

LearningOnWeb – development of a SCORM compliant Learning Management System

Norberto Henriques, Damyan Slavov and António José Mendes

Abstract: *This paper describes in general terms a new SCORM compliant learning management system (LMS), currently under development at the University of Coimbra. This project pretends to develop an environment that provides a set of tools that will support a richer experience for both students and teachers, not only allowing the utilization of high quality contents, but especially facilitating the creation of good learning contexts where students and teachers can collaborate and create real learning communities.*

Key words: *LMS, SCORM, sharable content objects, reusable learning objects, Java.*

INTRODUCTION

The e-learning research area has been very active in these days. The ever growing need to share knowledge without frontiers introduced the need to create new and innovative tools and frameworks that allow the seamless interchange of information throughout different systems.

The availability of wireless Internet has driven the e-U initiative in Portugal. It consists in an integrated structure that involves Services, Contents, Applications and Mobile Communication Networks (within and outside the university) for college students and teachers, aiming to promote the production, access and sharing of knowledge [7].

The project we present in this paper is a response to the challenge proposed by the Unidade de Missão, Inovação e Conhecimento (UMIC – Unit of Mission, Innovation and Knowledge), the government body that coordinates the e-U initiative. Its main purpose is to create a fully integrated learning management system that can support distance learning, but also to study and develop some innovative features, such as e-portfolio support and a framework to support context-based learning.

LearningOnWeb is being developed on top of other ongoing project, WebOnCampus. This is a web based tool to support conventional academic activities, also developed after the e-U initiative and already serving several departments at our university.

In this paper, we first introduce the main technologies behind LearningOnWeb. We will then describe the architecture of our system, as well as the current state of the project. Finally we will discuss what we expect from the system once it is concluded.

SCORM, THE STANDARD BEHIND IT

Although we are not trying to base our work mainly on contents, there was the need to have a standard procedure to deal with them. Nowadays SCORM presents itself as a *de facto* standard. SCORM is being adopted all over the world as the framework to manage and (re)use content. It provides the tools necessary to categorize, organize, share and deploy content.

SCORM promises that all the compliant content will be *RAID* enabled [5]:

- **Reusability** – the content is independent of the context. It can be used on numerous training situations with any development tools or delivery platforms.
- **Accessibility** – the content can be identified and located when it is needed.
- **Interoperability** – the content functions in multiple applications, environments, and hardware/software configurations, regardless of the tools used to create it and the platform on which it is delivered.
- **Durability** – the content will not require any modification when the platforms are changed or upgraded.

Although, in our view, the full promise of SCORM hasn't been delivered yet, we decided to make LearningOnWeb SCORM compliant, since that is a requirement of the e-U initiative.

OPEN-SOURCE – A MAIN CHOICE

Many of the platform's technical choices were already decided from the beginning of the project, since we wanted that LearningOnWeb can be fully integrated with WebOnCampus and also it seemed wise to take advantage of all the work previously done in this project. So LearningOnWeb is being developed in JAVA (using TomCat as its servlet container) and uses MySQL as its database engine.

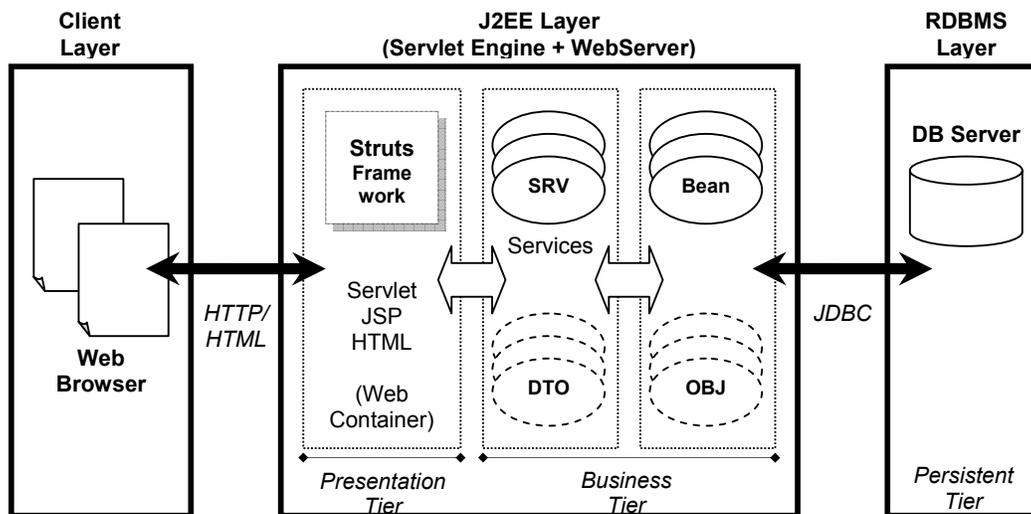
The fact that we had to adopt these technologies by inheritance isn't really a handicap, since the two systems have several common features and, hence, we can reuse some WebOnCampus components. Moreover, it presents itself as a competitive feature, due to the fact that these technologies are low cost and completely platform-independent. For instance, the WebOnCampus project, although running on a Linux server on the instances already deployed, is developed on Windows machines.

It is also a fact that JAVA is widely used as the choice for server applications. The platform independence and its performance and flexibility make it a smart choice.

The MySQL database engine also features some very interesting points, such as its continuous support and update by their creators and the support for well known database mechanisms (such as transactions and table spanning over several files).

A MAINTAINABLE ARCHITECTURE

LearningOnWeb is being developed in a three-tier architecture. Although this implies an overhead on the complexity of the system, it provides a way to centralize the control of the application and the separation in terms of features and roles of the different layers. Figure 1 describes this architecture:



- Client Layer** – presentation layer;
- J2EE Layer** – logic and business layer;
- RDBMS** – Relational Database Managing System;
- SRV** – Service classes, contain the application logic;
- DTO** – Data Transfer Objects, transfer data between the different layers;
- Obj** – data mapping objects – map the data of database tables (associated with beans);
- Web Container** – provides an execution environment for servlets, JSP components and JAVA classes;

Figure 1 - architecture

As it can be seen in the figure, the three-tier architecture is composed by the following layers:

- **Presentation Tier** – responsible for the data flow control of the application. It has the handling and routing of requests as its main task. We use the Struts framework (this framework is based on Action classes that handle requests and forward the correspondent responses) and Servlets (using TomCat as the servlet engine) in order to build this tier [1].
- **Business Tier** – responsible for the application logic. It is composed by classes of type *Services* that contain the business logic. This layer operates on the parameters (coming from either sessions or requests) passed by the web container, as well as on the obtained data from the persistent layer using *data transfer objects* (DTO) for this purpose. These DTO objects send and receive data through *object* (OBJ) and *data access objects* (DAO) classes that map (OBJ) and handle (DAO) data on the persistent layer.
- **Persistent Tier** – responsible for the data management in a persistent manner, using a *relational database management system* (RDBMS), MySQL in our case.

A MODULAR APPROACH

For our project we identified several areas that required a modular approach. Figure 2 represents LearningOnWeb general architecture:

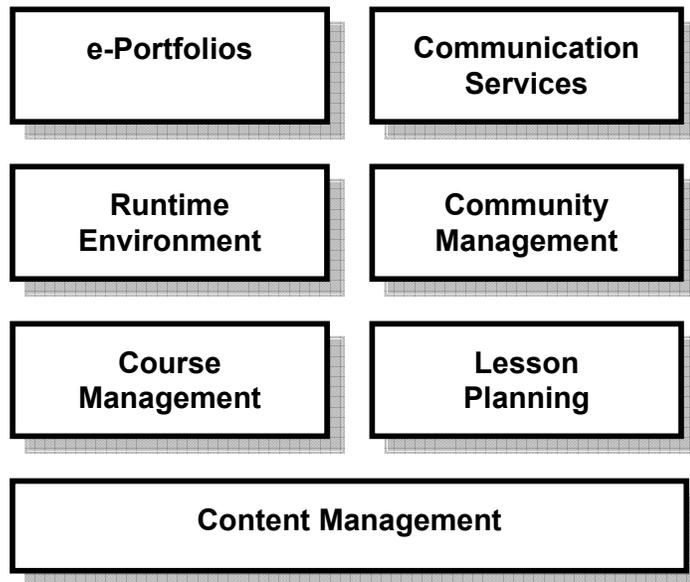


Figure 2 – LearningOnWeb modular structure

When compared to WebOnCampus, these are rather new modules in the application. Nevertheless, one of the main challenges that we are faced with is to reengineer the existing work that was inherited from the WebOnCampus project, in order to comply with SCORM. This involves a continuous evaluation of the existing features and the impact of any changes on those features, as we want to maintain WebOnCampus as untouched as possible.

E-Portfolios

This module allows building and managing the portfolios of a particular course or student. Within this module the student can create a showcase of all his/her work, whereas the teacher can have a vision of the portfolio regarding a specific course (all the student's portfolios for that course).

This module also supports a set of concepts to enable the creation of rich learning contexts, such as pedagogical contract (where the student states what he/she proposes to do), tasks (what the students must do) and reflections (short student reflections the work done).

Communication services

This module provides the necessary means for participants to communicate. This includes standard forums, webmail, as well as some more innovative features such as structured forums [6] and chats [2].

Runtime Environment

This module provides the proper execution environment for the contents to be presented to the student. It consists on the implementation of the run-time environment specified by the SCORM model [4].

Community Management

This module provides the tools to manage groups of people. It enables the teachers and/or students to create groups.

Course Management

This module allows managing courses. This is where the information about courses is inserted, such as the course presentation, frequently asked questions, available contents, among others.

Lesson Planning

This module allows the teacher to build course lessons (re)using existing contents. It basically provides a way to define the sequencing of several SCORM contents packaged in a single course lesson. This sequencing will be later interpreted by the runtime environment and the content will be delivered accordingly to the sequencing rules.

Content Management

This module is responsible for cataloguing, importing and managing contents.

At the present time we already have an implementation of this module, which is able to import SCORM content (packaged in a ZIP file) and to create SCORM compliant packages from authors' materials. It is also possible to search for any content that is already on the system.

The content creation isn't in fact a tool to create content from scratch, but rather a way to wrap regular assets as SCORM content. This option enables the user to fully categorize the asset with the attributes defined on the content aggregation model (CAM). [3].

All the gathered information is stored both in SCORM manifest file and in the database. We decided to have the information stored in these two different places because the manifest file will be part of the package (zip file) that stores the asset itself (including the wrappers that provide the minimal compliance with SCORM – an html that embeds the asset and calls both Initialize() and Finish() API functions [4]), while the information stored on the database is used to perform queries on content that is on the system.

Our system also provides a folder-based system that allows the content creators to organize their work. Instead of having a single list of unsorted items, the user can move any content to a specific folder that is shown in the web interface.

Figure 3 shows a screenshot of the content creation section:

Figure 3 – content creation

When our project started, SCORM 1.2 was the latest version available and this module was developed having that model as a reference. In the meantime SCORM 1.3 was released and the new features introduced by this new version were strong enough for us to decide to support it. This means that some rework had to be done in order to ensure that the module imports and create SCORM packages compliant with version 1.3.

This module is being developed with the collaboration of a University of Rouse student who is making a semester long Erasmus stay at the University of Coimbra.

CONCLUSIONS AND FUTURE WORK

As we are in the development stage of our project, we are still unable to provide practical results of the utilization of this new platform at our University or elsewhere. However, we are confident that our project will surely support teachers and learners in creating and taking advantage of well contextualized courses where the expression learning community can be fully implemented. On the student's side we are expecting to provide a learning environment that is highly personalized and that includes the necessary tools to enable efficient activities and focused communication.

Adopting SCORM imposes the need to follow ADL's effort to enhance their model. We already felt that need during this development stage, when we decided to support SCORM 1.3, although we already had some work done based on SCORM 1.2. The (future) upcoming 2.0 version is being discussed for quite some time. It will be a complete rewrite of the current model and will focus on task simulations. This will surely bring a new opportunity for enhancing the contexts available to the students.

Another challenge that takes place is that the simple fact of providing a great e-learning platform doesn't necessarily mean it will be successfully used. At this level you have to take in consideration that the technology must drive the people involved in the process to take the step forward and look at the technology as an important help.

REFERENCES

- [1] Apache Jakarta Project, "Jakarta Struts", <http://jakarta.apache.org/struts>, May 15, 2004.
- [2] C. Bravo, A. Mendes, M. Marcelino and M. Redondo, "Integrating collaboration with animation and simulation in computer-supported Programming learning", in *Proc. of the 33rd International Symposium IGIP / IEEE / ASEE*, Fribourg, Switzerland, 2004 (accepted for publication).
- [3] P. Dodds and S.E. Thropp. (2004, Jan). "Content Agregation Model (CAM) Version 1.3". [Online]. Available: <http://www.adlnet.org>
- [4] P. Dodds and S.E. Thropp. (2004, Jan). "SCORM Run-Time Environment Version 1.3". [Online]. Available: <http://www.adlnet.org>
- [5] Learning Systems Architecture Lab. (2003, Feb.). "SCORM Best Practices Guide for Content Developers", 1st Edition. Carnegie Mellon University. Pittsburgh, Pennsylvania, USA. [Online]. Available: <http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide>.
- [6] M. Redondo, C. Bravo, M. Marcelino and A. J. Mendes, "Tools for programming learning: an approach to provide a social perspective using collaborative planning of design", in *Proc. of IADIS International Conference e-Society*, Avila, Spain, 2004 (accepted for publication).
- [7] UMIC (2004, May), "O que é a E-U". [Online] Available: <http://www.e-u.pt/conceito/index.asp?areaid=1>.

ABOUT THE AUTHORS

Norberto Henriques, MSc student, Department of Informatics Engineering, University of Coimbra, Portugal, Phone: +351 239 790 000, E-mail: norberto@dei.uc.pt.

Damyan Slavov, BSc student, Department of Computer Systems, University of Rousse, E-mail: saint_dumbo@yahoo.com.

Assist. Professor António José Mendes, PhD, Department of Informatics Engineering, University of Coimbra, Portugal, Phone: +351 239 790 000, E-mail: toze@dei.uc.pt.