

Learning Objects in Open Source environments

Lic. Francisco Javier Díaz

Laboratorio de Investigación en Nuevas Tecnologías Informáticas – Universidad Nacional de La Plata
La Plata - Buenos Aires – Argentina - Calle 50 esq. 115 (1900)

Tel/Fax: 54 - 221 - 4236609/11

jdiaz@unlp.edu.ar

Lic. Alejandra Schiavoni

Laboratorio de Investigación en Nuevas Tecnologías Informáticas – Universidad Nacional de La Plata
La Plata - Buenos Aires – Argentina - Calle 50 esq. 115 (1900)

Tel/Fax: 54 - 221 - 4236609/11

ales@info.unlp.edu.ar

Abstract: In the recent years, e-learning has become most important due to the extended use of ITs. This has given place to several standards that allow the development of high-quality educational content. The SCORM content creation process requires a series of stages for the identification of learning objects and the incorporation of metadata and communication functions with LMS. Through this communication, it is possible to track the learner's progress during the course. However, incorporating this communication is not a simple task, since to do so, it is necessary to know the standard implementation in detail. This article proposes an Open Source tool, with the goal of making easier the incorporation of functions implementing communication between learning objects and LMS. The tool extends the functionality of an HTML editor, so that the author of a web page has the possibility of adding communication-oriented predefined components at the stage of creation.

Introduction

In the recent years, the learning process has evolved considerably towards what is known today as e-learning (electronic learning), due to the wide use of Information and Communication Technologies that support this new method of education, also called *distance learning*. For this reason, several international organizations are working with the only goal to produce e-Learning standard that would led to the development of e-Learning systems, tools and methodologies which combined would produce high-quality e-Education. (Radoslav, Ganchev, O'Droma 2005). The most popular standards in today's market are: IEEE LTSC LOM [<http://ltsc.ieee.org/wg12/>], AICC [<http://www.aicc.org>] and ADL SCORM [<http://www.adlnet.org/>], although there are some others such as Dublin Core [<http://dublincore.org>] and EML [<http://eml.ou.nl/eml-ou-nl.htm>]. In the evolution of these standards there has been a convergence towards ADL SCORM, that comprises IMS, LOM and AICC and is nowadays the best known international standard.

The fact of having educational material that is reusable, durable, accessible and interoperable and that can therefore be used with any learning platform or LMS (Learning Management System) is very encouraging. However, although the mentioned standards appeared years ago, the creation of standards compliant learning objects is still a hard task which requires certain specific technical knowledge. The generation of reusable content adds complexity to the sole content generation process (González Barbone, 2006). This is why it has taken so long for e-learning to be taken as a universal learning method when speaking of new trends in education. When using the SCORM standard, there are tools that make the creation and validation of objects easier, but do not cover all aspects. Organizations attempting to create their own material come across many difficulties along the way, because they must be familiar with the standards characteristics in great detail, and this very rarely happens.

Implementing a way of communication between LMS and content is not a simple task, since the developer must know both the JavaScript language and the data model the standard works with to achieve an information exchange. This communication is of great help to record the data of the students and their progress during the course.

This article provides a detail of the stages to be followed in the creation and use of learning objects and presents an Open Source environment based tool to facilitate one of the aspects of this process. The goal of this tool is to make easier the incorporation of functions implementing communication between learning objects composed by HTML pages and LMS. The following sections show the stages of the SCORM content creation process, its difficulties and the functionality of the proposed tool.

The task of creating SCORM content

The SCORM standard is focused especially in the creation of educational material which is reusable, interoperable, durable and accessible regardless of which content delivery and management system is being used (Carnegie Mellon, 2004). For this, it suggests the use of *Shareable Content Objects (SCOs)*, known as *assets*, which are composed by educational material and can include objectives, self-assessments, simulations, etc. These objects can be launched in a SCORM run-time environment, provided by a LMS or a repository of learning objects capable of fulfilling SCORM requirements. Moreover, the use of metadata allows identifying and finding the educational material, for example, in online content repositories.

SCORM makes a clear distinction between the role of the LMS and the role of the content itself. Confusing these roles is one of the main causes of the lack of interoperability and reusability (Forth, Childs, 2003). For example, when using SCORM, all information in relation to the student is stored in the LMS and not within the educational material.

The creation of SCORM compliant content is a process that consists of several stages, as described in the Learning Systems Architecture Lab (LSAL) Carnegie Mellon guide (Carnegie Mellon, 2004). In first place, the task is to identify and design the SCOs. It is important to take into account that one way of assuring the instructional integrity of the SCORM content is to make each SCO represent an independent learning unit.

If we are working with a course previously designed and we want to transform it into SCORM standard, it must take place a reengineering process. The first step is to organize the content by establishing a hierarchy composed by modules, which are in turn composed by lessons. The following step is to identify the potential audiences. This is very important with regard to future reusability of contents. And then, to identify each of the SCOs that will afterwards constitute the material. The paper “Standardizing educative content with SCORM: a case study” describes how a course which was originally designed specially for the “Plan of Training Teachers on the usage of ITs” required by the National Ministry of Education was converted to SCORM. On this plan, the Faculty of High Computer Sciences of the National University of La Plata works as trainee agent and had organized the first pilot project (Díaz, Schiavoni, Osorio, Amadeo, 2006).

Once the SCOs have been defined, the incorporation of metadata is of great importance, since it allows the objects to be efficiently discover. The metadata scheme used by SCORM is based on the LOM (Learning Object Metadata) standard, that includes nine categories of fields divided in subcategories. The choice of metadata to be included depends on the type and volume of the content being created.

There are other aspects to be taken into consideration, such as the implementation of sequencing and content navigation implemented through a set of rules, and the creation of assessments or evaluations within the SCORM objects, all quite difficult tasks for non specialized authors (González Barbone, 2006) (Jun-Ming Su, 2005).

One of the most interesting aspects of the SCORM standard is that it allows communication between learning objects and LMS. Through this communication it is possible to track learner’s progress information. In the following sections we will go deeper into this aspect together with the features of the proposed tool.

The need of having metadata, packaging and a sequencing which is not ambiguous, leads to the need of the setting of a normative and its correct implementation in LMSs. These are the goals of the SCORM normative, but this normative stumbles upon difficulties when the teachers/instructional designers are trying to adopt it, because of the complexity of metadata, the communication with the LMS and the changes from one version to another (González Barbone, 2006).

In SCORM, the development stage requires additional technical experience to fulfil the reference model agreement. The programmers and developers must have basic knowledge of the required technology to apply SCORM, based on the Aggregation Model, Run-time Environment, and Sequencing and Navigation Rules for leaning objects. They must be able to organize the data accordingly and add metadata to organizations, aggregations, activities, SCOs and assets. (Pisel, Lindsey, 2004).

SCORM communication between learning objects and LMS

As we have mentioned previously, one of the great advantages of the SCORM model is that the content, represented through learning objects, can communicate information about the student to any LMS SCORM compliant. The SCORM specification determines exactly which are the pieces of information that can be recovered and updated: student ID, name, test score, time invested on each learning object and visualization preferences.

The communication is performed through an interface that standardizes the communication protocols, and provides methods to LMS to know the current objects states (initialized, finished, etc) and to exchange data.

The use of an Application Program Interface (API) provides a standard for the contents to communicate with LMS, although the implementation of this communication is transparent to the contents developer. The way in which the API is implemented is not important to the developers, but all of them must use the external interface that represents the API functionality. This interface hides the implementation details from the contents.

The API Adapter is the JavaScript-based software which enumerates the functions of the API. This adapter makes communication between the SCOs and the LMS possible, since it is the means offered to SCOs to talk with LMS.

This adapter was designed to achieving independence between the contents and the execution environment. Hence, the teaching environments developers must only take the API functions into account, and the content creators must not worry about the type of system or platform in which the contents will be displayed. The use of this adapter represents the total separation of contents and systems.

To establish communication, the SCORM standard uses, as well as the API, a data model that defines the data flow between the LMS environment and content. The purpose of establishing a shared data model is to assure the information on the SCO can be shared by different LMSs.

There are a number of data models under development in several standard organizations and communities. The data model of the SCORM run-time environment derives directly from the data model of AICC CMI. The elements of the data model are divided in two categories: mandatory and optional. The AICC CMI001 guide delivers recommendations and guidelines to be followed to achieve interoperability between LMSs and specifies which of the data elements are mandatory and which are optional. The mandatory data elements must be admitted by our LMS. Besides, the LMS can also provide implementation to admit some or all of the optional data elements.

According to the previously exposed, to be able to perform this type of communication it is necessary to be familiar with certain implementation details of the SCORM standard, such as RTE Management, the communication requirements exposed by the API and the management and behaviour requirements for the data model. The goal of the proposed tool is to facilitate the incorporation of the communication functions between SCOs and LMS.

The Open Source proposed tool

The task of creating SCORM content improved with all the functionalities provided by this standard requires, on the one side, to understand the characteristics and implementation of the standard itself, and on the other side, to be familiar with a set of technologies including XML, HTML, and JavaScript.

Following the stages of the LSAL Carnegie Mellon guide (Carnegie Mellon, 2004) described previously, the identification of the SCOs composing the course appears to be quite simple. In addition, anyone who has built a SCORM course will have been able to notice that when using a tool such as ReloadEditor [<http://www.reload.ac.uk/>], it is possible to effortlessly import the objects (HTML documents, images, multimedia files, etc.) and specify the structure of the course and its basic metadata. Then it is possible to produce a SCORM package which can be visualized and test on SCORM compliant LMS, such as Moodle [<http://www.moodle.org>].

Although SCORM package construction and visualization are simple tasks which are nowadays performed by Open Source tools, such as the mentioned above, one of the major difficulties is regarding the incorporation of the JavaScript functions. These functions allow the content to send and receive information from the LMS. The complexity of these functions grows in accordance with the number and detail of the data to be sent.

There are currently a number of tools that facilitate this process but they are proprietary and therefore have special requirements for their use. The goal of our project is to obtain a tool which is fully functional under the Open Source philosophy. In this case, we have focused our attention in learning objects composed by HTML pages, because they are the most commonly used ones. The idea is to extend a HTML code editor so, at the moment of creation or edition of a page, the developer can add the necessary JavaScript functions to establish communication between SCO and LMS. This functionality will be transparent for a user who has not been familiar with JavaScript

and even HTML programming. In this way, it provides the user with the possibility of incorporating communication-oriented predefined components to the page. Figure 1 shows a scheme with the role of the proposed tool within SCORM package creation process.

The methods available currently are the basic Session and Data Transference methods, implemented by the API. The first make it possible to start and close the communication session between a SCO and the LMS through the API. The latter are used to exchange values of the data model, such as: personal data of the student, tests scores, time invested on each SCO.

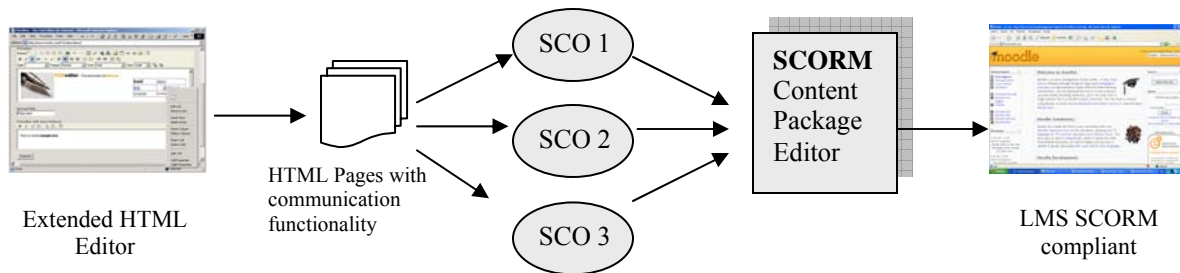


Figure 1: SCORM package creation process using the extended HTML Editor

For the implementation of this tool, different alternatives were evaluated. On the one hand, we tested the HTML editor which comes with Moodle, a WYSIWYG graphic editor. It is a very simple editor which only allows entering plain text with HTML tags. Besides composing the text directly on the editor, it also allows copying and pasting text with formats and structures from documents generated in another word processor, inserting images, tables and links. With these characteristics it represents a convenient, direct method for composing a web page without knowing the language. The disadvantage it presents for our project, is that it does not allow to write scripts, and they are automatically deleted when the page is saved. Because of these inconveniences, the editor included in Moodle was not used and we decided on an Open Source HTML editor, which could be modified to include the desired functionality. The FCKEditor [<http://www.fckeditor.net/>] was chosen for this purpose, since it has all the basic characteristics of this type of editors, offering functions for text formatting, image management, insertion of links, XHTML, etc. Moreover, it supports ASP, ASP.NET, PHP, Java and JavaScript. The current version is stable, and it can be easily integrated with any existing application.

The incorporation of the mentioned functionality consists of modifying the upper toolbar by adding new buttons that allow the insertion of JavaScript code in the page being edited. The code functions will be selected from a list and incorporated directly, requesting the user the parameters necessary for each method. In this way, the user can create web pages with all incorporated communication components and prepared to be a part of a SCO, capable of exchanging information with LMS.

Conclusions

The construction of courses based on e-learning standards seems to be very significant to the creation of new stronger and powerful kinds of educative material (Jones, 2002). The possibility of having instructional SCORM compliant material making use of all the functionality provided by the standard, requires a series of stages to be followed that involve a detailed knowledge of the standard implementation and a set of technologies, that seem to be difficult to understand to teachers. The goal of the proposed tool is to facilitate one of the stages of the SCORM content creation process, related to the implementation of communication between learning objects and LMS. This tool is Open Source based, and this fact represents a very significant contribution, because today not much tools exist under this philosophy. Because there is a gap between the pure technical nature of the SCORM model and the global process of courseware development for teachers/instructional designers, the following stages in this research will be to aim at the most difficult tasks, attempting to facilitate the construction and use of e-learning content.

References

Carnegie Mellon (2004). *Best practices Guide for Content Developers*. Carnegie Mellon, Learning Systems Architecture Lab. 1st. edition including updates to Section 7, dated 2004-04-05. <http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/>

Díaz, J.; Schiavoni, A.; Osorio, A.; Amadeo, P. (2006). Standardizing educative content with SCORM: a case study, *Proceedings of 17th International Conference of Society for Information Technology and Teacher Education (SITE)*, Association for the Advancement of Computing in Education (AACE), Orlando, Florida, March 20th, 2006

Forth, S.; Childs, E. (2003). *White Paper on e-Learning Specifications and Standards*. Recombo Inc, Innovative Training Solutions Inc.

González Barbone, V. (2006). Creación de Cuestionarios en Objetos de Aprendizaje Reutilizables. *Proceedings of Congreso Regional de Telecomunicaciones, I*. MVD Telcom. Montevideo, Uruguay.

Jones, E., (2002). Implications of SCORM and Emerging E-learning Standards On Engineering Education. *Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference*, The University of Louisiana at Lafayette, March 20 – 22, 2002. Available online at <http://falcon.tamucc.edu/~ejones/papers/ASEE02.pdf>

Jun-Ming Su (2005). An Object based Authoring Tool for Creating SCORM Compliant Course. *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)*, March 28th, 2005, Vol. 1, pp 209-214.

Pisel, K.; Lindsey, A. (2004). Developing SCORM-Compliant Media-Rich Graduate-Level Distance Education: A Case Study of Best Practices, *Proceedings of the 20th Annual Conference on Distance Teaching and Learning*, August 4-6, 2004, Madison, Wisconsin, Monona Terrace Convention Center. Available online at http://www.uwex.edu/disted/conference/Resource_library/proceedings/04_1333.pdf.

Radoslav, A.; Ganchev, I.; O'Droma, M. (2005). Content Metadata Application and Packaging Service (CMAPS) – Innovative Framework for Producing SCORM-compliant e-Learning Content. *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, Kaohsiung, Taiwan, July 5, 2005 pp. 274-278