

## Adapting LMS architecture to the SOA: an Architectural Approach

Miguel Ángel Conde González, Francisco José  
García Peñalvo  
Universidad de Salamanca  
Salamanca, España  
{mconde,fgarcia}@usal.es

María José Casany Guerrero, Marc Alier Forment  
Universitat Politècnica de Catalunya  
Barcelona, España  
{mjcasany,malier}@lsi.upc.edu

**Abstract**— Service oriented approaches (SOA) are currently used to develop new software-as-a-service applications, but they can also be used in the reengineering of existing legacy systems, such as Learning Management Systems (LMS).

The introduction of a SOA to reengineer current LMS can provide LMS with the ability to deliver internal functions as services as well as the ability to integrate external applications as services.

This paper presents an architectural approach to adapt the Moodle LMS to the SOA and some important issues involved in the adaptation are analyzed. Taking into account interoperability specifications, all SOA to LMS adaptation drawbacks are solved by the application of the new architecture.

**Keywords:** *SOA, Web Services, IMS LTI, OKI, interoperability*

### I. INTRODUCTION

Service-oriented technologies and management have gained attention in the past few years, promising a way to create the basis for agility in business processes. Companies can deliver new and more flexible business processes that harness the value of the services approach from a customer's perspective. Service-oriented approaches are used to develop software applications and software-as-a-service that can be sourced as virtual hardware resources, including on-demand and utility computing [1].

But the use of service-oriented approaches (SOA) is not limited to the design and development of new software systems. It can be used to support the reengineering of existing legacy systems, such as Learning Management Systems (LMS).

Current web based LMS are focused on meeting the needs of the institution in providing a basic, common educational platform. Most of universities worldwide have successfully integrated the use of a LMS, where all the academic information services, online contents and learning application are centralized and managed. LMS are a consolidated online learning environment already adopted by learners, teachers and institutions.

Some of the problems of current LMS are:

- LMS often become close communities, leaving no room for an inclusion of additional user preferred features. Often learners are not even allowed to upload additional content. Instead learners are forced to use what has been allocated by tutors, with no flexibility neither with respect to functionality, nor to contents. In a world where the learning process is becoming learning-on-demand many e-learning systems are no longer appropriate.
- There are lots of learning applications, such as mobile learning applications, living outside the LMS ecosystems. Teachers willing to innovate are using applications and technologies not supported by their institution LMS, and by doing so they are taking their students outside the virtual campus. Thus the students need to go to several different sites (using different usernames and passwords) in a scrambled learning environment. This may cause confusion and frustration to students.

There are several SOA applications to learning environments. Most of them have some drawbacks like unidirectional application, complex integration and sometimes a LMS linkage, definition for a unique application scenario, etc.

The purpose of architecture is to define an architecture that allows the export of information from the LMS and the inclusion of applications in the platform. This follows a set of interoperability specifications to ensure integration into different learning platforms. Also, in order to optimize and test this architecture some application domains must be defined.

In the next sections we will discuss some promising standards that are being defined, some SOA initiatives and its drawbacks. Finally will be exposed an architecture proposal to adapt the open source LMS Moodle to the SOA.

### II. LEARNING INTEROPERABILITY STANDARDS AND SOA INITIATIVES

The problem of delivery of educational contents through the Web has been addressed successfully in different ways, as well as its integration in web based Learning Management Systems. There was a clear interest for the industry to port all the contents being created for CD-ROM to the online world. Standards such as Advanced Distributed Learning (ADL SCORM [2] have been widely implemented and adopted. So,

there are standards implemented to create, share, and use educational contents.

One of the big challenges of e-learning systems is the exchange of data between different systems. Current approaches are file-based (i.e, copy files from one system to another). To achieve a better compatibility, e-learning standards and specifications have been developed. However, too many standards already exist for one same problem [3]. Besides, education is not only about content, as the last trends in online pedagogy models make explicit connectivism [4], social constructionism [5]. For the previous reasons, the service orientation approach is used to deal with the interoperability problem between different e-learning systems.

Thus the goals of the interoperability we seek are not bound only to content interoperability, but to a wider scope of features and services that the learning applications can offer.

Interoperability is defined by IEEE as “the ability of two or more systems, or components to exchange information and to use the information that has been exchanged”[6]. The IEEE definition for interoperability is 16 years old, and nowadays software systems can do more things together than just exchange information, for example share functionality. So from the Open Knowledge Initiative (OKI) offers a new definition for interoperability: “the measure of ease of integration between two systems or software components to achieve a functional goal. A highly interoperable integration is one that can easily achieved by the individual who requires the result”. According to this definition, interoperability is about making the integration as simple and cost effective as technologically possible [7].

The Service Oriented Architecture (SOA) is a software engineering approach that provides a separation between the interface of a service, and its underlying implementation. For consumer applications of services, it does not matter how services are implemented, how contents are stored and how they are structured. There is not even exchange of contents but just inclusion of services, so data synchronization between systems and data exchange are not problems. In the SOA approach consumer applications can interoperate across the widest set of service providers (implementations), and providers can easily be swapped on-the-fly without modification to application code.

SOA preserves the investment in software development as underlying technologies and mechanisms evolve and allow enterprises to incorporate externally developed application software without the cost of a porting effort to achieve interoperability with an existing computing infrastructure.

There have been several initiatives for the adaptation of SOA services for LMS and to join LMS to other applications. As an example some initiatives could be considered:

- The adaptation of a part of LMS services to mobile devices [8].
- The definition of service-oriented architectures for the semantic search and retrieval of learning information as the LUISA project [9].

- The integration between different learning tools and systems.[10]

In any case, these initiatives are constrained by the following problems:

- A defined application domain. Not all LMS services are provided, only those which are useful to a specific application domain.
- Unidirectional Interoperability. Architectures work only in one sense, that is to say, provide information from the LMS or integrate it with other tools. But is not possible provide that information and integrate tools in the LMS transparently to users.
- Interoperability Specifications. Definition of a service structure that does not use specifications for interoperability.

In any case, what is intended with this approach is to solve these problems defining a flexible, open and bidirectional architecture.

#### A. The Open Knowledge Initiative

The Open Knowledge Initiative (OKI) was born in 2003 with the purpose of creating a standard architecture of common services that learning software systems need to share, such as Authentication, Authorization, Logging [7]. The OKI project has developed and published a suite of interfaces know as Open Service Interface Definitions (OSIDs) whose design has been informed by a broad architectural view. The OSIDs specifications provide interoperability among applications across a varied base of underlying and changing technologies. The OSIDs define important components of a SOA as they provide general software contracts between service consumers and service providers. The OSIDs enable choice of end-user tools by providing plug-in interoperability. OSIDs are software contracts only and therefore are compatible with most other technologies and specifications, such a SOAP, WSDL. They can be used with existing technology, open source or vended solutions.

Each OSID describes a logical service. They separate program logic from underlying technology using software interfaces. These interfaces represent a contract between a software consumer and a software provider. The separation between the software consumer and provider is done at the application level to separate consumers from specific protocols. This enables applications to be constructed independently from any particular service environment, and eases integration.

An OSID is a description of a logical service between a server provider and a consumer, with independence from the communication framework or data definition language.

For example, services such as authentication are common functions required by many systems. Usually each application has built this specific function. As a result the authentication function is implemented in many ways and this results in information being maintained in different places and being unable to easily reuse. OKI would separate the authentication function from the rest of the systems and provide a central authentication service for all the applications.

OKI describes with OSIDs the basic services already available in e-learning platforms. Among others, these basic services used by many e-learning platforms are described in the following OKI OSIDs:

- The authentication OSID is used to register a new user or to know if the user is connected to the system. This is a basic service in any software system.
- The authorization OSID is used to know if a user has rights to access a service or function. This service is necessary in any system using roles.
- The logging OSID is used to capture usage information. It is useful to know how the system is working for system diagnostics and performance.
- The internationalization OSID is used to change the language of the application or add new languages.
- The configuration OSID is used to change configuration parameters.

Thus using the OKI OSIDs has the following advantages:

- Ease to develop software. The organization only has to concentrate in the part of the problem where they can add value. There is no need to redo common functions among most of the systems.
- Common service factoring. OKI provides a general service factory so that services can be reused.
- Reduce integration cost. The current cost of integration is so high that prevents new solutions from being easily adopted. OSIDs are a neutral open interface that provides well understood integration points. This way there is no need to build a dependency on a particular vendor.

Software usable across a wider range of environments, because OKI is a SOA architecture.

But OKI still has a long way to go before becomes a de facto standard of interoperability. Nowadays up to 75 projects have implemented the OSIDs and given feedback to the OKI community process.

#### *B. The IMS Global Learning Consortium initiatives for interoperability in learning systems*

The IMS Global Learning Consortium is also working since 2005 in standards towards interoperability and integration of learning services and systems.

The IMS Abstract Framework is set of (abstract) specifications to build a generic e-learning framework, which might be able to interoperate with other systems following the IMS AF specifications. IMS AF describes a e-learning system as the set of services that need to be offered. IMS AF is a standard that can be complemented by the OKI OSIDs because OKI provides more specific information about the semantics of the services, how to use them and in what kind of situations they could be used.

IMS also defines the IMS Learning Technologies for Interoperability. While IMS AF and OKI work on the exchange of information and services, IMS LTI developed under supervision of Dr. Charles Severance, focuses on the process on how a remote service is installed on a web based learning system [11].

The OSIDs tells us how to exchange information between the LMS and an external learning application, but how will the teacher and the student reach the application from the LMS? These kinds of proxy bindings are described by the IMS LTI 1.0 and 2.0 standards.

#### *C. IMS Learning Tools for Interoperability*

The basic idea of IMS LTI is that the LMS has a proxy tool that provides an endpoint for an externally hosted tool and makes it appear if the externally hosted tool is running within the LMS. In a sense this is kind of like a smart tool that can host lots of different content.

The proxy tool provides the externally hosted with information about the individual, course, tool placement, and role within the course. In a sense the Proxy Tool allows a single-sign-on behind the scenes using Web services and allows an externally hosted tool to support many different LMS's with a single instance of the tool.

The IMS LTI 2.0 architecture focuses on the launch phase of the LMS-to-tool interaction. The launch accomplishes several things in a single Web service call:

- Establish the identity of the user (effectively like a single sign-on).
- Provide directory information (First Name, Last Name, and E-Mail address) for the user.
- Indicate the role of the current user whether the user is an Administrator, Instructor, or Student.
- Provide information about the current course that the Proxy tool is being executed from such as Course ID and Course Title.
- Provide a unique key for the particular placement of the Proxy Tool.
- Securely provide proof of the shared secret.
- Hints as to display size.
- An optional URL of a resource, which is stored in the LMS – which is being provided to the external tool as part of a launch.

#### **III. A SERVICE-ORIENTED ARCHITECTURE FOR MOODLE: DFWS ARCHITECTURE**

The integration of a Service Oriented Architecture over a LMS requires considering what are the elements to adapt. Depending on the different kinds of LMS, modules and functionalities can change. Instead of this, there are some core functionalities shared between the learning platforms that must be considered in first place. OKI OSIDs will provide some of those common functionalities, but other elements like basic information and basic interaction must be provided. This information could be represented as activities and resources in a LMS.

Moodle adaptation implies consider the different modules in which functionalities and information are distributed, the existing dependences between them and how to adapt interaction.

### A. Initial issues

In this case the adaption of a LMS (Moodle) to SOA requires a process of refactoring. The usage of new technologies introduces new problems in the maintenance and reengineering of the systems developed using them, requiring further new solutions to well face this evolution [12]. In this particular case we meet not only a problem of software engineering: open source community dynamics and governance issues need to be addressed also. The resulting solution needs to be valid for the whole community of users and developers.

In a previous project the authors of this paper had successfully refactored the Moodle LMS to SOA implementing the OKI OSID and the IMS LTI standards [13]. It is a good example of adaption of an LMS to an interoperability standard involving a project of several thousands of hours of software engineering. But this project was only a third party contribution to the Moodle community, since it is not part of the core distribution, it will not be compatible with higher versions of Moodle.

Moodle needs a way to be easily extended and customized in a way that the developments maintain compatibility with new versions of the software that are released periodically. Another important issue in the refactoring of Moodle, is the selection of the services that have to be accessible using SOA. Once these services have been selected they can be used by a variety of applications such as mobile applications to access the LMS.

The refactoring entails the solution of both a decision problem for establishing what can be migrated from the original legacy system, and of a technical problem concerning how the migration can be executed. As to the first problem, structured approaches to find candidates for a Web Service are needed [14].

### B. A layered approach

From the architectural point of view Moodle is based on a model-view-controller controller. This pattern is common in interactive applications that evolve rapidly. This architecture is complemented by other patterns that provide flexibility to the system.

The adoption of the SOA and its integration in Moodle requires a deep knowledge of a system core library that, due to an evolutive development is not particularly consistent. The core system is structured in modules, each of them providing a wide set of functions. Each module has a connection and access policy based on roles. This policy has to be considered in the design of the services.

The Moodle lead developer and founder Martin Dougiamas, assigned in early 2008 to the team in UPC [15] the task of developing a new API to access the services of the Moodle core system, with independence of its implementation, that may remain stable in the following versions of Moodle. This task is described in the Moodle tracker [16] and in Moodle Docs [17]. This API consists on a set of Web services that encapsulate most of the services that an external (and even internal) application shall need

from a Moodle server. In October 2008 this Web services layer has been integrated in the Moodle standard distribution for Moodle 1.9.3 and is going to be the standard interoperability subsystem for the future versions of Moodle.

This layer is intended to be useful for all developers who want to build applications for Moodle, because this development can lead to a documented and stable API to hack into Moodle that should overcome new versions of Moodle.

This API is the base to develop a set of Web services served by Moodle: Moodle-DFWSs.

Moodle needs to be accessible using any transport protocol present or future. So it cannot depend on a concrete Web services protocol, name it XML-RCP, SOAP, REST etc. Moodle-DFWSs be implemented in the present version of Moodle (Moodle 1.9) and in the future versions as well (appearing as a core feature in Moodle 2.0 expected early 2009).

Moodle-DFWSs architecture need to be extendable, so each Moodle Module can be a service provider. The proposed architecture consists in 3 layers described in Fig 1:

- Connectors Layer: Contains the connectors that implement services to local or remote applications.
- Integration layer: This layer consists on The API (being implemented) that provides a one-point access to the Moodle plus contrib functionalities.
- Services Layer: Is where real things happen. The API knows how to deal with the Moodle core, and in future posts we will deal on how the activity modules, course formats and plugins can offer their services to the clients.

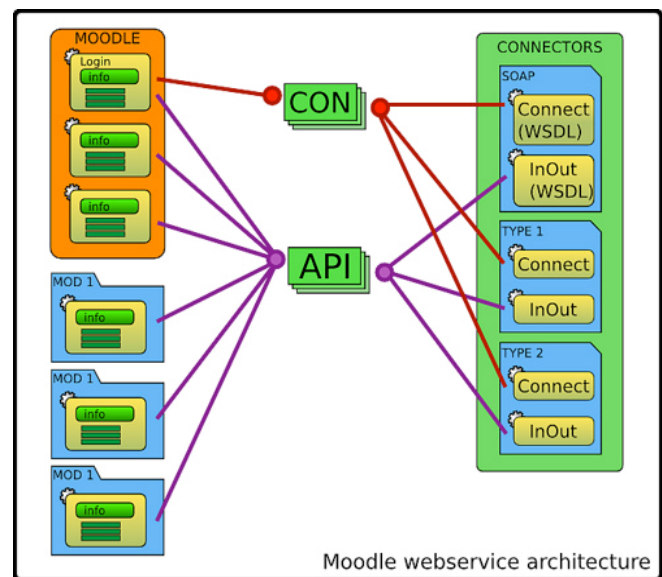


Figure 1. Moodle DFWS architecture.

The Connect layer can implement connectors adjusted, without hacking inside Moodle and creating code that will survive the new releases of Moodle for some years, to behave according to different standards. One of these connectors will be a Campus Gateway clone, so Moodle will implement the OKI OSIDs v3 on 2009.

How could this components work together? It depends on the finality of the use. Considering an use domain like a mobile consumer, the mobile client would be connected to the system by using the connector layer. After being authenticated, the client must be authorized, and would be able to use the part of the API proper to its role. This API will provide the information and/or interaction required by the client.

Other use of the architecture could be the integration of an external application by using a proxy tool based on IMS LTI.

### C. Future work

The Moodle core team agreed that for the version 2.0 of the system a refactoring of the core functionalities is required, structuring a clean access layer to the core (Fig 2. (2) Internal layer). This layer provides a clean interface where a layer of SOA services can be provided to external applications (Fig 2, 4)). It is a PHP API that can be wrapped in custom connectors to implement different standards of Web services and semantic behaviors like the OKI OSID's.

The moodle internal API is an extension of the DFWS architecture for Moodle 2.0

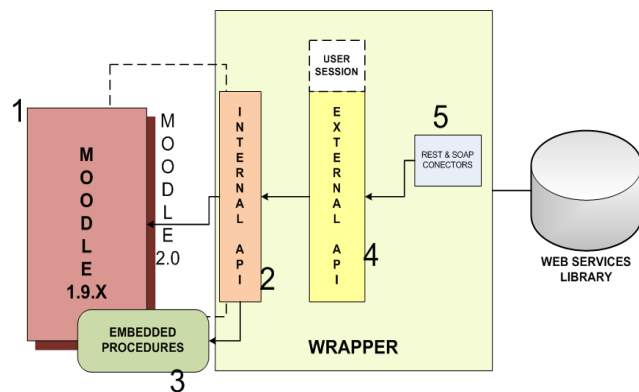


Figure 2. Moodle 2.0 core refactoring

## IV. CONCLUSION

The architecture proposed for the Moodle LMS provides the necessary technologies and development to adapt Moodle to the SOA approach. This is a first step so that Moodle can offer students and teachers services inside the LMS. These services will provide access to the LMS system from the Moodle interface. These services that provide access to the Moodle core system can also be used from a variety of external applications such as mobile-learning applications. This may be a way to extend the LMS to the mobile scenario.

Taking into account the existing shortcomings in other SOA adaptation initiatives, the purposed architecture settles them. To do this a layer architecture which considers specifications is proposed. The use of OKI OSIDs and IMS LTI guarantees the portability of the architecture and the bidirectional work. LMS information could be used by external applications, and external applications could be integrated transparently to users.

## REFERENCES

- [1] H. Demirkan and R. J. Kauffman, J. A. Vayghan, HG. Fill, D. Karagiannis, P. P. Maglio, "Service-oriented technology and management: Perspectives on research and practice for the coming decade", *Electronic Commerce Research and Applications*, Volume 7, Issue 4, Winter 2008, Pages 356-376.
- [2] ADL- Advanced Distributed Learning. ADL SCORM: Retrieved February 20, 2009 from <http://www.adlnet.gov/>
- [3] G. Vossen and P. Westerkamp, Why service-orientation could make e-learning standards obsolete. In *Int. J. Technology Enhanced Learning*, Volume. 1, Nos. 1/2, 2008, Pages 85-97.
- [4] T. Arina, Learning Zeitgeist: The Future of Education is Just-in-Time, Multidisciplinary, Experimental, In *Masternewmedia*. Retrieved July, 20, 2008 from [http://www.masternewmedia.org/news/2008/02/13/learning\\_zeitgeist\\_the\\_future\\_of.htm](http://www.masternewmedia.org/news/2008/02/13/learning_zeitgeist_the_future_of.htm)
- [5] M. Alier, A social constructionist approach to learning communities: Moodle. In *Militiadis D. Lytras (Ed.), Open Source for Knowledge Management: strategies beyond tools*. Hersley: Idea Group Publishing, pp. 369-381, 2008.
- [6] IEEE standard computer dictionary : a compilation of IEEE standard computer glossaries. IEEE Computer Society Press, New York, NY, USA, January 1991
- [7] J. Merriman, Redefining interoperability. The Open Knowledge Initiative (OKI). Retrieved May 5, 2008, from <http://www.okiproject.org/view/html/node/2916>
- [8] S. Kurz, M. Podwyszynski and A. Schwab. A "Dynamically Extensible, Service-Based Infrastructure for Mobile Applications" Springer Berlin / Heidelberg Editors. *Advances in Conceptual Modeling – Challenges and Opportunities*. Volume 5232/2008. 10 de octubre de 2008. ISBN: 978-3-540-87990-9
- [9] LUISA. Learning Content Management System Using Innovative Semantic Web Services Architecture. Retrieved February 20, 2009 from <http://luisa.atosorigin.es>
- [10] S. Pätzold, S. Rathmayer and S. Graf. "Proposal for the Design and Implementation of a Modern System Architecture and integration infrastructure in context of e-learning and exchange of relevant data". *ILearning Forum 2008*. European Institute For E-Learning. pp 82-90, ISBN: 2-9524576-4-6.
- [11] IMS Global Learning Consortium. IMS Tools Interoperability Guidelines. (2006). Retrieved June 10, 2008 from <http://www.imsglobal.org/ti/index.html>

- [12] G. A. Di Lucca, N. Gold and G. Vesaggio, "Guest editor's introduction: 10th Conference on Software maintenance and Reengineering, Journal of Systems and Software, Volume 81, Issue 4, April 2008, Pages 461-462.
- [13] Projecte Campus. Retrieved February 20, 2009 from <http://www.lafarga.org/campus/en/index.php>
- [14] G. Canfora, A.R. Fasolino, G. Frattolillo and P. Tramontana, "A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures", Journal of Systems and Software, Volume 81, Issue 4, April 2008, Pages 463-480.
- [15] DFWikiLABS. Retrieved February 20, 2009 from <http://www.dfwikilabs.org/>
- [16] Moodle Tracker. Retrieved February 20, 2009 from <http://tracker.moodle.org/browse/MDL-12886>
- [17] Moodle Docs. Development: Web services. Retrieved February 20, 2009 from [http://docs.moodle.org/en/Development:Web\\_services](http://docs.moodle.org/en/Development:Web_services)