

# Integrating rich learning applications in LMS

Ricardo Queirós<sup>1</sup>, José Paulo Leal<sup>2</sup> and José Paiva<sup>2</sup>

<sup>1</sup> CRACS & INESC TEC & DI/ESEIG/IPP, Porto, Portugal

ricardoqueiros@eseig.ipp.pt

<sup>2</sup> CRACS & INESC TEC, Faculty of Sciences, University of Porto, Porto, Portugal

{zp@dcc.fc.up.pt, up201200272@alunos.dcc.fc.up.pt}

**Abstract.** Currently, a Learning Management System (LMS) plays a central role in any e-learning environment. These environments include systems to handle the pedagogic aspects of the teaching-learning process (e.g. specialized tutors, simulation games) and the academic aspects (e.g. academic management systems). Thus, the potential for interoperability is an important, although over looked, aspect of an LMS system. In this paper we make a comparative study of the interoperability level of the most relevant LMS.

We start by defining an application and a specification model. For the application model, we create a basic application that act as a tool provider for LMS integration. The specification model act as the API that the LMS should implement to communicate with the tool provider. Based on researches we select the Learning Tools Interoperability (LTI) from IMS. Finally, we compare the LMS interoperability level defined as the effort made to integrate the application on the study LMS.

**Keywords:** LMS, e-learning, interoperability, standards.

## 1 Introduction

Interoperability is the ability of different computer systems, applications or services to communicate, share and exchange data, information and knowledge in a precise, effective and consistent way [1]. In the e-learning realm, interoperability is one of the most important aspects during the construction of an e-learning environment. Usually, system designers identify the environment requirements and select the tools needed to achieve the proposed objectives. However, many times it is forgotten, that environmental tools may need to interact with each other due to several reasons such as, implement a Single Sign-On system (SSO), harvest Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) from digital repositories or, even, personalize the frontend graphical interface with data from a recommendation service.

Nowadays, a learning environment includes a plethora of systems and services that need to communicate to fulfil goals. One of the most important and central systems are the Learning Management System (LMS). Still, the LMS cannot afford to be isolated from other systems in an educational institution. Thus, the potential for interoperability is an important, although frequently overlooked, aspect of an LMS system [2].

In this paper we make a comparative study of the interoperability level of the most relevant LMS. This study is part of an effort to select an LMS on which to base the development of an e-learning environment for the teaching-learning process on the computer programming domain. We chose several LMS vendors that combined have a significant share of the LMS market. We analyse and validate the interoperability features of these LMSs based on two models: specification and application models. The former presents the Learning Tools Interoperability (LTI) specification defined as an API for the standardization of the communication of LMS with external applications. The latter, provides the guidelines for the integration of an external application using the specification model. In the final section we draw conclusions on the results of this study.

## **2 The specification model**

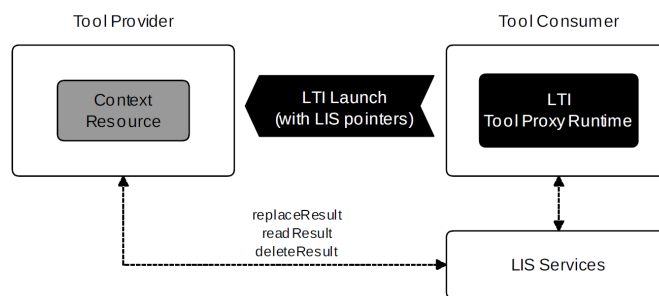
A common interoperability standard that is increasingly supported by major LMS vendors is the IMS LTI specification. The IMS LTI provides a uniform standards-based extension point, allowing remote tools and content to be integrated into LMSs. The main goal of LTI is to standardize the process of building links between learning tools and the LMS. There are several benefits from using this approach: educational institutions, LMS vendors and tool providers by adhering to a clearly defined interface between the LMS and the tool, will decrease costs, increase options for students and instructors when selecting learning applications and also potentiate the use of software as a service (SaaS). The LTI has 3 key concepts [3]: the Tool Provider, the Tool Consumer and the Tool Profile.

The tool provider (TP) is a learning application that runs in a container separate from the LMS. It publishes one or more tools through tool profiles. A tool profile is an XML document describing how a tool integrates with a tool consumer. It contains tool metadata, vendor information, resource and event handlers and menu links. The tool consumer (FC) publishes a Tool Consumer Profile (XML descriptor of the Tool Consumer's supported LTI functionality that is read by the Tool Provider during deployment), provides a Tool Proxy Runtime and exposes the LTI services.

A subset of the full LTI v1.0 specification called IMS Basic LTI exposes a single (but limited) connection between the LMS and the tool provider. In particular, there is no provision for accessing run-time services in the LMS and only one security policy (OAuth protocol) is supported. For instance, to export content from Moodle to Mahara using the Basic LTI, the teacher (or LMS administrator) must first

configure the tool (Mahara) as a Basic LTI tool in the course structure. When a student selects this tool, Moodle launches a Mahara session for the student. The web interface for this session can either be embedded in Moodle's web interface as an iframe or launched in a new browser window.

In March 2012 IMS launched the LTI v1.1 (final version) merging both specifications (Basic LTI and LTI). This version includes updates and clarifications as well as support for an outcomes service and bidirectional communication support. This version also includes the support for an outcomes service based on a subset of the IMS Learning Information Services (LIS) - the LTI Basic Outcomes Service.



**Fig. 1.** Specification model based on IMS LTI

The LIS specification is the definition of how systems manage the exchange of information that describes people, groups, memberships, courses and outcomes within the context of learning. Figure 1 shows how the bidirectionality of the LTI specification is performed. The TC provides launch data with LIS pointers to the TP. It is not required for the TC to provide these services. The LIS services could even be provided by a third system such as a Student Information System. Then, the TP calls the LTI Basic Outcomes Service if available. The service supports setting, retrieving and deleting LIS results associated with a particular user/resource combination. The following functions are supported:

- The `replaceResultRequest` sets a numeric grade (0.0 - 1.0) for a particular result;
- The `readResultRequest` function returns the current grade for a particular result;
- The `deleteResultRequest` function deletes the grade for a particular result.

LTI v2 has been under development for several years and, in July 2013, the CC/LTI APMG approved a revision to the LTI v2 Public Draft and replaced the early public draft released in November 2012. The LTI 2.0 supports two types of connection defined between the Tool Consumer and the Tool Provider: message-based and service-based. A message-based connection involves the user with data being transferred as a signed HTTP POST request via their browser.

### 3 The application model

The application model defines the guidelines for the integration of an external application (tool provider) with an LMS (tool consumer). The integration relies on the LTI specification. The LTI specification recommends REST as the web service flavour for exchanging data between the LMS and external tools. The LTI functions are summarized in Table 1.

**Table 1.** LTI functions.

Function	REST
Launch	POST APP_URL < LTI_PARAMETERS
ReplaceResult	POST LIS_OUTCOMES_URL < LIS_SOURCE_ID + GRADE
ReadResult	POST LIS_OUTCOMES_URL < LIS_SOURCE_ID > GRADE
DeleteResult	POST LIS_OUTCOMES_URL < LIS_SOURCE_ID

The Launch function allows the execution of a particular external application within the LMS. Before the launching, two steps are required: 1) the teacher (or LMS administrator) should configure the application as an external tool in the LMS control panel by setting the name and the URL of the application; 2) the teacher should add an activity into the course structure referring to the external tool. Later on, when a student selects the external tool, the LMS uses the URL to launch the external application through an HTTP POST. This request includes a set of launch parameters (LTI PARAMETERS) as hidden form fields.

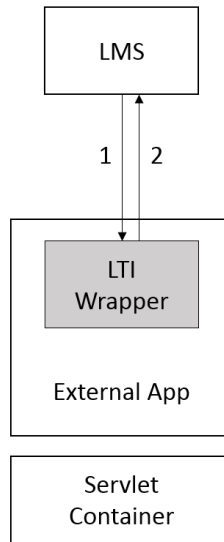
Listing 1 shows a subset of the launch parameters that the LMS (Tool Consumer) sends to the external application (Tool Provider).

```
resource_link_id=2
resource_link_title = MathGamify
lis_person_name_full= Silva Rui
lis_result_sourceid={"data":{"iid":"1","userid":"2"}}
lis_outcome_service_url=http://server:8080/service.php
custom_game_level = 3
```

Table 1 also refers to three functions included in the IMS LIS Outcomes Service. These functions use the `lis_result_sourceid` parameter included in the launch request that is unique for every combination of `resource_link_id/user_id` parameters and identifies a unique row and column within the TC gradebook. After computing a grade, the external application calls the LTI Basic Outcomes Service using the URL stated in the `lis_outcome_service` url launch parameter. The service supports setting, retrieving and deleting of LIS results associated with a particular user/resource combination (`lis_result_sourceid` parameter).

One of the key components of this integration is the LTI Wrapper (created by the authors) that implements both sides of the LTI communication. This component

receives LTI requests from LMS and issues LTI requests to LMS. This Java package can be used by any application requiring LTI communication. Both the LTI Wrapper and the external application are supported by a Java servlet container [4]. The following diagram shows the architecture of the application model:



**Fig. 2.** Application model based on the LTI Wrapper

A typical use case starts with a HTTP message replied by the LMS to the student's browser that starts an LTI request processed by the LTI wrapper (1). This request starts the external application (e.g. a course) on the browser where the student interacts with the system. Finally, the results obtained by the student in the course are reported to the LMS using LTI (2).

## 4 Validation

In this section we evaluate the previous models with seven LMS. For this validation a minimal external application was created - a simple multiplication game called MathGamify. This game can be used by primary school children to learn multiplication tables. MathGamify generates two random numbers. The first number between 1 and the current game level and the second number between 1 and 10. Then the student/player has the opportunity to answer the multiplication value of the two numbers. The score is accumulated in the ratio of the player's level until player misses, in which case the score is reset to zero. The game was tested in seven LMS. The results are presented in Table 2.

**Table 2.** LMS interoperability comparative study.

LMS	Specification Model	Application Model
Moodle	YES	YES
Blackboard	YES	YES *
Sakai	YES	YES
Dokeos	NO	NO
Desire2Learn	YES	NO
eFront	NO	NO
ATutor	YES	NO

Based on the previous table, and despite the support of the LTI specification by several LMS, only Moodle and Sakai successfully ran the MathGamify game. The Blackboard LMS (\*) was able to launch the external game, but the grade results were not received with success.

## 5 Conclusion

This paper presents a basic study on the LMS interoperability facets. We defined two models to base the study: the specification model defines the communication specification that LMS should support to interact with external applications. Based on previous studies, the LTI specification was chosen; the application model presents the architecture environment for the integration of external applications. This integration is managed by a LTI Wrapper that encapsulates all the complexity of the LTI API.

Based on these two models, seven LMS were chosen and validated through the integration of a LTI compatible game named MathGamify. Although, the majority of the LMS support the LTI specification, only Moodle and Sakai were able to communicate easily with the game and get the grade results.

## References

- [1] Martínez, J. Á. & Navarra, P. L. (2007). Content Interoperability on e-learning platforms: standardization, digital libraries, and knowledge management, *Revista da Universidad y Sociedad del Conocimiento*.
- [2] Queiros, Ricardo and Leal, José Paulo. A comparative study on LMS interoperability. In *Higher Education Institutions and Learning Management Systems: Adoption and Standardization*, IGI-Global, 2011.
- [3] Gilbert, T.: Leveraging sakai and ims lti to standardize integrations. In: *10th Sakai Conference* (2010).
- [4] Queiros, Ricardo, Leal, José Paulo and Campos, José. Sequencing Educational Resources with Seqins. In *Computer Science and Information Systems*, 2014.