Enhancing Web Supported Learning in Higher Education by Adding a Management Layer to LMSs

Carlos Sousa Pinto*, Fernando M.S. Ramos**

Universidade do Minho, Portugal

csp@dsi.uminho.pt

Universidade de Aveiro, Portugal

fmr@ca.ua.pt

Abstract

There are many situations in the e-Learning experiences that can compromise the success of the courses. Many times simple reasons are great enough to motivate people to abandon them. For example, if someone does not execute a programmed activity inside the defined window of time, it can compromise the rest of the course to that person. In such situations it would be important that the teacher knew about the situation in useful time, to be able to take any corrective action.

Another example could be presented, involving the professor and the learners. Let us assume that an activity A2 is programmed to be executed by the learners and that it depends on the previous knowledge of the result of the evaluation of a work submitted by the learners to the teacher (activity A1). If the teacher doesn't inform the learners about their classification in useful time, that can compromise the execution of the activity A2.

It seems to be necessary to use mechanisms of automatic management, in real time, of the envolvement of each participant in a distance learning course using LMS (Learning Management System). Such a functionality allows the detection of deviations to the scheduled activities planned for each actor. If it is the case, the referred mechanism can initiate the process of sending notifications to the relevant entities, enabling the correction of these deviations.

Several organizations and consortiuns, involving the industry, governmental institutions and universities, are developing projects of standardization. It seemed important to us to see how the referred aspects were covered by those projects, and to perceive how it could be possible to articulate our work with the ones that are available from these organizations and consortiuns.

This article describes the work that the authors are developing towards the specification of a layer for real-time management of user interactions with LMSs, during the operationalization of a course, and also includes a management meta-data model, related to that management layer.

Keywords: e-Learning**;** Standards; SCORM

1. Introduction

The University of Aveiro, Portugal, has a large experience on offering distance learning courses over the Web, 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. What would be the reason for that? A hypothesis was considered: The level of 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 [1].

We should highlight that the work behind this article is mainly technical and related to projects of standards definition under development by international organizations and consortiums like IMS (IMS Global Learning Consortium, Inc), ADL SCORM (Advanced Distributed Learning Sherable Content Object Reference Model), IEEE LTSC LOM (Institute of Electrical and Electronic Engineers Learning Technologies Standard Committee Learning Object Metadata), ARIADNE (ARIADNE Foundation for the European Knowledge Pool), AICC CMI (Aviation Industry CBT Committee Computer Managed Instruction), etc [4], [5], [6], [7], [8], [9].

The principal objectives of these organizations and consortiums have been the definition of guidelines to the developing of e-learning platforms and learning objects that could be interoperable and reusable.

Our work focuses on some aspects that are not covered by those projects and can compromise some types of teaching/learning approaches.

As examples, we can point out the inclusion of other types of actors – not only single learners. We are talking about teachers, elements of the support teams and groups of learners having to realize group works.

At the same time, we propose the inclusion of a real-time management component, for the interactions between actors and platforms/courses, with an automatic functionality that could notify the relevant actors about their abnormal performances during the courses.

So, for us, it is irrelevant what pedagogical approach is to be used (instructivistic, constructivist, etc). Our concerns are to contribute to the design of a global architecture that could be used to support any pedagogical theory. Web based e-learning platforms must be viewed as tools to help teachers and learners in the distance teaching/learning process. They should not impose limitations on how and what teachers and learners want to do.

However, it is clear for us that the technologies are introducing great changes in the process of learning and teaching. Teachers must change the traditional way of creating, organizing and delivering contents and activities, having in mind the fact that the trend is to see learners searching the information on a self-paced way.

On the other hand, the different actors, namely, teachers and learners, need to know how to work with the technologies in order to integrate themselves in the teaching/learning process. They have to learn how to use the technologies before to teach and learn the contents of the courses [3].

2. Our conceptual model

Our proposal for the management layer lies in the automatic 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 structure of the course while the first reflects the interactions of the actors with the LMS, in what concerns the execution of the activities foreseen for the course.

The subsystem of automatic management complets itself with the inclusion of a component of notifications and with the definition of a set of rules that regulate the notification process.
This functionality foresees the existence of three different instants where the sending of messages can occur:

- Before the beginning of the activity;
- Before being reached the limit defined for the execution of the activity:
- After this limit have been exceeded.

It can then occur a "warning", a "first alarm" and a "second alarm", as shown in the Figure 1.

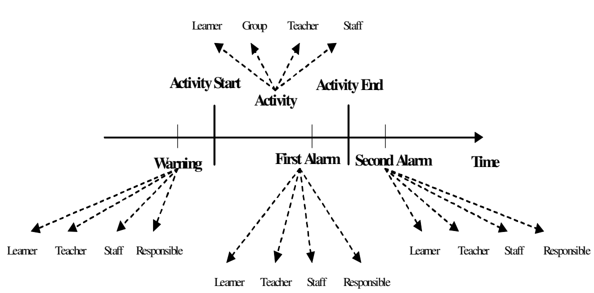

Figure 1. Atomic Unit of Management of Activities in Real-Time

Figure 1 represents what we call "Atomic Unit of Management of Activities in Real Time", on the basis of which all the courses can be architected.

In accordance with Figure 1, each activity has a "warning" emitted before the instant defined for the beginning of the activity, to alert the actors to the proximity of the beginning of that activity. This type of notification makes sense only if the activity is not a random one. In these cases the activity is initiated by the choice of the actor and not for the occurrence of a defined trigger.

When an activity is initiated, its conclusion must occur inside the defined window of time.

Before reaching the deadline to the execution of the activity it must be tested if the activity was already terminated or if it is still running. If this is not the case, a "first alarm" should be generated. This way it can be prevented that the structure of the course has to be redefined and the management subsystem will potentially contribute for the increase of the probability of success of actors' participation in the course.

Finally, once it is possible that an actor misses the execution of an activity inside the foreseen window of time, the management subsystem will have to emit a third type of notification, a "second alarm". The objective of this type of notification is to make possible the adoption of corrective actions, namely the reprogramming of the activity or of the entire course, for this actor.

For us, a course can be any combination of units of the type showed in Figure 1, organized in a sequential, parallel or random way and contemplating 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.

Figure 2 shows an activity composed by subactivities, each of them having exactly the same set of proprieties referred before to the simple activities. In that figure we represent the subactivities as sequencial but it is possible to include subactivities to be executed in a parallel way.

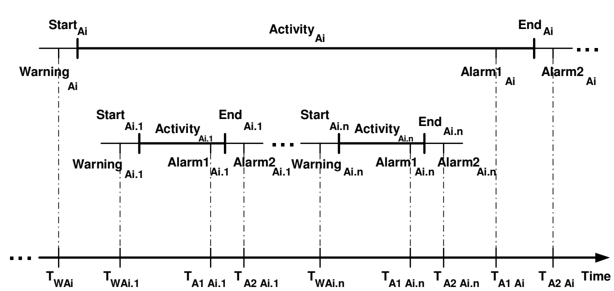

Figure 2. Activity composed by subactivities

3. Integrating our work into other projects

Given the existence of the already referred works of standardization (IMS, AICC, ARIADNE, ADL, IEEE) and once the project ADL SCORM is the one that congregates greater number of contributions from other projects [4], we thought that it would be interesting to make evolve our work to its possible integration in the SCORM. Being so, we made the identification of potential points of interface between our management layer and other layers referred in the documentation of SCORM 2004 specification [4], [8], to allow the monitoring of the interactions with the LMSs.

This work led to the identification of SCORM processes that need to be complemented and to the definition and inclusion of procedures in our subsystem of management, capable to make compatible this new layer with the foreseen functionalities that already exist in the SCORM project.

Figure 3 represents the integration of the different modules of an LMS, and the way they must relate to each other. The shadowed blocks in the diagram are our contribution to the global architecture proposed for an LMS.

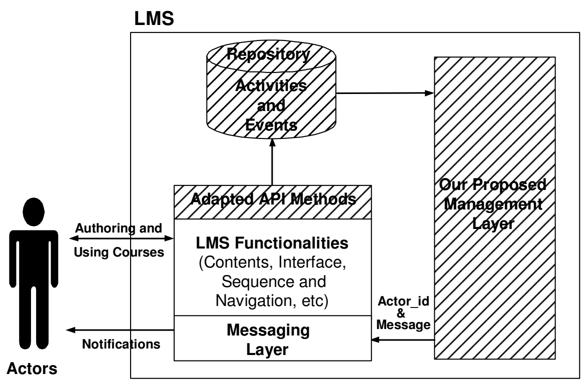

Figure 3. Architecture of the relationship between the management layer and the other LMS components

We should read the scheme of Figure 3 as follows:

1. The authors of the courses interact with the platform in order to construct the courses, registering among other information, the one that implements the structure of the course itself, that is, the activities.
2. Later, the actors to whom exist "defined activities", will interact with the LMS and, during this interaction, the LMS promotes the register of the diverse "corresponding events".
3. The actors will be able to use the mechanisms of synchronous and/or assynchronous communication, to communicate informally between them.
4. Permanently, our proposed management layer will consult the repository of activities and events to identify situations that justify notifications. If there are this type of cases, the management layer will request the "messaging" layer of the LMS, passing to it the following information:
    - Actor_id;
    - Message
5. Finally, the LMS using its functionalities of "messaging", after identifying the preferential way of communication of each actor for whom a message must be delivered, will send the notifications, according to the information received from the management layer, or it will create the conditions so that these notifications are sent in a non electronic form.

It should be highlighted that we can have more than one destination for a notification, namely when sending messages for a group of learners, for example. Even the case of destinations of different types, eventually receiving different messages, is well supported by the management layer as it can be inferred from the structure of informational pairs showed above in point 4.

In order to integrate our proposed management layer with the LMSs builded under SCORM recomendations, it is necessary that the LMSs can create the information about the execution of the activities in our informational entity “events”. The registration of that information must be done only if the activities are terminated successfuly. In our point of view, an activity for which there is no “event” registration, is an activity not executed and the management layer must generate notifications related to that fact.

The integration of our work with SCORM proposed guidelines can be done at several levels. We will present some examples.

In the SCORM RTE (Run-Time Environment) documentation [8], we can read that during the execution of a SCO (Sharable Content Object), that was launched by the LMS (Learning Management System), the SCO finds an instance of the API (Application Programming Interface) and iniciates the communication between itself and the LMS by calling the methods pertained to the API. Those methods are distributed by three main groups – Session Methods, Data-transfer Methods and Support Methods. The session methods “Initialize()” and “Terminate()” are used to initiate and terminate the communication, while the data-transfer methods “GetValue()”, “SetValue()” and “Commit()” are used to manage the storage and retrieval of data to be used in the actual communication session [8].

The method “SetValue()” is used to send information from SCO to LMS, for storage.

We think that it is possible to extend the behavior of this component of the API so that it could promote the insertion of right information in our “events” informational entity, in the cases that it is required.

Accordingly to SCORM documentation [4], LMSs must use SCO reported information, so that it could be possible 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 activity to which that SCO belongs must be considered terminated too [8]. So, we propose the extend of this mechanism, in order to create a valid entry in our proposed informational entity “events”. On the other hand, the data model element

“cmi.time_limit_action”, indicates what the SCO should do, when “cmi.time_limite_action” is exceeded [8].
In our work, we also have considered an instant, after the defined end, to test if the activity was executed and to decide what to do next.

We can identify a third possibility of integration of our work with SCORM proposals. In the SCORM Sequencing Behaviour Pseudo Code [8], we can read that the attribute “Objective Satisfied Status” must be set to true when an objective is reached. It is also a good time to potentially create an instance in our informational entity “events”.

4. The meta-data model

Figure 4 is the hierarchic meta-data model corresponding to our vision of what a course should be. In that model, we represent more than the elements strictly related to the problem of management we are discussing in this paper.

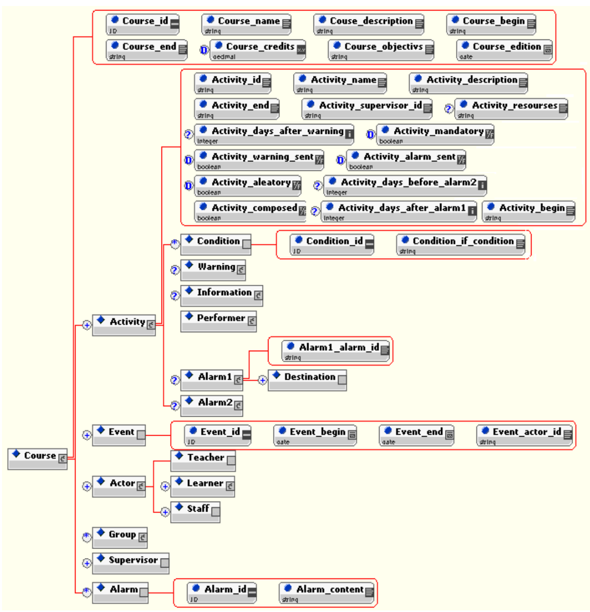

Figure 4. Parcial view of the meta-data model elements

In fact, the model represented in Figure 4 is a meta-data model that could support a complete LMS, accordingly to our perspective of what an LMS should be.

The symbols before the elements and attributes have the following meaning:

“+” - The element can exist one or more times.
“*” - The element can exist zero or more times.
“?” - The element is opcional.
“D” - The attribute has a default value.

All the elements and attributes without any precedent symbol are mandatory and must exist only once.

In the model of the Figure 4 we have included the elements “alarm1” and “alarm2” without a “content” attribute because there are no conceptual differences between the two types of alarms. Only the timing of eventual appearance in the process is different. This way, the two elements have an attribute (Alarm1_alarm_id and Alarm2_alarm_id) that points out to the meta-data element “alarm” where all the possible alarms must be stored.

It is clear in the meta-data model (by using the symbol “+” before the elements) that an alarm (first alarm or second alarm) can have more than one destination, as referred above.

The main works in this area, under development, such as the SCORM project, don’t cover aspects related with groups of learners. This is, in our point of view, an incorrect approach. In fact, there are many situations in the teaching/learning process, based on the work of groups of learners, cooperating to reach some common objectives.

Figure 5 shows an excerpt of the meta-data model highlighting the way we can define groups of learners. Each group has an identifier and a set of learners. Each of those learners is identified by a “learner_id”.

On the other hand, each activity has a “performer-type” associated and that atribute could have the value “Group”. In that case, we should also have defined a set of identifiers of actors – the learners that must execute the activity.

However, as we saw in Figure 2, an activity could be composed by subactivities, each of them to be executed by a single learner and, in that case, the atribute “performer-type” will have the value “learner”, and the atribute “actor_id” should have the identifier of the learner that must execute the subactivity.

One of the permitted values for the atribute “performer_type” is “all_learners”. In this case, the system can identify all the performers of the activity by retriving all the instances of the element “learner”.

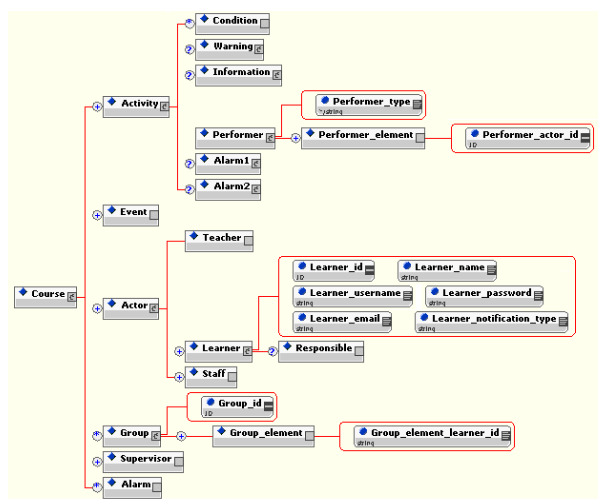

Figure 5. Subset of meta-data elements allowing groups definition

5. Conclusions

The standardization works being developed by the organizations and consortiums referred above [4], [5], [6], [7], [8], [9] are very important because they will allow the uniformization of the development of LMSs and contents.

This is a key aspect in order to obtain greater levels of reuse and interoperability among different systems.

However, it is clear that those works have as principal concerns, the contents development, the schedulling of the activities to be executed inside the courses and mechanisms for sequencing and navigation over the contents and the activities. Aspects that we consider important, like real-time monitoring of the participation of the different actors are not considered.

Our experience in Web based distance learning indicates that when there are not an effective follow up of the activities, by the responsibles for the courses, the probability of insuccess grows up. On the other hand, it seems to be an incomplete approach to consider only the learners as actors participating in a course and that is what we can see in the documentation about the different projects, namely in the SCORM documentation.

Teachers and members of the support teams are also important actors to be considered in the execution of some activities of the courses and it is very easy to identify several activities to be executed by them.

Based on these considerations we have developed the work presented in this paper. We have done it having in mind the proposal of a reference model and functionalities towards a specification of a layer for real-time management of user interactions with LMSs.

A possible integration in the ADL SCORM standard is also a goal for us and we will continue our work towards that integration.

Our proposed management layer can detect deviations to the course scheduled activities, enabling the correction of these deviations in useful time. This is possible due to a component of automatic notifications that is also responsible for the detection of abnormal situations.
The validation of the work is not complete at this time. It is necessary to effectively integrate our management layer in a SCORM compliant LMS and to use this e-learning platform in a significant number of experiences of distance learning. After these experiences it will be possible to compare the results with those known from passed experiences, so that we can conclude that our hypotesis is or is not correct.

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] Ramos, F., Moreira, A., Santos, A., Conde, A., Neves, L., Peixinho, F., Pinto, C. S., "Tracing Dynamic Behaviours in Distributed Learning Communities", Proceedings of 16th Australian International Education Conference, Hobart, Tasmania, Australia, (2002), September 30 – October 4.

[3] Pinto, C. S., "Cooperative Work Environments in the Distance Learning", MSc Dissertation, University of Minho, Braga, Portugal, (1999), May 17.

[4] ADL, Advanced Distributed Learning, "SCORM (2004), ADL Sharable Content Object Reference Model, Version 1.3.", http://www.adlnet.org, (2004).

[5] IMS Global Learning Consortium, Inc., "IMS Learning Design Information Model", http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html, (2003), January 20.

[6] IEEE LTSC, Learning Technology Standards Commitee, "Draft Guidelines for Learning Technology", http://ltsc.ieee.org, (2001), January 30.

[7] IEEE LTSC, "P1484.11.1 Draft Standard for Learning Technology - Data Model for Content Object Communication.", http://ltsc.ieee.org, (2003), November 10.

[8] ADL, Advanced Distributed Learning, "SCORM Run-Time Environment Version 1.3.", http://www.adlnet.org, (2004).

[9] AICC, "CMI Guidelines for Interoperability", http://www.aicc.org/docs/tech/cmi001v4.pdf, (2003), August 16.

[10] IMS Global Learning Consortium, Inc., "IMS Simple Sequencing Information and Behavior Model", http://www.imsglobal.org/simplesequencing/ssv1p0/imsss_infov1p0.html, (2003), March 3.