

A NEW SOFTWARE ARCHITECTURE FOR LEARNING MANAGEMENT SYSTEMS WITH SCORM SUPPORT

Nicolas Casel
nicolas.casel@uni.lu

Marc El Alami
marc.elalami@uni.lu

Damien Garot
damien.garot@uni.lu

Denis Zampunieris
denis.zampunieris@uni.lu

*University of Luxembourg
Faculté des Sciences, de la Technologie et de la Communication
Campus Kirchberg
6 rue Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg*

ABSTRACT

Multimodality and standardization are the main requirements to achieve in order to deliver powerful and successful Learning Management Systems (LMS). This paper shows how we designed such a LMS in the context of the “e-FSTC” project, which insures reusability and portability of its features and of learning contents. We sum up current trends in the e-learning domain, and we explain to what extent our LMS matches with them, as well as how it implements a flexible architecture, and how it adopts the SCORM Run-Time Environment for interoperability issues.

KEYWORDS

Learning management systems, Web-based course delivery systems, design issues, SCORM, SCORM run-time environment, standardization

1. INTRODUCTION

As the main purpose of Learning Management Systems (LMS) is to supply a virtual training environment online, the issues of content reusability and portability are currently maturing. Along the same line, as the widely used LMS shows, feature modularity is also a very relevant concept to take into account: the long-term use of the platform depends on it.

But even if functionalities or services provided by LMS can be easily managed by software developers, the design of an easily maintainable and upgradeable architecture is not so obvious.

Moreover, some LMS can only provide training contents specifically designed for them. Indeed, despite various attempts to standardize learning content formats, e-learning platforms usually work with proprietary and closed content. Hence, this content cannot be supported by other LMS, and content imported from outside cannot be played.

In this paper, we introduce a new LMS, free and open-source, designed to be evolutionary both for its innovative software architecture and for its interoperable learning content.

The following text is made of three consecutive parts. First, we sum up the current trends in the e-learning domain. Then we describe the complete software architecture as well as the technologies we used to produce

our LMS in the context of the “e-FSTC”¹ project. Finally, we explain what the SCORM [1] standard is and how our e-learning platform implements it in order to support external learning contents without having to modify them.

2. OVERVIEW

As any university, company, or other organization is able to produce its own learning content, the next step is to share and diffuse it as widely as possible. Consequently, learning content, or more generally every Learning Object, must be flexible and adaptable, so that people can access to it independent of the LMS they are using. In the same way, the SLOOP project [2] promotes the sharing of free and open learning objects. Currently, in web-based learning the main reference is SCORM, a collection of standards and specifications which defines communications between LMS and client side content (Shareable Content Object: SCO). Moreover, semantic web, with OWL [3] and RDF [4], tends to create a universal medium for information exchange.

Among different e-learning systems families, our system belongs to LCMS because it proposes web-based management of learning content. Moreover, it offers tutoring functionalities and a proactive system [5], so that it makes interactivity stronger and generates added-value. Obviously, application domain definition requires us to restrict ourselves to a set of services.

Two main issues guided our architecture design. First, object-oriented programming imposes itself by definition: we want to take advantage of encapsulation, inheritance, modularity, and polymorphism for more secure and flexible code. Then, MVC design differentiates data model, user interface and control implementation. By making atomic algorithmic and just-in-time execution, *i.e.* no need to wait for a result to continue processing, it gives more flexibility with easily maintained code.

Open-source, free, and large support community determined, as much as practical, the technical specifications. The LMS runs on Linux Ubuntu with classical technologies to be cross-platform. The server is Apache 2.0, with no need for Tomcat because there is no client-server java code. The main language programming is PHP 5.1.1 for its support of object-oriented programming and its usual association with MySQL, used here in version 5.0. Ajax, JavaScript, DOMXML methods, XHTML, and CSS implement the controllers and graphic user interface, which can be displayed in Firefox 2 and Internet Explorer 6 browsers.

In order to implement proactive functionalities, we keep constant contact with clients, by using C sockets between the server on one side and the Flash interface on the other (client) side.

3. LMS ARCHITECTURE

After defining customer requirements chart, we opt for a "tutored" and proactive LMS. The most relevant advantage of LMS are flexible architecture and the provided packages modularity. Meeting these two requirements calls for division into two different types of layers: horizontal layers for packages, and vertical layers like Java n-tier architecture.

First, our LMS groups delivered set of functionalities, which can be divided in several horizontal layers. This division is implementing the same logic as Java packages, which enables adding, updating and removing any functionality from a given set, without having to modify basic system structure. For instance, different packages are user management, content management with content creation by authors, content delivering, tracking/tracing, monitoring management, tutoring, and sequencing. In comparison with other LCMS like ATutor [6], Claroline [7] or Moodle [8], the e-fstc platform works with total packages modularity as well.

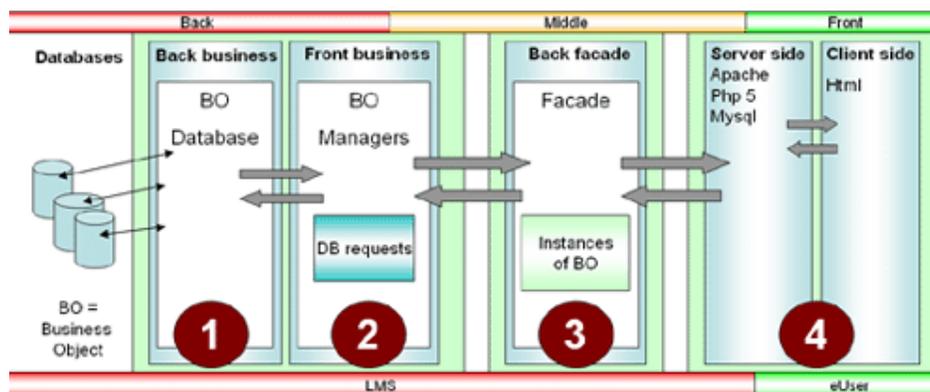
Moreover, another division, with vertical layers, is needed to meet three requirements. Upgradeability gives the LMS the capacity to be maintained independently from the technology changes. For instance, if we change our database management system, only a few coding modifications are required. Interoperability enables adding easily any external module to our LMS, as well exporting our own module to use it with an external system. Platform complexity is widespread for a stronger coherence, and it enables parallel coding development. Indeed, database queries, algorithms and styling are written in different dedicated files. We try

¹ This project is supported by the University of Luxembourg under the grant nr. R1F105K21.

to avoid heavy maintenance like in others LMS, where queries, PHP, HTML, and sometimes CSS pieces of code are grouped in the same file.

To this aim, we project J2EE logic to the PHP architecture: our LMS keeps database and front-end layer abstraction. For the middleware layer, we innovate with four abstract layers, as it is shown in Figure 1.

Figure 1. The e-FSTC platform architecture with four vertical layers.



Layer 1 gives access to database via a specific API: it is the Business Object Database.

Layer 2 guarantees access to the data model stored in the database. It works with a set of instructions to create, modify, delete and list objects classes available in the data model. They are the Business Objects Managers and mainly implement models in the MVC design pattern.

Layer 3 is the "facade" layer with system algorithm and intelligence. These objects group potential use cases and conditional connections, and implement listener controls in the MVC design pattern.

Layer 4 displays the graphic user interface. It implements views and works with instantiations of listenable controls. For instance, JavaScript events are calling controllers-facade methods.

4. SCORM IMPLEMENTATION

ADL – Advanced Distributed Learning (<http://www.adl.net>) defines SCORM specification which enables distributed learning accessible through a web browser. Dominating the market, it is being adopted all over the world to use portable learning content [6, 7]. Communication between client side and server side is managed thanks to the run-time environment implemented by the LMS. However, SCORM RTE is still difficult to understand and to implement properly [8].

SCORM 1.2 was the first robust version. It brings metadata to describe learning content and enables packaging to make it portable. However, there is no sequencing between content objects and no guided learning. Indeed, adaptive learning strategies must be built inside the content objects.

The current version, SCORM 1.3 (3rd edition, October 2006), can create sequencing rules based on success and/or completion. Moreover, it supports tracking and assessment of competencies, so that adaptive learning strategies can be defined for all levels of the activity tree. Due to this new complexity, high level design and authoring tools that take advantage of SCORM 1.3 will take a while to appear.

Nevertheless, up-to-date LMS's support the SCORM standard. For instance, ATutor, Claroline and Moodle enable the import of existing SCO packages into the platform, to be included into online courses. So, in order to be SCORM compliant, the e-FSTC platform deals with interoperable learning objects (SCO). SCO works with two kinds of metadata that are important not to confuse. LOM [9] is a data model that describes a learning object to enables its reusability through different LMS. So a SCO can properly work without these metadata. In addition, the SCORM Metadata Model defines a set of information about SCO tracking in different LMS environments. Its specifications group information sets to be exchanged between SCOs and LMS. Indeed, this data model has to be implemented by the LMS to benefit from interactivity. We choose to support both, by stocking metadata in a database rather than by using XML files, thus we earn from slowness

due to text parsing. Regarding SCORM Metadata Model, our platform currently works with several data like "cmi.core.score.raw" or "cmi.core.session_time", and other ones will be used to improve tracking. Besides, owner metadata can be added in order to meet teacher or author requirements, they are differentiated from SCORM ones thanks to a flag in the database.

SCORM run-time environment uses frames system in HTML code, although W3C – World Wide Web Consortium (<http://www.w3.org/>) recommends not to use them, or at least to provide an alternate version for browsers that do not support frames or are configured not to display them (see <http://www.w3.org/TR/html401/present/frames.html#h-16.4>). Nevertheless, regular websites are more concerned by frames drawbacks than LMS.

Another disadvantage seems to be the implementation of the run-time environment with JavaScript functions. W3C advises supplying alternative content in case of technology unsupported or disabled by the browser (see <http://www.w3.org/TR/WCAG10/wai-pageauth.html#tech-scripts>). If JavaScript is not enabled, the LMS cannot play a SCO.

5. CONCLUSION

Reusability, interoperability, flexibility, and sharing are keywords to take into account in order to design learning objects and up-to-date LMS.

In this paper, we explained the design of an e-learning management system that meets these requirements. Its software architecture enables easy-maintenance, provides modularity to manage additional features and supports the SCORM standard. Its implementation is based on free and open-source technologies. This LMS also includes innovative features, like proactive behaviors.

Future work includes the implementation of security and trust mechanisms into the platform. This concerns, for instance, rights management in order to determine for a given user to which part of the platform s/he has access. Another example is the management by the users of trust and of its delegation with respect to the proactive behaviors run by the system.

REFERENCES

- [1] SCORM – Sharable Content Object Reference Model, <http://www.adlnet.org>
- [2] Shareable Learning Objects in an Open Perspective project, <http://sloop.tes.mi.it/sloop/>
- [3] Web Ontology Language, <http://www.w3.org/TR/owl-features/>
- [4] Resource Description Framework, <http://www.w3.org/RDF/>
- [5] Zampunieris, D., 2006. Implementation of a Proactive Learning Management System. *Proc. of the E-Learn 2006 – 11th World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education*. Honolulu, USA, pp. 3145-3151.
- [6] ATutor, <http://www.atutor.ca>
- [7] Claroline, <http://www.claroline.net>
- [8] Moodle, <http://moodle.org>
- [9] Henriques, N., Slavov, D., Mendes, A.J., 2004. LearningOnWeb – development of a SCORM compliant Learning Management System. *Proc. of CompSysTech'04 – 5th int'l conference on Computer systems and technologies*. Rousse, Bulgaria.
- [10] Bohl, O., Schellhase, J., Sengler, R., Winand, U., 2002. The Sharable Content Object Reference Model (SCORM) – A Critical Review. *Proc. of the Int'l Conference on Computers in Education (ICCE'02)*. p. 950.
- [11] Nakabayashi, K., Kubota, Y., Yoshida, H., Shinohara, T., 2001. Design and Implementation of WBT System Components and Test Tools for WBT content standards. *Proc. of ICALT'01*. Madison, USA, pp. 213-214.
- [12] Learning Object Metadata, <http://www.imsglobal.org/metadata/>