

This document is published in:

Computers & Education (2011). 57 (1), 1160-1170.
DOI: <http://dx.doi.org/10.1016/j.compedu.2010.12.007>

© 2011 Elsevier Ltd.

Generic service integration in adaptive learning experiences using IMS learning design

Luis de-la-Fuente-Valentín*, Abelardo Pardo, Carlos Delgado Kloos

Department of Telematics Engineering, University Carlos III of Madrid, Av. Universidad, 30, 28911 Leganés (Madrid), Spain

*Corresponding author. Tel.: +34 91 6246234; fax: +34 91 6248749.

E-mail addresses: lfuentes@it.uc3m.es (L. de-la-Fuente-Valentín), abel@it.uc3m.es (A. Pardo), cdk@it.uc3m.es (C.D. Kloos).

Abstract: IMS Learning Design is a specification to capture the orchestration taking place in a learning scenario. This paper presents an extension called Generic Service Integration. This paradigm allows a bidirectional communication between the course engine in charge of the orchestration and conventional Web 2.0 tools. This communication allows the engine to configure external services so as to adjust their behaviour, and also retrieve information about their activity and use it to influence the orchestration process. This extension impacts all the phases of the life cycle of a learning activity: authoring, deployment and enactment. The paper analyses this impact and proposes appropriate adjustments. A pilot test was conducted and the obtained results show that learning experiences that adapt their flow-based on third party tools while maintaining interoperability, reusability and self-containment requirements are feasible.

Keywords: Architectures for educational technology system, Distributed learning environments, Multimedia/hypermedia systems

1. Introduction

Education has undergone significant changes especially in the area of teaching strategies. The widespread use of information and communication technology allows teachers to access a larger variety of resources. Furthermore, students may use Internet as a vast catalogue where information can be searched and used to complete the activities proposed by the teacher. The variety of learning scenarios deriving from this change is breathtaking. As a consequence, students may take now a much more active role in educational experiences, and these experiences can be tailored to their preferences.

Ideally, instructors may prepare hypermedia material such that the resources offered to a student depend on her interests, profile, previous knowledge, past performance, or many other possible factors. This material is called Adaptive Hypermedia (Brusilovsky, 2001). It is important to establish the difference between adaptable and adaptive content (Frias-Martinez, Chen, & Liu, 2009). Adaptable material is meant to enable users to adapt themselves the content layout and navigation support to their preferences. On the contrary, in the case of adaptive material, these adjustments are performed automatically based on observations of the user behaviour. There are different types of adaptive learning schemes depending on the elements to modify (He, Kinshuk, & Patel, 2002): content adaptation, where the same learning activity is presented to all the students but with different material; flow adaptation, where a different set of activities is chosen for each student; and interface adaptation, where the same information is presented to all the students but with different layouts.

There are multiple aspects to consider when adapting the content of a learning experience. For example, Felder and Silverman (1988) defined a set of learning styles as the basis to select different types of resources. In general, adaptation can be performed based on student goals, preferences, background, personal interests, etc. (del Puerto Paule Ruiz, Fernández Díaz, Ortín Soler, & Pérez Pérez, 2008). Adaptation does not always depend exclusively on personal data. Learning experiences can also be changed based on the surrounding environment. For example, Muntean (Muntean, 2008) studied how to adapt learning material based on the Quality of Experience of the used network, and Brown (Brown et al., 2011) studied the adaptation of an experience based on the learners' spatial location.

Another relevant factor in the adaptation of a learning experience is the delivery mode. A first approach is to use a dedicated platform providing a totally self-contained and fully functional adaptive learning environment to the users independently of any other tool. An example of this type of tool is AHA (De Bra & Calvi, 1998). A different approach is to use a pedagogically neutral framework allowing the use

of adaptive learning schemes as well as any other learning strategy. The IMS Learning Design specification (hence IMS LD) (Koper and Tattersall, 2005) falls into this category. IMS LD is a formalism conceived to create, deploy and enact interoperable, reusable and self-contained learning experiences. The specification allows the use of conditional expressions and properties to define multiple sequences of resources, and therefore, it is a suitable formalism to define and deliver adaptive learning material, as claimed by Burgos, Tattersall, and Koper (2007). A learning experience described in IMS LD is then enacted by a so-called engine, a computer program that deploys the appropriate sequence of resources and environments to the participants.

But the properties used in a learning experience described with IMS LD can only be modified through expressions in the resources included in the course. In other words, the engine interpreting the choreography is self-contained, all properties are interpreted and modified within the engine. From this point of view, IMS LD can be defined as closed to external properties, understanding "external properties" as those managed outside the engine.

The main consequence of this limitation is the impossibility for IMS LD courses to obtain information from external sources. That is, a learning experience cannot be defined, for example, to use conventional third party Web 2.0 tools hosted in external platforms and adapt the environment depending on the activities that took place in these platforms. If, for example, an IMS LD learning experience uses an external collaborative virtual whiteboard, its use may be scheduled, but the adaptation cannot take into account any of the (potentially useful) events that occurred in this third party tool.

This paper presents the Generic Service Integration paradigm (henceforth GSI), an extension of the IMS LD specification that allows the inclusion of conventional Web 2.0 services (third party tools) in the context of courses defined with IMS LD, and to bidirectionally exchange information among the course flow engine and the third party tools. GSI defines a method to assign the information gathered in the external tool to property values, so the course flow can be adapted depending on these external information sources. The feasibility of the proposed approach has been evaluated in a pilot experience attended by 22 teachers from primary and secondary education with no previous knowledge of IMS LD. These teachers were told to manage a course containing adaptive material based on the results of a questionnaire hosted in Google Forms (external tool).

The remaining of the document is organized as follows: Section 2 analyses the relationship between IMS LD, adaptive hypermedia and Web 2.0 tools. Section 3 describes the proposed GSI model. In Section 4 the results of the pilot experience are presented and discussed. Finally, Section 5 presents the main conclusions of this work.

2. Adaptive hypermedia orchestration with IMS learning design

The IMS Learning Design specification was released in February 2003 (IMS Global Learning Consortium, Feb 2003) with the aim of providing a framework to express and deploy generic learning flows. The data model included in the specification allows its usage as a pure modelling language. The publication of the XML binding for this data model and the consequent implementation of several run-time engines, such as CopperCore (Vogten, Tattersall, et al., 2006) or GRAIL (Escobedo-del-Cid, de-la-Fuente-Valentín, Gutiérrez, Pardo, & Kloos, 2007) turned IMS LD into a tool to create and deploy learning flows in on-line, blended and face-to-face scenarios. IMS LD is said to be "pedagogically neutral", that is, the vocabulary is not tied to a particular learning theory, but instead allows the definition of a wide range of pedagogical models.

The structure of IMS LD is highly influenced by the so-called theatrical metaphor: an analogy used to make the IMS LD vocabulary understandable for non proficient users. According to this metaphor, a so-called "Unit of Learning" (UoL) is composed by one or more acts. Each act includes a set of activities to be performed by people assuming a set of roles in a previously defined environment. Compared to other flow definition specifications such as IMS Simple Sequencing (IMS Global Learning Consortium, March 2003), Learning Design introduces the concept of role, so that each role in a course has a different relationship with the content and with the learning flow. Therefore, it can be used in the scaffolding of collaborative learning scripts (Valdivia, Nussbaum, & Ochoa, 2009; de-la-Fuente-Valentín, Pardo, Pérez, Dimitriadis, & Kloos, 2008; Hernandez-Leo et al., 2007).

The creation of adaptive learning material is supported by IMS LD with the inclusion of properties and conditions. The former are used to store certain values and define the state of the course. The latter allow performing different actions depending on a set of expressions evaluated over the properties.

IMS LD is considered the de-facto standard (Martinez-Ortiz, Sierra, & Fernandez-Manjon, 2009) by learning designers. However, this consideration did not push yet the specification to a widespread adoption. Neumann, Oberhuemer, and Derntl (2010), (Durand and Downes, 2009), argue that creating a Unit of Learning is too complex and there is a lack of easy-to-use authoring tools. Several initiatives have been started with the objective of filling this gap.

Hernandez et al. defines collaborative learning flow patterns and maps them into IMS LD terminology (Hernández-Leo et al., 2006). The pattern based approach is also considered and analysed by Rodriguez, Rifon, and Nistal (2004). In a similar way, (García-Robles, Díaz-del-Río, Vicente-Díaz, & Linares-Barranco, 2009) focus the authoring efforts on the support of a restricted scenario, creating an ad-hoc tool that allow the creation of concrete Units of Learning. Initiatives presented in Karampiperis & Sampson (2007), Neumann et al. (2009) and Martinez-Ortiz et al. (2009) take a more graphical – and therefore human understandable – approach on the design of Units of Learning.

Current research is also focused on finding the limits of the specification. Case studies found in the literature show the strengths and weaknesses of IMS LD: Valdivia et al. (2009) implement collaborative flow patterns concluding that the provided vocabulary is expressive enough for their purpose, but claiming that collaborative services are required to completely support the scenario. de-la-Fuente-Valentín et al. (2008) also focus on supporting the design and deployment of Computer Supported Collaborative Learning on distant scenarios. A different approach is taken by Zeng and Zhou (2007), which analyses how game-based learning can be implemented with IMS LD, showing how Learning Design elements have a correspondence with game design ones.

Towle and Halm (2005) point out that, thanks to properties and conditions, IMS LD is suitable to be used in the context of adaptive hypermedia (AH) orchestration. According to He et al. (2002), adaptive learning material can be categorized in the following types, which are argued to be supported by IMS LD in Burgos et al. (2007):

- Interface-based: IMS LD defines how a course must behave during its enactment, but says nothing about how compliant players must present the information. Therefore, interface modifications are not supported by the adaptation inside a UoL.

- Learning flow-based: The visibility (and therefore availability) or the finalization of an activity can be directly defined by conditional actions.
- Content-based: The shown-or-hidden attribute of IMS LD is applied to all activities, but it can be used with a finer level of granularity. The visibility of any portion of an XHTML document can be controlled by assigning a class attribute and controlling their visibility with a property.

Any other type of adaptation can be generalized as one of the three types described above. Consequently, any adaptation that includes the modification of flow and content is also supported by IMS LD.

Adaptive Hypermedia development, as described in del Puerto Paule Ruiz et al. (2008), is a process composed by the following steps: (1) Select a good taxonomy of learning styles, (2) develop techniques to introduce the adaptation into the system, (3) design the adaptation and (4) design the content. Adaptation by means of IMS LD provides step 2, and there exist several tools to facilitate step 3. Therefore, it can be concluded that adaptation is not only supported, but also enhanced by IMS LD.

An adaptive Unit of Learning described with IMS LD must be self-contained, meaning that all resources involved in the experience must be included in the UoL package (which ultimately is a zip file). Although this requirement improves reusability, it imposes a severe restriction when considering the use of third party tools as part of the learning flow. In fact, the current version of IMS LD only supports a simplified mechanism to include external tools in a course. A UoL may contain a URL of a tool external to the engine. However, the orchestration and adaptation contained in the UoL is totally independent of the events occurring in that external resource.

As stated in Chatti, Jarke, and Frosch-Wilke (2007), Web 2.0 techniques have changed the way people access and manage information in Internet. Learning is also affected by this new paradigm. E-learning courses may now interact with different available tools. Joining learning environments and new trends in social software is explored in Ebner, Lienhardt, Rohs, and Meyer (2010), where micro-blogging is used for sharing thoughts among learners. Cole (2009) show the benefits of using wikis to create documents collaboratively. Andreas, Tsiatsos, Terzidou, and Pomportsis (2010) investigate on how Second Life suits the requirements of collaborative learning patterns. From these examples, it seems clear that instructional design techniques should be compatible with this trend where resources and services are distributed and loosely coupled, and the orchestrating engine is the element that organizes all the different parts of the course.

On the way to support the retrieval of external property values to be used as IMS LD properties, the IMS consortium has defined a few guidelines but they are restricted to achieve interoperability between IMS QTI and IMS LD (Vogten, Martens, et al., 2006). Taking advantage of the same guidelines, Moreno-Ger, Burgos, Martínez-Ortiz, Sierra, and Fernández-Manjón (2008) propose a framework where serious games are combined with IMS LD for the enrichment of Units of Learning. A game is played in the context of a UoL and certain values are stored as properties. These values allow the UoL to react based on game results and adapt course material.

But the common aspect of these approaches is that they force the inclusion of the resources (game, QTI, etc.) as part of the resources in the packaged UoL. Web 2.0 tools, on the other hand, do not fit in this paradigm. They are provided by external vendors, and it is not possible to "include" them as resources in the UoL. A new approach is needed to handle this increasingly common type of resources. Services such as the conference or the send-mail facility are already supported in IMS Learning Design, but with a behaviour that depends on the run-time environment functionality. The list of supported services is not sufficient to cover the increasing number of web tools that are suitable to be included as part of a learning flow.

In order to preserve the spirit of service integration into UoL, the proposed extensions should fulfill the following requirements

- Self-containment: All the information required to deploy and enact a learning flow must be included in the Unit of Learning. This does not require for the service functionality to be in the zip file, but rather the instructions to find and instantiate the service.
- Adaptability: One of the strengths of IMS LD is the possibility to adapt the learning flow depending on the state of the course. This feature must be extended to all parts of the course, including third party services. Furthermore, external services should be able to react to changes in the course, so the learning flow must be able to operate external tools as a response of changes in the course.
- Reusability: It should be possible to replicate a learning experience in more than one instance. The concept of reusability encourages an agile re-instantiation process at the same time that guarantees the same initial conditions for all course replicas. Thus, the functionality provided by third party web tools needs to be instantiated at the time that the UoL is also instantiated.
- Interoperability: An IMS LD course must be able to be instantiated on any compliant platform. Any definition of service integration must follow the same principle, avoiding proprietary-based protocols that restrict interoperability.

In the next section the Generic Service Integration framework is described. GSI allows the inclusion of certain web tools as part of an adaptive learning flow while preserving the previously described requirements.

3. Integration of third party services in the course flow

As stated previously, the adaptive capabilities of IMS Learning Design are limited by its inability to obtain information from distributed sources, as in the case of Web 2.0 tools. This section is devoted to present Generic Service Integration, a specification that complements IMS LD so that it makes possible to integrate third party services as part of a scripted learning flow. The inclusion of GSI in a course conducted by IMS LD does not result in the loss of any of the properties of self-containment, adaptability, reusability and interoperability.

There are three requirements to be satisfied to integrate a third party service using GSI:

1. The UoL must contain a description of the third party service as part of the IMS manifest. The vocabulary to make the description is defined by GSI.
2. The run-time environment must be compliant with both the IMS LD and GSI models. Since the latter is an extension of the former, GSI compliance implies IMS LD compliance.
3. A GSI module specific for the requested external tool has been installed in the run-time environment. This module connects the engine enacting the UoL with a service that matches the functionality described in the UoL.

The inclusion of a service in a course follows the same philosophy than the course itself. First, the expected behaviour of the service is described in the authoring phase and packaged in the Unit of Learning. Secondly, this description is used by the run-time environment to allocate the required resources during the deployment phase. Finally, the course participants interact with material as described in the activity descriptions. The life-cycle of a service is then divided into three phases: authoring, deployment and enactment (see Fig. 1).

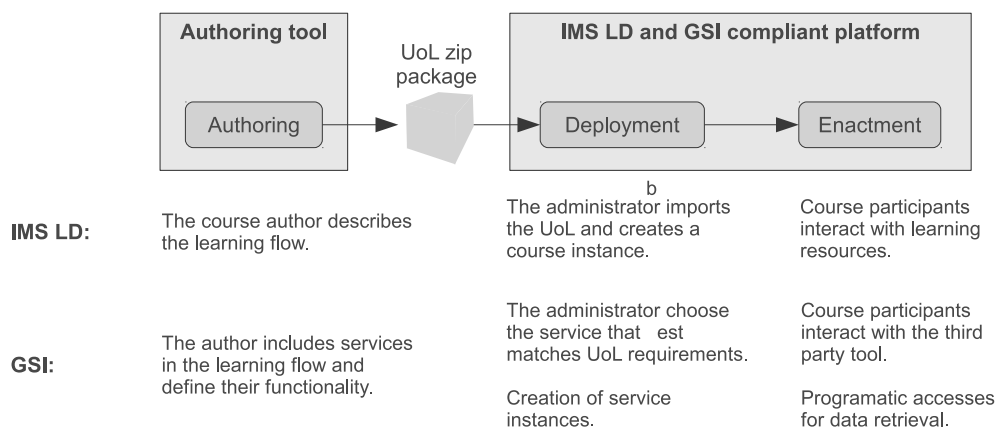


Fig. 1. Life-cycle for courses that integrate GSI services.

3.1. Course life-cycle

When course authors include a Web 2.0 service as part of the learning flow, they have in mind one specific tool from a certain provider. The problem is that such service cannot be guaranteed to be available at run-time and, if not available, the proposed learning method may not achieve the expected outcomes.

Let us consider a course hosted in a conventional LMS using the existing assessment functionality. In most cases, this binding to the platform and the offered functionality the course is neither interoperable nor reusable. Solutions such as import/export functionality or packaging the course in a format such as SCORM are not optimal. In contrast with this scenario, for the remaining of this section we consider a course similar to the one described but using the assessment functionality provided by Google Forms (<http://docs.google.com/support/bin/answer.py?answer=87809>) (a generic external service integrated in the course).

In GSI, a service is described by its expected functionality and behaviour. In practice, the author lists all the capabilities the service must satisfy and a set of metadata that complements this description. To do so, GSI provides a generic vocabulary to describe the communication between the course flow manager and the external service and integrate this communication with course events. In later phases of the course life-cycle, this description is bound to an actual tool, which is selected depending on the deployment scenario. Following the previous example, the course author would express the expected functionality as “tool to deliver questionnaires and manage answers”, instead of “Google Forms questionnaires tool”.

A GSI compliant deployment platform must have a catalogue of available services, which is a list of the tools whose usage is supported in the run-time environment. When a Unit of Learning is uploaded, its service descriptions are compared to the features of the available services. The tool that best matches the requirements is chosen to be included as part of the course. The election can be manual (done by tutors), or automated by processing the metadata defining the service. In the example considered in this section, the platform informs, during the instantiation process, that an assessment service is required, and offers a catalogue of the supported services that match this requirement. After examining the alternatives, the administrator would then choose the Google Forms module.

To increase reusability, the course participants will be granted access to a “clean instance” of the service specifically created for the course. Thus, the responses gathered from each instantiated course are stored in different spreadsheets in Google Forms so that it is empty at the time the course starts, regardless of the previous editions of the same course. The meaning of a clean instance depends on each service’s functionality. When a new instance of a course is created, the IMS LD engine requests one or more service instances, which will then be accessed during the enactment.

A GSI service is included in the course as part of the environment element described in the specification. Therefore, the tool is not an activity by itself, but a mean to perform the actions specified in the activity description. The service is shown to the user as an additional link placed in the corresponding environment. During the enactment phase, when course participants perform their activities, the IMS LD engine interacts with the external service in two different ways as shown in Fig. 2.

The first type of interaction is conducted programmatically. IMS Learning Design defines when conditions must be evaluated: at the beginning of a course, when a property value has changed, or when an activity is finished. We refer to these moments as events. If the evaluation of a condition is true the course reacts by changing the value of a property, changing the visibility of any activity, or by completing an activity. The GSI model extends this interaction as follows:

- The course may react to an event by sending a command to the external service. Thus, the configuration of a service can change depending on course events.
- IMS LD properties can take their value from the external service, instead of local information. The service becomes a new source of course events.

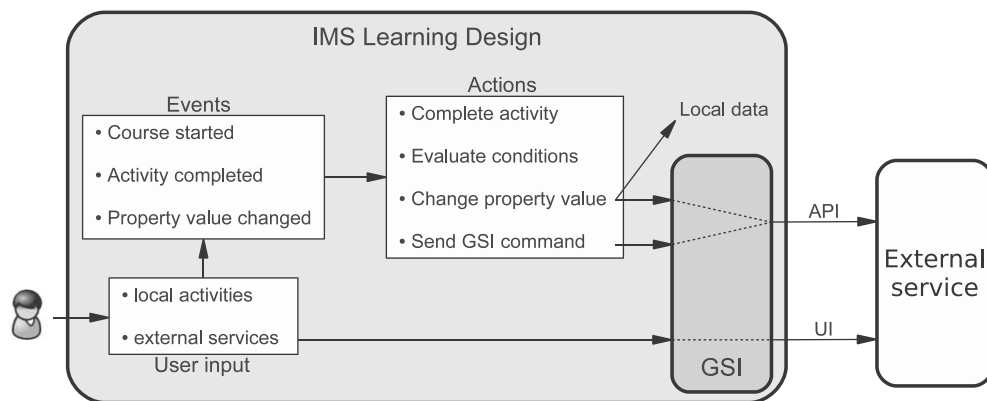


Fig. 2. Interactions between IMS LD engine and external service.

The second type of interaction is directly derived from the user activity. The course material contains a set of hyperlinks to local and external resources. The course participants use these links to access the service and perform the requested activities. Their actions are supervised such that they can be processed and values assigned to properties. Additionally, the regular interaction of students with local resources also produces events that are programmatically processed by the engine.

When the GSI compliant platform interacts with the third party tool, the data exchange that takes place has certain considerations with respect to the user privacy. An in-depth discussion of this aspect is beyond the scope of this paper. A detailed discussion can be found in de-la-Fuente-Valentín, Leony, Pardo, and Kloos (in press).

3.2. Generic service integration architecture

Generic Service Integration was conceived as an extension of the IMS Learning Design functionality. Therefore, the GSI layer is built on top of an IMS LD compliant platform as depicted in Fig. 3. The proposed model does not impose any technical restriction over the implementation of the underlying engine. That means that, in practice, GSI can be build over any existing IMS LD compliant platform.

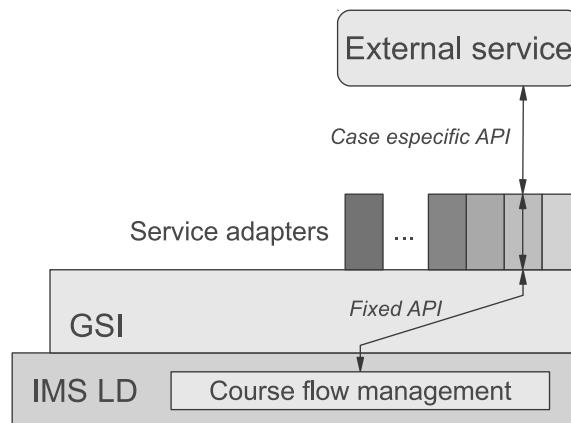


Fig. 3. Architecture of the Generic Service Integration framework.

The GSI layer mediates the communication between the IMS LD engine and the third party service. From the point of view of the IMS LD engine, GSI offers a fixed API that allows the request of external actions and values. Part of these methods are used during the deployment phase, while others are used during the enactment. For example, during course instantiation, the engine needs the URL to access the service, so the request-configured-instance method is called. This method returns a collection of URLs to be inserted in the corresponding environments. Analogously, when the engine needs to obtain a value from the external service, the get-external-value method is called. Table 1 shows the methods used in GSI and the different phases in which they are invoked.

Regardless of the configured third party tool, the communication between the IMS LD engine and the GSI layer is determined by the defined API. However, the procedure triggered in the GSI layer depends on the third party tool in use; that is, depends on the configured service. In other words, the GSI layer offers a unified API, whereas the specific adaptation to each service is done at the level of the service adapter. For example, data from a Google Form can be access through a public API that uses http requests for data retrieval, so the corresponding GSI module uses this API to access the information and translate it into IMS LD properties. Different services offer different access methods. Instead of requiring an agreement on a common protocol to access information, the service adapters in GSI are designed to interact with a particular service.

During the course enactment, GSI acts as a layer that mediates the communication between the IMS LD run-time environments and the external services. The actual behaviour of that layer depends on the service chosen in the deployment phase. That is, the layer has to be

Table 1
Methods to request actions and information from the GSI layer.

Deployment		Enactment
UoL import	UoL instantiation	
get-documentation	initialize-user	perform-startup-actions
get-zero-act-roles	request-configured-instance	zero-act-scripts
get-zero-act-urls	get-zero-act-visibility	get-external-value
	action-list-execute	action-list-execute

configured so that it uses the functionality that corresponds to the desired service. When the learning flow needs to interact with the external service, the proper command is send to GSI, which is in turn redirected to the corresponding functionality module.

A GSI compliant run-time environment has a collection of adapters, each one allowing the communication with a different service. The support for a new service implies the implementation of the corresponding adapter. When a course is loaded, the available service with the best match in requirements is chosen to configure the GSI layer. Hereby, the configured adapter dictates the behaviour of the course regarding services interaction. For the sake of simplicity, the vocabulary provided by GSI is limited to a small set of generic terms. Each of these terms can be interpreted in several ways, so that each adapter makes its own implementation, which depends on the supported service. This limited set of terms facilitates the implementation process while not imposing restrictions to any available service.

3.3. Discussion of the model

The development of the GSI model has been affected by several design decisions. When possible, the simplicity has been the concept that led this design. The most relevant considerations are discussed here.

The vocabulary used to describe the services in the UoL could be either generically defined, or be tool specific. The second option allows making finer grain definitions of the expected behaviour of a given service, but at the cost of handling a more complex set of terms. This complexity hinders the implementation of both new service adapters and GSI authoring tools. The first option, a generic vocabulary, allows the inclusion of existing and future services at the expense of a more abstract syntax. The generic vocabulary was chosen because of the benefit gained in the implementation process, which is a key factor in the support of new type of services. The same reason was considered when designing the actions introduced by GSI. The required functionality could be implemented by using the current IMS LD actions. For example, a possible reaction to changes in the external service could be to finish one activity. But such action is already considered in IMS LD, thus the GSI event is the modification of a property value. The completion of an activity can then be implemented by relating the already existing complete-activity element with the externally modified property using the syntax provided by IMS LD.

Course properties can be modified by information obtained from third party services but, how long does the course engine take to react to these external changes? According to the GSI design principles, a service is assumed to behave passively. That is, the IMS LD run-time environment is in charge of starting the communication to detect any changes. As a consequence, an event needs to occur within the course in order for the process of requesting data from the service to be executed.

The ultimate goal of the GSI model is to integrate external services in the context of a UoL while preserving the principles of IMS Learning Design. It follows a discussion of how are these requirements satisfied.

A course is *self-contained* if the UoL package includes all resources and data required to successfully perform the enactment phase while preserving the defined pedagogical model. Instead of including a service inside a distributable package, which is unrealistic, GSI includes a service description and delegates to the run-time environment the task of finding an appropriate tool. Considering this delegation, the information included in the original Unit of Learning is enough to satisfy the self-containment requisite.

The straightforward consequence of self-containment is *reusability*, which is the capacity of a course to be used in different scenarios with no modifications of the original package. On one hand, the tool that will be used in the course is chosen in the actual scenario; this fact protects from unavailability of certain tools and ensures that the course can be played. On the other hand, course instantiation steps include the creation of a clear instance of the external service, allowing all students to start the learning flow under the same circumstances.

Units of Learning allow the *adaptation* of the course based on students profile and behaviour. Conditions imposed on property values are used to model the adaptation, and they are evaluated when an IMS LD event occurs. The same events and conditions are used in GSI as triggers to request remote command execution, so that the service can behave depending on the course rules. Additionally, property values can be mapped to values on the external service, allowing the engine to react using the service as a data source.

Based on the same design principles, GSI is *interoperable* in the same way that IMS LD is: the vocabulary used during authoring defines a behaviour that is the same in any compliant run-time environment. The information model is not tied to any particular technology, being GSI the only underlying requirement.

4. A case of study to evaluate the approach

The model proposed in this document has been validated by a case study. In the pilot experience, instructors without prior knowledge of the IMS LD specification were presented with a Unit of Learning with adaptation based on third party services embedded in the flow. The experience used GRAIL (Escobedo-del-Cid et al., 2007), the IMS LD engine integrated in the .LRN (<http://www.dotlrn.org>) Learning Management System, as supporting software. This engine includes a module implementing the exchange of information between the engine and generic external services as described in the GSI architecture (Frias-Martinez et al., 2009).

This section explains the details of the enactment of the case study: first, a description of the audience and performed activities is given; then the methods to evaluate the experience are presented; finally, the conclusions obtained from the performed qualitative and quantitative analysis.

4.1. Description of the experience

4.1.1. Demographic data

The experience was deployed in the context of a series of science oriented workshops, where the presented one was advertised as Computer support for the creation of adaptive content. Considering the title and the topic of the workshop, the expected audience was teachers from K-12 or high schools interested on new methods of teaching and, more specifically, on using computers to support these methods.

Participants were volunteers that had to register days before workshop was held. A total of 22 persons registered for the experience: 17 women and 5 men. Among them, there were 9 teachers from K-12, 3 from high school and 6 were counselling psychologists. 38.9% of them claimed to have advanced computer skills, while the rest had basic computer skills. Their computer skills averaged 3.38 on a scale of 1–5. Each participant was provided with a laptop to work individually in the activities. The participant profiles are summarized in Table 2.

4.1.2. Workshop activities

The activities were carried out in a single 4-h session. The objective for participants was to understand a previously-created UoL, and create a new one based on a similar template. The team of tutors guided the participants through the activities in the workshop. That is, there was neither written description of the activities nor an automated orchestration system. The UoL under study contained a flow with three acts where the second one was adapted depending on a questionnaire obtained in the first act. The structure was quite similar to the What is Greatness UoL (Dalziel, 2003), but obtaining the data from a Google Spreadsheet-based form, using GSI (de-la-Fuente-Valentín et al., 2009). Each participant was given an independent user space in the platform. UoLs were already deployed and instantiated so the first required step was to subscribe the students to the course.

The workshop activities were divided as follows: first, researchers' team introduced the IMS LD framework to the audience, focusing on the adaptive capabilities of the specification. This explanation was followed by a demonstration of the example UoL, showing how the adaptation was implemented in the working case. Then, participants replicated the same sequence of activities in their own computers, assuming the corresponding roles of the UoL. They changed from the teacher role to the student one when the flow suggested so, both roles were being played in parallel. At this point, the participants interacted freely with the learning flow, asking for help when required.

In the second part of the workshop, the participants used GRAIL to create a new version of the adaptive flow: they started from a UoL where the flow was identical to the one in the previous activity, but with no content. They used the editing functionality offered by GRAIL to add contents to the course. The objective of this second stage was to consolidate the acquired knowledge of the specification, seeing if the provided example would fit in their personal working environment and, if not, what new features would be required.

Before finishing, participants were invited to take part in a group discussion moderated by the researchers. The goal was for participants to reflect on the possibilities of adaptive schemes and the integration of third party services in the course flow. This final discussion also provided feedback about the contents of the workshop and its methodology.

4.2. Evaluation methodology

4.2.1. Research questions

The workshop was intended to study two different aspects of the proposed architecture: from the technical point of view, the experience helped to evaluate if the model really satisfies the requirements discussed in Section 3.3. In particular, the research questions about the architecture characteristics were:

- RQ1) Can UoLs using GSI be reused in completely independent course instances?
- RQ2) Is it feasible to deploy such an scenario at a reasonable cost?

The workshop also evaluated the acceptance of the model by the participants. In order to keep improving the proposed architecture and the provided features, it is important to determine how teachers understand the model, its perceived usefulness and the most valued features. The specific research questions in this area were:

- RQ3) How difficult it is for teachers to understand the “script-based adaptation” paradigm?
- RQ4) How teachers ponder the feasibility of IMS LD based adaptation?
- RQ5) Does spreadsheet-based question management lower the adoption threshold of IMS LD based adaptation?
- RQ6) Are Web 2.0 tools currently considered as teaching tools by instructors?

Table 2
Summary of participant profiles.

Variable	Number	%
Men	5	22.73
Women	17	77.27
Primary education teachers	9	40.91
Secondary education teachers	3	13.64
Counselling Psychologists	6	27.27
Unknown occupation	4	18.18
Average computer skills	3.38 (out of 5)	
Standard deviation in computer skills	0.83	

Table 3
Summary of the questionnaires used in the workshop.

Id.	Question	Comment
<i>Q1: before the workshop</i>		
Q1.a	Computer skill of the participants	1 = Low-Skilled, 5 = High-Skilled
Q1.b	Occupation of the participant	Open question
<i>Q2: before the final activity</i>		
Q2.a	Difficulty/benefit ratio of adaptation	1 = Too difficult, 5 = High benefit
Q2.b	Comprehensibility of the example	1 = Uncomprehensible, 5 = Very comprehensible
Q2.c	Difficulty/benefit ratio of the example	1 = Too difficult, 5 = High benefit
Q2.d	Comments about the adaptive script	Open question
Q2.e	Realistic in a real scenario?	1 = Unrealistic, 5 = Very realistic
Q2.f	Useful in a real scenario?	1 = Unuseful, 5 = Very useful
<i>Q3: after the workshop</i>		
Q3.a	Difficulty/benefit ratio of the example	1 = Too difficult, 5 = High benefit
Q3.b	Feasible to be deployed in your working scenario?	1 = Not feasible, 5 = Very feasible
Q3.c	Is the access to the spreadsheet easy to use?	1 = Not easy, 5 = Very easy
Q3.d	Does the spreadsheet provide any added value?	1 = No added value, 5 = High added value
Q3.e	What is the spreadsheet useful for?	Open question
Q3.f	Web 2.0 tools to include in an adaptive course	Open question

4.2.2. Data sources

The case of study offered several data sources: the surveys filled by participants, a final group discussion with the observations of the researchers and the instantiation process itself.

Participants received a total of three questionnaires, referred as Q1, Q2 and Q3. First, they were contacted by e-mail prior to the workshop and asked to fill a questionnaire to determine the participants' profile. The second questionnaire was presented before the break between activities, and was aimed at capturing their first impression of the adaptive learning scheme. Finally, in Q3 they expressed their conclusions about the feasibility of the learning method to be deployed in real scenarios. The questionnaires had a mixture of quantitative and qualitative questions and their analysis followed the principles of a mixed evaluation method (Martínez, Dimitriadis, Rubia, Gómez, & De La Fuente, 2003). A summary of the most relevant questions presented to the participants is given in Table 3.

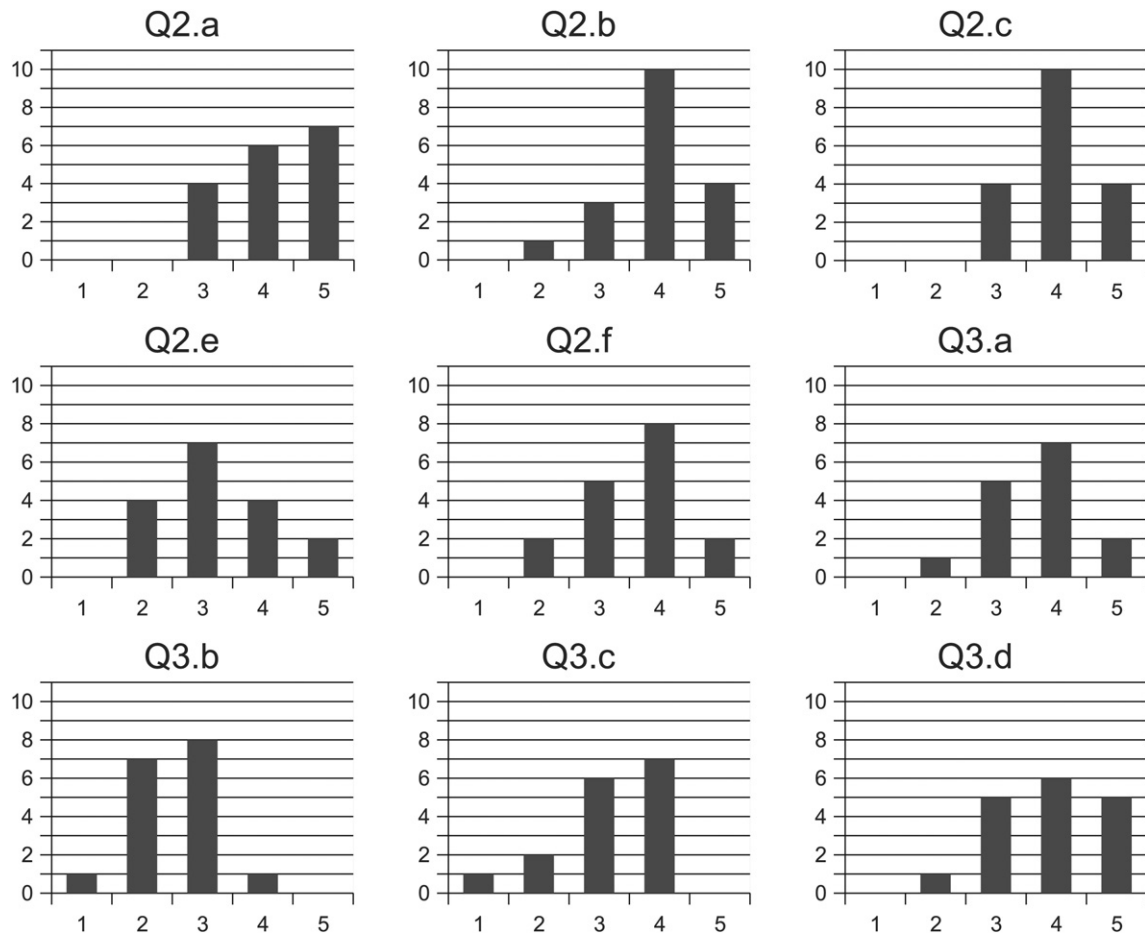


Fig. 4. Histograms of the obtained answers. Y axis represents the number of responses, X axis is the option number.

Table 4
RQ3 relative questions.

Question	Mean	Standard deviation
Q2.a	4.18	0.81
Q2.b	3.94	0.8
Q2.c	4	0.69
Q3.a	3.67	0.82

When course activities had finished, participants were invited to discuss or comment what they found interesting in the workshop. A group of five participants took part in this discussion, which offered relevant reflections to better understand the participants' view of the used tools and methods. Researchers methodologically guide the discussion between an informal conversation interview and guided interview (Creswell, 2002). That is, there was an informal plan that guided the conversation and, depending on how talkative the participants were, the interview reminded more one of the other methods.

Finally, the issues of the instantiation and enactment process itself offered the researchers data to determine the technical feasibility of the model.

4.2.3. Analysis methodology

Considering that the research questions given in Section 4.2.1 can be divided in two groups (technical feasibility and participants' view of the model), they were analysed using different datasets and following different methodologies. First, technical feasibility was qualitatively evaluated by considering the observations of the instantiation process and the technical issues occurred during the workshop. Second, the analysis of the participants' view of the model followed the principles of a mixed evaluation method so that qualitative and quantitative techniques were in use. This approach allows expanding an understanding from one method to another, at the time that it allows triangulating data so that the evidences shown by one method can be confirmed or refuted by the other (Creswell, 2002).

The questionnaires included numeric questions with a Likert-scaled response, used for qualitative analysis; and open-ended ones where participants freely expressed their opinions for qualitative analysis. The qualitative analysis also considered the final group discussion as a valid source of data.

The goal of the case study is to see how teachers understand the IMS LD courses, its adaptive capabilities and the connection of the framework with Web 2.0 systems. The aim is to reveal an overview of the GSI model's feasibility in current education, i.e., map the phenomenon. This is why qualitative analysis was used. In such type of analysis, the size of the sample is not an issue because the factors and/or opinions tend to repeat after a low number of questionnaires (Johnson & Christensen, 2007). It goes without saying that there is no guarantee of identifying all factors that affect the phenomenon, but the most relevant ones are usually identified with such analysis. Since a case study is influenced by environmental factors and cultural issues, there is no room for the generalization of the results (Stake, 1995). However, this is not an issue for the sake of this study because generalizing to the whole population of teachers is not a goal of the research. It is worth to mention that the statistical analysis described later suggests that some of the results could have been obtained also in larger samples. The worthwhile assertion here is that there is a chance of discussion, rather than asserting the generalization itself.

There is an intrinsic subjectivity on qualitative analysis. It could be argued that the results of the presented study are biased by different factors of the workshop, and that researchers have emphasized some factors more than others. To promote objective results as much as possible, the quantitative analysis is used to reinforce the findings of the qualitative one. The numeric questions included in the questionnaire allow extracting more objective conclusions than the text-based ones. In the context of this study, they were included to confirm the findings of qualitative analysis. To ensure that the mean and the standard deviation of a sample has not been randomly produced, statistical analysis allows detecting those situations in which the results are significative or not, even when the size of the sample is small (Rees, 2000). In the presented study, quantitative data is represented by their mean and standard deviation. In those cases where two paired samples are compared, the *T*-test was used to examine the significance of the results.

4.3. Results

The instantiation process and the workshop itself serve as response for research questions RQ1 and RQ2, formulated in Section 4.2.1. First, each participant interacted with a different course instance. All the proposed activities were successfully completed by the participants, which means that they could work with their individual course instance. Since there were no reported problems specifically related to the interaction among IMS LD and Google Forms, it can be said that GSI-shaped UoLs are as reusable as IMS LD courses are (RQ1). The deployment of the UoLs was performed by the researchers prior to the workshop, so the first participants' activity was to instantiate the courses. The process was repeated once per each participant, so the steps given in Section 3.1 were repeated a total of 25 times. This task revealed to be highly repetitive and therefore easy to automate. The creation of an automation script reduced the time cost of administrative tasks so they remained into reasonable values (RQ2). Considering the deployment and enactment process it can be said that GSI-shaped UoLs maintain the self-containment and reusability of IMS LD.

Table 5
RQ4 relative questions.

Question	Mean	Standard deviation
Q2.a	4.18	0.81
Q2.c	4	0.69
Q2.e	3.24	0.97
Q2.f	3.59	0.87
Q3.a	3.67	0.82
Q3.b	2.53	0.72

Table 6
RQ5 relative questions.

Question	Mean	Standard deviation
Q3.c	3.19	0.83
Q3.d	3.88	0.86

The answers gathered from the participants were used to analyse the response to the rest of the research questions. We now analyse how participants perceived the usefulness of the architecture proposed in Section 3, when applied to adaptive course material. The analysis is based on the 18 valid collected answers. A graphical representation of the obtained answers is shown in Fig. 4, where the corresponding questions are presented in Table 3.

Research question RQ3 regards how well the participants understood the script-based adaptation paradigm. The straightforward facts that support the comprehensibility of the approach are, first, that all participants successfully developed the proposed activities and, second, that they expressed a positive opinion of the workshop. Examples of this positive opinion in the questionnaires are: “reasonable difficulty, and interesting”, or “it seems easy to use and with a fast deployment for teachers with no experience with computers”. The participants agreed on the argument that they had preferred to have more time to complete workshop activities, but the questionnaires do not contain negative comments on the comprehensibility. The perception of the researchers is that the model was quite well understood by participants. This perception is also supported by quantitative data, summarized in Table 4.

The value obtained in question Q2.b shows that the example, the UoL with GSI, was quite comprehensible (3.98 out of 5). Considering the values obtained in questions Q2.a and Q2.c, it can be said that course participants perceived the potential of adaptation and use of third party tools, while they felt optimistic about the difficulty of the implementation process. This perception on the difficulty of the creation process evolved after performing the final activity of the workshop: The mean value of the answer to Q3.a shows that the participants realized the difficulty in the creation of adaptive material. The *t*-test for the comparison of Q2.b and Q3.a samples ($H_0 : \mu_{Q2.c} = \mu_{Q3.a}$; $H_1 : \mu_{Q2.c} > \mu_{Q3.a}$) reveals that there is statistical significance of the result (*p*-value = 0.03408).

Half of the participants explicitly mentioned that the implementation of the UoL in an authentic situation would be beneficial for students' learning and that it would increase their motivation, at the same time that they recognize that teacher training would be mandatory to success in such implementation. The perceived feasibility of such type of learning material in a real situation is the matter of research question RQ4. Despite the positive view of the enacted example, they expressed their lack of confidence on its effective adoption. This view is supported by some remarks such as “It's nice, but it appears to be too abstract, and far from its application on specific material in the context of K-12 Education”, or “I think it is very interesting, but maybe the difficulty appear while creating the material”. Data shown in Table 5 reinforces this view: they considered the example to be fairly useful for a real scenario (Q2.f, 3.59 out of 5), but not feasible to be effectively deployed (Q3.b, 2.53 out of 5).

The previous observations confirm the results of other studies in the field of IMS LD: the most significant obstacle for the adoption of the specification resides in the complexity of course authoring, rather than other problems like the understanding of the specification or the required training process of instructors (Neumann et al., 2010).

While performing the workshop activities, participants' comments and questions were oriented towards understanding the process and the conditions imposed to the adaptation. The researchers' perception was that the participants did not make a clear distinction among the course flow manager and the spreadsheet tool. That is, they did not perceive the spreadsheet as something external. Regarding the complexity of the authoring phase, rules for questionnaire data management are easier to be edited with a specialized tool like a spreadsheet, rather than using IMS LD vocabulary. This is the conclusion extracted from Table 6, which provides an answer to research question RQ5. With no previous knowledge of the specification and no experience using IMS LD conditions, course participants had a positive opinion about the use of the spreadsheet. For example, “It's easy to analyse the received data” show the relevance of a good interface; “Store registers so the teacher can view students' progress, and maybe obtain a graphical representation” emphasizes the potential on the use of specialized tools. This positive perception of conditions management contrasted with other experiences on the use of IMS LD (Derntl, Neumann, Griffiths, & Oberhuemer, 2010), where conditions authoring were considered error prone and difficult to understand.

To evaluate research question RQ6, the number and variety of tools proposed in answers to Q3.f were considered. If it is taken into account that the participants were volunteers interested in supporting teaching methods with computers, the researchers found surprising that only two of the participants proposed tools to incorporate in learning material: wiki, e-mail and blogs were mentioned. The prevailing opinion can be summarized with “Any tool is suitable for me, because I have so poor formation in the field that any tool is a novelty”. Despite that educators perceived the potential of Web 2.0 tools in their learning context, they do not currently consider these tools to be included as part of their courses and the lack of training is perceived as one of the major determining factors.

5. Conclusions

This paper presents the Generic Service Integration model to allow a learning experience to interact with external services. A pilot experience was conducted to study the soundness of the approach by integrating the use of a questionnaire provided through Google Forms.

The GSI architecture allows the integration of third party services in the context of courses described with IMS LD. According to GSI, the course definition includes the description of one or more services. This description is a collection of the desired features used in the learning activities. In the deployment stage, a GSI compliant platform detects the requested features and searches for a supported service that fulfills the requirements. If successful, the course participants are allowed to interact with a third party tool during the enactment of the learning flow. The information exchange between the IMS LD engine and the external service is bidirectional: the course can be adapted depending on information from the third party service, which itself can be configured through commands sent from the IMS LD compliant platform.

The practical experience consisted of a workshop with 22 participants with teaching background and basic computer skills. There were two major aspects to evaluate: the feasibility of the deployment and the understanding of the model. The results show that GSI does not impose any loose in the course replication process, and Units of Learning keep their adaptability, self-containment, interoperability and

reusability. Data gathered from the audience reveals that the main obstacle for IMS LD adoption resides in the authoring process, while teachers acknowledged the potential Web 2.0 tools integration. According to participants' opinion, the inclusion of spreadsheets to grade student responses helps on data analysis and to view student's progress.

Acknowledgements

Work partially funded by the Learn3 project, "Plan Nacional de I+D+I" TIN2008-05163/TSI, the Best Practice Network ICOPER (Grant No. ECP-2007-EDU-417007), the Flexo Project "Plan Nacional de Investigacin Cientfica, Desarrollo e Innovacin Tecnolgica" (Ref. TSI-020301-2008-19), and the "Emadrid: Investigacin y desarrollo de tecnologas para el e-learning en la Comunidad de Madrid" project (S2009/TIC-1650).

References

- Andreas, K., Tsiatsos, T., Terzidou, T., & Pomportsis, A. (2010). Fostering collaborative learning in second life: metaphors and affordances. *Computers & Education*, 55(2), 603–615.
- Brown, D. J., McHugh, D., Standen, P., Evett, L., Shopland, N., & Battersby, S. (2011). Designing location-based learning experiences for people with intellectual disabilities and additional sensory impairments. *Computers & Education*, 56(1), 11–20. (serious Games).
- Brusilovsky, P. (Mar. 2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1), 87–110.
- Burgos, D., Tattersall, C., & Koper, R. (2007). How to represent adaptation in e-Learning with IMS – learning design. *Interactive Learning Environments*, 15(2), 161.
- Chatti, M. A., Jarke, M., & Frosch-Wilke, D. (2007). The future of e-Learning: a shift to knowledge networking and social software. *International Journal of Knowledge and Learning*, 3(4/5), 404–420.
- Cole, M. (2009). Using wiki technology to support student engagement: lessons from the trenches. *Computers & Education*, 52(1), 141–146.
- Creswell, J. W. (2002). *Research design: Qualitative, quantitative, and mixed methods approaches*. Edition (2nd ed.). Sage Publications, Inc.
- Dalziel, J. (2003). Implementing learning design: the learning activity management system (LAMS). In *Interact, integrate, impact: Proceedings of the 20th annual conference of the Australasian society for computers in learning in tertiary education* (pp. 7–10). Adelaide: Citeseer.
- De Bra, P., & Calvi, L. (1998). AHA! An open adaptive hypermedia architecture. *New Review of Hypermedia and Multimedia*, 4(1), 115–139.
- Derntl, M., Neumann, S., Griffiths, D., & Oberhuemer, P. (2010). Investigating teachers' understanding of IMS learning design: yes they can!. In *European conference on technology enhanced learning* (pp. 62–77) Springer Verlag.
- Durand, C., Downes, S., 2009. Toward simple learning design 2.0. In: *2009 4th International Conference on Computer Science & Education*. Nanning, China, pp. 894–897.
- Ebner, M., Lienhardt, C., Rohs, M., & Meyer, I. (2010). Microblogs in higher education – a chance to facilitate informal and process-oriented learning? *Computers & Education*, 55(1), 92–100.
- Escobedo-del-Cid, J. P., de-la-Fuente-Valentín, L., Gutiérrez, S., Pardo, A., & Kloos, C. D. (Sep. 2007). Implementation of a learning design run-time environment for the .LRN learning management system. *Journal of Interactive Media in Education*, (Adaptation and IMS Learning Design. Special Issue).
- Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering Education*, 78(7), 674–681.
- Frias-Martinez, E., Chen, S. Y., & Liu, X. (2009). Evaluation of a personalized digital library based on cognitive styles: adaptivity vs. adaptability. *International Journal of Information Management*, 29(1), 48–56.
- de-la-Fuente-Valentín, L., Leony, D., Pardo, A., Kloos, C. D. User identity issues in mashups for learning experiences using IMS learning design. *International Journal of Technology Enhanced Learning*, in press.
- de-la-Fuente-Valentín, L., Pardo, A., & Kloos, C. D. (Oct. 2009). Using third party services to adapt learning material: a case study with Google forms. In *Learning in the synergy of multiple disciplines* (pp. 744–750). Niza: Springer.
- de-la-Fuente-Valentín, L., Pardo, A., Pérez, J. I. A., Dimitriadis, Y., & Kloos, C. D. (2008). Collaborative learning models on distance scenarios with learning design: a case study. In *ICALT '08: Proceedings of the eighth IEEE international conference on advanced learning technologies* (pp. 278–282). Spain: Santander.
- García-Robles, R., Díaz-del-Río, F., Vicente-Díaz, S., & Linares-Barranco, A. (2009). An e-Learning standard approach for supporting PBL in computer engineering. *IEEE Transactions on Education*, 52(3), 328–339.
- He, S., Kinshuk, H., Patel, A., 2002. Granular approach to adaptivity in problem-based learning environment. In: *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2002)*, pp. 3–7.
- Hernandez-Leo, D., Bote-Lorenzo, M., Asensio-Perez, J., Gomez-Sanchez, E., Villasclaras-Fernandez, E., Jorrin-Abellan, I., et al. (2007). Free- and open-source software for a course on network management: authoring and enactment of scripts based on collaborative learning strategies. *IEEE Transactions on Education*, 50(4), 292–301.
- Hernández-Leo, D., Villasclaras-Fernández, E., Jorrón-Abellán, I. M., Asensio-Pérez, J. I., Dimitriadis, Y., Ruiz-Requies, I., et al. (2006). Collage, a collaborative learning design editor based on patterns. *Educational Technology & Society*, 9(1), 58–71, Special Issue on Learning Design.
- IMS Global Learning Consortium. (Feb. 2003). IMS learning design specification. <http://www.imsglobal.org/learningdesign/> [On line]. Last visited: October 2010.
- IMS Global Learning Consortium. (March. 2003). IMS Simple Sequencing. information model, "best practice and implementation guide". <http://www.imsglobal.org/simplesequencing/> [Online], Last visited: September 2010.
- Johnson, R. R. B., & Christensen, L. B. (2007). *Educational research: Quantitative, qualitative, and mixed approaches*. Sage Publications, Inc.
- Karampiperis, P., & Sampson, D. (2007). Towards a common graphical language for learning flows: transforming BPEL to IMS learning design level a representations. In *IEEE international conference on advanced learning technologies* (pp. 798–800). Los Alamitos, CA, USA: IEEE Computer Society.
- Koper, R., & Tattersall, C. (Eds.). (2005). *Learning design. A handbook on modelling and delivering networked education and training*. Springer Verlag.
- Martínez, A., Dimitriadis, Y., Rubia, B., Gómez, E., & De La Fuente, P. (2003). Combining qualitative evaluation and social network analysis for the study of classroom social interactions. *Computers & Education*, 41(4), 353–368.
- Martínez-Ortiz, I., Sierra, J., & Fernández-Manjón, B. (2009). Authoring and reengineering of IMS learning design units of learning. *IEEE Transactions on Learning Technologies*, 2(3), 189–202.
- Moreno-Ger, P., Burgos, D., Martínez-Ortiz, I., Sierra, J. L., & Fernández-Manjón, B. (2008). Educational game design for online education. *Computers in Human Behavior*, 24(6), 2530–2540.
- Muntean, C. H. (2008). Improving learner quality of experience by content adaptation based on network conditions. *Computers in Human Behavior*, 24(4), 1452–1472, (including the Special Issue: Integration of Human Factors in Networked Computing).
- Neumann, S., Klebl, M., Griffiths, D., Hernández-Leo, D., de-la-Fuente-Valentín, L., Hummel, H., et al. (2010). *Report of the results of an IMS learning design expert workshop*. Neumann, S., Oberhuemer, P., Derntl, M., 2009. Visualizing learning designs using IMS learning design: the position of the graphical learning modeller. In: *2009 Ninth IEEE International Conference on Advanced Learning Technologies*. Riga, Latvia, pp. 732–733.
- del Puerto Paule Ruiz, M., Fernández Díaz, M. J., Ortín Soler, F., & Pérez Pérez, J. R. (2008). Adaptation in current e-learning systems. *Computer Standards & Interfaces*, 30(1–2), 62–70.
- Rees, D. (2000). *Essential statistics* (4th ed.). Chapman and Hall/CRC.
- Rodríguez, M., Rifon, L., Nistal, M., 2004. Towards IMS-LD extensions to actually support heterogeneous learning designs. A pattern-based approach. In: *IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings*. Joensuu, Finland, pp. 565–569.
- Stake, D. R. E. (1995). *The art of case study research*. Sage Publications, Inc.
- Towle, B., & Halm, M. (2005). Designing adaptive learning environments with learning design. In *Learning design* (pp. 215–226). Springer Berlin Heidelberg.
- Valdivia, R., Nussbaum, M., & Ochoa, S. (2009). Modeling a collaborative answer negotiation activity using IMS-based learning design. *IEEE Transactions on Education*, 52(3), 375–384.
- Vogten, H., Martens, H., Nadolski, R., Tattersall, C., van Rosmalen, P., & Koper, R. (2006). Coppercore service integration – integrating IMS learning design and IMS question and test interoperability. In *ICALT '06: Proceedings of the sixth IEEE international conference on advanced learning technologies* (pp. 378–382). Washington, DC, USA: IEEE Computer Society.
- Vogten, H., Tattersall, C., Koper, R., van Rosmalen, P., Brouns, F., Sloep, P., et al. (October 2006). Designing a learning design engine as a collection of finite state machines. *International Journal on E-Learning*, 5(4), 641–661.
- Zeng, M., Zhou, Y., 2007. The application of IMS learning design to develop compute-based educational game. In: *Seventh IEEE international conference on advanced learning technologies ICALT 2007*. Nigata, Japan, pp. 726–727.