

Enhancing Web Supported Learning by Adding a Management Layer to SCORM Compliant LMSs

Carlos Sousa Pinto
Universidade do Minho, Portugal
csp@dsi.uminho.pt,
Fernando M.S. Ramos
Universidade de Aveiro, Portugal
fmr@ca.ua.pt

Abstract

Many Web based learning experiences fail due to bad or absent support. LMSs (Learning Management Systems) must incorporate mechanisms for real time monitorization of the involvement of each participant in a course, allowing the detection of deviations to the scheduled activities, enabling the correction of these deviations [1].

The principal standardization projects in the area do not cover this type of aspects. Those projects are mainly focused on contents and its delivery to the learners participating in the courses [2], [3], [4].

This article describes a proposal of a reference model and functionalities towards a specification of a layer for real-time management of user interactions with SCORM (Sharable Content Object Reference Model) compliant LMSs.

1. Introduction

The University of Aveiro, Portugal, has a large experience on offering Web based courses, using e-learning platforms. Experience showed that different editions of a same course, using the same contents and structure, and having similar target learners, had different success rates.

We think that success could be directly related with the remote follow-up of the learners' participation in the courses. The best results usually occur when the follow-up is closer.

We believe that LMSs should include mechanisms for automatic monitoring of the participations, so that the probability of success of the teaching/learning process could be enhanced.

2. Our conceptual model

Our proposal for the management layer (ML) lies in the monitoring of an informational entity that we call

"events" and in its comparison with another one that we assign as "activities". This last one implements the pre-defined structure of the course while the first reflects the interactions of the actors with the LMS. The proposed ML completes itself with the inclusion of a notification component and with the definition of a set of rules regulating its behaviour.

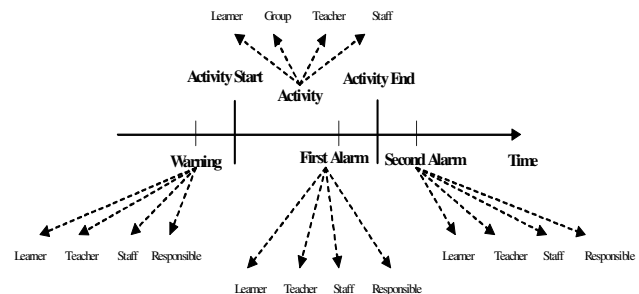


Figure 1. Atomic Unit of Management

For us, a course can be any combination of units of the type showed in Fig 1, organized in a sequential, parallel or random way and including the possibility of recursive application of this concept to the decomposition of an activity in subactivities, to be executed by an actor or a group of actors. Documentation about the most principal projects on the area only refers learners and we can't read anything about the participation of groups, teachers and members of support teams. In our model we consider these types of actors and an actor type named "Responsible" that plays the role of somebody having some level of responsibility over the learner process of learning.

In accordance with Fig 1, each activity has a "warning" to alert the actors to the proximity of that activity. Before reaching the deadline to the execution of the activity it must be tested if it was already terminated or if it is still running. If this is not the case, a "first alarm" will be generated. A "second alarm" must be sent to an actor if he didn't execute a programmed activity.

3. Integrating our work into ADL SCORM

SCORM is the project that congregates greater number of contributions from other projects (IMS, AICC, ARIADNE, IEEE) [2]. We thought that it would be interesting to develop our work towards its possible integration in the SCORM project.

Fig 2 represents our perspective of that possible integration.

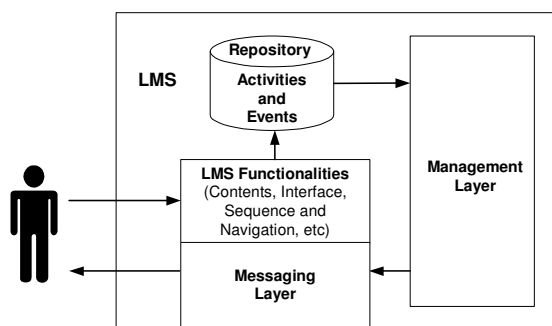


Figure 2. ML relationship with LMS components

Authors interact with the platform in order to construct the courses (storing data into the information entity “activities”). Then, the actors will interact with the LMS to execute the programmed activities. During this interaction, the LMS promotes the updating of the informational entity “events”.

Permanently, the ML will consult the repository of “activities” and “events” to identify situations that justify notifications and, in those situations, it will request the “messaging” layer of the LMS, passing to it, pairs composed by the identification of the destination and the message itself. The LMS using “messaging” functionalities, will send the notifications, according to the information received from the ML.

In order to integrate our proposed ML with the SCORM compliant LMSs, it is necessary that the LMSs can update our informational entity “events” whenever an activity is successfully executed.

SCORM RTE (Run-Time Environment) documentation asserts that during the execution of a SCO (Sharable Content Object), the SCO finds an instance of the API (Application Programming Interface) and initiates the communication between itself and the LMS by calling API methods [5].

The data-transfer methods – “GetValue()”, “SetValue()” and “Commit()” – are used to manage the storage and retrieval of data to be used in a communication session [5]. Using “SetValue()”, information is sent from SCO to LMS, for storage. Extending the behaviour of this component of the API it could be possible to insert relevant information in our “events” informational entity.

SCORM documentation [5], refers that LMSs must use SCO reported information, to take decisions about the sequence of the next activities to be delivered. If the SCO, using the SCORM RTE Data Model element “cmi.completion_status”, informs that the learner has completed that SCO, the associated activity must be considered terminated too. So, SCORM specification can be extended so that this mechanism could create a valid entry in our proposed informational entity “events”.

We can identify another possibility of integration. SCORM Sequencing Behaviour Pseudo Code [6], refers that the attribute “Objective Satisfied Status” must be set to true when an objective is reached. It is also a good time to update our informational entity “events”.

5. Conclusions

It is clear that SCORM project has as principal concerns, the contents, the scheduling of the activities and the mechanisms for sequencing and navigation over activities and contents. Real-time monitorization of the different actors participation, is not considered and it seems to us to be an incomplete approach to consider only learners as relevant actors.

Our proposed ML foresees the existence of other actors and can detect deviations to the course scheduled activities, enabling some kind of intervention in order to correct these deviations in useful time.

At the moment the validation of our work is not complete. It is necessary to integrate the ML in a SCORM compliant LMS and to use this e-learning platform in a significant number of experiences. After these experiences it will be possible to compare the results with those known from passed experiences.

6. References

- [1] Ramos, F. Moreira and Santos, A., “Towards a Reference Framework for eLearning Management”, Proceedings of DLA’2001 - Distance Learning Administration Conference, University of West Georgia, GA, USA, (2001), June 6-8.
- [2] ADL, Advanced Distributed Learning, “SCORM (2004), ADL Sharable Content Object Reference Model, Version 1.3.”, <http://www.adlnet.org>, (2004).
- [3] IEEE LTSC, Learning Technology Standards Committee, “Draft Guidelines for Learning Technology”, <http://ltsc.ieee.org>, (2001), January 30.
- [4] AICC, “CMI Guidelines for Interoperability”, <http://www.aicc.org/docs/tech/cmi001v4.pdf>, (2004), August 16.
- [5] ADL, Advanced Distributed Learning, “SCORM Run-Time Environment Version 1.3.”, <http://www.adlnet.org>, (2004).
- [6] ADL, Advanced Distributed Learning, “SCORM Sequencing and Navigation Version 1.3.”, <http://www.adlnet.org>, (2004).