A Framework for Helping Developers in the Integration of External Tools into Virtual Learning Environments

Emilio Julio Lorenzo
LTCS group at UNED
C/ Juan del Rosal, 16 28040 Madrid
Spain
emiliojulio@lsi.uned.es

Roberto Centeno
LTCS group at UNED
C/ Juan del Rosal, 16 28040 Madrid
Spain
rcenteno@lsi.uned.es

Miguel Rodriguez-Artacho
LTCS group at UNED
C/ Juan del Rosal, 16 28040 Madrid
Spain
martacho@lsi.uned.es

ABSTRACT

In this paper we present a lightweight abstract framework to help developers to integrate tools into a learning environment. This approach is based on an abstract declarative layer and an integration library. The proposal follows the cloud computing paradigm, defining and storing the different integration models remotely. Finally, the proposed integration framework has been successfully used for integrating a tool called JBraindead into a real educational web environment. This integration showed how developers reduced the coding time, due to the fact that they did not have to deal with e-Learning standards and techniques and could focus on coding the own tool.

Categories and Subject Descriptors

K.3.1 [**Computer Uses in Education**]: Computer-assisted instruction (CAI) D.2.3 [**Coding Tools and Techniques**]: Standards D.2.11 [**Software Architectures**]: client/server

General Terms

Design, Standardization, Languages.

Keywords

Educational tools, virtual learning community, integration, metamodel, cloud computing.

1. INTRODUCTION

In the last few years, learning environments have evolved from a monolithic application perspective to a modular application based on web services and cloud computing [12]. The new trend of cloud computing and mush-ups, makes virtual learning community users deal with several specialized tools with complex knowledge representations for instructional purposes, such as search and classification using concept maps, semantic annotation, and personalized user interfaces. This scenario forces instructors and pedagogical content authors to hire expert developers. Sometimes, those experts are not familiarized with e-learning topics and standards and, thus, they are not able to take advantage of the e-learning state-of-the-art [1] that would allow them to improve the interactions between their own tools. Common problems related to this approach such as persistence, searching, sharing and, creating pedagogical scenarios with learning objects have been deeply discussed in the e-learning community. Educational experiences, like [18], show how it is necessary to integrate new tools into a certain learning environment. These proposals are usually composed of a Learning Management System (LMS), to manage users and activities, along with a Learning Object Repository (LOR), to manage educational objects and their storage. Unfortunately, developers are sometimes reluctant to learn and implement new protocols. The large number of standards and their continuous update make it difficult for developers to follow the latter trends (Figure 1).

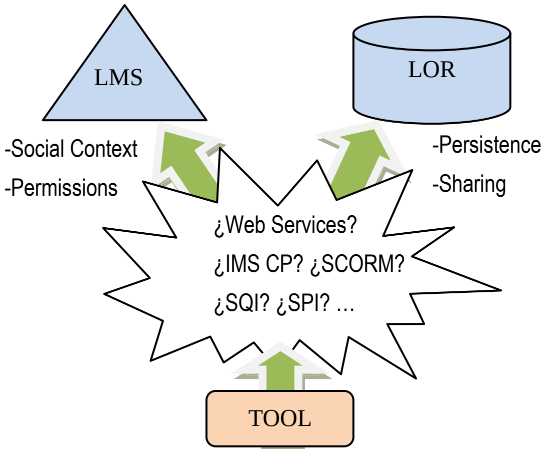


Figure 1. Proposed virtual learning environment

In order to solve these problems to a degree, we propose a lightweight abstract architecture partially based on cloud computing fundamentals. That is, the Virtual Learning Environment is not only a LMS but also it is compound of a LMS as the social provider, a LOR as Learning Object manager, coupled with the educational tools.

The goal of this approach is twofold: on the one hand, it aims to help developers to build educational authoring tools and; on the other hand, it aims to integrate them into a virtual community, together with other tools and resources. In order to do that, standards or ad-hoc protocols are used, depending on the case. Thus, developers can focus on the core of the application and reduce the time they spent on designing, coding, training and learning the new framework

ISELEAR'13, November 14–25, 2013, Salamanca, Spain.

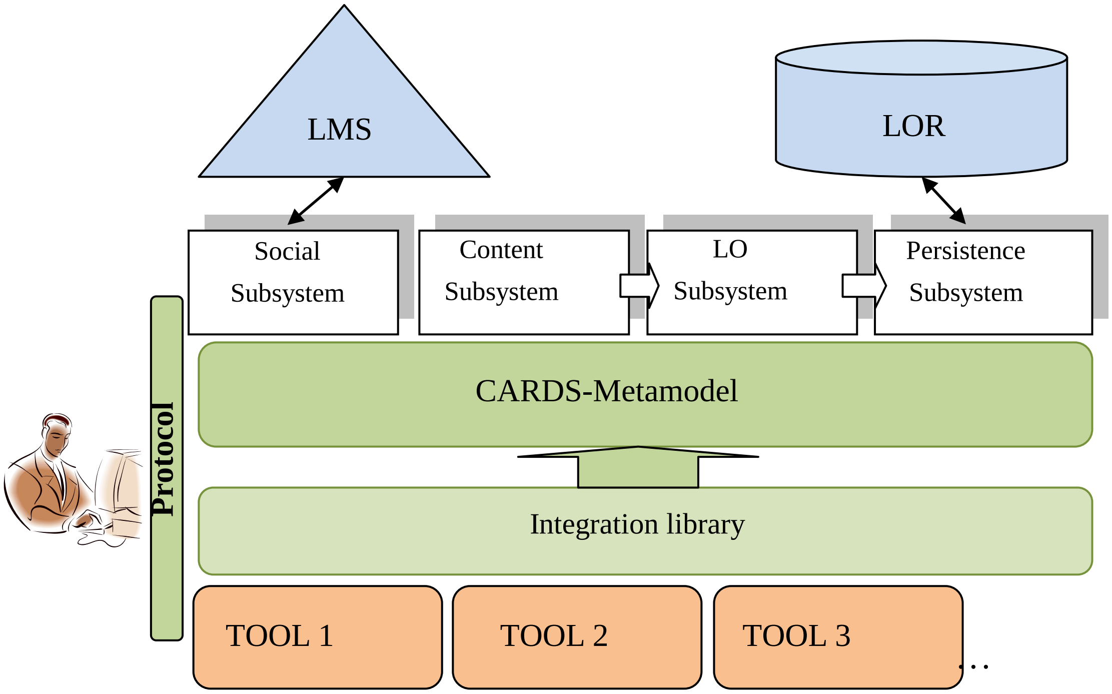


Figure 2. Framework architecture

In the context of this research, students are seen as 'prosumers' involved into constructivist long-term ubiquitous activities; and the classic Learning Object concept of learning units is extended so that a variety of artifacts such as students' results, or the required resources to set a tool for an activity, are included [7].

The rest of the paper is organized as follows: first, in Section 2 we present some related work. Second, section 3 presents the proposed integration framework, describing its main components. Then, section 4 describes a real case study testing the proposal. Finally, Section 5 puts forward some conclusions and future work.

2. RELATED WORK

Integration is an important issue in learning and there are a lot of standards related to this issue. To the best of our knowledge, regarding integration with a LOR the two most relevant standards are IMS SQI [17] and OAI [14]. However, both of them only cover how to query the repository but not how to store new content. In order to cover this, some additional protocols are required, for example IMS SPI [16]. SCORM and IMS CP describe how learning objects together with their metadata might be packaged into a LOR. However, not all LORs are able to store both types of objects.

Regarding the integration of LMS, IMS LD [9] is mainly focused on describing activities paths. Other proposals such as IMS LTI [10] or OKI [15] try to integrate tools within LMS. However, they are usually very complex and, thus, their usage requires a great deal of effort. Besides, few LMS follow standards recommendations and they have their own ad hoc API.

On the other hand, there are several reference models (e.g. IEEE LTSA [8]) helping the design and development of educational tools. However, they are usually just recommendations that are not supported by an implementation. On the other hand, when an implemented proposal is presented, it is usually too much centered in a particular standard and/or technology [20], suffering from a strong dependence.

With these limitations in mind, new approaches have emerged with the objective of behaving as proxies or bridges between tools and LMSs, like GLUE [1] or [4]. However, they are usually based on complex architectures or they do not make a clear difference between the social context and the persistence of the results.

In contrast with other approaches, our proposal put the focus on the tool and makes a clear separation between LMS and LOR, but taking into account that both are necessary in order to support constructivist activities [19].

3. INTEGRATION FRAMEWORK

As it has been previously introduced, the aim of this research is to include tools into an eLearning environment, composed by a LMS and a LOR like the backbones of the environment. In order to reach this goal a framework is proposed, including several integration layers, to cover the necessities that a developer should have when dealing with eLearning.

Figure 2 shows the architecture of the framework. As it may be seen, it is composed of different integration layers, allowing the developer to use different subsystems to integrate the tool into a VLE. On one hand, the social subsystem allows the integration with a LMS; and on the other hand, the content, Learning Object and persistence subsystems make it possible to wrap the result of the tool and store it as a Learning Object into a repository.

In order to use the subsystems, the developer must follow a defined protocol using an integration library to program the necessary code inside the tool. Depending on the functionalities of each subsystem, the developer should have to design integration models, such as the content model, to wrap the result of the tool.

3.1 Technical architecture

As we can observe in Figure 2, the main components of the technical architecture that support this integration framework are the following: *i)* a meta-model engine; *ii)* an integration library; and *iii)* an integration protocol.

The meta-model tool, called CARDS-Metamodel [12], is a Web oriented application, which allows the creation of declarative models, via an online interface. These models are based on key-value fields with semantic about the type of the value and how to fill and show them properly. Therefore, it clearly works like an authoring tool based on cloud computing. That is, the different models, containing the information about the needs of the tools, are stored in an external server, as learning objects as well.

The integration library is coded on java, providing objects and methods to help developers to code the necessary parts for the integration of the tools. This includes an easy way to make the communication with Web services. The library allows managing the models, which are created with CARDS-Metamodel, using different levels of abstraction. Thus, developers do not have to be concern about how to store and retrieve the models, nor about how to install the new framework.

The framework protocol can be seen as an integration contract that allows developers to correctly use the integration library, as well as the CARDS-Metamodel, in order to cover each proposed subsystem. This contract depicts the data models and technologies to implement a subsystem, as described in [5]. An example of the protocol is showed in section 3.6.

The functionalities provided by the previous components are grouped into four subsystems depending on the kind of integration the developer chooses. Each subsystem is explained in the following sections.

3.2 Social Subsystem

This subsystem is responsible for the communication between a learning tool and a LMS. Thus, it provides social context knowledge to the tool from the LMS. The basic social information provided is:

- Current user. Username of the current user that is using the tool.
- Permissions. The basic permissions that the user has within the tool.
- Group. The group within which the user is included in the current activity.
- Activity. The current activity within which the user is involved.
- Task. The current task within the activity that the user is carrying out.

All these fields are involved with the use of the tool. The values of these fields depend on the kind of the LMS, which the tool is related to. This is so because not every LMS has the whole information [1]. This information is modeled by special social templates in CARDS-Metamodel (see Figure 3)

3.3 Content Subsystem

This subsystem encapsulates the functionalities in charge of assisting in the aggregation of the tool outcomes and incomes. It also offers a web interface acting as content authoring tool. In the previous work presented in [11], the main features of this subsystem are showed. Mainly, it is in charge of providing an internal or final result model to the educational tool.

Those models are declared by CARDS-Metamodel interface and stored in a repository outside the tool context. Thus, the fundamentals of cloud development are followed. The tool must get the models to create their result by web services, although supported by a java library.

This way of developing tools requires a “contract” between the fields declared and the code in the tool. The developer must know the name of the template, representing the final result of the tool, and the name of the fields, which will be filled with the result content.

3.4 Learning Object Subsystem

This subsystem is in charge of creating tool results as learning objects and adding semantic information via metadata automatically generated by other models. Therefore, the tool can choose between using one of the several package models in the literature, like SCORM and performing an ad-hoc packaging. The best way to use these functionalities is after using the content subsystem, because several features are related.

Regarding metadata, several sources, apart from the user, are taken into account and supported:

- The system: This is automatically gathered from the system when the object is created. For example, the date.
- The tool itself: The tool looking at its internal model gathers metadata. For example, if the tool is related to images, the metadata resolution of a particular image is easier to get from the tool than from the user. This process is fully supported by libraries, so it has to be programmed into the tool.
- The final result content: Our proposal allows the generated models to map their content fields to metadata, when designing in a declarative way. Thereby, when the learning object is created, the selected content will be copied into the metadata file. In order to use this functionality the content subsystem must be previously implemented.

3.5 Persistence Subsystem

It deals with storing, searching and recovering tool outcomes and incomes in a learning object repository. The way to communicate with the repository is by means of web services. The standard SQI [17] is the principal communication protocol used by repositories. However, due to their limitations, like compulsory anonymous sessions, several repositories have their own protocol, or a combination of them.

These two different options are supported, in this proposal, by the java library. In particular, the following methods are defined, as an extension of the SQI ones:

- saveToolResult. It saves the tool final result into a particular repository. This feature is not included in SQI, but in SPI [16].

- getToolResultByLOID. It gets a tool final result from the repository, searching by an identifier.
- getToolRestultListBySearch. It gets a list of tool final results, searching by metadata matching.
- getToolResultBySearch. It gets one tool final result from the repository, searching by metadata matching. It is used when the programmer knows there will be just one result, in order to optimize the query.
- deleteToolResultByID. It removes a tool result (learning object) from the repository by its identifier.
- getAllMetadataValues. It returns all values of a concrete metadata field. This method is useful in some occasions. For example, to show an options list in a search web page.

3.6 Explaining the framework protocol

In this section it is explained how developers can use the proposed framework, following a particular protocol. Figure 3 shows an example of the sequence of that protocol. ❶ The tool developers declare the models required by their tools. This is made online using CARDS-Metamodel. In the example a particular LMS, called Organization2, is chosen, along with information about the current user and their permissions. ❷ The meta-model tool fills the models with information from the specific source. ❸ The educational tool can take the information from the model. In the example showed in Figure 3, JSMITH is the current user with read only permission. This information can be sent to the tool by means of web services. In the figure, two tool examples are shown: *tool 1* uses e-Learning standards, IMS LD to communicate with LMS and SQI with the LOR; *tool 2* uses ad-hoc APIs, like *moodle* as LMS and *Flicker* as *Repository*, but the model declaration is similar to *tool 1*. An example of the implementation protocol for the content subsystem is showed in [11]

4. FRAMEWORK IN ACTION: THE JBRAINDEAD EXAMPLE

In this section a real case study is presented. In it a particular tool, called JBraindead [13], is integrated into an educational Web environment by using the proposed framework.

JBraindead is an online classifier of unstructured content using Formal Concept Analysis (FCA), along with didactic information to get educational resources from Internet, as web links.

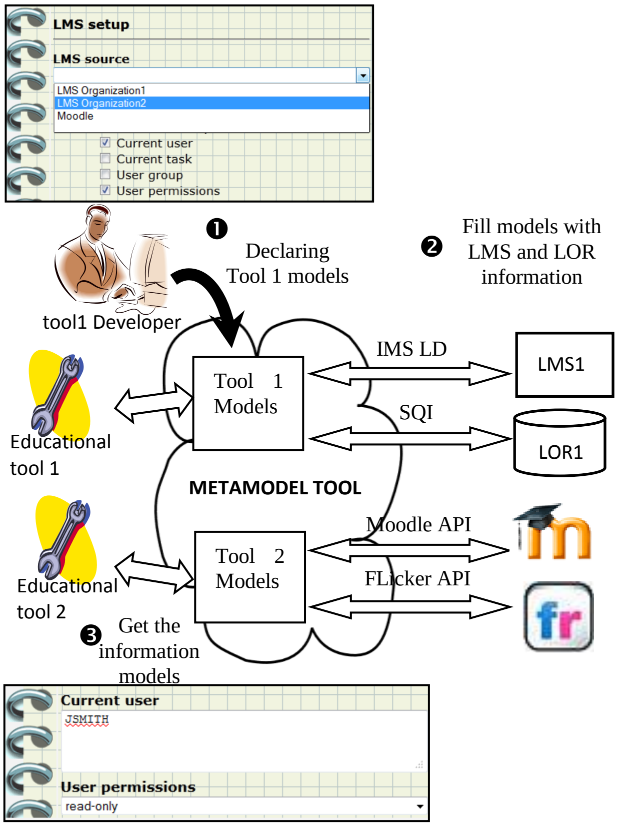
Although educational activities were not the main goal of the tool, it can be used in that kind of activities. For example, to gather information related to a concrete topic, similar to [19]. Therefore, it was interesting to include it into a virtual learning environment. Although the tool included e-learning concepts, such as searching guided by didactic goals, they are not developed by programmers with e-learning knowledge. Also, the code was not oriented to be integrated into an educational environment, for example there is no way to save the results.

The first necessity that derives from using the tool was to store resources previously found, as URL links, into a repository for their later use. Therefore, a communication protocol and a package method had to be implemented. When using standards as SCORM or IMS CP, it is not clear where to store only URLs, without creating a typical learning object. The estimated time for the developer to accomplish this task was one month. Later, another problem appeared, because the tool had not registered users and there was not control of the current user or activity.

In order to solve these problems, the proposal architecture was used. Firstly, by using the content subsystem, a new content model was created. This model was compound of a card with two fields:

- url. Defined as a multivalued url field. That is, one or several urls can be stored and the semantic associated is to display a hyperlink when the content of the object is showed.
- comment. This is an extra field, suggested by an educational instructor, to include more information about the resources and to be displayed along with the content and not only as metadata. For example, the didactical orientation of the object and the user opinion about that. In future implementations more fields will be included to better classify the comments.

Once the model was declaratively created, the programmer had to include new code into the tool. Using the java library, a web service connection was created to import the template with the new content model, when the tool started. Then, when the user wants to store the resources, a new content is created and stored, as a learning object, into a repository with ad-hoc communication protocol [3].



LMS setup

LMS source

LMS Organization1
LMS Organization2
Moodle

☑ Current user
□ Current task
□ User group
☑ User permissions

Current user
JSMITH

User permissions
read-only

Figure 3. An example of the framework protocol

When the tool is launched from a LMS, it is possible to get the user and current activity, using the social subsystem and CARDS-Metamodel social templates. This way, the code is similar to the previous one and the developer does not need to learn more APIs or protocols.

With this solution, the intrusion in JBraindead code is slight and the development time is cut down to two days. Now the tool is within a virtual learning community and students can use, at home, the LOR Web interface to recover and modify their outcomes. The content model also allows other tools to show the results, through the meta-model Web interface, without JBraindead intervention.

5. CONCLUSIONS

This paper has proposed a new lightweight abstract integration framework, with the aim of helping developers to create and integrate tools into virtual learning communities (with a LMS and a LOR). The framework is partially based on the cloud-computing paradigm. Thus, its functionalities, encapsulated by different subsystems, allow us to create different integration models for the tools and manage them properly. A first prototype has been developed in order to add external tools to an existing educational environment. The results show, on the one hand, that the coding time decreases significantly and on the other hand, that those tools can be improved by taking advantage of educational systems. In particular, with the proposed framework results can be stored into a LOR with an understandable content, enabling to find and reuse them in other activities or by other tools.

Therefore, the strength points of this proposal are i) the possibility of changing the integration data models in an easy way and; ii) that every model is stored in a repository, so it is possible to check or reuse a model. Another important contribution is that there is not necessity for the tool developer to implement an interface to view the result generated.

This approach also separates the content of a learning object from the packaging method. The tool can package all the elements of the final result into a card, which has enough semantic of the kind of content, and then be stored like a SCORM object. If another standard, like IMS CP, is chosen, there is not necessity to change the package. This is an important reason to make a clear differentiation between content and the packaging method.

Future work involves the study of developers and final user satisfaction, when working in a virtual community following eLearning protocols. In order to do that, metrics and questionnaires have to be designed.

6. ACKNOWLEDGMENT

We thank Ministerio de Ciencia e Innovación for the support of Project TIN-2009-14317-C03-03 and Region of Madrid for the support of e-Madrid Network of Excellence S2009 TIC-1650.

7. REFERENCES

[1] Alario-Hoyos, C., Bote-Lorenzo, M. L., Gómez-Sánchez, E., Asensio-Pérez, J. I., Vega-Gorgojo, G., and Ruiz-Calleja, A. 2013. GLUE!: An architecture for the integration of external tools in Virtual Learning Environments. *J. Computers & Education*, 60(1), 122–137

[2] Buyya, R., Yeo, C., Venugopal, S., Broberg, and J. Brandic, I. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *In Journal of Future Generation Computer Systems Elsevier* Volume 25, Issue 6, 599-616

[3] Celorrio, C., and Verdejo, M. F. 2008. An Interoperable, Extensible and Configurable Service Architecture for an Integrated Educational Networking Infrastructure. In *Proceedings of International Conference on Advanced Learning Technologies (ICALT 08)*, 207-211

[4] Conde, M. A., García-Peñalvo, F. J., Piguillem, J., Casany, M. J., and Alier, M. 2012. Interoperability in eLearning Contexts. Interaction between LMS and PLE. In *1st Symposium on Languages, Applications and Technologies (SLATE 2012)*, 205–223

[5] Ghiglione, E., and Dalziel, J. 2006. Design principles for LAMS version 2 and the LAMS "Tools contract". In *Proceedings of the TenCompetence Conference Workshop*, 1–15

[6] Gibson, B. 2007. Enabling an Accessible Web 2.0. IBM Emerging Technologies. *Proceedings of W4A'07*, pp. 1-6

[7] Hoppe, U., Pinkwart,N, Oelinger, M, Zeini, S, Verdejo, F, Barros, B and Mayorga, J.L.2005. Building Bridges within Learning Communities through Ontologies and Thematic Objects. *In CSCL 05, Taipei, Taiwan*

[8] IEEE Std 1484.1-2003 In IEEE Std 1484.1-2003, (2003)

[9] IMS Learning Desing. http://www.imsglobal.org/learningdesign/index.html (Accessed 2013)

[10] IMS LTI (2012) "IMS Learning Tool Interoperability v1.0 Project Group" http://www.imsglobal.org/toolsinteroperability2.cfm (Accessed 2013)

[11] Lorenzo, E. J., and Verdejo, M. F.2011. Modelling the Aggregation of Multimedia Data to Connect the Inputs and Outcomes of a Variety of Tools. In *Proceedings of International Conference on Advanced Learning Technologies (ICALT 11)*, 381-383

[12] Lorenzo, E.J. and Verdejo, M.F. 2010. CARDS: A Metamodel Approach to Aggregate Outcomes of Learning Tools. In *Proceedings of International Conference on Advanced Learning Technologies (ICALT 10)*, 156-157

[13] Mayorga, J. I., Reyes, J. V., Rodriguez-Artacho, M., & Cigarran, J. M. 2012. An architecture for retrieving and organizing web resources for didactic purposes. *Journal of Research and Practice in Information Technology*, 44(2), 183-201

[14] Open Archives Initiative http://www.openarchives.org (Accessed 2013)

[15] Open Knowledge Initiative. http://www.okiproject.org (Accessed 2013)

[16] SPI: The Simple Publishing Interface Specification, CEN Workshop Agreement (CWA 16097). ftp://ftp.cen.eu/cen/Sectors/TCandWorkshops/Workshops/CWA16097.pdf Accessed (2013)

[17] SQI: http://nm.wu-wien.ac.at/e-learning/interoperability/SQI_V1.0beta_2005_04_13.pdf Accessed July (2013)

[18] Verdejo, F., Celorrio, C., Lorenzo, E.J., Millán, M., Prades, S., and Vélez J.,2009. Constructing mobile technology-enabled environments for an integrated learning approach. *Innovative Mobile Learning. IGI GLOBAL*, 145-171

[19] Verdejo, M.F., Celorrio, C., Lorenzo, E.J. and Sastre, T. 2006. An Educational Networking Infrastructure Supporting Ubiquitous Learning for School Students. In *Proceedings of International Conference on Advanced Learning Technologies (ICALT 06)*, 174-178

[20] Wen H., Yi-Xia Z., Yun X. and Xing-Hua S. 2011. A Cloud Service Environment Framework for SCORM Compatible Content. In *Proceedings of PACCS*, 1-4