Derntl, M., Neumann, S., & Oberhuemer, P. (2014). Lost in Interaction in IMS Learning Design Runtime Environments. *Educational Technology & Society, 17* (3), 332–342.

Lost in Interaction in IMS Learning Design Runtime Environments

Michael Derntl[1*], Susanne Neumann[2] and Petra Oberhuemer[2,3]
[1]RWTH Aachen University, Advanced Community Information Systems (ACIS), Informatik 5, Aachen, Germany // [2]University of Vienna, Center for Teaching and Learning, Vienna, Austria // [3] University of Vienna, Teaching Affairs and Student Services, Vienna, Austria // derntl@dbis.rwth-aachen.de // susanne.neumann-heyer@univie.ac.at // petra.oberhuemer@univie.ac.at
[*]Corresponding author

(Submitted March 15, 2013; Revised October 11, 2013; Accepted December 5, 2013)

ABSTRACT

Educators are exploiting the advantages of advanced web-based collaboration technologies and massive online interactions. Interactions between learners and human or nonhuman resources therefore play an increasingly important pedagogical role, and the way these interactions are expressed in the user interface of virtual learning environments is assumed to greatly impact the learning process. This paper presents an evaluation of how interactions for learning are expressed in the user interface of an IMS Learning Design (LD) runtime environment. Specifically, we selected Reigeluth and Moore's set of interactions for learning was selected and tested how these interactions are visually expressed and supported in the user interface of the Service Based Learning Design (SLeD) player. The findings show several drawbacks in the current way of visualizing units of learning during runtime. For instance, the synchronization of participants in corresponding learning and support activities is not clearly expressed, the current role remains unclear in multi-role settings, and the nested display of unit-of-learning contents impedes navigation. Drawing from these findings, the paper offers recommendations regarding future development of IMS LD runtime user interfaces.

Keywords

Learning design, Learning interactions, IMS LD, SLeD player, Virtual learning environments

Introduction

Learners today use a multitude of web-based tools and services to support their interactions during learning experiences. While interactions for learning have always been an important aspect of education, several recent trends complicate the teachers' roles as providers and orchestrators of learning interactions today. An example of such a trend is the rapid pace of development and advancement of web-based technologies that support interaction. Since the advent of Web 2.0 and successful web-based business models, new collaboration tools and services become available to web users on a daily basis for free. Another recent trend on a global scale, propelled by rapid innovation in web technology and web pedagogy, is the offering and support of learning opportunities that build on openness and interaction on a massive scale, e.g., in massive open online courses (Yuan & Powell, 2013). A decade ago, the learning-technology world was much simpler, limited to classic forms of interaction support such as chat rooms and asynchronous message boards to support learners and their interactions with other learners, tutors, and teachers. At that time, the IMS Learning Design (LD) specification was developed as an interoperability instrument that allowed course authors to describe a pedagogical approach in a so-called "unit of learning" (Koper & Olivier, 2004) using a well-defined set of concepts specified in the IMS LD information model (IMS Global, 2003). A unit of learning can be a formal model of a course, a seminar unit, a self-study unit, or any other teaching and learning activity. IMS LD differentiates between the design time (when the learning design models are being authored) and the runtime (when these models are instantiated and executed). This enables the transfer of designs between different learning design authoring tools and the reuse of designs and materials in any IMS LD-compliant runtime environment. This kind of interoperability of units of learning that are specified with IMS LD enables learners and teachers to use the unit of learning in the virtual learning environment (VLE) of their choice, as long as this system is implementing the IMS LD specification. On the one hand, the need today to preserve and transfer the designs of units of learning (e.g., successful MOOCs) is stronger than ever. On the other hand, the specification and its conceptualization of interaction was set in stone in 2003 and never revised since then, and in the light of recent developments towards massive online interactions for learning it is indicated to analyze the support of interactions for learning in existing IMS LD runtime environments.

This paper puts the frequently discussed design-time issues of IMS LD aside and focuses exclusively on the runtime perspective of IMS LD. One general challenge that comes with the strict separation of design time (authoring) and runtime (execution) in IMS LD is the insecurity during authoring about how the designed unit of learning will look like in the VLE during execution, which was previously pointed out as a major problem by Neumann and colleagues (2010). Traditionally, teachers design the learning environment directly within the VLE, where they can immediately see the effects of their design choices because they are reflected in the user interface. For instance, when a teacher adds a new assignment in a Moodle course, s/he will immediately see the effects on the course page and can react accordingly. When using IMS LD software, design decisions have to be made beforehand without knowledge of the VLE, where the unit of learning will be deployed for learning.

Adding to these conceptual challenges, which can nonetheless be met by unit-of-learning authors (see Derntl, Neumann, Griffiths, & Oberhuemer, 2012) and appropriate runtime management tools (e.g., Leony, de La Fuente Valentín, Pardo, & Delgado Kloos, 2008), a severe shortcoming on the runtime side is that there are only few VLEs that are able to read and execute IMS LD units of learning. There have been various attempts to implement an IMS LD runtime in mainstream VLEs. An implementation in Moodle was discussed (e.g., in Burgos, Tattersall, Dougiamas, Vogten & Koper, 2007) but never realized. An IMS LD implementation as an extension of .LRN called GRAIL succeeded (Escobedo del Cid, de la Fuente Valentín, Gutiérrez, Pardo & Delgado Kloos, 2007); however, the resulting tool has not been adopted widely. Most of the other existing IMS LD players are either commercial products like CLIX (http://www.im-c.de/germany/en/solutions/learning-management/clix-learning-suite) or have been developed in European R&D projects, for example, the SLeD player (http://sourceforge.net/projects/ldplayer/) and the more recently developed Astro Player (http://tencompetence-project.bolton.ac.uk/ldruntime/index.html).

To identify how interactions for learning are expressed in current IMS LD runtime systems, we analyzed a representative IMS LD player, the SLeD player, according to relevant pedagogical principles. One goal of the analysis was to understand interaction issues in IMS LD runtime systems beyond pure technical interpretation and visualization of IMS LD units of learning. Drawing from this analysis, a second goal was to distill recommendations for IMS LD runtime environment developers.

Our paper is structured as follows. In the following section, we present the theoretical background of the study and the study objectives. In the third section, we present the methodology to obtain and analyze IMS LD units of learning at runtime. In the next section, we present a detailed account of the results obtained during runtime analysis. Following that, we discuss the technological restrictions and remedies to support runtime interaction with IMS LD. In the final section, we outline findings and recommend actions for future work.

Background and objectives

In this paper, we assume that the reader is familiar with the main concepts of the IMS LD specification. An excellent introductory article by Koper and Olivier was previously published in this journal (Koper & Olivier, 2004). Since then, a considerable body of research has been produced regarding the design time, that is, the authoring of IMS LD units of learning. (See Griffiths, Blat, García, Vogten and Kwong [2005] for an early discussion and Lockyer, Bennett, Agostinho and Harper [2009] for a more recent, comprehensive overview.) Given the current forces in web-based education as outlined in the introduction, it is equally important to view IMS LD from the runtime perspective. Less research was published on this aspect, since it is primarily an implementation effort. Because IMS LD, with its focus on roles and activities, uniquely set up among learning-technology specifications, it is particularly suited to be tested for the expression of pedagogical aspects in VLEs. For this reason, Reigeluth and Moore's framework for comparing instructional strategies was chosen as an analysis framework (Reigeluth & Moore, 1999). This framework is appropriate because the contained aspects are well accepted and can be classified more precisely than with other frameworks, such as Reeves' pedagogical dimensions of computer-based education (Reeves, 1997). Of the six aspects in Reigeluth and Moore's framework, "interactions for learning" was selected for the evaluation because it appeared to be most meaningful for evaluating VLEs' ability to deal with multiple roles and active engagement, which are considered to be strengths of IMS LD compared to other learning activity-focused specifications (Koper & Olivier, 2004; Griffiths & Liber, 2008).

In the framework proposed by Reigeluth and Moore (1999) human interactions are categorized into human and nonhuman interactions. The types of interactions that students engage in during the learning process are distinguished as follows in the framework, with the first three representing human and the last four representing nonhuman interactions:

- Student-teacher: interacting with the teacher or instructor
- Student-student: working with or utilizing other students as resources, individually or in a group
- Student-other human: interacting with a community member, parent, or other individual (or group)
- Student-tools: using tools that enable completion of tasks
- Student-information: working with, and making sense of, the information that is available or found
- Student-environment/manipulatives: utilizing and working with resources and simulations, both within and outside the classroom environment
- Student-other nonhuman: working with any other conceivable nonhuman resource.

Each of these interactions pursues a pedagogical purpose such as creating dependence between students for student/student interactions, going beyond the traditional classroom for learning challenges in student/environment interactions, or producing high quality products during student/tools and student/information interactions.

This analytical study was set up to analyze IMS LD units of learning (from here on simply referred to as "units of learning") according to the (visual) expression of the above-mentioned interactions for learning in VLEs. This is not a test of the IMS LD specification itself (e.g., for its ability to express certain pedagogical aspects), but rather for the ability of VLEs to support and express pedagogical aspects contained in IMS LD units of learning and whether the chosen expression fosters or hinders the learning process.

Methodology

Material selection

Units of learning were solicited from several European organizations that were part of the ICOPER consortium, a European Commission-funded best-practice network that investigated, among other issues, contemporary learning design standards and interoperability specifications (cf., Simon, Pulkkinen, Totschnig, & Kozlov, 2011). These organizations had built units of learning for specific disciplines, such as architecture, or for specific learning settings like collaborative learning. Criteria for eventual selection among collected units of learning were that they needed to exhibit characteristics relating to the interactions for learning. Six units of learning were selected that exhibited at least two different types of interactions (see Table 1). Two types of interaction listed in the previous section were not considered in the analysis (namely the "other" interactions) and the student-tools and student-environment or manipulatives interactions were subsumed under the student-tools interaction.

Table 1. IMS LD units of learning used for analysis.

IMS LD unit-of-learning title	Short description	Interactions*			
		SS_H	ST_H	ST_N	SI_N
DECONSTRUCTIVISM	There are two roles, learner and teacher. Learners are to explore a project called Villa dall'Ava, and then create their own project. The teacher supports learners when using the resources and gives them feedback on their project.	Yes	Yes	Yes	Yes
MODERN ARCHITECTURE	There are two roles, learner and teacher. Learners brainstorm their previous knowledge on modern architecture, then read some selected content and prepare a presentation on this content. The teacher supports learners when they use tools and resources. After learners' presentations, the teacher gives feedback.	No	Yes	Yes	Yes
SKYSCRAPERS AND RESIDENTIAL HOMES—LEVEL A	The unit of learning features three roles: "interested in architecture", "interested in buildings" and "teacher," but the teacher role was not assigned any activities. Depending on their assigned role, learners move through the unit of learning according to architects and their styles or according to famous buildings. Some reuse of learning objects and activities can be seen. Another special feature of the unit of learning is that it comprises two plays.	No	No	Yes	Yes
SKYSCRAPERS AND RESIDENTIAL	The unit of learning features one role, which is a learner role. Learners first choose whether they wish to learn about skyscrapers	No	No	Yes	Yes

HOMES—LEVEL B	or about residential homes. Each activity and interaction with learning content gives learners also a chance to reflect and summarize what they've learned. At the end, learners summarize everything they learned.				
SHARED OUTCOME	There are five roles. In addition to the teacher role, learners can either be Team A or Team B, and each team has an additional coordinator role. The teacher introduces the topic, then learners split into two groups A and B, around an assigned theme, for the project work phase. Each group selects a person to be the coordinator, who aggregates the group's work. After group work, students answer a questionnaire. If they pass, the activity is finished. Otherwise, the group has to re-work its project to find the correct answers to all the questions. The teacher has the option to support students when needed.	Yes	Yes	Yes	Yes
BLOG COLLABORATION	There are two roles, learner and teacher. Learners select a topic from one of two lists, which the teacher provided. One list contains learning technologies; the other contains learning contexts. Students keep a blog on the chosen technology or context, and discuss with each other how to apply learning technologies in various learning contexts. In a final report, learners describe scenarios for applying learning technologies in specific learning contexts.	Yes	Yes	Yes	Yes

* SS_H … Student-student (human) interactions ST_N … Student-tools and student-environment/manipulatives (nonhuman) interactions
ST_H … Student-teacher (human) interactions SI_N … Student-information (nonhuman) interactions

Runtime player

From the VLEs that are capable of interpreting IMS LD units of learning, the system that has been established the longest, the Service Based Learning Design player (SLeD, version 3.0) (http://sourceforge.net/projects/ldplayer/files/SLeD%20Player/Sled%203.0/sled3.0.zip/download), was chosen for the analysis. The SLeD player is an enhanced version of the CopperCore LD player. The latter is also known as the player module of the CopperCore Run Time (CCRT) environment (http://coppercore.sourceforge.net/documentation/ccrt.shtml). While the CCRT-integrated player can be used to run an LD, the distribution page of CCRT clearly states that it was "not intended to be used in a real-life deployment." The SLeD player, on the other hand, builds upon CCRT and enhanced the state of the art by "separating out the player functionality from the underlying engine" (McAndrew, Nadolski, & Little, 2005). SLeD is the only player to date that fully implements IMS LD based on CCRT and is freely available for educational and research purposes. To substantiate this, we performed pre-study trials with other available players. GRAIL requires a dotLRN host system and is visually similar in its setup to SLeD but architecturally more tightly integrated with the host VLE. We realize that an analysis using GRAIL may have provided different results than the ones presented in this paper. Trials with the other available players CLIX and the Astro Player showed that these systems did not support all features used in the units of learning used for the study, as presented in the previous sub-section. For instance, the Astro Player did not support the displaying of multiple activity descriptions, which is however possible in the IMS LD specification.

IMS LD players are usually set up in a similar manner. To access activities, participants click on one of the links in the navigation tree on the left (see Figure 1). When an activity is selected from the navigation tree, the content area of the browser window provides access to the activities' descriptions and learning objectives. If the activities or learning objects contain several items, players can access them by selecting the activity or learning object in the navigation tree on the left and choosing 1, 2, 3, etc., within the main frame on the right. For instance, the currently selected learning object "Skyscrapers—Seagram Building" in Figure 1 contains three subordinate items as indicated by "1 2 3" under the Description tab.

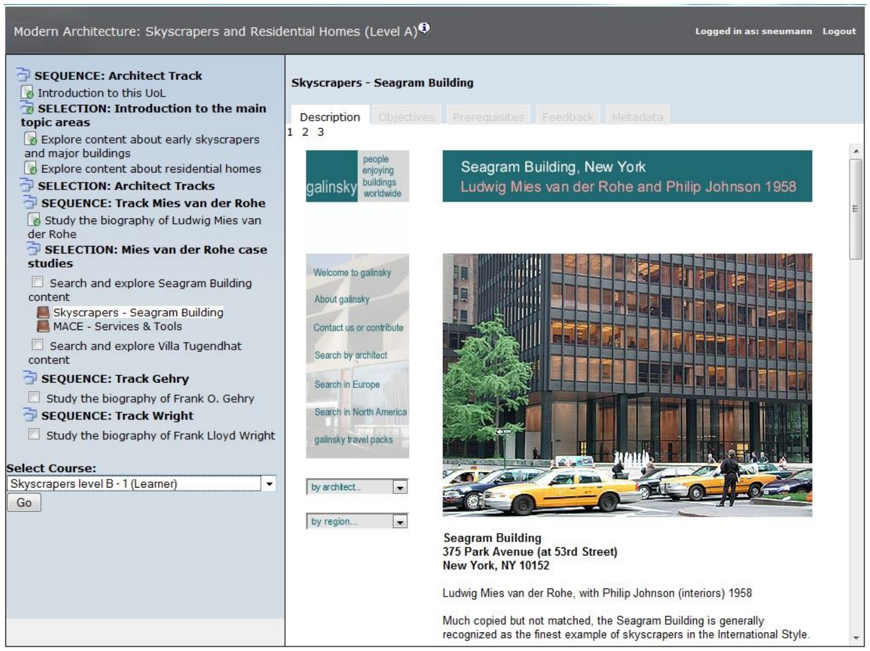

Figure 1. Screenshot of IMS LD player SLeD used in the evaluation

Analysis procedure

Two IMS LD experts conducted the following analysis procedure independently. The analysis results were discussed and integrated in a face-to-face session. The interpretation of the results in the light of the research question tackled in this article was done by the authors. The analysis procedure proceeded as follows. First, the IMS LD conformant XML representation of the units of learning, including their packaged resources, were analyzed structurally in relation to the interactions for learning outlined in the previous section. That means, IMS LD elements used to provide these interactions were noted down. Then, one after the other, each unit of learning was played in the SLeD player. In each unit of learning each role was impersonated; all activities were performed in all possible permutations of the activity sequence through the unit of learning, and all included services and resources were accessed and used. While doing so we recorded how Reigeluth and Moore's interactions for learning were expressed visually and/or metaphorically in the user interface of the SLeD player. These expressions of the interaction were then analyzed by the authors of this paper for the ability to support the interactions during runtime. The results of this analysis procedure are presented in the following section.

Results

The results are grouped by the interaction typology introduced before. "Student-other human" and "Student-other nonhuman" were excluded from further analysis as interactions since these were not featured in any of the units of learning selected for this study. These interactions, while representable with IMS LD, generally appear to be out of the scope of capturing and analysis in an IMS LD runtime player.

Student-student

DECONSTRUCTIVISM, SHARED OUTCOME, and BLOG COLLABORATION (Text in small caps refers to the units of learning presented in Table 1) featured student-student interactions. Learners in these units of learning could only identify a chance for student-student interaction if they received explicit instructions in the activity, and if the author of the unit of learning included corresponding services where the interaction takes place, such as a chat or forum. Students were at times asked to post an idea, link, or other information, but there were no instructions for working with or responding to one another. Next to the missing instructions for posting or interacting, it was not apparent who else had access to and participated in the forum. What other students had access? Were forums meant for student access and discussion only, or did teachers have access as well? To support student-student interactions, it should be made clear with whom students are to interact. Most modern VLEs offer features to support awareness who has access to and who is contributing in a VLE. The SLeD player, a VLE by definition, does not display this information, although the IMS LD conceptual model would easily allow revealing it.

Shared Outcome featured four different learner roles: Team A, Team B, Team Leader A, Team Leader B. During the course of the unit of learning, the team members chose a team leader, who was then assigned this additional role. For a person assigned to the Team Leader A role, it was not clear in the user interface how and when to act as team leader and when to act as a regular member of team A (because the person likely acted in both roles). The player did not offer explicit support for this change of roles. Switching roles was not obvious because the player displayed all units of learning a person was currently assigned to in one drop-down box; the role the person took within each unit of learning was written in parentheses behind the unit of learning's title.

Student-teacher

Student-teacher interactions were mostly realized using the IMS LD concept of support activities in the units of learning. In the player, this interaction was not well supported. The view offered to learners differed from the view offered to teachers: Each role saw only those activities that were assigned to this very role. Deconstructivism, Modern Architecture, Shared Outcome, and Blog Collaboration featured student-teacher interactions and thus encountered this problem. For illustration, see the screenshots below (Figure 2 and Figure 3), which were taken from the Deconstructivism unit of learning. As the two navigation trees indicate, there was no guidance for learners or teachers regarding when the teacher was to support the students during their activities. The player did not support the teacher in finding out about learners' current status.

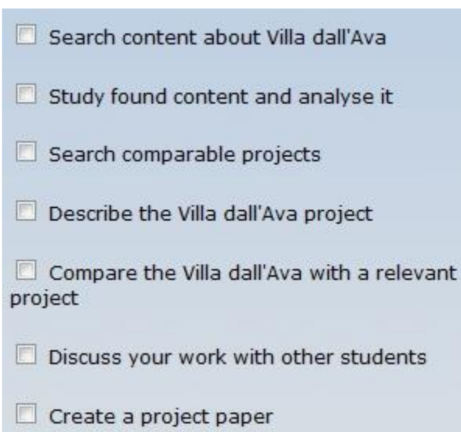

Figure 2. Student navigation within Deconstructivism in SLeD

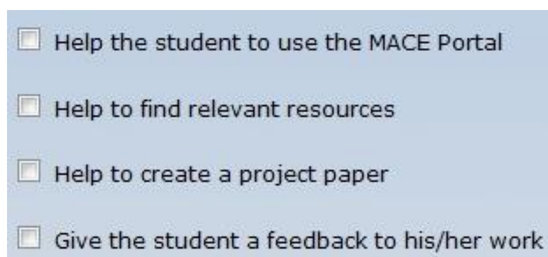

Figure 3. Teacher navigation within Deconstructivism in SLeD

Having isolated views for each role's activities created an artificial separation in the VLE. It required some explaining within the activity descriptions to overcome this separation, that is, to make clear that teachers and students were to work together at this point. This is a difference between currently available IMS LD players and non-IMS LD VLEs. In non-IMS LD VLEs, teachers and learners see the same interface in the learning environment, and all instructions in the interface are targeted towards learners, while the teacher implicitly assumes any support tasks. The SLeD player interprets IMS LD units of learning in a way that learners and teachers are presented with separate views onto the (same) unit of learning. Each role receives own activities with own instructions and is not aware of other roles and their activities. This makes it difficult to identify when the different roles will interact.

In the units of learning Deconstructivism, Modern Architecture, Shared Outcome, and Blog Collaboration, the teachers' activities were IMS LD support activities. This special type of activity did not appear any different in the player than a learning activity. The teacher was not offered additional functions to execute this support activity and may thus not even know that s/he was involved in a support activity. What appears to create further uncertainty is that in the IMS LD specification support activities are by definition not linked to another activity but to another role. The specification is unclear whether the indicated role is only valid during an act, during a play, or always. This leaves room for interpretation for player vendors and poses obstacles for student-teacher interactions.

Student-tool and student-environment/manipulatives

In a VLE, as opposed to a face-to-face learning environment, these two interactions are hard to distinguish because the line between the environment and its tools diminishes. These two interactions were thus observed together for the purposes of this analysis. Interactions with tools and environments can be regarded as interactions with the VLE itself and with provided resources as well as tools. Such resources or tools are sometimes external to the unit of learning, for example, external webpages. In the player, learners received no hint regarding what role they were assigned and what role they currently carried out (what the role is called and a possible description of the role). This information, however, influences the learners' interaction with the tools and environment, and is thus important to display. For instance, the unit of learning SKYSCRAPERS AND RESIDENTIAL HOMES—LEVEL A features two different learner roles: "interested in architects" and "interested in buildings". Learners had no idea that they were assigned one of these special foci in the unit of learning since the role's title was not displayed. Acting in either one of the roles, however, affected how students interacted with the tools and the environment.

In the unit of learning SKYSCRAPERS AND RESIDENTIAL HOMES—LEVEL A, the nested activity structures are confusing and do not support navigational orientation. As can be seen from the screenshot of the unit of learning SKYSCRAPERS AND RESIDENTIAL HOMES—LEVEL A in Figure 4, keeping the overview of the different layers inside the activity structure could be quite hard. Adding to the structural complexity the activity structure "Selection: Architect Tracks" contained three nested activity structures, each containing additional activities and resources. In the unit of learning SKYSCRAPERS AND RESIDENTIAL HOMES—LEVEL B, navigation was further hampered because it was hard to distinguish between activities that were part of the activity structure "Sequence: Introduction into skyscrapers" and activities that were part of the regular sequence because they are nearly horizontally aligned (cp. Figure 5). Also, the distinction between activity structures of type sequence and of type selection should be better reflected in runtime interfaces so it is clear to learners what the difference is and what options for progression they have. In the screenshots, it might appear as if the SLeD player did actually make this distinction because "sequence" and "selection" were written in the navigation tree. In fact, unit-of-learning authors, not the runtime system, defined these titles; the distinction would not be clear without the explicit inclusion of "sequence" and "selection" in the activity structures' titles.

Figure 4. Screenshot of the nested activity structures in the unit of learning SKYSCRAPERS AND RESIDENTIAL HOMES—LEVEL A

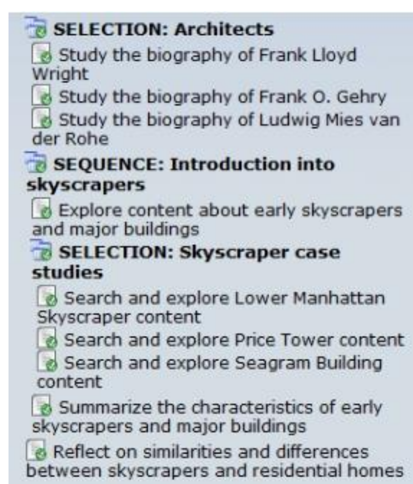

Figure 5. Screenshot of the nested activity structures in the unit of learning SKYSCRAPERS AND RESIDENTIAL HOMES—LEVEL B

Next to the nested activity structures, some activities included additional activity descriptions or learning objects, which in turn had further nested objects. For instance, the learning object “Explore content about early skyscrapers and major buildings” (cp. Figure 4) provided access to 12 subordinate objects (note that the actual subordinate objects were only displayed in the main frame and are not depicted in the navigation tree shown here). Some of these objects again featured links to other sites. Learners would have to navigate through up to five levels. The student-environment interaction was greatly impeded through this complicated setup. Although IMS LD supports the use of nested structures, one of the goals of VLEs should be to ease navigation by modifying the display of activities and learning objects.

Student-information

The difference between the information and the environment was not easily drawn because the information is highly embedded in the environment. For the purposes of this analysis, student-information interactions were interpreted as the interactions taking place with activities or learning objects. From a learner’s point of view, it was hardly understandable what the difference between learning objects and activity (descriptions) was in the player interface. The activity description stated to “do something” but the place to do this was not immediately provided where the activity description was displayed. The separation between activities, whose description was displayed in the main frame on the right, and the referenced learning objects, which were only accessible via the navigation tree on the left, introduced an additional layer of complexity. The use of activities and learning objects as seen in the units of learning under analysis suggests that IMS LD activities were at times used as an additional, yet nearly useless, container for referencing learning objects. If used this way, the line between the IMS LD concepts’ activity and environment (with contained learning objects) diminishes. This may also have been an inspiration to the developments of Simple Learning Design 2.0 (Durand & Downes, 2009), where only one of the concepts (the activity) is included, and the concept environment is not present at all.

Interaction services

The investigation so far has shown that interaction between learners and other human or nonhuman entities is an important pedagogical aspect of VLEs. IMS LD as a prominent interoperability specification should therefore be able to express such collaboration and communication interactions. This section discusses in more detail how this can be achieved in the runtime perspective.

In the IMS LD information model (IMS Global, 2003), activities take place in environments, which can contain learning objects, services, and further environments. A service element in an IMS LD unit of learning is an abstract declaration of the required facility or tool, for example, a synchronous conferencing service. During runtime, the IMS LD player will create and offer an instance of the service for use within the environment. Since different services like discussion forum, chat, announcement, etc., offer different functionalities and require different steps and parameters during setup and use, there is a dedicated description schema for each service. When the IMS LD specification was conceived, the authors decided to include the four “most widely implemented and used services” (Olivier & Tattersall, 2005, p. 32) into the specification. These are send-mail (for emailing), conference (synchronous, asynchronous communication, and announcements), monitor (observing property values), and index-search (searching in the unit of learning). From the teaching practitioner’s point of view, this limitation comes with drawbacks: The creation of units of learning with intense virtual communication and collaboration as in the study presented in the previous section may become problematic since the only services that are available for this purpose are send-mail and conference. There is no service description schema in IMS LD for currently popular applications supporting collaboration and communication like blogs and microblogs (Twitter), wikis (Wikipedia), shared virtual calendars, document sharing (Google Drive), polling (Doodle), and so forth. While these are gradually being integrated as external services in popular VLEs like Moodle, IMS LD (conceived a decade ago) was not built to cater to these new options.

A simple workaround that does not require any modification or extension of the IMS LD information model would be to provide a link to or description of the required service as well as the instructions for setup and use of the service in the IMS LD activity description element. There are several potential problems with such an approach. For one, the external service likely requires login with a separate account (or signing up for a new one). There is currently no way to map the user's account in the service with his/her account in the IMS LD player. Also, different users have different roles within the same activity/environment (e.g., moderator and participant in discussions). These roles need to be mapped to the roles offered by the service (if any), but the mapping cannot be enforced because there is no protocol for transferring any data between the IMS LD player and the external service. Additionally, the design-time provision of a static hyperlink to a service instead of an abstract service description means that different runs of a unit of learning will use the same instance of the service.

Other approaches to using services that are not offered by IMS LD potentially require a modification to the IMS LD information model and/or a customized interpretation of the existing XML binding and will therefore neither be portable to nor runnable on all existing IMS LD players. In recent years, several approaches have been proposed in this area. The perceived trend upon which these developments build is that of a move towards personal learning environments that are composed of small, autonomous, loosely coupled applications, for instance based on widget technology (cf. Wilson, Sharples, & Griffiths, 2009). Another avenue to broaden the spectrum of available interaction services is based on generic service integration (de la Fuente Valentín, Miao, Pardo and Delgado Kloos, 2008), which supports the semi-automatic identification appropriate services at runtime based on a design-time description of what a service is supposed to achieve during runtime, rather than describing which specific service to use.

As it currently stands, the IMS LD specification and, therefore, most runtime players offer only two services that can be used for collaboration and communication: the send-mail service for email communication and the conference service for synchronous and asynchronous communication/collaboration and one-way communication. The concrete appearance and handling of these services is left to the player. Unfortunately, there is currently no agreed way of providing richer communication and collaboration services in IMS LD. In the light of the large number of interaction services available and used for educational purposes, this shortcoming will continue to diminish the role of IMS LD.

Discussion and conclusion

The task of IMS LD compliant VLEs is to interpret a unit of learning so that the learning environment is set up and the learning process is best supported. The VLE must work with the information provided in the IMS LD unit of learning. What information can or should be derived from this unit of learning at runtime to support interactions for learning? In this paper, we explored answers to this question in an IMS LD runtime environment, namely the Service Based Learning Design (SLeD) player, and selected and adopted interactions for learning from Reigeluth and Moore's (1999) set of pedagogical aspects. We demonstrated that the expression and implementation of interactions of learners with human actors and nonhuman elements in the learning environment are currently far from optimal and far from fit for current pedagogies and diverse communication channels during learning experiences. Crucial issues encountered include no explicit link between activity descriptions in the main content area and the environment objects in the navigation area, unless the author provides a descriptive linkage, which would however contradict the basic idea of IMS LD by requiring knowledge of the runtime user interface at design time. The first recommendation to developers, therefore, is to provide in-place access to information pertaining to the current activity, which would enhance student-information interactions by integrating the activity description with the learning object more closely in the user interface or by providing instant previews common in popular VLEs like Moodle.

The second recommendation is that items that conceptually belong together be displayed together in the user interface to facilitate the learners This is the case for activities and environments that are linked to one another as well as for their subordinate items, such as multiple activity descriptions and multiple learning objects. Currently, the display of conceptually connected items is separated in the observed runtime environment: The participant has to switch back and forth between the navigation tree and the main frame to get the wanted items to display. The hierarchical nesting as the display metaphor of all unit-of-learning content/activities additionally impedes easy navigation since there is little learning-process-related information contained in such a structure. The third recommendation, therefore, is that IMS LD player developers devise a meaningful depiction of nested activity structures in the user interface to guide learners in progressing along the learning path. Although we don't know why, this kind of nested navigation structure is a visual design flaw present in all IMD LD players to some extent. Lambe acknowledges that nested depiction of the unit of learning's structure becomes hard to navigate beyond three levels of depth (2007).

We revealed that it is not clear at runtime which role or roles the user is currently impersonating, and there are missing visual cues on supported and/or collaborating roles for the current activity. Thus, the fourth recommendation is that roles should be explicitly displayed in the user interface. Also, the runtime system does not indicate places to interact with other roles (even if designed that way), such as when and how does a supporter actually support learners? Missing indication of interaction and awareness of other participants was also the case with any type of conference service included in the units of learning. Participants had no idea who was interacting with them within the conference service or widget. The fifth recommendation, therefore, is that the runtime system make participants aware of who is interacting (or meant to interact) with them during an activity or in an environment.

The concluding overall recommendation in light of these issues is that developers of IMS LD VLEs take more caution in observing the user perspective. Instead of exposing the XML hierarchy defined in the IMS LD unit of learning to the users, runtime environments should display units of learning in a way that lessens the cognitive load currently needed for navigating and understanding how the different parts of a unit of learning relate to each other. Thus far, many crucial elements that set IMS LD apart from other learning-technology specifications — such as roles, activities, and environments — are not expressed with their full potential in the runtime. Especially a lack of support regarding human-to-human interactions can be observed. Currently, the responsibility lies with the author of a unit of learning to make instructions for interactions explicit at design time, requiring prior knowledge about how the unit of learning will be rendered at runtime. The runtime environment did not offer support in this regard as the technical differentiation of concepts such as activities, environments, or activity structures hardly affected their display in the runtime's interface. Rather, there appeared to be a crucial connection between decisions the unit-of-learning author made at design time and their influence on the appearance of a unit of learning in the runtime system. Not knowing where the unit of learning will be deployed, however, may serve for poor design decisions.

This paper offered bits of advice to IMS LD VLE developers about what pitfalls to avoid in terms of student interactions. In future work, the expression of additional relevant pedagogical aspects and additional IMS LD players need to be taken into account. Understanding how pedagogy is expressed visually and navigationally in units of learning at runtime helps to identify shortcomings in the way runtime environments have been conceived to date. Building on knowledge about these shortcomings, work can then proceed towards interoperable runtime support of interactions for learning based on evolving pedagogies and contemporary web-based interaction technologies.

Acknowledgments

This work was partially funded by the European Commission in the ICOPER network (ECP-2007-EDU-417007); we are grateful to ICOPER partners for providing the units of learning. MD acknowledges further support by the European Commission through the TEL-Map support action (FP7-257822) and the Lifelong Learning Programme multilateral project METIS (531262-LLP-2012-ES-KA3-KA3MP).

References

Burgos, D., Tattersall, C., Dougiamas, M., Vogten, M., & Koper, R. (2007). A first step mapping IMS learning design and moodle. *Journal of Universal Computer Science, 13* (7), 923–931.

de la Fuente Valentín, L., Miao, Y., Pardo, A., & Delgado Kloos, C. (2008). A supporting architecture for generic service integration in IMS Learning Design. In *Proceedings of the Third European Conference on Technology Enhanced Learning, EC-TEL 2008* (pp. 467–473). Berlin / Heidelberg: Springer.

Derntl, M., Neumann, S., Griffiths, D., & Oberhuemer, P. (2012). The conceptual structure of IMS Learning Design does not impede its use for authoring. IEEE Transactions on Learning Technologies, 5 (1), 74–86.

Durand, G., & Downes, S. (2009). Toward Simple Learning Design 2.0. *Proceedings of the 4th International Conference on Computer Science & Education* (pp. 894–897). Nanning, China: IEEE. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5228214

Duval, E., Verbert, K., & Klerkx, J. (2011). Towards an open learning infrastructure for open educational resources: Abundance as a platform for innovation. In C. S. Calude, G. Rozenberg, & A. Salomaa (Eds.), *Rainbow of Computer Science* (pp. 144–156), Berlin, Germany: Springer.

Escobedo del Cid, J. P., de la Fuente Valentín, L., Gutiérrez, S., Pardo, A., & Delgado Kloos, C. (2007). Implementation of a learning design run-time environment for the .LRN learning management system. *Journal of Interactive Media in Education*, September 2007, 1–12.

Griffiths, D., Blat, J., García, R., Vogten, H., & Kwong, K. L. (2005). Learning design tools. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 109–135), New York, NY: Springer.

Griffiths, D., & Liber, O. (2008). Opportunities, achievements, and prospects for use of IMS LD. In L. Lockyer, S. Bennett, S. Agostinho, & B. Harper (Eds.), *Handbook on Learning Design and Learning Objects* (pp. 87–112). Hershey, PA: IGI Global.

IMS Global (2003). *IMS larning design lnformation model*. Retrieved February 25, 2013, from http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html

Koper, R., & Olivier, B. (2004). Representing the learning design of units of learning. *Educational Technology & Society, 7*(3), 97–111.

Lambe, P. (2007). *Organising knowledge: Taxonomies, knowledge and organisational effectiveness*. Oxford, UK: Chandos.

Leony, D., de la Fuente Valentín, L., Pardo, A., & Delgado Kloos, C. (2008). *Basic resource edition in GRAIL*. *Proceedings of the 7th OpenACS /.LRN Conference*, Valencia, Spain.

Lockyer, L., Bennett, S., Agostinho, S., & Harper, B. (Eds.) (2009). *Handbook of research on learning design and learning objects: Issues, applications, and technologies*. Hershey, PA: IGI Global.

McAndrew, P., Nadolski, R., & Little, A. (2005). Developing an approach for learning design players. *Journal of Interactive Media in Education, 2005* (14).

Neumann, S., Klebl, M., Griffiths, D., Hernandez-Leo, D., de la Fuente Valentín, L., Hummel, H., Brouns, F., Derntl, M., & Oberhuemer, P. (2010). Report of the results of an IMS Learning design expert workshop. *International Journal of Emerging Technologies in Learning, 5*(1), 58–72.

Olivier, B., & Tattersall, C. (2005). The learning design specification. In R. Koper & C. Tattersall (Eds.), *Learning Design* (pp. 21–40), New York, NY: Springer.

Reeves, T. (1997). *Evaluating what really matters in computer-based education*. Retrieved March 13, 2013, from http://www.eduworks.com/Documents/Workshops/EdMedia1998/docs/reeves.html

Reigeluth, C. M., & Moore, J. (1999). Cognitive education and the cognitive domain. In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory* (pp. 51–68). Mahwah, NJ: Lawrence Erlbaum.

Simon, B., Pulkkinen, M., Totschnig, M., & Kozlov, D. (2011). The ICOPER reference model for outcome-based higher education. ICOPER Deliverable D7.3b. Retrieved February 17, 2013, from http://www.icoper.org/deliverables/ICOPER_D7.3b.pdf

Wilson, S., Sharples, P., & Griffiths, D. (2009). Distributing education services to personal and institutional systems using widgets. In F. Wild, M. Kalz, M. Palmer, & D. Müller (Eds.), *Proceedings of Mash-Up Personal Learning Environments* (pp. 47–58). Retrieved from http://ceur-ws.org/Vol-388/wilson.pdf

Yuan, L., & Powell, S. (2013). *MOOCs and open education: Implications for higher education* (CETIS White Paper 2013:WP01). Retrieved March 8, 2013, from http://publications.cetis.ac.uk/2013/667