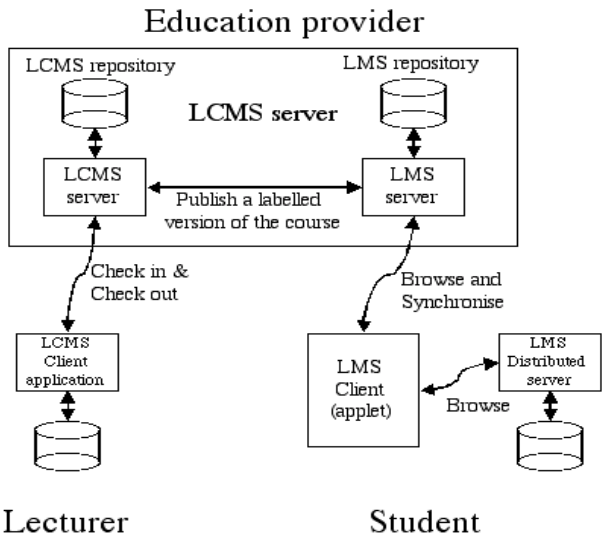

Figure 1 TILE LCMS architecture

TILE has two databases on the education providers computer(s), one is the LCMS or development repository, the other is LMS or publication repository. A history of all versions of course units, learning object and learning content are kept in the LCMS database. Only one version of any course unit will be published to the LMS database.

DESIGN OF TILE LCMS

User roles

There are three user roles in the TILE LCMS system. They are *course administrator*, *course controller* and *reviewer*. A course administrator is able to set up new courses, and specify course controller. A course controller is the person who has the responsibility for the course unit, e.g. author or maintainer of the specific course unit. A course reviewer is the user who can access the LCMS but has no responsibility with the specific course unit. A reviewer for example will be able to reuse material but will not be able to modify it, although he will be able to modify some part of the corpus of material so that he can copy structure there. Course units of TILE LCMS are organized in a tree like structure. Learning objects are located in the leaves of the tree. Sharing is achieved by copying the structure of a course unit but by sharing the learning objects. The data requirements of the structure are small compared to the learning objects, which may be multimedia. Also reuse may require re-annotating nodes with different meta data for different purposes. All copying of course units is deep, i.e. a node will always be copied with all of its sub-structure right down to, but not including, the learning objects.

Authority control applies to every node of the structure tree. Access authority propagates down the tree, i.e. if the user is the controller of one node, he is the controller of all the descendent nodes. A controller of one node can also add other controllers to its descendent nodes.

This division of user roles helps to control the integrity of the course. At the same time, the mechanism of applying authority control to every node of the course structure increases the flexibility of concurrent, multi-person development of the course.

Check in and check out mechanism

TILE LCMS uses version control method to manage learning contents. The system allows a controller to check-out and check-in course units. While course reviewer can only check out a course unit for reference or re-use. For example a controller may develop a complete course locally and then check it in to the repository. The controller may also check out that course unit, edit it and check it back in again. Learning objects may be shared between courses and it is possible to author without creating any content at all by

simply reusing existing learning objects. In this case only structure will be created and even this structure may be a copy of structure found in another course. During the editing process, as material is checked-out and back in again, the server database will keep track of all changes made to course material with an audit trail to identify who has made what change and to what data and when. The implication of this requirement is that the metadata for a course structure being developed may contain multiple instances of a node if that node has been changed. The system must therefore keep track of a version number, associated with each node, automatically updates it whenever a change is made.

Since one course unit might have multiple controllers, a locking mechanism is used to ensure a consistent update. Checking out is divided into two types: lock then check out and check out without locking. Any LCMS user may check out a course unit without locking. But only unit controller can lock the unit and then check out. Locking a unit implied locking all sub-nodes below it. After that, only the single controller holding the lock will be able to perform a check-in. The protocol for checking-in, locking and checking-out is as follows:

- Any TILE LCMS user may check out a unit of a given course.
- Only one of the course controllers for a unit node may lock that node.
- The lock is propagated down the structure tree but not to the learning objects. A separate locking mechanism is provided for the learning objects
- Only one controller can hold a lock for node at any given time. Any other controller attempting to lock this entry will be notified the identification of the person currently holding the lock.
- Only the lock holder may check in updated contents.

An author, who has changed some checked-out structure that is subsequently updated by someone else, by locking and checking-in, is responsible for checking out the new material and transferring their changes to the new structure. Warnings will be given by email to anybody who has structure checked out that is updated by somebody else checking in.

Labeling

The TILE LCMS repository keeps multiple version for each course structure node. By default the user will only see the latest version in the sequence of changes, although an interface will be provided to view previous versions. The system supports storing author-defined labels for given versions of the structure node. This label is used for defining an author-specified version of the course; it may be a development version, it may be a test version or it may be a version that has been identified for publication to the course delivery system.

Open Environment for Learning Content Authoring

The TILE LCMS client does not put any restriction on what kind of learning content and raw media segment authoring tool the user should use. The user can choose their favorite tool and only need to let the TILE LCMS client know the tool when it is used for the first time. After that the system will automatically launch the authoring tool for the user, whenever they wish to edit the raw media or learning objects that ahs been created with it. There must also be a mechanism for uploading any files involved, this may include both source and raw media data. If an application is not scriptable, the user will have to locate the information locally for the LCMS client to store it in the repository, when it is checked in.

Course Structure and Learning Object Authoring Tool

Course units are organised into a tree like structure, with the root node being a that unit that is identified by the learning institution as a component of its programs. It is at this level that the administrator passes over control to the unit controller. Course units below this root node are created by the course controller to structure the course pedagogically or administratively, for example the unit may be shared between many instructors, each having control over their own sub-units and each having permission to edit their own unit. In this way a course unit is recursively defined.

There may also be precedence relationships established between the various units, even though they not be a part of the same root unit. These relationships form a set of prerequisite constraints and may be used to control the capabilities of students to view material.

Metadata is used in TILE for indexing, delivery, management and re-use purposes. It is used to describe both course units and the learning objects. The authoring application and the delivery client both use metadata searches for locating content.

Course structures are managed separately from the learning objects. This allows the user to insert and remove learning objects to/from a course unit s structure quite freely. It also means that a learning object can be shared by many course structures without duplicating the learning object. Indeed a learning object may be an external URL. The update of a learning object therefore, will be visible in all the courses that reference it.

The course structure and metadata in the TILE LCMS are SCORM compatible.

XML and SCORM and Cross-platform issues

When the TILE LCMS user constructs a new course structure, they will first construct it locally on their own computer s disc. Alternatively the user may check out an existing structure from the LCMS server, in this case, all the related information needed is downloaded from the LCMS server and also stored locally for browsing and updating. XML files are is used on the client side to manage the course structure. This decision allows a standard application to access the data without the requirement for installing a database server. The XML data structure is, in any case, the preferred method of communicating the data between the LCMS server and the authoring application. XMLs characteristics of being totally text based and conforming to a well established standard provide simplicity in dealing with both firewall and cross platform issues. SCORM also uses an XML course structure format to move a course from one LMS to another. The XML file will be compliant with the SCORM XML DTD.

The TILE LCSM is completely implemented in Java. So it can be supported across many platforms. Careful design of the system has also made it firewall friendly.. Communication between the client and the server will use HTTP protocol, which may embed XML data. Because most firewalls pass traffic on port 80, which is used for the http protocol, then this allows a means by which a client server application can work through the firewall.

CONCLUSION

We have described a learning content management system to be used in conjunction with the TILE learning management system. This paper has described the general requirements for such a system, borrowing from the area of software development we have added further issues to those generally regarded as being desirable in such a system, such as version control and concurrent development of material. We have described our planned implementation and described standards issues, which will determine the schemas used. A prototype implementation of the TILE Authoring client has already been undertaken (Wang 2002).

ACKNOWLEDGEMENTS

We gratefully acknowledge the support from the New Zealand New Economy Research Fund, NERF, for support of the Technology Integrated Learning Environments, TILE, project.

REFERENCES

EML (2001) EML downloads, http://eml.ou.nl/

Gehne, R., Jesshope, C.R. and Zhang, J. (2001), Technology Integrated Learning Environment - A Web-based Distance Learning System. Proceedings of IASTED International Conference 2001, Internet and Multimedia Systems and Applications. Hawaii, USA. ISBN 0-88986-299-0. pp1-6.

IEEE (2002) IEEE Learning Technology Standards Committee (LTSC) IEEE P1484.12 Learning Object Metadata Working Group home page, http://ltsc.ieee.org/wg12/

IMS (2001) IMS specifications, http://www.imsproject.org/specifications.html

Koolen, R (2001), Knowledge Mechanics , Learning Content Management System; the 2nd Wave of e-Learning!, http://www.internettime.com/itimegroup/lcms/lcms2ndwave.pdf

Lennox, D (2001) , Managing Knowledge with Learning Objects, The Role of an e-Learning Content Management System in Speeding Time to Performance, http://www.internettime.com/itimegroup/lcms/wbt_Mngknw.pdf

Raghavan R (2001) , LCMS and LMS, Taking Advantage of Tight Integration, http://home.click2learn.com/en/downloads/lcms_and_lms.pdf

Rosenberg, M (2001), E-learning Basics: A guide to the e-learning industry, http://www.elearningmag.com/elearning/article/articleDetail.jsp?id=2779 retrieved date January 26, 2002.

Rover, R (2002), Shareable Content Object Reference Model Initiative (SCORM), http://xml.coverpages.org/scorm.html

Ryann K. Ellis (2001), LCMS Roundup http://www.learningcircuits.org/2001/aug2001/ttools.html

Singh, H (2000), Achieving Interoperability in e-Learning, http://www.learningcircuits.org/mar2000/singh.html

S ss, C (2000) A Meta-Modeling Adaptive Knowledge Management: Approach and its Binding to XML (2000) (http://daisy.fmi.uni-passau.de/db/literatur.php3?key=S00)

Wang, Y (2002) *An authoring tool for structuring and annotating on-line educational courses*, M.Sc. Thesis, Massey University.

Zhang, Z (2001) *A Feasibility Study for the Design of a Web-based Course Delivery System*, M.Sc. Thesis, Massey University.