Interactions for Learning as Expressed in an IMS LD Runtime Environment

Michael Dertl*, Susanne Neumann†, and Petra Oberhuemer‡

*RWTH Aachen University, Advanced Community Information Systems, Aachen, Germany
†University of Vienna, Center for Teaching and Learning, Vienna, Austria
‡University of Vienna, Educational Affairs, Vienna, Austria
derntl@dbis.rwth-aachen.de

Abstract—The way pedagogical aspects are expressed in the user interface of virtual learning environments greatly impacts the learning process, in particular the interactions between students and other human or non-human actors in the learning environment. This paper presents an evaluation of how the 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 and tested how these are visually expressed and supported in the user interface of the SLeD Player. The findings show that there are several drawbacks in the current way of visualizing units of learning during runtime: the synchronization of participants in corresponding environment interactions, or producing high quality products without knowledge of the VLE, where the unit of learning will be instantiated for learning.

I. INTRODUCTION

IMS Learning Design (LD) was developed as a specification that can describe any pedagogical approach [7]. One of the strengths of IMS LD is the interoperability of the created units of learning; it enables teachers and learners to use the unit of learning in the virtual learning environment (VLE) of their choice, as long as this system is capable of interpreting IMS LD units of learning. The challenge that comes with this setup is the insecurity of what the designed unit of learning will look like in the VLE. The separation of environments for designing units of learning (i.e. the authoring environment) and running units of learning (i.e. the runtime environment) was previously pointed out as a major problem by IMS LD experts [9]. Traditionally, teachers design the learning environment directly within the VLE, where they can immediately see the effects of their design choices as 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 instantiated for learning.

There are few VLEs that are able to read IMS LD units of learning. Among them are GRAIL [6], the SLeD Player [12], the commercial system CLIX [3], and the recently developed Astro Player [1]. In order to identify how interactions for learning are expressed in current IMS LD runtime systems, we analyzed one representative IMS LD player, namely 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.

II. BACKGROUND AND OBJECTIVES

Although considerable research has been performed regarding the design time, i.e. authoring of IMS LD units of learning (e.g. [4]), it is equally important to view IMS LD from the runtime perspective. Because of IMS LD’s unique setup among learning-technology specifications with focus on roles and activities, it is particularly suited to be tested for the expression of pedagogical aspects in VLEs. For this purpose, we chose Reigeluth and Moore’s Framework for Comparing Instructional Strategies [11] as an analysis framework. 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 [10]. Of the six aspects in Reigeluth and Moore’s Framework, we selected “Interactions for Learning” for the evaluation as this aspect appeared 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.

In the framework proposed in [11] human interactions cover interactions of students with teachers, other students (working with or utilizing other students as resources, individually or in a group), or other humans (a community member, parent, or other individual or group). Students’ interactions with non-humans include tools (to complete tasks), information (working with, and making sense of, the information that is available or found), environment/manipulatives (utilizing and working with resources and simulations, both within and outside the classroom environment) and any other non-humans. 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.

III. METHODOLOGY

We solicited units of learning from several European organizations which had built units of learning for specific disciplines, e.g. for architecture, or for specific learning settings, e.g. for collaborative learning. Criteria for eventual selection among collected units of learning were that they needed to exhibit characteristics relating to the interactions for learning. We selected six units of learning that exhibited at least two different types of interactions (see Table I).

From the VLEs that are capable of interpreting IMS LD units of learning, we chose the system that has been established the longest, i.e. the SLeD Player [12] as the system for our analysis. The SLeD Player, which uses the CopperCore LD engine [2] for managing runtime data, is similar in its setup to GRAIL, thus removing the necessity to study GRAIL in addition to SLeD. We recognize, however, that an analysis using GRAIL may have provided different results than the ones presented in this paper. Our pre-study trials with CLIX and the Astro Player showed that these systems did not support all features used in the selected units of learning.

We first analyzed the units of learning in relation to the interactions for learning and took note of the IMS LD elements used in this regard. We then played the units of learning in the SLeD Player and recorded how the interaction was expressed visually and/or metaphorically. The expression of the interaction was then analyzed by the authors of this paper for its ability to support users during runtime.

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 activity’s related activity descriptions and learning objectives.

Table I

<table>
<thead>
<tr>
<th>Unit of Learning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deconstructivism</td>
<td>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 to their project.</td>
</tr>
<tr>
<td>Modern Architecture</td>
<td>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.</td>
</tr>
<tr>
<td>Skyscrapers and Residential Homes-Level A</td>
<td>The unit of learning features three roles: “Interested in Architects”, “Interested in Buildings” and “Teacher”. Depending on their assigned role, learners move through the unit of learning according to architects and their styles, or according to famous buildings. Another special feature of the unit of learning is that it comprises two plays, where some reuse of learning objects and activities can be seen between plays.</td>
</tr>
<tr>
<td>Skyscrapers and Residential Homes-Level B</td>
<td>The unit of learning features one role, which is a learner role. Learners first choose whether they wish to learn about skyscrapers or about residential homes. Each activity and interaction with learning content gives learners also a chance to reflect and summarize what they have learned. At the end, learners summarize everything they learned.</td>
</tr>
<tr>
<td>Shared outcome</td>
<td>There are five roles: Next 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, and then learners split for the project work phase into two groups A and B around an assigned theme. Each group selects a person to be the coordinator, who aggregates the group’s work. After group work, students answer a questionnaire. If it is passed, then the activity is finished. Otherwise, the group has to re-work its project in order to find the correct answers to all the questions. The teacher has the option to support students when needed.</td>
</tr>
<tr>
<td>Blog collaboration</td>
<td>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.</td>
</tr>
</tbody>
</table>

Figure 1. Screenshot of IMS LD player SLeD used in the evaluation.
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. Clicking on a number displays this item. Only one item can be displayed at a time.

IV. RESULTS

The results are grouped by interaction type. Student–Other Human and Student–Other Nonhuman were excluded since none of the units of learning featured this type of interaction.

Student–Student The units of learning Deconstructivism, Shared outcome, and Blog collaboration 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.

Missing instructions for using the forum or for interacting within the forum was one reason that impeded student-student interactions in the observed units of learning. 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 always be clear to students, with whom they are to interact. The SLeD player did not offer such orientation.

The unit of learning 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 (as the person likely acted in both roles). The SLeD player did not offer explicit support for this change of roles. Switching roles was not as obvious because the SLeD 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. Acting in different roles could be supported at design-time if the unit of learning author provides instructions about switching roles.

Student–Teacher Student-teacher interactions were often realized using support activities in the units of learning. In the SLeD Player, this interaction was not well supported. Main problems occurred because the view offered to learners differed from the view offered to teachers: each role had their own view onto the unit of learning, only depicting those activities that were assigned to this very role. The units of learning Deconstructivism, Modern Architecture, Shared Outcome and Blog collaboration featured student-teacher interactions and thus encountered this problem.

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, i.e. to make clear that teachers and students were to work together at this point in time. This is a difference between the SLeD player (or other 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. At this point, only straightforward descriptions could help the roles to interact since the SLeD player does not offer technical support such as indicating the roles who are to interact with each other.

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 interface 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”. The meaning of support activity could not become clear to teachers as the SLeD player did not communicate specific possibilities for acting in a support activity. What appears to create further uncertainty is that support activities are by definition not specific to another activity but to another role. The IMS LD specification is unclear whether the role support via role-ref is only valid during an act, during a play, or always. This leaves room for interpretation and poses obstacles for student-teacher interaction within IMS LD VLEs.

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. We thus observed these two interactions 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 were sometimes external to the unit of learning such as the MACE portal [http://portal.mace-project.eu], which was referenced in the architecture-related units of learning. The MACE portal is a repository with architecture related content and can be considered an environment that provides manipulatives and tools.
In the SLeD 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 2, keeping the overview of the different layers inside the activity structure could be quite hard. The activity structure “Selection: Architect Tracks” contained three other activity structures: track Mies van der Rohe, track Gehry, and track Wright.

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 which were part of the regular sequence because they are nearly horizontally aligned (cp. Figure 3). 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” was 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.

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 2) 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 had to navigate through up to five levels. The student-environment interaction was greatly impeded through this complicated setup. Even though 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. Nested structures become hard to navigate if there are more than three levels in the navigation tree [8].

There was no indication in the SLeD Player when learners had actually finished the unit of learning. The only indication might be that no further activities appeared in the navigation tree. In order to support student-environment interactions, the runtime system should indicate to all participating roles when the end of the unit of learning was reached.

**Student–Information** The differentiation between the information and the environment was not easily drawn because the information is highly embedded in the environment when learning in virtual spaces. For the purposes of this analysis, we interpreted student-information interactions 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 SLeD Player. The borders were fluent in the observed units of learning because the true “content” to interact with was often contained in the learning objects. The related activity was a one-line activity description of the sort “Study [title of learning object]”. 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. Learners had to figure out that they had to click in the navigation tree (instead of somewhere in the main frame) to access the learning objects connected to the current activity.

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 (SLD) 2.0 [5], where there is
only one of the concepts included, i.e. the activity, and the concept environment is not present at all.

V. DISCUSSION AND CONCLUSIONS

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 manifest of the unit of learning. What information can or should be derived from this manifest? In this paper we have presented an analysis of this expression of interactions for learning in an IMS LD runtime environment, namely the Service Based Learning Design (SLeD) Player, and the interactions for learning were selected and adopted from Reigeluth and Moore’s [11] set of pedagogical aspects.

It was demonstrated that the expression and implementation of interactions of learners with human actors and non-human elements in the learning environment are currently far from optimal. Crucial issues encountered include:

(1) No explicit linkage between activity description (main window area) and environment objects (navigation area), unless the author provides a descriptive linkage, which would however contradict the basic idea of learning design by requiring knowledge of the runtime user interface at design-time. Thus, in-place access to information pertaining to the current activity would enhance student-information interactions, e.g. by integrating the activity description with the learning object more closely in the user interface.

(2) Uncertainty which role(s) the user is currently impersonating, and missing visual cues on supported and/or collaborating roles for the current activity. Roles should therefore be explicitly displayed in the user interface. Also the differentiation between units of learning a person was assigned to and the roles that a person acted in within the same unit of learning should be clearly distinguished rather than meshed into one single user interface element.

(3) Hierarchical nesting as the display metaphor of all unit of learning content/activities, which impedes easy navigation since there is little learning process-related information contained in such a structure. In order to allow better progression of learners through nested activity structures, a meaningful depiction that guides learners along the learning sequence would be helpful.

In light of these issues we recommend that developers of IMS LD VLEs take more care in observing the user perspective. Instead of exposing the XML hierarchy in the IMS manifest to the users, runtime environments could display units of learning in a way that lessens the cognitive loads currently needed for navigating and understanding how the different unit of learning parts relate. Thus far, many crucial elements that set IMS LD apart from other learning-technology specifications, such as roles and activities, are not expressed with their full potential in the observed VLE. Especially a lack of support regarding participant interactions can be observed. Currently, the responsibility lies with the unit-of-learning author to make instructions for interactions explicit. 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. Thus, a more meaningful interpretation and arrangement of IMS LD elements in VLEs is needed to improve interaction and learning support.

Future work requires to evaluate the expression of additional relevant pedagogical aspects and to take different IMS LD VLEs into account. A more thorough understanding about how pedagogy is expressed in units of learning at runtime can lead to increased user-centeredness and thus—in the long run—to wider adoption of IMS LD as an infrastructure for supporting learning.

Acknowledgments. This work was supported by the FP7 Support Action TEL-Map (257822) and the eContentplus Best Practice Network ICOPER (ECP 2007 EDU 417007).

REFERENCES