Integrating the LMS in Service Oriented eLearning Systems

José Paulo Leal, CRACS & INESC-Porto LA, Faculdade de Ciências, Universidade do Porto, Portugal
Ricardo Queirós, CRACS & INESC-Porto LA, Faculdade de Ciências, Universidade do Porto, Portugal

ABSTRACT

Learning management systems are routinely used for presenting, solving and grading exercises with large classes. However, teachers are constrained to use questions with pre-defined answers, such as multiple-choice, to automatically correct the exercises of their students. Complex exercises cannot be evaluated automatically by the LMS and require the coordination of a set of heterogeneous systems. For instance, programming exercises require a specialized exercise resolution environment and automatic evaluation features, each provided by a different type of system. In this paper, the authors discuss an approach for the coordination of a network of eLearning systems supporting the resolution of exercises. The proposed approach is based on a pivot component embedded in the LMS and has two main roles: 1) provide an exercise resolution environment, and 2) coordinate communication between the LMS and other systems, exposing their functions as web services. The integration of the pivot component in the LMS relies on Learning Tools Interoperability (LTI). This paper presents an architecture to coordinate a network of eLearning systems and validate the proposed approach by creating such a network integrated with LMS from two different vendors.

Keywords: *eLearning, Interoperability, Learning Management Systems (LMS), Learning Tools Interoperability (LTI), Service Oriented Architectures (SOA)*

1. INTRODUCTION

The architecture of eLearning platforms is moving from centralised, component based systems to decentralised platforms assembling multiple services (Dagger, O'Connor, Lawless, Walsh, & Wade, 2007). These services can participate in several learning processes that are easily reconfigured to meet changing requirements and demands. We are particularly interested in networks of eLearning systems providing services related to automatic evaluation.

The motivation for this work reflects the experience gained by the authors in several educational projects such as Mooshak and EduJudge. Mooshak (Leal & Silva, 2003) is a contest management system for computer programming contests that is being used since 2002 also as an e-Learning tool in programming language courses. EduJudge (Leal & Queirós, 2009) is a system developed for enabling the use by LMS of the collection of programming exercises of the UVA (University of Valladolid)

DOI: 10.4018/jksr.2011040101

on-line judge (Regueras, Verdú, Castro, Pérez, & Verdú, 2008). Both these systems require a programming exercise evaluation feature. This is a complex feature that cannot be easily integrated as a component in a general purpose Learning Management System (LMS) such as Moodle, Blackboard or Sakai, that already provide other important features such us content sequencing and grade books.

Our research objective is to manage and coordinate a network of eLearning system where students can solve exercises in complex domain such as computer programming. Networks of this kind include systems such as evaluation engines, repositories of learning objects and exercise resolution environments. The LMS has also an important role in this network of systems since it is the natural place to assign exercises to students and to collect grades. However, the coordination of a network of such disparate systems is rather complex. Some of these systems expose their functions as web services, such as the repository of learning objects or the evaluation engine, but others have their own web interfaces for students and teachers, such as the LMS and the exercise resolution environment.

The goal of the research described in this paper is to explore the possibility of embedding a pivot component in an LMS that acts as exercise resolution environment and coordinates the communications between the LMS and the web services. The integration of this component with the LMS is supported by the IMS Basic Learning Tools Interoperability (LTI). This standard provides a framework for integrating applications with educational platforms such as LMS, portals, or other systems from which applications can be launched. An integration component developed with LTI can be used in any LMS that supports this standard.

The remainder of this paper is organized as follows. Section 2 presents the state of the art of the automatic evaluation of programs, including the current application profiles used to describe programming exercises and the systems to manage them such as the learning objects repositories and the evaluation engines. In this section we focus also in the interoperability efforts made by educational organizations to connect these and other applications into the LMS. In the following section we start by presenting the proposed architecture for a network of eLearning systems supporting the resolution of programming exercise. Then, we highlight the pivot component that acts as exercise resolution environment followed by the other eLearning services. In the following section we validate the proposed solution by integrating the pivot component in two LMS - Moodle and Sakai. Finally, a summary of the main contributions and a perspective of future research conclude this paper.

2. STATE OF ART

The current generation of eLearning platforms values the interchange of learning objects and learners' information through the adoption of standards that brought content sharing and interoperability to eLearning. Learning Objects (LO) are units of instructional content that can be used, and most of all reused, on web based eLearning systems. Despite its success in the promotion of the standardization of eLearning content, it is not enough to ensure interoperability, which is a major user concern with the existing systems. The definition of common protocols and interfaces for the communication among systems is also an important issue to address.

In the last few years there have been initiatives (Leal & Queirós, 2010) to adapt Service Oriented Architectures (SOA) to eLearning. These initiatives, commonly named eLearning frameworks, had the same goal: to provide flexible learning environments for learners worldwide. Usually they are characterized by providing a set of open interfaces to numerous reusable services organized in genres or layers and combined in service usage models. Other eLearning interoperability initiatives (e.g. NSDL, POOL, OKI, EduSource, IMS DRI, IMS LTI) appeared in the last decade.

While eLearning frameworks are general approaches for eLearning system integration, several authors proposed service oriented approaches specifically targeted to the LMS. In fact, there are several references in the literature to middleware components for LMSs integration in SOA based eLearning systems. Apostolopoulos proposes a middleware component (Apostolopoulos & Kefala, 2003) to address the lack of integration of eLearning services. In this approach the eLearning components are implemented as agents maintained in a local management information base, and can communicate with the agent manager through the SNMP protocol. Costagliola develop an architecture (Casella, Costagliola, Ferrucci, Polese, & Scanniello, 2007) based on a middleware component and use Web Services to integrate different software components and improve interoperability among different systems. The middleware component enables the student learning process traceability since it has been developed to be compliant with SCORM. Al-Smadi presents a service-oriented architecture (Al-Smadi & Gutl, 2010) for a generic and flexible assessment system with cross-domain use cases. All these approaches have in common the need of a modification of LMS for each specific vendor, with the implementation of a new module or building block. To the best of the authors' knowledge there are no references in the literature to the use of a common standards supported by the major LMS vendors as a means to integrate the LMS in a service oriented network of learning environments.

A common interoperability standard that is increasingly supported by major LMS vendors is the IMS Learning Tools Interoperability (IMS LTI) specification. It provides a uniform standards-based extension point in LMS allowing remote tools and content to be integrated into the LMS. The main goal of the LTI is to standardize the process for 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, increases options for students and instructors when selecting learning applications and also potentiates the use of software as a service (SaaS).

The LTI has 3 key concepts as shown in Figure 1 (Gilbert, 2009): the Tool Provider, the Tool Consumer and the Tool Profile.

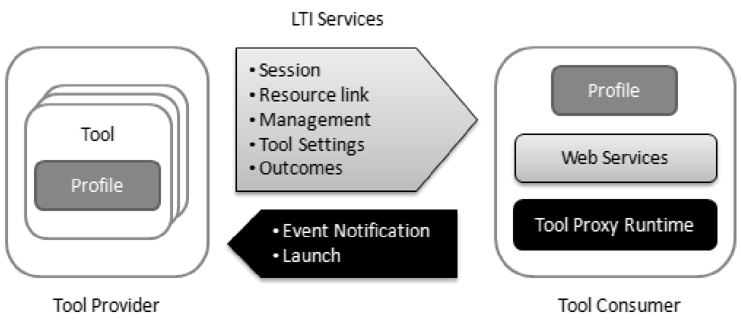
The Tool Provider is a learning application that runs in a container separate from the LMS. It publishes one or more tools through the Tool Profiles. The Tool Profile is an XML descriptor that describes how a tool integrates with a tool consumer. It is composed by information about the tool metadata, vendor information, resource and event handlers and menu links. The Tool Consumer 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.

The IMS launched also a subset of the full LTI v1.0 specification called IMS Basic LTI. This subset exposes a single (but limited) destination between the LMS and the application as shown in Figure 2.

For instance, there is no provision for accessing run-time services in the LMS and only one security policy is supported. Basic LTI also supports a basic security model based on the OAuth protocol. This protocol aims to secure the message interactions between the Tool Consumer and the Tool Provider. It requires a key and a shared secret to sign messages. The key is sent with each message, as well as an OAuth-generated signature based on the key. The Tool Provider verifies the secret based on the provided key and re-computes the signature and compares the recomputed signature with the transmitted signature to verify the sender's credentials (IMS, 2010).

As the IMS LTI specification, the IMS Digital Repositories Interoperability (IMS DRI) specification was created by the IMS Global Learning Consortium (IMS GLC). The IMS DRI provides recommendations for common repository functions, namely the submission,

Figure 1. The IMS LTI framework



search and download of LOs. It recommends the use of web services to expose the repository functions based on the Simple Object Access Protocol (SOAP) protocol, defined by W3C. Due to their growing popularity other web service interface flavours, such as Representational State Transfer (REST) (Fielding & Taylor, 2002) should be considered, since they are not excluded from the recommendation. This will improve interoperability with systems that adhere to a more informal style of development.

3. NETWORK ARCHITECTURE

In this section we present the overall architecture of a network of eLearning systems participating in the resolution of programming exercises based on a pivot component embedded in the LMS. Then, we exemplify the integration of these services in a pedagogical learning process.

3.1. Architecture

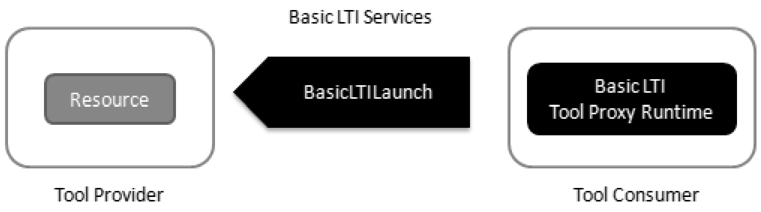
The proposed architecture is described by the UML components diagram shown in Figure 3. The architecture is divided in four components:

- **Learning Objects Repository (LOR)** to store the exercises and to retrieve those suited to a particular learner profile;
- **Evaluation Engine (EE)** to automatic evaluate and grade the students attempt to solve the exercises;
- **Programming Exercises Resolution Environment (PERE)** to recover the exercises descriptions from the repository and provide an exercise resolution environment allowing students to solve the exercises and submit them to the evaluator;
- **Learning Management System (LMS)** to sequence the presentation of exercises to learners and to collect the students' grades.

PERE plays an important role in this network. Therefore, it is important to emphasize its integration with all other components. This integration relies on the following communications standards:

- **IMS DRI** to allow the access to the programming exercises stored in repositories implementing the IMS DRI functions;
- **Evaluate service** to allow the submission and evaluation of students' attempts to Evaluation Engines implementing the Evaluate service - recently submitted to the e-Framework;
- **IMS Basic LTI (bLTI)** to embed the PERE in the LMS (e.g. Moodle, Sakai, Blackboard) by assigning information in the launch process such as user identity, course information and role information;
- **IMS Full LTI (fLTI)** to report an evaluation result back to the grade book of the LMS.

Figure 2. The IMS Basic LTI framework



3.2. Communication Model

The communication model of the proposed network of eLearning systems defines the interaction between all its components. The Figure 4 shows an UML sequence diagram to illustrate the sequence of the functions invocations from these eLearning components.

This diagram could be applied to a typical pedagogical learning process such as a classroom assignment in a computer science course. In this scenario, the teacher sets a number of activities (exercises) in the LMS. To select the relevant programming exercises the teacher searches for them in the repository. Then, the learner tries to solve the exercises assigned by the teacher using the PERE (launched by the LMS). The PERE recovers the exercise description from the repository and shows it to the student. After coding the program the student send an attempt to the Evaluation Engine. The student may submit repeatedly, integrating the feedback received from the Evaluation Engine. In the end, the Evaluation Engine sends a grade to the PERE and consequently for the LMS that records it and reports the Learning Object usage data back to the repository. This last task will provide data for future adaptability services that will adjust the presentation order in accordance with the effective difficulty of programming exercises (not the difficulty stated on the LO) and the needs of a particular student.

4. PROGRAMMING EXERCISES RESOLUTION ENVIRONMENT

PERE is a pivot component that is embedded in the LMS and has two main roles: 1) to provide an exercise resolution environment, and 2) to coordinate the communication between the LMS and the other systems (e.g. learning objects repositories and evaluators) exposing their functions as web services.

PERE is organized in two main packages: the back-end (used by the teacher) and the front-end (used by the student). In the back-end the teacher configures a work assignment by searching for programming exercises in the repository and associating the most relevant. In the front-end, the student reads the exercise description, solves it in the exercise resolution environment and gets the evaluation report that will help him to refine the exercise and, if necessary, resubmit it. In order to model the front-end three classes were defined:

- **Description** recovers the exercise description from the repository;
- **Solver** provides an exercise resolution environment allowing the students to solve the exercise and submit it to the evaluator;
- **Report** presents the final evaluation report of the students' attempt to solve the exercise.

The web interface of PERE was developed with the Google Web Toolkit (GWT), an open source framework for the rapid development of AJAX applications in Java. When the application is deployed, the GWT cross-compiler translates Java classes of the GUI to JavaScript files and guarantees cross-browser portability. The framework supports also asynchronous remote procedure calls. This way, tasks that require high computational resources (e.g., complex searching within the repository) can be triggered asynchronously, increasing the user interface's responsiveness. The specialized controls are provided by SmartGWT, a GWT API's for SmartClient, a Rich Internet Application (RIA) system. The graphical interface of the front-end is composed by three panels reflecting the classes previously explained. The Figure 5 depicts the Solver panel.

The integration of the pivot component in the LMS relies on the LTI specification. The basic workflow for using Basic LTI starts when the Teacher (or LMS administrator) adds the tool (PERE) as a Basic LTI tool into their course structure as a resource link using the LMS control panel. The Teacher sets the URL, secret, and key as metadata for the resource link. When the students select the tool, the LMS uses the URL, secret, and key information to launch the student into the PERE in an iframe or new browser window. The PERE receives a launch request that includes user identity, course information, role information, and the key and signature. The launch information is sent using an HTTP form generated in the user's browser with the Basic LTI data elements in hidden form fields and automatically submitted to the external tool using JavaScript. The following is a subset of the information that the LMS (Tool Consumer) sends to the PERE (Tool Provider):

resource_link_id=1 //An unique identifier for the resource in the LMS.
resource_link_title= My First Exercise // Title of the resource
resource_link_description= Description... //Description of the resource.
user_id=2 // User identifier
user_image=myPhoto.gif // Profile picture.
roles= Instructor,Administrator //List of one or more user roles.
context_title=Course Fullname 101 //A title of the context (e.g. course information).

All these data items are included on the POST data when a Basic LTI launch is performed. These data items can be used, for instance, to personalize the frontend of the tool providers. To extend these fields it is necessary to prefix all fields not described herein with "ext_".

5. ELEARNING SERVICES

In this section we detail the design and implementation of the complementary services of the proposed network - Learning Objects Repository and Evaluation Engine.

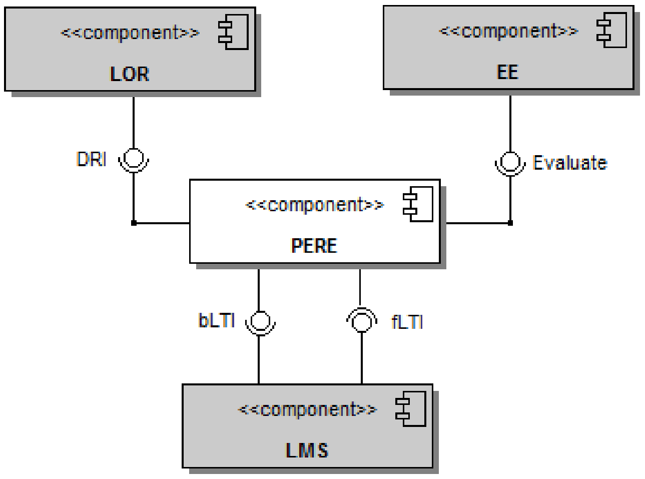
5.1. Learning Objects Repositories

Learning objects repositories are an essential part of service oriented platforms in eLearning since they provide content to several types of services. The need for this kind of repositories is growing as more educators are eager to use digital educational contents and more of it is available. Several surveys (Holden, 2004) show that users are concerned with issues that are not completely addressed by the existing systems, such as interoperability. Thus, a desired feature of a repository is the support for a standard and automatic communication with other systems.

The repository used in this work was the crimsonHex. It was developed as part of the EduJudge project to act as a programming problem repository service. The Core component of the repository exposes the main features, both to external services, such as the LMS and the EE, and to internal components, respectively: the Web Manager, to allow the creation, revision, uploading/downloading of LO and related metadata, enforcing compliance with controlled vocabularies; and the Importer, to populate the repository with existing legacy repositories.

The Core component of the crimsonHex repository provides a minimal set of operations

Figure 3. UML components diagram



exposed as web services – in SOAP and REST flavours - and based in the IMS DRI specification. The syntax of function requests in SOAP flavour is summarized in Table 1.

More details about the implementation of crimsonHex can be found elsewhere (Leal & Queirós, 2009).

5.2. Evaluation Engine

The purpose of a programming exercise evaluator is to mark and grade exercises in computer programming courses and in programming contests. By exposing its functions as services, an evaluator of this kind is able to participate in business processes integrating different system types such as Programming Contest Management Systems, LMS, Integrated Development Environments (IDE) and LOR.

To formalize the definition of this service we used an eLearning framework. An eLearning framework aims to adapt SOA to eLearning providing flexible learning environments for learners worldwide. The new service - Evaluate Programming Exercise - models the evaluation of an attempt to solve a programming exercise defined as a learning object and produces a detailed report. This evaluation report includes information to support exercise assessment, grading and/or ranking by client systems.

This service exposes its functions as SOAP and REST web services. The syntax of function requests in SOAP flavour is summarized in Table 2. The three types of request handled by this service are:

- **ListCapabilities -** provides the client systems with the capabilities of a particular evaluator;
- **EvaluateSubmission -** allows the request of an evaluation for a specific programming exercise;
- **GetReport -** allows a requester to get a report for a specific evaluation using a ticket.

More details about the definition of this service can be found elsewhere (Leal & Queirós, 2010).

6. INTEGRATION AND VALIDATION

In this section we validate the proposed approach by creating such a network integrated with LMS from two different vendors: Moodle and Sakai.

To recreate the proposed network using Moodle (version 1.9.9) it was necessary to install the XAMPP package that includes the Moodle basic requirements such as the Web

Figure 4. UML sequence diagram

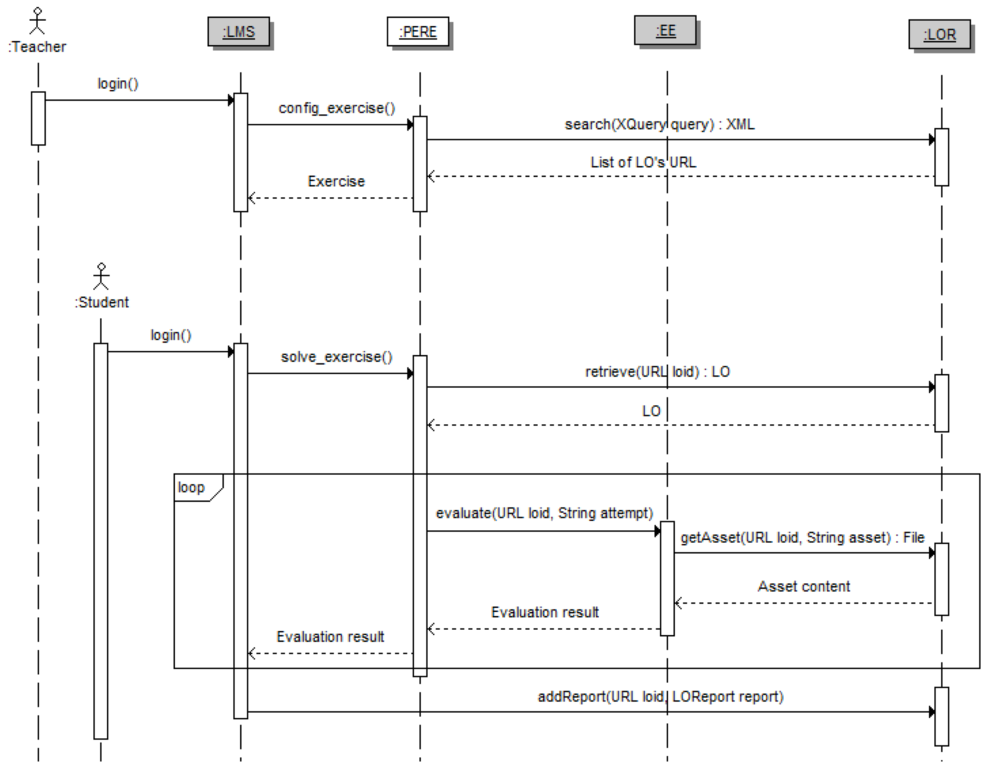
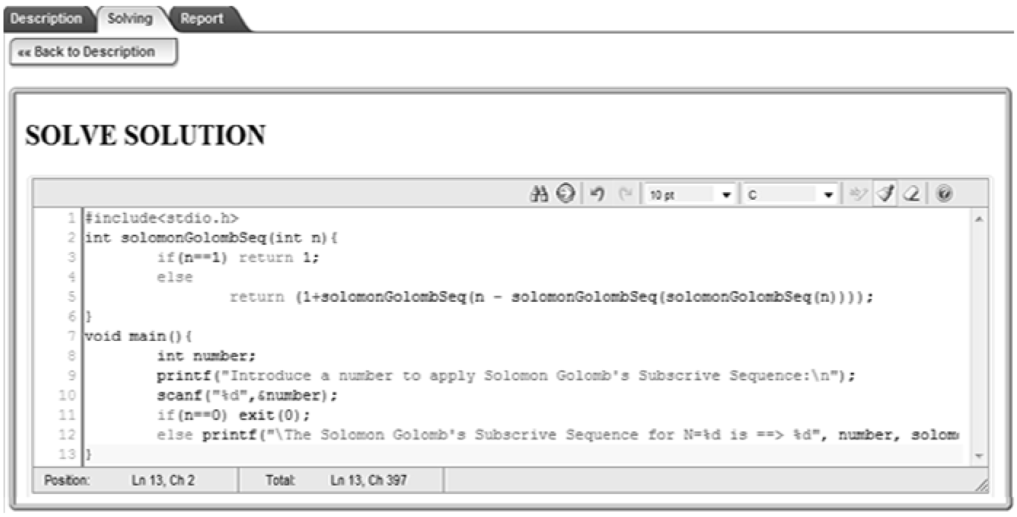


Figure 5. The PERE graphical interface



SOLVE SOLUTION

```
#include<stdio.h>
int solomonGolombSeq(int n){
        if(n==1) return 1;
        else
                return (1+solomonGolombSeq(n - solomonGolombSeq(solomonGolombSeq(n))));
}
void main(){
        int number;
        printf("Introduce a number to apply Solomon Golomb's Subscrive Sequence:\n");
        scanf("%d",&number);
        if(n==0) exit(0);
        else printf("\The Solomon Golomb's Subscrive Sequence for N=%d is ==> %d", number, solom
}
```

Position: Ln 13, Ch 2 Total: Ln 13, Ch 397

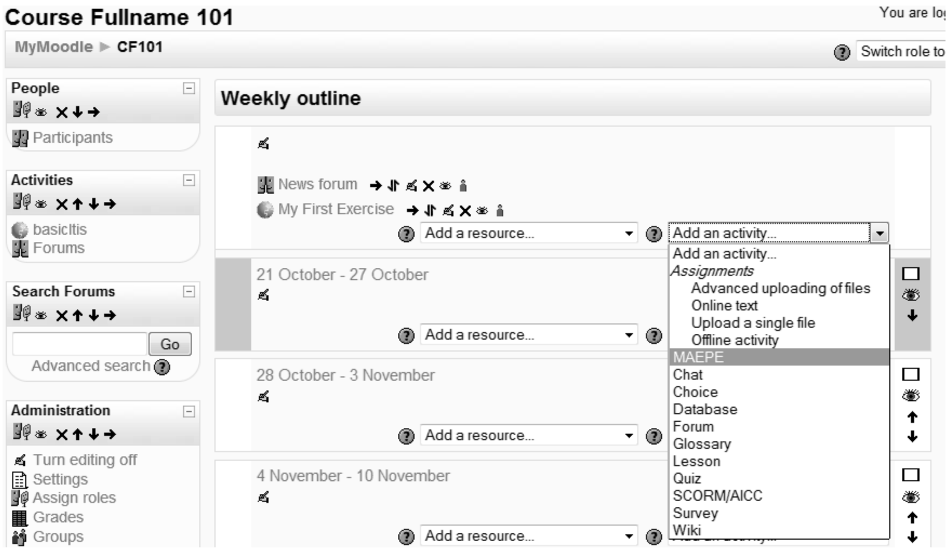
Table 1. Core functions of the Repository

Function	SOAP
Reserve	XML getNextId(URL colectionl)
Submit	XML submit(URL loid, LO lo)
Request	LO retrieve(URL loid)
Search	XML search(XQuery query)
Alert	RSS getUpdates()
Report	XML Report(URL loid,Report rp)
Create	XML Create(URL collection)
Remove	XML Remove(URL collection)
Status	XML getStatus()

Table 2. Core functions of the Evaluation Engine

Function	SOAP
ListCapabilities	ERL ListCapabilities()
EvaluateSubmission	ERL Evaluate(Problem, Attempt, Capability)
GetReport	ERL GetReport(ticket)

Figure 6. Assign of a PERE activity in a Moodle course



server (Apache), the database (MySQL) and PHP. This Moodle distribution needs the further installation of an IMS bLTI consumer.

In relation to Sakai (version 2.7.1) it was necessary to install the Java 1.6 and a Servlet container (Apache Tomcat 5.5). Like the previous case, the Sakai distribution needs the further installation of an IMS bLTI consumer.

After the LMS installation, the PERE tool must be configured as a basic LTI tool in the LMS Control panel. Then, the PERE tool can be included as an activity in the course structure. In Figure 6, an activity associated with the PERE tool is defined in a Moodle course.

In both cases it was possible to configure the PERE as an IMS Basic LTI tool in the LMS and allow its standard launch by the students. The main advantage was the fact that there weren't significant differences between both LMS in the configuration and launching of the PERE tool. The main disadvantage was the fact that both LMS still do not support the full LTI. For this reason it was impossible to implement the report of an evaluation of a student's attempt back to the LMS grade book.

7. CONCLUSION

In this paper we presented an approach to integrate the LMS in a network of services that participates in the automatic evaluation of programming exercises. This is a heterogeneous network connecting different kinds of systems such as evaluation engines, repositories of learning objects and resolution environments. The approach we propose is based on a pivot component, acting as an exercise resolution environment, connected to the LMS using the Learning Tools Interoperability (LTI) and responsible for the coordination of the network.

The main contributions of this paper are the following. Firstly, we defined architecture for networking heterogeneous based on a pivot component. Secondly, we designed and implemented this component using LTI to manage the communication with LMS. Thirdly we assembled the network hosting this pivot component on two reference LMS vendors to test the maturity of the LTI standard.

We concluded that a pivot component integrated in the LMS is a promising approach to the task of coordinating a heterogeneous network of eLearning systems. The pivot component can have its own user interface for interaction with students, as is required for the resolution environment, that is embed in the LMS user interface. It can also control the invocation of remote web services, such as those exposed by the repository of learning objects and evaluation engine. Finally, it can summarize the activity of the student as a grade and report it back to the grade book of the LMS.

Unfortunately, we must conclude also that the LMS support of LTI standard in not mature enough for using this approach in the near future. Most LMS vendors, and in particular those we tested, support only the Basic LTI. Thus, the grade reporting feature could not be fully implemented in our integration and validation setup. Moreover, we had to perform custom installations of both LMSs, mixing code from a stable distribution and code under development just to have Basic LTI.

A full and stable support of LTI in major LMS vendors will encourage us to implement a more sophisticated version of the approach described in this paper. Instead of embedding the resolution environment on the LMS the student should be able to use and Integrated Development Environment (IDE) such as Eclipse. We plan to develop an IDE plug-in to create a programming exercise resolutions environment. It will complement the standard code programming features of an IDE with reading exercise descriptions from the repository, submitting code them to the evaluation engine and displaying feedback to the student. In this future work we will split the coordination task among the pivot component integrated in the LMS and the plug-in on the IDE. Also, the LMS must communicate with a local service on the student's machine (hosted on the IDE) rather than on the cloud. Still, this variant is a step towards to integrate in this network the best of bread for each task.

REFERENCES

Al-Smadi, M., & Gutl, C. (2010). SOA-based architecture for a generic and flexible e-assessment system. In *Proceedings of the IEEE EDUCON Conference on Education Engineering*, Madrid, Spain (pp. 493-500).

Apostolopoulos, T. K., & Kefala, A. (2003). An e-learning service management architecture. In *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies*, Athens, Greece (pp. 140-144).

Casella, G., Costagliola, G., Ferrucci, F., Polese, G., & Scanniello, G. (2007). A SCORM thin client architecture for e-learning systems based on web services. *International Journal of Distance Education Technologies*, *5*(1), 13–30. doi:10.4018/jdet.2007010103

Dagger, D., O'Connor, A., Lawless, S., Walsh, E., & Wade, V. (2007). Service oriented elearning platforms: From monolithic systems to flexible services. *IEEE Internet Computing*, *11*(3), 28–35. doi:10.1109/MIC.2007.70

Gilbert, T. (2009). Pearson Education: Leveraging Sakai and IMS LTI to standardize integrations. In *Proceedings of the 10th Sakai Conference*, Boston, MA.

Holden, C. (2004). *What we mean when we say "repositories" user expectations of repository systems*. Madison, WI: Academic ADL Co-Lab.

IMS. (2010). *IMS basic learning tools interoperability specification v1.0- final specification*. Retrieved from http://www.imsglobal.org/lti/blti/bltiv1p0/ltiBLTIimgv1p0.html

Leal, J. P., & Queirós, R. (2009). CrimsonHex: A service oriented repository of specialized learning objects. In *Proceedings of the 11th International Conference on Enterprise Information Systems*, Milan, Italy.

Leal, J. P., & Queirós, R. (2010). eLearning frameworks: A survey. In *Proceedings of the International Technology, Education and Development Conference*, Valencia, Spain. Fielding, R. T., & Taylor, R. N. (2002). Principled design of the modern web architecture. *ACM Transactions on Internet Technology*, *2*(2), 115–150.

Leal, J. P., & Queirós, R. (2010). Specifying a programming exercises evaluation service on the e-framework. In X. Luo, M. Spaniol, L. Wang, Q. Li, W. Nejdl, & W. Zhang (Eds.), *Proceedings of the 9th International Conference on Web-based Learning* (LNCS 6483, pp. 141-150).

Leal, J. P., & Silva, F. (2003). Mooshak: A web-based multi-site programming contest system. *Software, Practice & Experience*, *33*(6), 567–581. doi:10.1002/spe.522

Regueras, L. M., Verdú, E., Castro, J. P., Pérez, M. A., & Verdú, M. J. (2008). Design of a distributed and asynchronous system for remote evaluation of students' submissions in competitive e-learning. In *Proceedings of the International Conference on Engineering Education*, Budapest, Hungary.

José Paulo Leal is assistant professor at the department of Computer Science of the Faculty of Sciences of the University of Porto (FCUP) and associate researcher of the Center for Research in Advanced Computing Systems (CRACS). His main research interests are eLearning system implementation, structured document processing and software engineering.

Ricardo Queirós is a Ph.D. student of the Doctoral Program in Computer Sciences in the Faculty of Sciences of the University of Porto (FCUP) and an associate member of the Center for Research in Advanced Computing Systems (CRACS). He is also a lecturer at the School of Industrial Studies and Management (ESEIG) in Vila do Conde, which is responsible for courses in the area of Programming Languages and Databases. His scientific activity is related with e-Learning, Languages for XML, Architectural Integration with focus on the development of e-Learning Systems.