# A SCORM-Conformant LMS

Ch. Bouras,
Computer Engineering and Informatics Dept., Univ. of Patras & Research Academic Computer Technology
Institute, GR-26500 Rion, Patras, GREECE
bouras@cti.gr

M. Nani
Computer Engineering and Informatics Dept., Univ. of Patras & Research Academic Computer Technology
Institute, GR-26500 Rion, Patras, GREECE
nanim@cti.gr

Th. Tsiatsos
Computer Engineering and Informatics Dept., Univ. of Patras & Research Academic Computer Technology
Institute, GR-26500 Rion, Patras, GREECE
tsiatsos@cti.gr

**Abstract:** In this paper we propose a sample Learning Management System (LMS) that will be conformant with the SCORM v1.3 Specification. In particular, the sample LMS we propose will make use of both, the SCORM Run Time Environment, in order to communicate with the learning content, and a sequencing engine so as to sequence discrete learning activities in a consistent way, based on the learners' interactions with the content. Even if the SCORM v1.3 Specification is not yet finalized, the proposed LMS intends to be an attempt to test and evaluate the specification in its present form.

## Introduction

The need to have learning content that can be easily accessed, reused for different educational purposes and managed by different LMSs, led the scientific community and industry in an attempt to create learning technology specifications. The most significant outcomes of these efforts are the specifications regarding learning resource metadata, aiming to fully describe a learning resource regardless of a specific learning context or educational purpose, learning content packaging, for moving whole instructional units, together with information about content's intended behavior, from one system to another and communication protocols between learning content and LMSs as a result of the need to be interoperable.

The most significant specifications for metadata are the IEEE 1484.12.1 Learning Object Metadata or LOM (IEEE Learning Technology Standards Committee), approved as an IEEE-SA standard, the IMS Learning Resource Metadata Specifications (IMS Global Learning Consortium) and the ADL's SCORM (Advanced Distributed Learning) that, both, reference LOM. As regard content packaging, AICC (Aviation Industry CBT Committee) firstly introduced the concept of the Content Structure Files, which describe the structure of an entire course. The IMS Content Packaging Specifications describe a whole package though a special file called manifest, whereas the IMS Simple Sequencing (SS) Specification enable content authors to specify more robustly the content's intended behavior. Finally, SCORM references the aforementioned IMS Specifications, whereas previous versions have made use of the course structure concept introduced by AICC. Concerning the communication between LMSs and learning content, AICC firstly presented Run Time functionality by means of an API and a Data Model. ADL, based on the AICC's experience, has presented its own API and adopted an extended version of AICC's Data Model in order to integrate the IMS SS Specification. ADL's API presents faster rate of adoption, as the learning content needs to support only two API calls.

From the above it is clear that the SCORM Specification has precedence in comparison with the other ones, as it refers to them while making one step further. This is, actually, the main reason why more and more learning technology products tend to be compliant with it. In this paper we are mainly interested of a SCORM-compliant LMS, as the LMS is considered to be the most critical component of any e-learning solution (THINQ). LMSs contribute to the improvement of the speed and effectiveness of a training process and provide many benefits for both, organizations and individuals (THINQ), (Bean, 2002).

In the following sections we shortly describe the SCORM Specification and our proposal for the LMS. In particular, we present the LMS's expected functionality and architecture as well as our vision regarding the system implementation. We end up with conclusions and plans for future work.

## SCORM Description

SCORM Specification has been developed by the ADL initiative with the intention to meet the DoD's requirements for reusable, accessible, interoperable and durable Web-based learning content. It specifies a Content Aggregation Model (CAM) and a Run Time Environment (RTE) for learning objects.

The CAM distinguishes the learning content into four components: a) *Assets*: They concern learning content in its most basic form that can be delivered in a Web client. b) *SCA*: A Sharable Content Asset (SCA) is a collection of one or more assets that can be launched by an LMS. c) *SCO*: A Sharable Content Object (SCO) is, also, a collection of one or more assets. However, as opposite to a SCA, it can communicate with the LMS using the RTE, thus allowing the LMS to track down the learners' interactions with the content. d) *Content Aggregation*: It concerns a content structure that can be used to aggregate learning resources into cohesive instructional units, apply structure and associate learning taxonomies.

CAM specifies five types of metadata application profiles to describe Assets, SCAs, SCOs, Activities and Content Aggregations. Moreover, it concerns the packaging of the manifest and the related physical files into a zip file (called Package Interchangeable File or PIF) or on a CD-ROM. With the integration of the IMS SS Specification, CAM supports the consistent sequencing of learning activities. These may be instructional events in a content resource or aggregation of activities and they can be associated with author defined sequencing information. In run tine, each activity experienced by the learner is associated with tracking status data, which may affect the sequencing process. A Tracking Model captures information regarding the learners' interactions with the resource associated with an activity.

The RTE includes a launch mechanism, an API and a Data Model. The launch mechanism refers to the LMS's responsibility to determine the learning resource (SCA or SCO) that will be next delivered to the learner. The API is a communication mechanism between the LMS and the content. It is provided by the LMS, but it is SCO's responsibility to locate the API adapter and make the appropriate API calls in order to communicate and exchange information with the LMS according to the Data Model. Some of the Data Model elements relate to the Tracking Status Model for the respective activity.

## Our Proposal for LMS

The LMS we propose intends to be an asynchronous learning system for delivering and managing learning content, which adapts accordingly to the learner's progress and performance. This means that learners with difficulties in satisfying a predefined learning objective associated with a learning activity should be able to experience additional activities to improve their cognitive skills. Learners could, occasionally, not allowed to experience an activity for more than a predefined period of time or a number of attempts. Objectives and limits are set by content authors so as to serve specific instructional strategies.

According to the aforementioned goal, the LMS is expected to provide a suite of functionality, such as the ability to maintain a list of registered users; provide login/logout services; allow registered users to change their preferences; import SCORM v1.3 learning content that can be grouped into a variety of categories; inform learners of the new available categories and courses; allow learners to view the courses contained within a category and register/un-register for a particular course; allow learner to view what is contained within a course; provide learners with the means that will help them navigate through a course's learning activities; inform learners in the case an attempt to experience a learning activity is not allowed; monitor the users' interactions with the content, as specified within a resource; conditionally affect, in run time, the sequence of the activities according to the learner's progress and performance; provide bookmark facilities and report on learner's performance on assessments, if any, of a course.

As mentioned above, the LMS will maintain a list of registered users. LMS's administrator will be responsible for adding to or deleting learners from the list, motivated by the users' intention to register to or un-register from the LMS. He will, also, be responsible for defining the different categories the imported courses can be divided into, as well as for importing new courses. These could be developed using any SCORM v1.3-compliant authoring tool and they will be in the form of a PIF (zip format). Registered learners will be able to enter the LMS through a login session and explore a list of the non-empty categories of the available courses. They will, also, be able to change their preferences and be informed of the new available categories and courses. A learner can select the course(s) he wishes to attend/delete and make the appropriate registration(s)/deletions. When he selects to view a course, he will be provided with the course's table of contents as well as other means that will help him navigate through the learning activities contained in that course (e.g. "next"/"previous" buttons). If any of the learner's navigation request violates the sequencing rules specified by the content developer, the learner is being informed of the wrong navigation event he has triggered. Otherwise, the learner can view the learning resource associated with an activity, as well as to interact with it.

The sequencing rules are based on learner's progress and performance and affect the availability of the activities the learner is allowed to experience. In particular, if a learner does not succeed in satisfying a learning objective associated with a learning activity (e.g. pass a pretest), he should be prompted to experience additional activities that will help him to succeed. Moreover, if an activity is associated with an attempt or time limit, the LMS will inform the users of this bound as well as whether this bound has been met or exceeded. The LMS will, also, present a list of auxiliary resources, which could be engaged with a particular activity, and provide bookmark facilities. The overall sequencing process will be controlled by a sequencing engine, which is provided by the LMS in order to assure that the sequence of the learning activities conforms to the instructional strategy. This engine will make use of the information gathered from the LMS, which concerns the status of the learning activities, as well as information from the manifest file.

The LMS will be able to report on learner's progress and performance on assessments contained in a course. This will help learners to be informed of their cognitive skills in various instructional units as well as to be motivated to achieve a better performance in subsequent tests. In order to track the learners' interactions with the content, the LMS will group the information it can exchange with the traceable learning content (SCO) according to the SCORM Data Model. This information concerns, among others, student's profile and preferences, questions and tests, objectives and status. The information exchange will be realized by means of the API adapter, provided by the LMS, and the appropriate API calls. A SCO will, then, be able to locate and use the API adapter so as to initialize or finish communication with the LMS, read/write data from/to the LMS, commit the changes made and be informed of the possible error states.

Being SCORM-compliant, the proposed LMS has a standardized way to be interoperable with the learning content. This means that the LMS is able to manage, track and deliver learning content developed by any SCORM-compliant authoring tool or exported by any SCORM-compliant LMS. This is due to the fact that the learning content and all the required information about its structure and intended behavior is contained in the manifest file of the PIF. With the integration of the IMS SS Specification, SCORM enables instructors to translate learning strategies into sequencing rules and actions, which in turn are associated with the activities a learning experience consists of. Any SCORM-compliant LMS is then able to sequence activities in a way consistent with the instructional purposes and deliver learning resources based on the learner's interactions with them. This allows the development of content that adapts student's progress and performance, thus serving our LMS's goal and approach. Following SCORM, the proposed LMS is able to exploit the SCORM API and Data Model as the standardized means for monitoring the students' interactions with the content. Information that can be exchanged between itself and the content concerns the session and total time a student spends on an activity, the accomplishment or not of the learning objectives, the student's performance on assessments, how, why and where a student has left off a SCO, student' s preferences, SCO enabled navigation events as well as comments from/to the LMS.

## System Architecture and Implementation Issues

The LMS we propose will be based on a client-server architecture, as shown in Figure 1. In particular, the LMS will consist of the following components: a) *A Web client*: Learners will be able to access the learning content and other LMS's services, through a browser window. The LMS will come with a variety of user interfaces in order to manage the different functions, such as registrations, reports, student's profile management and course administration. The LMS will provide a browser window that will encapsulate a SCORM API adapter. The content will be delivered in a frame of this API-enhanced window. In another frame, the LMS will provide learners with navigation controls to help them navigate through the content and trigger the overall sequencing process. b) *A server*: The LMS's server will deliver learning resources in a consistent way, according to the learners' requests and based on the information associated with each available course. It is expected to consist of a *Package Handler*, responsible for accessing and extracting the manifest file contained in the package, a *Manifest Handler*, for accessing information contained in the manifest file and storing it into the database, a *RTE Data Model*, for providing the communication mechanism between the server and the SCOs and, finally, the *Sequencer*. The last, triggered by various learner- or SCO- enabled navigation events, will capture information gathered through the RTE Data Model (which then is mapped into information regarding learning activities) or from the manifest file in addition to start the overall sequencing process. c) *A database:* The database will be used to provide data persistence. It will store information regarding data extracting form the manifest file, data resulted from the communication between the server and the various SCOs, as well as information about the registered users, the different courses and categories and the learner's performance on assessments and exercises.

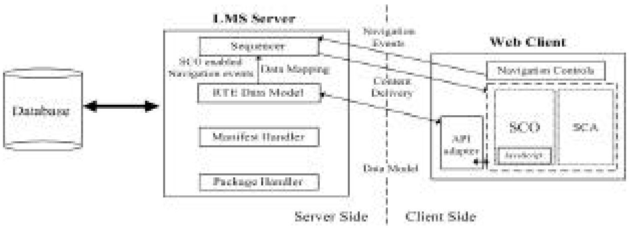

**Figure 1:** Architecture of the proposed SCORM-compliant LMS

For the implementation, we plan to exploit a JSP- and Java Servlet-enabled Web server, such as the Jakarta-Tomcat 4.0.1. JSP will be used to implement the user interface and provide connectivity with the database through a JDBC connection. Java Servlets will be used for handling the communication between the RTE and the SCO and for controlling the sequencing of the learning activities. Java language will be also used for implementing the remaining server components. The database is planned to be a mySQL database. On the client side, the API adapter is planned to be a Java applet embedded into a frame of the LMS's window, which will be responsible for presenting the learning content to the students. SCOs will make use of JavaScript functions in order to communicate with the LMS through the API adapter. The implementation is expected to employ the Sample RTE v1.2.1 developed by ADL.

## Conclusions and Future Work

SCORM constitutes an important attempt towards learning content's reusability and interoperability. The interest it gains from the scientific community and industry as well as the fact that it references the most significant learning technology specifications, ensures its widespread acceptance in the near future. The LMS we propose aims to be a first attempt to test and evaluate the upcoming SCORM version and an early exploitation of its promising potential. Our next steps involve the implementation of the proposed LMS, based on the ADL's Sample RTE. In this phase, we will keep sight of the changes will be made in SCORM v1.3 Application Profiles, till the release of the SCORM v1.3 Specification, in order to adopt the changes. Thereby, we will assure full compliance of our LMS to the upcoming SCORM version.

## References

IEEE Learning Technology Standards Committee (LTSC), http://ltsc.ieee.org
IMS Global Learning Consortium, http://www.imsproject.org
Advanced Distributed Learning, http://www.adlnet.org
Aviation Industry CBT Committee, http://www.aicc.org
THINQ's Research Department, LMS, Backbone of Your Training System, THINQ White paper available at: http://www.thinq.com/pages/white_papers_pdf/LMS%20The%20Backbone%20of%20Your%20Training%20System.pdf
Martin Bean (2002), Final Thoughts - The Real Advantage of an LMS, *Certification Magazine*, http://www.certmag.com/issues/jun02/contrib_bean.cfm