

The Conceptual Structure of IMS Learning Design Does Not Impede Its Use for Authoring

Michael Derntl, Susanne Neumann, David Griffiths, and Petra Oberhuemer

Abstract—IMS Learning Design (LD) is the only available interoperability specification in the area of technology enhanced learning that allows the definition and orchestration of complex activity flows and resource environments in a multirole setting. IMS LD has been available since 2003, and yet it has not been widely adopted either by practitioners or by institutions. Much current IMS LD research seems to accept the assumption that a key barrier to adoption is the specification's conceptual complexity impeding the authoring process. This paper presents an empirical study to test this assumption. Study participants were asked to transform a given textual design description into an IMS LD unit of learning using 1) paper snippets representing IMS LD elements and 2) authoring software. The results show that teachers with little or no previous IMS LD knowledge were able to solve a design task that required the use of all IMS LD elements at levels A and B. An additional finding is that the authoring software did not facilitate people in producing better solutions than those who used paper snippets. This evidence suggests that conceptual complexity does not impede effective IMS LD authoring, so the barriers to adoption appear to lie elsewhere.

Index Terms—E-learning standards, IMS Learning Design, authoring tools, computer-managed instruction.

1 INTRODUCTION

SEVERAL initiatives, organizations, and projects around the globe have been investigating the state-of-the-art and future developments of interoperability standards and specifications dealing with learning design issues in technology-enhanced learning, e.g., SynergiC3,¹ TENCCompetence,² and several projects funded in the *eContentplus* programme.³ In the area of learning activities a key specification is IMS Learning Design (LD) [1], [2], as it is the only existing interoperability specification which supports the definition and orchestration of teaching and learning activities involving multiple user roles and complex activity flows. Even though this specification has been available since 2003, its adoption by practitioners and institutions is poor and largely restricted to research projects and pilot implementations (see, e.g., [3], [4]).

IMS LD, like any other educational modeling language, needs to strike a balance between 1) high expressiveness to cover a broad spectrum of applications and 2) manageable complexity to be understandable by authors and users [2], [5]. As a modeling language, IMS LD is intended to provide

a means of defining a wide range of pedagogic strategies. Its performance in this regard has been evaluated and verified in [6]. This is largely confirmed by practical experience, as few reports have been published of designs which it is not possible to represent in IMS LD. We, therefore, take as our point of departure the assumption that a lack of expressivity is not a significant barrier to the adoption of IMS LD. If this is the case, then what factors can provide explanations for the lack of adoption?

The different ways in which IMS LD can be used are discussed by Griffiths and Liber [3], who identify four aspects to adoption of the specification, considered as a modeling language, interoperability specification, infrastructure, and methodology. In this paper, we focus on the first of these, seeking to separate it from the influence of the other factors, and to clarify if the characteristics of the specification lead to difficulties in using it for modeling learning activities. We recognize that there are uses of IMS LD which are purely technical, and which do not require any comprehension by the author of the modeling principles which underlie the specification. It is also clear that the specification documents themselves are highly technical and are intended primarily for the developers of authoring applications. The information model [1], which contains the conceptual structure, is lengthy and formal, and this is still more true of the XML binding [7]. Consequently, the specification itself is inaccessible for the typical designer of learning activities. However, we distinguish this complexity of presentation from the underlying conceptual complexity of the metamodel, which is the object of our investigation. The metamodel informs many of the existing authoring applications and runtime systems [8], and some critics—most notably [9]—have maintained that while the specification may be sufficiently expressive, the underlying metamodel is too conceptually complex to be comprehensible to authors, even when mediated by appropriate tools.

1. <http://www.synergi3.com>.
2. <http://www.tencompetence.org>.
3. <http://ec.europa.eu/econtentplus>.

- M. Derntl is with the Chair of Information Systems and Databases, RWTH Aachen University, Ahornstrasse 55, 52056 Aachen, Germany. E-mail: derntl@dbis.rwth-aachen.de.
- S. Neumann and P. Oberhuemer are with the Center for Teaching and Learning, University of Vienna, Porzellangasse 33a, 1090 Wien, Austria. E-mail: susanne.neumann-heyer@univie.ac.at, petra.oberhuemer@univie.ac.at.
- D. Griffiths is with the Institute for Educational Cybernetics, University of Bolton, Deane Road, Bolton BL3 5AB, United Kingdom. E-mail: d.e.griffiths@bolton.ac.uk.

Manuscript received 12 Jan. 2011; revised 25 July 2011; accepted 13 Aug. 2011; published online 18 Aug. 2011.

For information on obtaining reprints of this article, please send e-mail to: lt@computer.org, and reference IEEECS Log Number TLT-2011-01-0003. Digital Object Identifier no. 10.1109/TLT.2011.25.

If this were the case, then this would have major implications for the way in which IMS LD can be used.

IMS LD was not primarily intended to be read and understood by teachers. In fact, neither the specification [1] nor the introductory papers (e.g., [2]) pinpoint teachers—or any other particular group of people—as the “learning designers.” Nevertheless, we believe that it is desirable that teachers (or learning designers who have experience in teaching) should be able to engage with IMS LD authoring tools, so that they can participate in the authoring process. In the Learning Design Handbook [10], the chapters are organized according to a distinction of roles into “course developers” and “tool developers.” In most higher education institutions the teachers usually are in charge of both design and delivery of educational activities, hence we assume that the teachers are the largest group of (potential) course developers. The trend of developing educational modeling languages also shows that it is perceived as valuable if teachers express their pedagogical decisions using a structured format. IMS LD can provide such a format. We, thus, seek to answer the question: *Does the conceptual structure of IMS LD present a serious challenge for teachers to understand?*

The challenge in answering this question is to separate out the various potential causes of confusion and obscurity which could be experienced by authors. However, the interfaces of authoring applications introduce usability issues which are hard to quantify. It is easy to compare two applications and conclude that one is more usable or effective than the other [11], but it is hard to assess the degree to which they transparently provide access to the conceptual structure of IMS LD. Moreover, in investigating this question, we should recognize that authoring applications differ from each other not only in their usability and effectiveness, but also in the way in which they represent the metamodel. They may take the modeling task out of the teachers hands, e.g., by providing templates [12] or by hiding parts of the specification, or they may extend the metamodel with additional metaphors. There is no calibration available which can enable us to distinguish which aspects of modelers’ difficulties are due to usability issues in the tool implementation, and which are inherent in the metamodel of the specification. The study reported in this paper seeks to gain leverage on this problem by using two fundamentally different forms of interacting with the specification without concealing the underlying metamodel: a paper-based tool and a software-based tool.

The rest of the paper is structured as follows: in Section 2, we introduce the IMS LD specification. Sections 3 and 4 present the study methodology and results, respectively. Section 5 discusses the findings in the light of current research and concludes the paper.

2 IMS LEARNING DESIGN IN A NUTSHELL

2.1 Structure of the Specification

IMS LD is an interoperability specification that enables the exchange and execution of learning designs in computer-managed environments. The core idea is to model learning and teaching activities in a way which is both formal and

abstract, making it possible to use one learning design on numerous occasions, with different learning management systems, users, and tools [2]. The specification’s metamodel follows a stage-play metaphor. That is, the learning and teaching process is conceptually modeled as a play comprising a sequence of acts, with each act containing a number of role parts that connect the roles to the activities they perform and to resources they use.

IMS LD elements are divided into two parts: components and method. The components may be seen as the learning design ingredients, while the method part corresponds to the recipe which combines the components. The core components are:

- **Role.** A role expresses the function that a person carries out in a learning design. Example: learners could take the roles of moderator, group member, peer reviewer, or learner.
- **Activity.** Activities are used to express actions that learners or instructors perform during learning and teaching. Example: learners are to discuss a topic. For this, they engage in the activity called “Discuss topic XYZ,” which contains descriptions of how to carry out the discussion.
- **Activity structure.** Activity structures combine several activities in order to create a sequence or a selection. Example: learners can choose two out of four activities. The four activities up for choice are part of an activity structure of type selection. The number to select is 2.
- **Environment.** Environments are containers which hold learning materials (learning objects) and/or services (chat, forum, etc.). Example: when learners discuss topic XYZ, they need two articles, which cover the topic, and access to a chat application to exchange messages. The author of the learning design places these articles and a chat service in an environment, which is associated with the discussion activity.
- **Property.** Properties are “containers” to store different kinds of data, which can be displayed and updated/changed during the teaching/learning process. Example: learners take a test. The score that learners achieve is stored in a property. The test score property is attached to the instructor’s activity “Score learner tests.”

To weave instances of these components into an executable learning design, the following method elements are defined in IMS LD:

- **Play.** A play contains a sequence of acts. An IMS LD unit of learning must include at least one play. If it includes more than one play, then the plays are executed concurrently. Typically only one play is used.
- **Act.** Acts are used to create consecutive, self-contained phases. All activities in one act are finished before the next starts. Example: the first act includes all activities related to “Introduction to hydraulic engineering.” The second act includes activities regarding “Forces on dams.”

- **Role-part.** Role-parts connect a role to an activity, activity structure, or environment. Example: to indicate that the role “Learner” is performing the activity “Discuss topic XYZ,” a role-part is used to connect these two elements.
- **Condition.** A condition is an IF-THEN-ELSE statement that controls the visibility of elements such as activities, activity structures, and environments, as well as the updating of properties. Example: IF learners score less than 50 percent on the test, THEN they will see an environment that offers additional learning material, ELSE they will not see this environment.

IMS LD is divided into three levels: A, B, and C. Of the elements described above, property and condition are level B elements, while all the others are level A elements. Level C adds the concept of notification. Level C is not dealt with in this paper, because most learning designs can be described using elements of levels A and B. It may also be noted that many authoring and runtime tools do not support level C.

2.2 Tool Support for Authoring

IMS LD has repeatedly been criticized in the past for its conceptual complexity (e.g., [13], [9]) and problematic tool support (e.g., [14], [4], [9], [15], [11]). However, the frequently stated presumption that IMS LD’s complexity is one of the main barriers to its adoption stands without empirical support to date. To shed light onto this matter, this paper presents a study that was aimed at empirically investigating how well teachers as learning designers and facilitators understand the structure and the elements of IMS LD.

To date, IMS LD authoring software has not been integrated with virtual learning environments, a development which would enable teachers to construct learning sequences in an environment that they are used to and that does not confront them with IMS LD terminology. There are authoring tools in which the IMS LD specification’s complexity is reduced by hiding or disguising particular elements of the specification (e.g., in Graphical Learning Modeler [12] or in the ASK-LDT system [16]) or by providing a particular implementation or visualization metaphor on top of the specification’s stage-play metaphor (e.g., a design pattern-based approach in Collage [17]). Other examples include the MOT+ system [18], which predates the IMS LD specification, or the LAMS system [19], which are capable of generating valid IMS LD units of learning without originally being conceived as IMS LD authoring tools. These applications are out of scope for the present study, which required authoring tools which make full use of the concepts of the specification’s metamodel in their interface. Tools that fulfilled this requirement include Reload, ReCourse [8], CoSMoS [20], and the eXact Packager,⁴ whose interfaces expose all supported IMS LD concepts to the user without simplifying or disguising the concepts.

3 METHODOLOGY

3.1 Participants

To test how teachers understand and work with IMS LD elements, they solved a given learning design task.

4. <http://exact-learning.com/en/products/learn-exact-suite/exact-packager-scorm-compliant-content-authoring>.

University teachers were recruited to participate. The study was performed in four workshops with participants from over 10 different European countries. Two of these workshops were organized as *paper*-based workshops, where participants ($N_{paper} = 23$) used paper snippets representing the elements of IMS LD to solve the design task; the other two workshops were organized as *software*-based workshops, where participants ($N_{software} = 17$) used IMS LD authoring software to solve the same task. The participants’ background data were collected through a survey at the beginning of the workshops. These data included: years of teaching experience, subject area of expertise, level of teaching (primary, secondary, higher education), and previous experience with IMS LD.

3.2 Workshop Setup

The four workshops followed a similar procedure, the only difference being in the authoring tools introduced and used for the design task. The objective of the study was to test participants’ understanding of the conceptual structure of IMS LD. To be able to control bias introduced by tool usability issues, two different tools were deployed: in two workshops the authoring tool was a set of paper snippets, while the two other workshops used IMS LD authoring software.

IMS LD introduction. Each workshop started with a demonstration of IMS LD. The objective of the demonstration was to acquaint participants with IMS LD at levels A and B and to provide guidance for the subsequent hands-on design task. The demonstration took the form of a 45-minute presentation with slides including 1) a brief general introduction to IMS LD, 2) a demonstration of IMS LD components and method elements based on a simple, one-role example that required the use of all IMS LD levels A and B elements, and 3) guidelines for using the authoring tool (paper or software) in the design task.

Learning design task. After the demonstration, every participant had the task of representing the given learning scenario with IMS LD elements without any assistance. The scenario was more complex than the example shown during the demonstration to ensure that participants had to perform more than a simple transfer of identical concepts from the example to the task. The scenario was handed out as follows:

The following scenario is an online learning scenario. The instructor gives a presentation to learners via a presentation conferencing service about coal-burning power plants as well as their positive and negative side effects. After the presentation, learners can choose to do one of the two following activities:

1. *set up controversial questions regarding coal-burning power plants (to be used during a later discussion round), or*
2. *collect credible sources on the World Wide Web regarding new developments in coal-burning power plants (to be used during a later discussion round).*

The outputs learners created in the previous activities will be used during the subsequent discussion activity: the learners engage together with the instructor in the discussion using the questions and materials. After the discussion, learners rate their own agreement with the statement “Coal-burning power plants have a future in Europe’s energy production” (scale of 5, where 5 is the highest agreement). Learners, who rated themselves with a level of at least 3, will next do an activity involving the creation of a

<input type="radio"/> LEARNING ACTIVITY <input type="radio"/> SUPPORT ACTIVITY			ACTIVITY		
TITLE:					
Description:					
Properties to display:		Properties to change/update:		Environments to display:	
[]		[]		[]	
ID:					
ENVIRONMENT					
TITLE:					
Learning objects:			Services:		
•			<input type="checkbox"/> Chat		
•			<input type="checkbox"/> Forum		
•			<input type="checkbox"/> Mail		
•			<input type="checkbox"/> Announcement		
ID:					
ACTIVITY STRUCTURE					
TITLE:					
TYPE: <input type="radio"/> Sequence (activities to be executed one after another)					
<input type="radio"/> Selection - How many activities must be selected: []					
ACTIVITIES: (separate referenced activity IDs with commas)					
[] IDs					
ID:					
PROPERTY					
TITLE:					
Description:					
Initial value:					
ID:			ROLE-PART		
TITLE:			Title:		
ROLE:			<input type="radio"/> ACTIVITY: []		
			<input type="radio"/> ACTIVITY STRUCTURE: []		
			<input type="radio"/> ENVIRONMENT: []		
ID:					
<input type="radio"/> STAFF					
<input type="radio"/> LEARNER					
ROLE					
TITLE:					
[]					

Fig. 1. Paper snippets representing the IMS LD components and method elements.

(digital) poster on one positive aspect of coal-burning power plants. Learners, who rated themselves at a level of 2 or lower, will next create a (digital) poster on one negative aspect of coal-burning power plants.

Paper-snippets introduction (paper-based workshops only). Building on the IMS LD information model, we identified the key elements which need to be assimilated in order to build a unit of learning. In the two paper-based workshops every participant received an envelope with paper snippets (see Fig. 1), each representing one instance of an IMS LD levels A or B element. Each paper snippet was divided into boxes which represent information about the element in conformance with the specification. The snippets with their information boxes were designed as an immediate, slightly simplified representation of the IMS LD elements as specified in the IMS LD information model [1]. The following simplifications were applied:

- The concept of play was not considered to avoid unnecessary overhead. In the example and for the design task, only one play was needed (as in most units of learning).
- Only one type of property was offered, to be used as needed. In IMS LD, properties can be used to store data for persons and roles either locally (i.e., only for the current running instance of the unit of learning) or globally (across different units of learning). This distinction was not required for the design task in this study.
- There were no self-contained snippets for activity descriptions, learning objects, services, and other concepts. These concepts were represented as information within the containing paper snippets (see, e.g., the learning object bullet-list within the environment snippet in Fig. 1). In the IMS LD information model, these concepts are represented as elements that need to be referenced by the containing elements. This simplification was made to keep the number of distinct paper snippets within reasonable bounds.

Each element had its own unique color. In order to guide participants in placing connections between elements (e.g., link an environment to an activity), the reference box of the

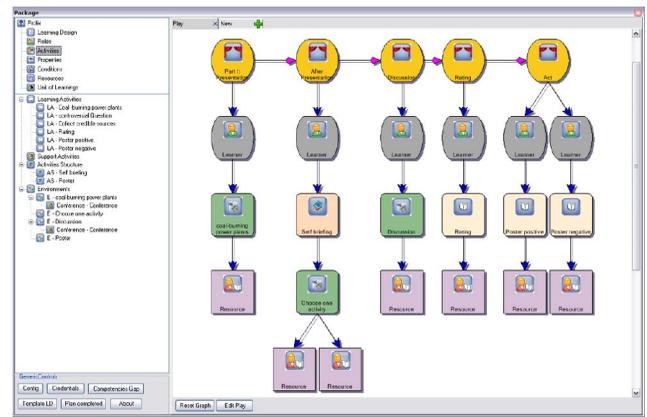


Fig. 2. Screenshot of the IMS LD authoring plugin for the eXact packager software.

snippet was colored with the color that represents the target element (e.g., the “environments to display” box on the activity snippet is colored with the signature color of the environment snippet, since it should reference an environment’s ID). Each act was represented as a letter-size sheet of paper. The assignment of a role-part to a particular act was achieved by sticking a role-part snippet onto the sheet representing the act with provided paper glue or tape. Participants had to number the acts consecutively. Conditions were also to be glued onto the act sheets to simplify the workshop setup, although we are aware that in IMS LD conditions are captured as part of the IMS LD method.

Authoring-software introduction (software-based workshops only). In the two software-based workshops, participants were individually given access to either a workstation computer or a laptop. On each machine, a version of the eXact packager software was installed. This software features a plug-in that is capable of authoring and exporting IMS LD compliant units of learning (see screenshot in Fig. 2). Since the conceptual structure of the IMS LD information model was the object of scrutiny in this study, we needed to deploy authoring software that does not conceal any portion of the specification. The eXact packager software was chosen for the workshops because it allows direct, unconcealed access to the entire IMS LD specification. We deemed the eXact packager to be a reasonable choice, as this tool strictly follows the naming conventions and structures defined in the IMS LD information model and does not add any additional user interface metaphors to the authoring process. It graphically depicts the basic IMS LD elements that correspond with the paper snippets used in the paper-based workshops. Besides, the eXact packager software has a similar order of complexity to other authoring applications that represent all IMS LD elements in the user interface (e.g., ReCourse).

Task protocol. Participants were not offered any help or guidance during the task other than 1) provision of a cheat sheet with a tabular overview of IMS LD elements and 2) personal answers to questions for clarification. They were asked to keep a task protocol by writing down any issue, problem or question they encountered during the task on a protocol sheet. They were also asked to indicate for each identified issue whether or not they were able to solve this issue on their own.

Post-task survey. After the design task was completed participants were asked to fill out a survey, which aimed to collect additional information about what gave them the most trouble during the task.

3.3 Data Analysis

Rating the solutions. Two IMS LD experts created a prototype solution for the design task by decomposing the overall task into solution items. Each solution item consisted of an action that needed to be performed with an individual IMS LD element, or a small group of elements, in order to obtain a correct solution. The output of this process was a checklist of 81 items. For example, the checklist for the activity structure where material for the discussion is to be collected includes the action items: create activity structure, define type (selection), and link activities (reference two activities to choose from). To test the checklist, the two experts independently analyzed the participants' solutions from the paper-based workshops and matched them with the prototype solution by assigning a correct/incorrect flag for each action item on the checklist. During this process, 2 of the 23 paper-based solutions had to be discarded, because they represented a self-defined design and not the one given for the design task. Hence, for the remaining 21 paper-based solutions with 81 checklist items each, the experts had to make a total of 1,701 judgments each. The two experts independently agreed on 1,617 (95 percent) of these judgments. The inter-rater reliability of judgments as calculated using Cohen's Kappa is extraordinarily high ($\kappa = 0.87$, $p < 0.001$). The 84 judgments, on which the experts had disagreed, were reconciled. Based on this procedure, a solution analysis model was conceived, which led to the discarding of 21 of the original 81 checklist items.

Using the consolidated checklist of 60 items, this procedure was then applied to the software-based solutions. Two experts again independently applied the checklist to the software-based solutions ($\kappa = 0.86$, $p < 0.001$). The judgments on which the raters had disagreed were again reconciled in a meeting. Of the 17 solutions, 3 had to be discarded because either the participants failed to properly save their designs in the software or the experts were unable to open the solutions using the authoring software.

Data samples. The resulting set of 21 paper-based and 14 software-based solutions ($N = 35$) represents the sample with which the data analyses reported in Section 4.2 were processed. For the data analyses reported in the section on process-related issues (Section 4.3), the original sample of 40 participants was considered, since people who failed to produce a valid solution still went through the design process and responded to questions in the surveys.

Solution analysis model. The resulting solution checklist data were analyzed using the layered model depicted in Fig. 3. The left-most (lowest) layer comprises the IMS LD element instances required for completing the task. These element instances are connected with general actions to be performed with IMS LD elements at the IMS LD Action Layer. Essentially, each connection between these two layers refers to one item on the solution checklist. Each action on the IMS LD Action Layer was assigned a level (A or B), and 18 of the 20 actions were connected with their

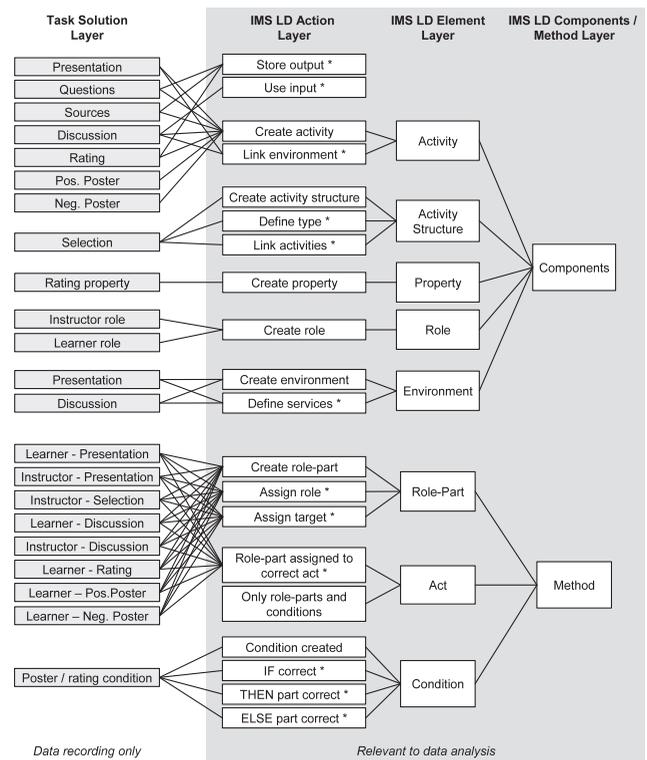


Fig. 3. Layers of data analysis and connections between the layers (solid lines). Dependent actions are marked with an asterisk (*); these could only be performed when an underlying element was created.

corresponding IMS LD elements at the IMS LD Element Layer. While most of these connections are trivial, only the use of input to and output from activities were not linked to one specific element, since the linkage can—depending on the context of the action—relate to activities, environments, or properties.

The actions on the IMS LD Action Layer were also used to identify the performance of participants on IMS LD levels A and B. The elements on the IMS LD Element Layer were used to analyze the performance of participants in regard to the different IMS LD elements. Using this data it was also possible to separately analyze the correct use of component elements and method elements. Finally, the actions at the IMS LD Action Layer were used to calculate the overall correctness of participants' solutions.

To compute a "score" for each connection between the layers, we proceeded as follows: on the Task Solution Layer, we recorded for each participant and each element instance whether a connected action from the IMS LD Action Layer was performed successfully or not ($0 = fail$, $1 = success$). From this data, an average score was computed for each connection between Task Solution Layer and IMS LD Action Layer across all participants. This score indicates the share of participants who were able to perform an action for a given element instance; the score could also be interpreted as the probability of one single participant successfully performing a given action with a given element instance. Note that some actions depend on others. For example, a role can only be assigned to a role-part, if the role-part was created first. These dependent actions (marked with an asterisk in Fig. 3) were only included in



Fig. 4 Word cloud of teaching backgrounds.

the analysis of a participant’s solution (and therefore in the calculation of the score) if the participant had created the underlying element.

When computing the score for the higher layers in the model (i.e., moving to the right in Fig. 3), for each item of the higher layer the scores of all connections with items from the lower layer were arithmetically averaged. For instance, at the IMS LD Action Layer, the scores for the two items “Create environment” and “Define services” are averaged to obtain a score for the item Environment at the IMS LD Element Layer. That is, a score for any item in the gray-shaded area labeled “Relevant to data analysis” in Fig. 3 represents how likely one participant was to successfully complete the item. In the following results section, we, therefore, refer to all scores computed this way as the “performance” of a participant, or “conformity” with the correct prototype solution, on a given layer. The scores are within the interval [0, 1], and in the figures and in the text we express them as percentage values.

4 RESULTS

4.1 Participants

The total sample consists of $N = 40$ participants. The average teaching experience is 7.2 years, so participants may generally be characterized as experienced teachers. The range of subject areas in which participants were teaching was quite diverse. However, most participants had a rather technical background: 26 of the 40 participants provided one or more subject areas with a technology focus, e.g., computer science, information (communication) technology, or engineering—see a word cloud of the provided teaching areas in Fig. 4. The vast majority (93 percent) of participants were teaching in the higher education sector, with a few additional nominations of secondary, primary, and other educational sectors (e.g., vocational training). Only 7 of the 40 participants (18 percent) had previous experience with IMS LD authoring tools, with only five of them having seen an IMS LD unit running in a player. That is, more than four out of five participants had never “done” anything with IMS LD.

4.2 Learning Design Solutions

This section presents quantitative data analysis results and figures according to the layered data analysis structure presented in Fig. 3.

4.2.1 IMS LD Action Layer

The results of the IMS LD Action Layer are captured in Fig. 5. For each action in Fig. 5, there are two bars and three percentage values: the bar labeled “Paper Snippets” represents the percentage of conformity of the action as

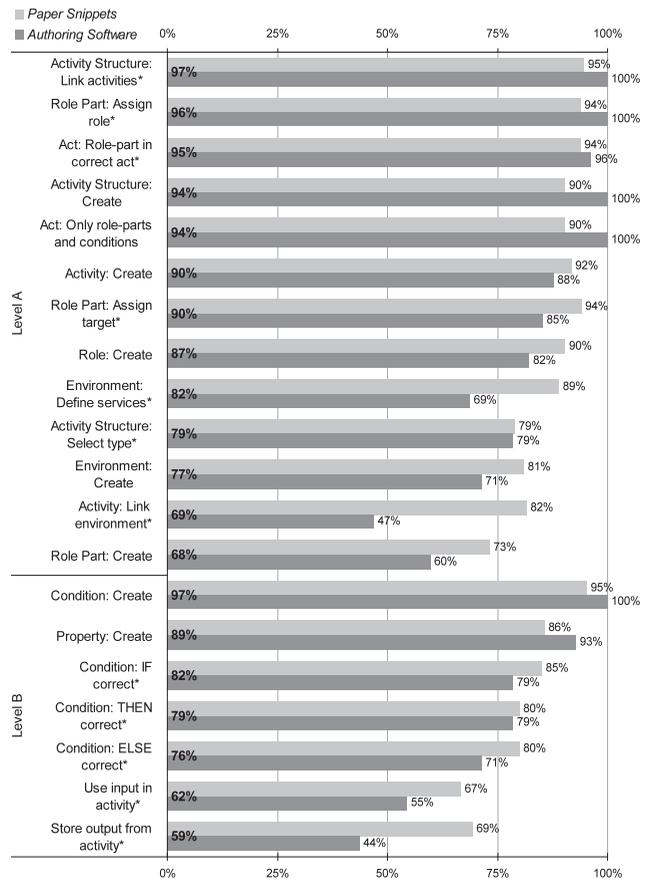


Fig. 5. Participants’ solutions’ conformity with the experts’ prototype solution at the IMS LD Action Layer. Legend: The value 92 percent for the action “Activity: Create” in the paper-based workshops means that on average each activity snippet required for the correct solution was created by 92 percent of the participants. Dependent actions are marked with an asterisk (*); these actions could only be performed when the underlying element was created. For instance, the value of 85 percent for the action “Role Part: Assign target” in the software-based workshops means that for each role-part required for the correct solution an average of 85 percent of the participants, who actually created the underlying role-part, managed to assign a target to the role-part.

compared with the prototype solution for participants’ solutions in the paper-based workshops. In the following presentation of results, we will refer to this sample as the “paper-based sample.” Analogously, data from the participants of the software-based workshops are referred to as the “software-based sample.” The percentages of conformity for these samples are displayed next to the bars. The large-font, bold-faced percentage value next to the vertical axis represents the conformity score in the total sample (paper- and software-based participants).

Creation of elements. The results on the IMS LD Action Layer show that participants’ solutions had a generally high conformity with the prototype solution. The percentage of conformity in the total sample varies between 97 and 59 percent, whereby only 4 out of 20 actions were rated at less than 70 percent conformity. The actions “[Element]: Create” rank, on average, among the highest actions that were successfully resolved. Participants were, thus, able to perform the most essential step of unit-of-learning authoring: they could recognize when an element needed to be created to represent a portion of the learning scenario,

and for most of the elements they could adequately define this element with necessary parameters or descriptions.

Role-parts. Role-parts were the elements whose creation caused participants the most trouble of all IMS LD elements. Several factors may contribute to this. First, a role-part cannot be created and placed into the (correct) act until activities and roles have been created. It is, therefore, reasonable to assume that paper-based participants dealt with role-parts in the final phase of authoring, and that some of them may already have run into time constraints. Software-based participants seemingly struggled with usability issues regarding the depiction of role-parts in the software's interface. Also, role-parts were complex elements since each role-part needs to be connected to one role and one target activity, activity structure, or environment. From a cognitive viewpoint, this was probably the most demanding element. Additionally, participants did not always understand the need to combine the role and the activity in yet another concept "role-part"; to them it seemed sufficient to have roles and activities. However, once created, it was clear that participants were easily able to assign a target (90 percent average conformity) and a role (96 percent average conformity), and put the role-part into the correct act (95 percent average conformity).

Differences. Some actions exhibited a considerable difference in conformity scores between the paper-based and the software-based sample. We focus our discussion on actions where this is evident with a score difference of more than 10 percentage points between the two samples:

- *Environment: Define services.* The paper-based workshop participants outperformed the software-based participants for defining the services to be used in an environment. The services predefined in the IMS LD specification could be checked both on the paper snippets and in the authoring software. However, the paper snippets were less restrictive in the selection of services than the authoring software, because participants could simply write down any additional service which they deemed useful in an environment, which is not as easy in the authoring software.
- *Activity: Link environment.* Software-based participants performed considerably worse when linking activities with environments, particularly when activities were part of an activity structure. One problem was that once an activity was assigned to an activity structure, it was no longer possible to use drag-and-drop to assign it to an environment. Instead they had to go via six mouse clicks to the dialog where an environment could be assigned. In the paper snippets, the participants would simply write the linked environment's identifier on the activity snippet.
- *Role-Part: Create.* Paper-based participants outperformed software-based participants for the creation of role-parts. The authoring software automatically depicted one role-part symbol for each act in the graphical workspace. An additional role-part could be added by right clicking on the act symbol, but participants may not have remembered how to do this, or may not have recognized that another

role-part was needed. Also in the software it was hard to visually distinguish whether a role-part had already been assigned a role.

- *Use input in activity and Store output from activity.* The software-based participants performed worse than the paper-based participants with these actions. A potential reason is that it was possible to annotate the paper snippets with arbitrary hand-written text. During the analysis of solutions (both paper-based and software-based) the experts agreed to give a positive judgment to solutions where it was clear that the participant intended to store input to or use output from an activity. In the authoring software, annotations are naturally not possible to include, and the "workaround" solutions (using an environment with a learning object to hold the output from an activity instead of properties) require more effort in the software than on paper.

A general explanation of the slightly better performance of paper-based participants for the majority of actions could be that the paper-based workshop offered lower entry barriers since paper is a highly accessible and familiar tool. Moreover, the paper-based setup offered more space for personalizing the tool: participants could conceive their own ways of arranging and rearranging the snippets on their tables, while the authoring software came with one predefined user interface that could not be changed.

Actions on levels A and B. Averaging the actions' scores for IMS LD levels A and B yields that there are 8 percentage points difference between conformity scores of actions at level A (86 percent average conformity) and level B (78 percent average conformity). A significant share of this difference is accounted for by the low conformity scores of the level B actions related to activities, namely considering input to activities (62 percent average conformity) and output from activities (59 percent average conformity) using properties. In the task scenario this was the case, e.g., when students were to produce materials to be used during a subsequent discussion activity as demonstrated in the task description. Other than that the use of level B elements added little to the task's complexity.

4.2.2 IMS LD Element Layer

Moving on to the IMS LD Element Layer, we averaged the conformity scores of all actions at the IMS LD Action Layer that map to a particular element. This way, we obtained data on participants' success to use IMS LD elements (see Fig. 6). All percentages are at a rather high level, with the most "difficult" elements being activities and environments with conformity scores of 80 and 79 percent, respectively. It, therefore, seems valid to state that participants performed well with all levels A and B elements. Interestingly, the "easiest" structural element of IMS LD appears to be the act (94 percent average conformity). Moreover, since the act as the primary structuring element of the method part was understood well by almost all workshop participants, it seems that the stage-play metaphor of IMS LD does not present a real obstacle to the orchestration of roles and activities.

While there is not much difference between the paper-based and software-based samples for the top six elements (max. 8 percentage points difference, cf. Fig. 6), the

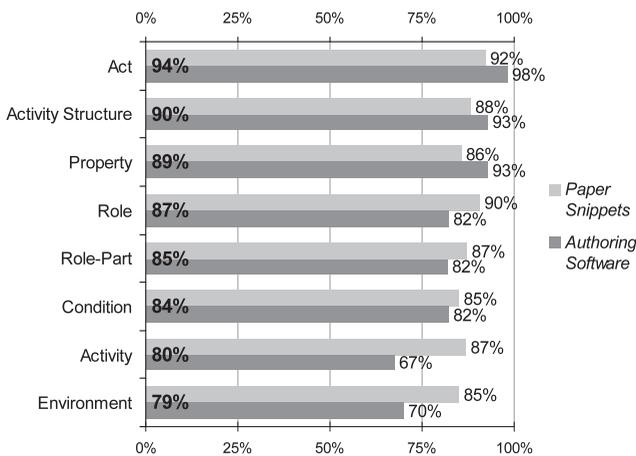


Fig. 6. Performance on the IMS LD Element Layer.

differences for activities and environments are considerably higher at 20 and 15 percentage points, respectively. Responsibility for the low activity-related performance of the software-based participants can be attributed to the action “Activity: Link environment,” whose outlying low value of 47 percent average conformity was discussed above.

4.2.3 IMS LD Components/Method Layer

IMS LD conceptually distinguishes elements belonging to the components part (i.e., activity, activity structure, environment, role, and property) or to the method part (i.e., role-part, condition, and act). While the component elements are frequently used concepts in several approaches to educational modeling and design, the method section is particular to IMS LD since it follows the metaphor of stage plays, where actors perform their role-parts in several consecutive acts. The average conformity scores for elements in the components and method sections were almost identical (85 and 88 percent, respectively), thus providing evidence that the stage-play metaphor of IMS LD does not pose any significant problems to users.

4.2.4 Overall Quality of Solutions

The average overall conformity of participants’ solutions with the experts’ prototype solution was computed using data from the IMS LD Action Layer for the following two reasons: 1) the data on the IMS LD Element Layer was abstracted from the actions that had to be performed and may thus under/over-represent particular elements and actions; and 2) Using the checklist on the Task Solution Layer would assign more weight to those element types that occurred more frequently in the solution (e.g., since there were two environments and eight role-parts in the solution, the weight for creating role-parts would be 4 times the weight of environments). As an additional measure, all checklist data for dependent actions (i.e., actions that prerequire another action to be performed) were included in this calculation. This was considered necessary, since otherwise those participants who created an element but failed with the element’s dependent actions could be “punished” more than participants who failed to create the element at all.

As a result, the average overall match of participants’ solutions to the prototype solution based on correctly performed actions at the IMS LD Action Layer was 75.6 percent in the total sample ($N_{total} = 35$; $s.d. = 17.2$ percentage points), 78.1 percent in the paper-based sample ($N_{paper} = 21$; $s.d. = 20.2$ percentage points), and 71.9 percent in the software-based sample ($N_{software} = 14$; $s.d. = 11.1$ percentage points). These values indicate that participants were on average able to perform three quarters of the actions required for the correct solution. Given the tight time-frame of the workshops, the generally low previous knowledge of IMS LD, and the tools that had to be assimilated, this appears to be a highly satisfactory value.

An independent-samples *t*-test shows that the difference between paper-based and software-based results is not statistically significant ($t = 1.045$, $df = 33$, $p = 0.303$). That is, the visual GUI of the authoring tool did not guide the users in a way that would enable them to create better solutions than participants who had to figure out how to correctly assemble the paper snippets without any guidance.

4.3 Learning Design Process

In this section, we focus on difficulties that participants encountered in their learning design process. Some of the difficulties were obtained when studying participants’ learning design solutions, while others were derived from participants’ task protocol and the post-task survey including participants’ top 3 reported troubles in solving the task.

Joint activities and role-parts. As the solution-related results in the previous section showed, participants were less successful at creating role-parts correctly. This was especially true for activities, which were jointly performed by two roles. In the task, this was the case for the presentation activity (instructor presents while students listen during the presentation) and the discussion activity (the instructor and students discuss together). The anticipated way to express a scenario with two roles joining in an activity is to create two role-parts (one for each role) within the same act that both link to the same activity or environment. Often, participants created just one of the role-parts.

The overall average for completing the task “Role-part: Create” was 68 percent. Two role-parts accounted for a large share of these difficulties: the instructor’s role-part in the discussion activity (only 49 percent of the participants created this role-part) as well as the learners’ role-part in the presentation activity (only 43 percent of the participants created this role-part). The most coherent explanation of these difficulties is that participants were tempted to link only the “dominant” roles to these activities, i.e., the instructor with the presentation, and the learners with the discussion, respectively.

Other role-parts that caused problems were the role-parts for learners’ creation of a positive poster (57 percent created this role-part) and a negative poster (49 percent created this role-part). This was a complex part of the task, since participants had to create the poster-creating activities and use a condition to control the visibility of these activities based on a property-value set during a previous rating activity. While participants were highly successful in solving the associated challenges in this setup (91 percent created both poster-creation activities, 89 percent created

the property, 98 percent created the condition, and 82 percent referenced the property in the condition) they may have simply forgotten to create the needed role-parts for these activities. Another explanation could be that participants viewed the condition as being responsible for showing the activities, and therefore did not deem it necessary to create role-parts for the poster creation activities. Some participants created the activities and additionally set up an activity structure of type selection referencing both poster-creation activities. Seemingly, they did not trust that the condition would automatically arrange the display.

Activity input and output. One of the more difficult actions for participants was to store the product or output of an activity. In the task learning scenario, learners were supposed to collect credible sources on the web or set up controversial discussion questions. These items were then to be used during a later discussion activity. The difficulty for the participants was to decide how to store and reuse the learners' output. Thirty-three of the 35 participants managed to create both output-generating activities. Of those, 22 participants stored the outputs of the prediscussion activities. Some participants used properties to store the output, while some used dedicated environments (e.g., for collecting the questions in a forum) to do so. Only a portion of them finally managed to link the property or environment as input to the discussion activity. Although properties are seen as the correct way of expressing this solution in IMS LD, we regarded both ways as correct, since the technical constraints on properties and environments—a highly complex part of the specification—were not covered in all detail in the introductory demonstration. The participants were obviously quite resourceful in finding a reasonable solution to this problem.

Referencing elements. The link between activities and environments caused uncertainty in the learning design process. Participants of the paper-based workshop correctly linked environments to activities as the analysis of solutions showed (82 percent conformity score for this action). However, the same environment was frequently linked twice: participants referenced an environment in an activity on the activity snippet (correct action), but then linked the activity and again the environment to the role within the role-part (incorrect action). The same pattern, though not as frequent, appeared with activities and the activity structure. Some paper-based participants correctly set up the activity structure and referenced the corresponding activities, but when constructing the role-part they repeated the links to the activities which were part of the activity structure. For users of the authoring software, this was not an issue as the software allowed only one role and one target element to be connected to the role-part.

Task protocol. On average, each paper-based participant reported two issues, while each software-based participant reported nearly six issues. Fig. 7 shows the results of recorded issues, whereby issues were clustered in order to identify common themes in issue reporting. Clustering was performed whenever an item was mentioned more than once. Issue reports that only appeared once in all protocols, such as "I am unsure when to use activities and when to use

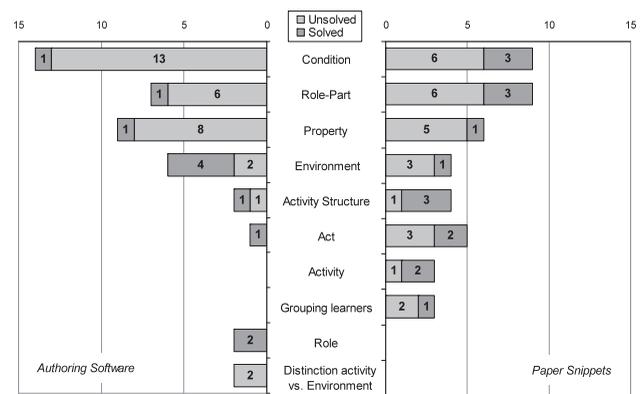


Fig. 7. Reported issues and problems encountered while solving the task.

activity structures" were omitted. Participants did not put down for every reported issue whether they were able to solve it or not. Therefore, for all nonmarked issues, we assume that these are unresolved issues. Note that about one-third of all reported issues related to usability matters with the software's interface. We have omitted usability issues in this figure, because in this study we were investigating participants' understanding of IMS LD rather than the usability issues of a specific authoring tool.

The most frequently reported issue was the condition element of IMS LD (23 nominations). Comments majorly focused on the connection between conditions and the other IMS LD elements. Specifically, participants' questions were:

- *When are conditions evaluated?* The IMS LD specification states that conditions must be evaluated "when entering the run of a unit of learning (new session); every time when the value of a property has been changed" [1]. Participants were not instructed about this.
- *To which other elements can I attribute conditions?* This question expresses uncertainty how conditions relate to other IMS LD elements. First, conditions are stored as self-contained elements in the IMS LD method. Second, conditions are not attributed but monitor other elements, most often properties. The introduction did not show the entirety of options for conditions.

Paper-based participants were faced with a simplified version of conditions: they could formulate the IF statement in the conditions with free text, and they only used a limited set of options for the THEN statement. Software-based participants were faced with the entirety of options available for setting conditions: they reported in their protocols that they were at times overwhelmed with the choices offered for defining conditions. Software participants thus reported conditions more frequently.

Role-parts received the second greatest number of nominations. Paper-based participants had most trouble with the strict matching of a role to either an activity, environment, or activity structure within a role-part. Paper-based participants reported that they were unsure whether to link the activity and the environment over again using role-parts even if the environment was already linked to the

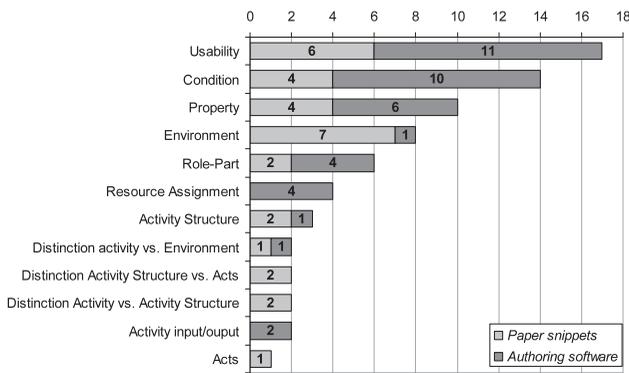


Fig. 8. Difficulties reported by participants.

activity using the “Environment(s) to display” referencing box. Software-based participants did not face this problem as the software automatically restricted the assignment within role-parts. Software-based participants, however, were most concerned with creating two role-parts when two roles jointly perform an activity. They were uncertain whether creating two role-parts was the correct way to proceed. They were further confused because if they created two role-parts that linked two roles to the same activity, the software depicted the mutually performed activity twice in the workspace. Software-based participants saw this as illogical, thinking that an activity should only appear once, even if it was assigned to a number of roles. This indicates that software-based participants did not understand the difference between an activity and the reference of that activity in the user interface.

Properties were the third most frequently reported element in the issues protocol. Paper-based and software-based participants alike reported that they were unsure whether they correctly understood this element. Specifically, they pondered whether the property is the appropriate element to use when storing the output of activities. The combination of properties with, for instance, activities caused confusion as well. Even though paper-based participants included references to properties in their activity snippets (either for updating/changing or for displaying properties), they did not see a clear connection between properties and activities. Seeing this connection was even harder for software-based participants since they had no guidance for connecting properties with activities (or other IMS LD elements); they were required to manually create a file containing the activity description and placing a remark inside that a property is used in one way or another within this activity (they were instructed by the workshop leader to do this). The use of properties in learning designs can be identified as a technical aspect that is hard to understand during authoring.

Greatest perceived difficulty. When asked to name three things about the task that gave them the most trouble, 35 percent of the paper-based participants and 70 percent of the software-based participants used this chance to actually name three specific difficulties (cf. Fig. 8), which may indicate that software-based participants perceived the task as more troublesome than paper-based participants.

Paper-based participants reported difficulties with the usability of the paper snippets (6 nominations), for

instance, “absence of a use guide,” “definition and layout unknown,” or “too many elements to understand for the first time.” Regarding IMS LD elements, the concepts environment (7 nominations), property (4 nominations), and condition (4 nominations) caused the most trouble, which is consistent with the analysis of actions and task protocols in previous sections. For software-based participants, software usability was the most frequently mentioned factor (11 nominations). Regarding IMS LD elements, conditions represented the concept causing the most trouble (10 nominations), followed by properties (6 nominations) as well as role-parts and resource assignment (4 nominations each).

The differences in troubles reported with IMS LD elements can be largely attributed to the tool that participants used. Conditions and properties were represented as simplified concepts for paper-based participants. Paper-based participants saw these concepts as problematic, but only every sixth paper-based participant reported this. In contrast almost two out of three software-based participants reported having difficulty with conditions, and one out of three reported properties as a troublesome concept. Software-based participants faced conditions and properties in all their complexity with all options. This had a strong influence on their perception of managing these concepts within the task they worked on. In addition, paper-based participants never asked when conditions would actually be evaluated during runtime. This is probably because they were told to put conditions into the act where the condition was relevant. Software-based participants saw conditions as disjointed from the rest of the design, and the point of evaluation of conditions was their most reported difficulty concerning conditions.

While participants easily understood the meaning of an IMS LD activity, the concept of an environment posed problems, particularly for paper-based participants who frequently reported that it was not clear to them. Paper-based participants wondered about the distinction between activities and environments, or the role of the environment as an “add-on” to activities. Software-based participants did not frequently report this difficulty. It is possible that the software is a better agent in guiding the user in understanding for what purposes the environment is used.

Apparently, the graphical depiction in the software gave software-based participants a greater chance to distinguish between the purposes of acts, activities, and activity structures. Paper-based participants reported problems in distinguishing these concepts whereas software-based participants did not perceive these troubles.

5 DISCUSSION AND CONCLUSION

5.1 Limitations of the Study

Before discussing the findings and their impact on future work related to IMS LD, some words on limitations inherent in the study design are due. In order to manage the available workshop time, the task scenario was kept moderate in terms of number and complexity of activities and was situated in a purely online context. Also, the demonstration of IMS LD at the beginning of the workshops was targeted at

enabling participants to solve the given task using the paper snippets or authoring software. Most of the actions required in the task were demonstrated. Actions not required in the task (e.g., employing several plays in a learning design, or employing a monitor service) were not demonstrated.

Another factor of potentially positive bias in this study could be that most participants had a high technology affinity regarding their teaching backgrounds. Although the tasks carried out were not technological and most of the concepts used in IMS LD are also not technological (e.g., environment, activity structure, act...), a technical background may have helped participants in understanding some details (for example in understanding the nature of properties). However, since IMS LD is meant for technology-enhanced learning and teaching, the workshop participants may in fact be representative of a typical audience that implements e-learning.

Compared to the IMS LD information model, the paper snippets representing the IMS LD elements were simplified in terms of structure (e.g., only one instead of five types of properties) and information content (e.g., textual instead of hierarchical IF-statements in conditions, or omitting to represent learning objects as paper snippets). This simplification was deemed as appropriate in seeking to understand comprehension of the basic structure underlying the specification.

Last but not least, the study included a particular set of paper snippets and a particular piece of authoring software, respectively. While the decision to use these two highly different “authoring tools” was taken to level out the bias stemming from tool usability issues, it is hard to estimate the effect of the provided tools and their user interface abstractions on the results. To minimize potential effects, we chose an authoring software whose user interface resembles the elements and structure of the IMS LD specification without simplifications or interaction metaphors.

5.2 Summary of Findings

In the introduction to this paper, the following question was stated: “Does the conceptual structure of IMS LD present a serious challenge for teachers to understand?” Based on the results in the empirical analyses this question can be answered with *no*. Participants performed well on the components and the method sections of their solutions, and they performed very well on levels A and B. They achieved a conformity of more than 75 percent with the correct prototype solution for 16 out of 20 distinct actions required to solve the design task.

From the analysis of the task solutions created and the issues and troubles reported by participants, the conceptual structures in the IMS LD specification that presented problems to teachers’ understanding were:

- **Linking of elements.** The use of role-parts to link roles with activities, activity structures and environments caused difficulties in deciding when to set up a role-part, especially in relation to joint activities and in relation to conditions controlling the visibility of activities. Another problem was the linking of an environment to an activity, particularly for the software-based participants.

- **Properties and document flow.** The handling of the “document flow” between activities, so that output from activities (user-generated content) is reused in other activities posed a problem for participants. The authoring software did not do a good job of supporting teachers in this endeavor; there are too many elements to connect and steps to take (and during which to get lost). This is a well-known IMS LD issue [13].
- **Conditions.** Participants reported conditions as the most difficult element. They had trouble grasping what elements conditions can control, and when conditions are evaluated at runtime.

The above items—in combination with the results displayed in Figs. 7 and 8—suggest that level B elements caused the most difficulties. However, these are *perceived* difficulties. While these perceived difficulties were reported in the task protocols, most study participants still managed to use the level B elements successfully as demonstrated in Fig. 6, which shows the *actual* success scores. These actual success scores were extraordinarily high for the core level B elements property (89 percent) and condition (84 percent). The parts of the solution where participants achieved lower success scores (cf. the scores for linking of elements and storing activity input/output in Fig. 5) involve elements from levels A and B, respectively. While we consider both actual success and perceived difficulties as relevant for drawing conclusions, we consider the actual success scores as the most important and robust indicator of gauging users’ understanding of IMS LD.

Looking at those concepts that were regarded as easy-to-handle, we found that participants experienced the least trouble with the concepts activity, role, activity structure, and act. The smallest number of problems occurred with these concepts and they were rarely reported in task protocols or top 3 trouble lists.

Participants in the paper-based workshops understood some of IMS LD’s conceptual structures well, while teachers using the authoring software did not show a good understanding of the same concepts. These include defining services to be offered within environments, linking activities with the environments in which they take place, creating role-parts, as well as using input in activities and storing output from activities.

5.3 Repercussions and Implications

The analyses presented in this paper have identified a number of conceptual structures which presented challenges to teachers’ understanding. One of these challenges, i.e., the management of role-parts, is largely resolved by existing authoring platforms. For example, the ReCourse LD Editor [8] and the Graphical Learning Modeler [12] hide the definition of role-parts from the learning designer with no loss of expressivity. The other principal aspects regarding the storage and reuse of user output should be investigated in greater detail. Also, there is a need to identify the best way to show learning designers the consequences of conditions (what is being shown and what is hidden) and the most effective way of representing environments and their linkage to activities.

The obstacles to teachers’ understanding of the IMS LD specification such as the challenges identified in this paper

as well as in previous research, for instance, regarding the management of roles and groups (e.g., [6], [12], [17], [21], [22]), the concepts of property (e.g., [13], [23]), and environment (e.g., [9], [12]), or the lack of comprehensive integration of communication and collaboration services (e.g., [23], [24]), have led some to conclude that radical changes to the specification are required or that a new or modified specification should be developed. For instance, König and Paramythis [13] propose to extend the IMS LD information model with conceptual structures addressing well-known shortcomings of IMS LD like runtime grouping and artifact handling. Other protagonists, for instance Durand and Downes [9], instead opt for a simplified specification: "it is complicated to create a learning design specification meeting both usability and technical needs. IMS LD tried to do that and it's maybe not the right way to proceed." In response they suggest that two new specifications should be developed. One of these, Simple Learning Design 2.0 [25], would be a technical interoperability specification, while the other would be "a real UML for learning design" [9].

However, extending or simplifying a mature specification like IMS LD may be an arduous path. As pointed out in [26], "...the decision to develop a new standard is often taken without proper due diligence. The enthusiasm and will to create something useful is of course positive, but channeling that energy in the development of a new standard is often misdirected," (p. 229) since developing a new standard and delivering the functionality to end users in a way that is useful and usable may take many years. The findings in this paper may be seen from the perspective of due diligence. The appropriate response to a critique of a specification from the user's point of view is to thoroughly test if there is a problem with the existing specification itself, or if innovative and creative use and implementation can meet the requirements set out by the critique. For example, in our case we have identified that some teachers do not find the relationship between environment and activity in IMS LD to be intuitive. Due diligence requires us to consider if this can be resolved at the level of implementation metaphors. In this case, it seems clear that a graphical interface, which enables authors to place activities within an environment, could be mapped onto the specification. The tool would represent this as an association between an activity and an environment. In any event, it seems clear that this problem does not necessarily require a change in the specification.

The study reported in this paper demonstrated the teachers' ability to assimilate the elements of the specification and to use these elements to create high-quality IMS LD representations of a given instructional design. Despite these positive indications, it is acknowledged that a comprehensive judgment of IMS LD as an interoperability specification must take the full spectrum of IMS LD uses and purposes (as discussed e.g., in [3]) into account. From the perspective of teachers' understanding of the specification, however, the study presented in this paper provides direct evidence that the complexity of the specification's underlying elements, structure, and metaphors is not an insurmountable barrier to its use for authoring.

ACKNOWLEDGMENTS

This work was supported by the ICOPER Best Practice Network (<http://icoper.org>), which was cofunded by the European Commission in the *eContentplus* programme, ECP 2007 EDU 417007.

REFERENCES

- [1] IMS Global, "IMS Learning Design Information Model," <http://tinyurl.com/imsld-im>, 2003.
- [2] R. Koper and B. Olivier, "Representing the Learning Design of Units of Learning," *Educational Technology and Soc.*, vol. 7, no. 3, pp. 97-111, 2004.
- [3] D. Griffiths and O. Liber, "Opportunities, Achievements, and Prospects of IMS LD," *Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies*, L. Lockyer, S. Bennett, S. Agostinho, and B. Harper, eds., Information Science Reference, pp. 87-112, 2009.
- [4] S. Neumann, M. Klebl, D. Griffiths, D. Hernández-Leo, L. de la Fuente Valentin, H. Hummel, F. Brouns, M. Dertnl, and P. Oberhuemer, "Report of the Results of an IMS Learning Design Expert Workshop," *Int'l J. Emerging Technologies in Learning*, vol. 5, no. 1, pp. 58-72, 2010.
- [5] I. Martínez-Ortiz, J.-L. Sierra, and B. Fernández-Manjón, "Authoring and Reengineering of IMS Learning Design Units of Learning," *IEEE Trans. Learning Technologies*, vol. 2, no. 3, pp. 189-202, July-Sept. 2009.
- [6] R. Van Es and R. Koper, "Testing the Pedagogical Expressiveness of IMS LD," *Educational Technology and Soc.*, vol. 9, no. 1, pp. 229-249, 2006.
- [7] IMS Global, "IMS Learning Design XML Binding. Version 1.0 Final Specification," http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bindv1p0.html, 2003.
- [8] D. Griffiths, P. Beauvoir, O. Liber, and M. Barrett-Baxendale, "From Reload to ReCourse: Learning from IMS Learning Design Implementations," *Distance Education*, vol. 30, no. 2, pp. 201-222, 2009.
- [9] G. Durand and S. Downes, "Toward Simple Learning Design 2.0," *Proc. Fourth Int'l Conf. Computer Science and Education (ICCSE '09)*, pp. 894-897, 2009.
- [10] *Learning Design: A Handbook on Modelling and Delivering Networked Education*, R. Koper and C. Tattersall, eds., Springer, 2005.
- [11] D. Griffiths, P. Beauvoir, and P. Sharples, "Advances in Editors for IMS LD in the TENCompetence Project," *Proc. IEEE Eighth Int'l Conf. Advanced Learning Technologies*, pp. 1045-1047, 2008.
- [12] S. Neumann and P. Oberhuemer, "User Evaluation of a Graphical Modeling Tool for IMS Learning Design," *Proc. Advances in Web Based Learning (ICWL '09)*, pp. 287-296, 2009.
- [13] F. König and A. Paramythis, "Towards Improved Support for Adaptive Collaboration Scripting in IMS LD," *Proc. Fifth European Conf. Technology Enhanced Learning Sustaining TEL: From Innovation to Learning and Practice*, pp. 197-212, 2010.
- [14] D. Griffiths and J. Blat, "The Role of Teachers in Editing and Authoring Units of Learning Using IMS Learning Design," *Advanced Technology for Learning*, vol. 2, no. 4, pp. 243-251, 2005.
- [15] W. Greller, "Managing IMS Learning Design," *J. Interactive Media in Education*, vol. 2005, no. 1, pp. 1-9, 2005.
- [16] D. Sampson, P. Karampiperis, and P. Zervas, "ASK-LDT: A Web-Based Learning Scenarios Authoring Environment Based on IMS Learning Design," *Advanced Technology for Learning*, vol. 2, no. 4, pp. 207-215, 2005.
- [17] D. Hernández-Leo, J. Asensio-Perez, and Y. Dimitriadis, "IMS Learning Design Support for the Formalization of Collaborative Learning Patterns," *Proc. IEEE Int'l Conf. Advanced Learning Technologies*, pp. 350-354, 2004.
- [18] G. Paquette, M. Léonard, K. Lundgren-Cayrol, S. Mihaila, and D. Gareau, "Learning Design Based on Graphical Knowledge Modelling," *Educational Technology and Soc.*, vol. 9, no. 1, pp. 97-112, 2006.
- [19] J. Dalziel, "Implementing Learning Design: The Learning Activity Management System (LAMS)," *Proc. Ann. Conf. Australasian Soc. for Computers in Learning in Tertiary Education*, pp. 593-596, 2003.
- [20] Y. Miao, "CoSMoS: Facilitating Learning Designers to Author Units of Learning Using IMS LD," *Proc. 13th Int'l Conf. Computers in Education (ICCE)*, pp. 275-282, 2005.

- [21] Y. Miao, K. Hoeksema, H.U. Hoppe, and A. Harrer, "CSCL Scripts: Modelling Features and Potential Use," *Proc. Conf. Computer Support for Collaborative Learning*, pp. 423-432, 2005.
- [22] S. Neumann and P. Oberhuemer, "Bridging the Divide in Language and Approach between Pedagogy and Programming: The case of IMS Learning Design," *Proc. 15th Int'l Conf. Assoc. for Learning Technology*, pp. 62-72, 2008.
- [23] L. de la Fuente Valentín, Y. Miao, A. Pardo, and C. Delgado Kloos, "A Supporting Architecture for Generic Service Integration in IMS Learning Design," *Proc. Times of Convergences. Technologies across Learning Contexts (EC-TEL '08)*, pp. 467-473, 2008.
- [24] S. Wilson, P. Sharples, and D. Griffiths, "Distributing Education Services to Personal and Institutional Systems Using Widgets," *Proc. Mash-Up Personal Learning Environments (MUPPLE '09)*, pp. 47-58, 2008.
- [25] G. Durand, L. Belliveau, and B. Craig, "SLD 2.0 XML Binding," <http://tinyurl.com/sld2-0-xml>, 2010.
- [26] E. Duval and K. Verbert, "On the Role of Technical Standards for Learning Technologies," *IEEE Trans. Learning Technologies*, vol. 1, no. 4, pp. 229-234, Oct.-Dec. 2008.



Michael Derntl received the PhD degree from the Faculty of Computer Science, University of Vienna. Currently, he is working as a research associate at the Information Systems and Databases chair, RWTH Aachen University. During recent years, his research mainly focused on learning technology standards and specifications, in particular IMS Learning Design. His general research interests include the interdisciplinary application of methods and

concepts from computer science, engineering, and social sciences to problems in the information systems and technology-enhanced learning domains. His most recent research project involvements include ICOPER (adopting standards for making digital content accessible and interoperable) and the EU FP7 projects TEL-Map (roadmapping for shaping the future of technology-enhanced learning) and GaLA (network of excellence on serious games).



Susanne Neumann is a researcher with the Center for Teaching and Learning, University of Vienna. In her research, she focuses on links between IMS Learning Design and teaching methods. Further foci include the description of teaching methods and how instructors adopt and use teaching methods. She also teaches courses on instructional design. Her dissertation investigates classifications for teaching methods.



David Griffiths is a professor at the Institute for Educational Cybernetics (IEC), where he coordinates research activities. In recent years, a major focus of inquiry has been the development, use, and implications of specifications for e-learning. In particular, he has published extensively on IMS Learning Design. He was coordinator of the UNFOLD project, funded by the IST programme, during which he led six international events for communities of practice involved in IMS LD. He led the IEC contribution to the TENCompetence project, within which IEC developed the Wookie widget server (Apache Wookie). He is now coordinating IEC work on two projects which make use of Wookie technology in the areas of telecoms mash-ups and in secondary school classrooms.



Petra Oberhuemer studied physics at the University of Vienna and information technology at Danube University. She has been working at the University of Vienna's Center for Teaching and Learning since 2003. Her main responsibilities include the development of services supporting innovative technology-enhanced learning and teaching concepts and the evaluation of new technologies with respect to their implementation at the university-wide level. Her main current interest lies in open educational resources, open access, e-learning standards, and in all aspects of output-based education triggered by the Bologna Process implementation. She has been and is presently involved in several R&D projects with European, national, and bilateral funding. She has published several articles in national and international conferences and journals, and is a reviewer for several journals.