Moodle 2.0 Web Services Layer and Its New Application Contexts

Miguel Ángel Conde, Diego Alonso Gómez Aguilar,
Alberto Del Pozo de Dios, and Francisco José García Peñalvo

Departamento de Informática y Automática, Universidad de Salamanca,
Plaza de los caídos S/N, 37008, Salamanca. España
{mconde,dialgoag,delpozo1988,fgarcia}@usal.es

Abstract. Owing to the intrinsic relation among actual education and new technologies, it results essential the fact to found the new ways to satisfy both sides of the modern eLearning platforms, the needs of students and tutors and the enough technologies to support it. Consequently, the possibility to interconnect the LMS with other external applications to enrich and strengthen the comprehension of learning process is one of the principal paths to follow.

Keywords: Moodle 2.0, Web services, PLE, backoffice, new learning moodels, SOA.

1 Introduction

Since the apparition of the web and other technologies related with it the manner of teaching and learning have been changing. Thus, when talking about eLearning its constant evolution must be considered. This development is no longer just due to technological changes that support new models of learning but also to new motivations, trends, educational models and so on. Some of those changes are:

1. With the emergence of the first PLE (Personal Learning Environment) is producing a shift in *elearning* approaches. The online course is no longer seen as an extension of face to face learning, but begins to focus on the student, their needs and preferences [1].
2. There is also a gap between the actors involved in the learning process. Many of the students were born in the digital age and all technological resources used in different contexts (digital natives). Moreover teachers should use some of those resources without understanding or having much knowledge of them (digital immigrants) [2].
3. The expansion of sozializing trends, constructivism, applications, tools and paradigms could play an important role in the so-called *eLearning* 2.0 [3].
4. New educational strategies such as those linked to the Bologna process requires support to new learning strategies, such as the recognition of non-formal and informal learning, support life-long learning, to make learning attractive and to strengthen links with working life and society [4].

M.D. Lytras et al. (Eds.): TECH-EDUCATION 2010, CCIS 73, pp. 110–116, 2010.

In this situation, the technology must provide support for different situations, defining, in each case, more comprehensive and specialized contexts and scenarios.

This paper will present one of these new scenarios such as a Service Oriented Architecture approach applied to eLearning. Specifically considering the web services layer for Moodle 2.0 and one of its possible uses, a backoffice application.

To do this, first step is an introduction to the theoretical basis and related works, then the backoffice client application and its funcionlaties will be described, finally some conclusions and future work will be exposed.

2 Related Works

By way of make easily to recognize and understand the related works, thus will be divided in SOA approach in education and Moodle web services layer.

2.1 SOA Approach in Education

As Dr Charles Severance [5] (founder of the Open Source LMS Sakai and currently working for IMS) states: LMS "are all mature enough that the majority of faculties and student users are generally satisfied regardless of which system chosen". It seems that LMS systems have achieved a stability and maturity, a Golden Age of LMS; LMS adopt each other's features and are slowly beginning to look like clones of one another. Nevertheless the LMS, as has previously been mentioned, it must be reinvented or be extinguished due to the new eLearning realities.

To incorporate the new generation of learning applications inside the LMS there is a need for interoperability between systems. One new approach to reach interoperability between different systems is the Service Oriented Architecture (SOA). 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. 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. [6]

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 [7].
- The definition of service-oriented architectures for the semantic search and retrieval of learning information as the LUISA project [8].
- The integration between different learning tools and systems.[9]

Each of these Initiatives are trying to obtain new benefits from the LMS. In the following sections another initiative. Also will be described a new backoffice client that

allows final user to exploit new contexts. Our approach is to propose and develop novel solutions that help different users of LCMS to extract specific knowledge related to the complex process of education and learning throughout the use of a web services. To do this a SOA approach based on Moodle is considered.

2.2 Moodle Web Services Layer

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 [10] 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 [11] and in Moodle Docs [12]. 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. These layers are:

- 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.

The Connectors 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. This will be the base of a great area of new LMS applications.

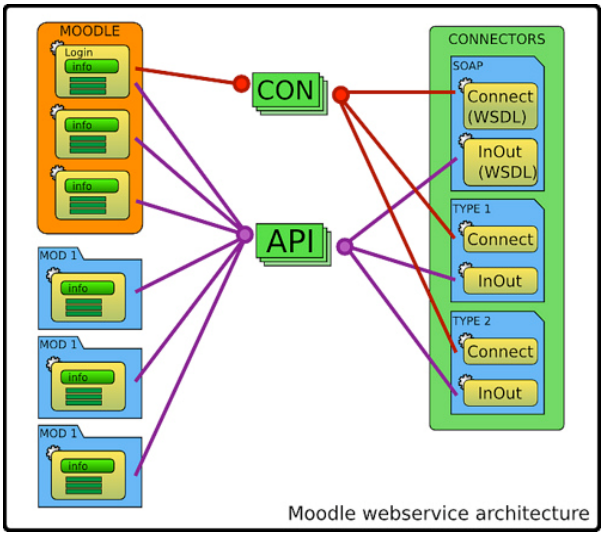

Fig. 1. Moodle Webservice Architecture

3 Backoffice Web Client

3.1 Main Reasons for This Development

Moodle is being consolidated more and more, as today shows that there are 45,494 registered sites using this platform [13]. Given this number, and since in many cases it is necessary to use them in institutions with proprietary systems, new needs appear. Among these new requirements can be the integration of Moodle with backoffice tools.

The idea behind is that the application can interact with several systems at once, performing the same action in several places, making it all work together. Instead of making a transaction twice, only an integrated operation is required.

To do this, a new tool has been defined. This tool lets users perform backoffice operations using Moodle Web services layer. Also, by way of example, the tool has other features not included in Moodle, which could be useful in managing activities. It also allows testing the operation of the web services layer and the detection of functional gaps.

3.2 Backoffice Tool Main Features

The creation of this tool shows that alternative interfaces and clients could be created. These clients can interact with Moodle, increasing greatly its capacities, from being a monolithic platform to an interoperable application.

To do this the Moodle Web services layer is required. This layer consists of a set of contracts that make use of certain functions defined in Moodle externallib. These contracts are described in Figure 2 using SOAml[14].

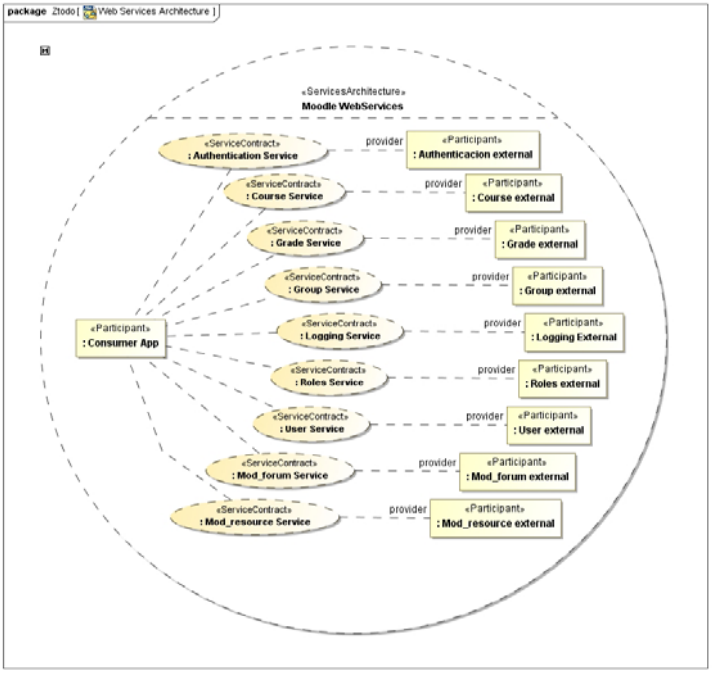

Fig. 2. Some of Web services Moodle contracts

Since the connectors corresponding functions are invoked, which use Moodle code to perform the appropriate tasks and, depending on its mission, return, create, modify or store information in the database. Specifically in this case, single management system will be considered. These may include:

- Client administration: This is the main part of the tool. By now it only allows choosing the protocol that is going to be used to connect with the platform (REST, SOAP, XML-RPC) and testing the functions existing in the externallib's code.
- Users administration: This client permits total management of Moodle users trough a simple and intuitive interface. The aim is to facilitate the users administration without accessing to the platform.
- Courses administration: This part provides the main features associated with courses such as creating, modifying, deleting and viewing their settings. The content management into the courses depends on the Web Services API development, but at the moment all the main contents can be accessed and managed from the client.
- Roles administration: Roles administration allows assigning capabilities based on the context where user is located. In addition to this, roles can be created, modified or delete from this part.
- Logs administration: It allows administrators to control the activity in Moodle, so that they can view the logs happened in a course, a date or made by a user.

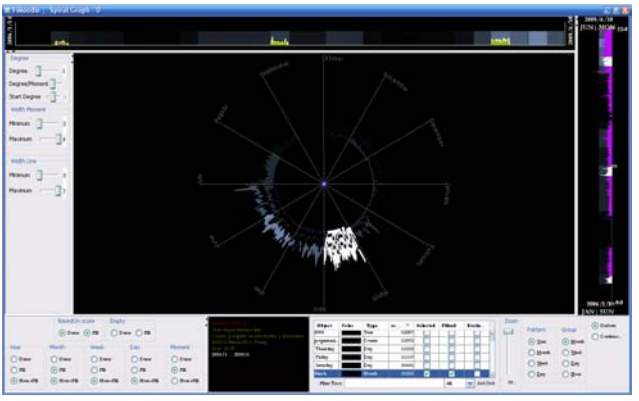

Fig. 3. Information Visualization Applet

3.3 Other Funcionalities

In addition to all the possibilities that web services allow, such as the creation of clients and tools that can access to Moodle from alternatives and adapted to new technologies interfaces, Moodle web services permits to create several new features that don't exist it the platform. This client have some of this new features in order to make some actions simpler and exploit the API potency:

- Alert inactive users: Every platform has a lot of inactive users that has never entered the platform or they don't do as usual as they should. Usually, teacher must access to user logs and find the last access to warn them (sending a message) if necessary. This client has a feature that allows teachers to choose the number of days that users must have been inactive and the text to send. The tool looks for the users and send the message. When finished, it displays a list with all the users that should have received the message.
- Logs visualization tool: According to [15], Visual Analytics is an emerging area of research and practice that aims to support analytical reasoning through interactive visual interfaces. This kind of analysis can feedback, with non-evident information from the LMS, reports that facilitate the learning, monitoring and making significant business decisions regarding courses. A visualization tool for temporal information analysis [16] is used in this application. The main target of the tool is to find behaving patterns in learners and teachers. This tool will consist in a JAVA applet included in the client, and it will connect to Moodle so as to get logs trough the Web services. After this, the tool will show the information in a suitable way Fig 3.

4 Conclusions

Taking into account the changes that are occurring in the learning processes, technology must provide solutions. Specifically, SOA approaches will provide some of them, as is the example of Moodle Web services layer. This approach will facilitate the

implementation of external Tools, as the previously described, which interact transparently with the LMS.

This kind of approach will provide reusability, flexibility, scalability and compatibility to eLearning platforms. It is also possible to extend the functionality of learning platforms as it does backoffice tool. Thus allowing the opening of learning to other technology trends as could be visualization of information. Such proposals open new ways for learning, in the search of the personalization of learning, with the PLE and iPLE.

References

1. Johnson, M., Liber, O.: The Personal Learning Environment and the human condition: from theory to teaching practice. Interactive Learning Environments 16(1), 3–15 (2008)
2. Prensky, M.: Digital Game-Based Learning. McGraw-Hill, New York (2001)
3. Ajjan, H., Hartshorne, R.: Investigating faculty decisions to adopt Web 2.0 technologies: Theory and Empirical Tests. The Internet and Higher Education 11(2), 71–80 (2008)
4. Ortiz, G.: Experiences with Virtual Environments in the EHEA. In: Proceedings of the 2008 Third international Conference on internet and Web Applications and Services, June 08 - 13, 2008, ICIW, pp. 54–59. IEEE Computer Society, Washington (2008)
5. Severance, C.: A Meteor Strike May Cause LMS Systems to become Extinct! Dr Chuck Severance Weblog, http://www.dr-chuck.com/csev-blog/000606.html
6. Forment, M.A., Guerrero, M.J.C., González, M.Á.C., Peñalvo, F.J.G., Severance, C.: Interoperability for LMS: The Missing Piece to Become the Common Place for Elearning Innovation. In: WSKS, vol. (1), pp. 286–295 (2009)
7. Kurz, S., Podwyszynski, M., Schwab, A.: A Dynamically Extensible, Service-Based Infrastructure for Mobile Applications. In: Song, I.-Y., Piattini, M., Chen, Y.-P.P., Hartmann, S., Grandi, F., Trujillo, J., Opdahl, A.L., Ferri, F., Grifoni, P., Caschera, M.C., Rolland, C., Woo, C., Salinesi, C., Zimányi, E., Claramunt, C., Frasincar, F., Houben, G.-J., Thiran, P. (eds.) ER Workshops 2008. LNCS, vol. 5232, pp. 155–164. Springer, Heidelberg (2008)
8. LUISA. Learning Content Management System Using Innovative Semantic Web Services Architecture, http://luisa.atosorigin.es (Retrieved January 19, 2010)
9. Pätzold, S., Rathmayer, S., Graf, S.: Proposal for the Design and Implementation of a Modern System Architecture and integration infrastructure in context of e-learning and exchange of relevant data. In: ILearning Forum 2008. European Institute For E-Learning, pp. 82–90 (2008) ISBN: 2-9524576-4-6
10. DFWikiLABS, http://www.dfwikilabs.org/ (Retrieved January 19, 2010)
11. Moodle Tracker, http://tracker.moodle.org/browse/MDL-12886 (Retrieved January 19, 2010)
12. Moodle Docs. Development: Web services, http://docs.moodle.org/en/Development:Web_services (Retrieved February 20, 2009)
13. Moodle Statistics, http://moodle.org/stats/ (Retrieved January 19, 2010)
14. Estefan, J.A., et al.: Reference Architecture for Service Oriented Architecture Version 0.3 (2008) (19 01 2010)
15. Thomas, J.J., Cook, K.A.: Illuminating the Path: The Research and Development Agenda for Visual Analytics: National Visualization and Analytics Ctr. (2005)
16. Gómez Aguilar, D.A., et al.: Semantic spiral timelines used as support for e-learning. Journal of Universal Computer Science (j-jucs) 6 (2009)