To appear in *Lecture Notes in Computer Science*

User Evaluation of a Graphical Modeling Tool for IMS Learning Design

Susanne Neumann[1] and Petra Oberhuemer[1]

[1] University of Vienna, Center for Teaching and Learning, Porzellangasse 33a, 1090 Wien, Austria
{susanne.neumann-heyer, petra.oberhuemer}@univie.ac.at

Abstract. IMS Learning Design is said to be highly technical language that is hard to understand and apply by teaching practitioners. The modeling tool Graphical Learning Modeler was built to bypass this problem for level A and partially level B of the IMS Learning Design specification. Its graphical interface with drag and drop allows easy setup of learning designs that can be made conformant to IMS Learning Design. This article presents the results of an evaluation of the Graphical Learning Modeler performed with instructors at a higher education institution. Results showed that instructors were generally successful in building learning designs, but that they still had problems with transferring concepts from their teaching environment to the concepts of IMS Learning Design. Furthermore, they had trouble grasping the meaning of an editor environment that is outside the runtime environment.

Keywords: IMS Learning Design, editor, modeling, graphical, usability

1 Introduction of the Graphical Learning Modeler

1.1 The IMS Learning Design Specification and Early Editing Software

The IMS Learning Design (IMS LD) specification [1] was introduced in 2003 to offer a counterpart to the regularly content-focused e-learning standards. Its purpose is to describe any pedagogic approach in standardized language for face-to-face as well as online learning situations [1]. Specifically, the language offers to express the actions that persons perform, the different roles they take during activities, as well as the artifacts they use and exchange during those activities. The eXtensible Markup Language (XML) builds the foundation of IMS LD. All needed elements (an imsmanifest.xml that holds the learning design as well as physical files used in the learning design) are captured in a unit of learning package.

The critique so far has been that the language IMS LD offers is hard to understand and that it takes considerable effort to apply [2], [3]. Before software applications for IMS LD editing were available, learning designers had to code in XML to produce units of learning. The Reload Learning Design Editor solves this problem by providing an interface that works without any XML coding. However, the Reload editor still requires its users to know the syntax and semantics of IMS LD, meaning

that instructors without this prerequisite knowledge can only create units of learning if they receive significant technical support [4]. To overcome this problem, the Graphical Learning Modeler (GLM)[1] was built.

1.2 Development of the GLM

The goal of the GLM development was that teaching practitioners could be provided with a software application that allowed them easy, i.e. without significant prerequisite knowledge, building of an IMS LD conformant unit of learning. A graphical interface with drag and drop functionalities was implemented to allow intuitive modeling. A screenshot of the GLM interface is depicted in Fig. 1.

To reach the goal of reducing instructors' prerequisite knowledge, the GLM hides at level A (basic IMS LD software development level) the time-related control elements *Play*, *Act*, and *Role-Part*, and at level B (advanced software development level) it hides the elements *properties* and *conditions*, which are used to store and monitor data and resources generated by persons participating in the unit of learning.

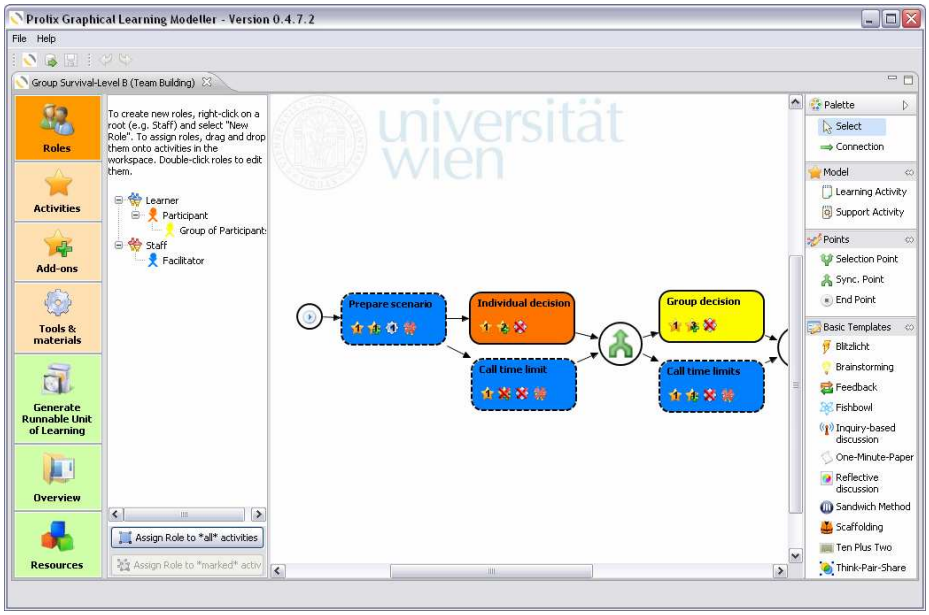

Fig. 1. Total View of Graphical Learning Modeler version 0.4.7.2 (post-evaluation version) depicting navigation menu, instance trees, workspace, and design palette (from left to right)

[1] http://sourceforge.net/projects/prolix-glm/

1.3 Main GLM functionalities

The largest part of the GLM is the workspace in the middle (cp. Fig. 1), where the learning designer creates the sequence of activities. The design palette in the upper right corner provides design and altering functions such as the creation of activities, or the drawing of connections between activities. Models (activities) and points for selection or synchronization of activities may be easily dragged and dropped from the palette onto the workspace. A special design area, formerly called *Interactions* now called *Add-Ons*, offers learning designers IMS LD level B functionalities for the integration of design elements that allow interactive contributions of the unit of learning participants during runtime. There are currently four types of Add-Ons available: text work, uploading files, question & answer, and multiple-choice test. The Add-Ons are integrated in the left-hand navigation bar and additionally within each dialog for editing an activity. A context-sensitive help system guides learning designers when applying Add-Ons to activity sequences. In order to enable users to systematically and efficiently manage all required resources, the GLM incorporates a centralized resource management that is accessible via the navigation bar's tab *Resources* (bottom left in Fig. 1). The resource management offers a taxonomical classification schema, which consists of pre-specified categories referring to IMS LD elements that are provided to store all learning design relevant files, e.g. activity descriptions and learning objects. In this way, the resource management facilitates the reuse of already existing resources and provides explicit information about the context a resource is used in.

The GLM interprets the graphically captured learning design from the workspace and automatically detects acts, role-parts and activity structures (bundled activities that are performed by the same role) as well as properties and conditions in order to generate the corresponding imsmanifest.xml. Conditions are needed for more complex learning designs with parallel, independent activity strands performed by several roles: In order to maintain the independence of all roles, level B conditions are used to provide each role access to its activities at the right time.

2 User Evaluation

2.1 Previous Evaluations of the GLM

Feedback during the GLM development was previously collected from three different communities of users: pedagogical experts, PROLIX[2] test bed partners, and IMS LD tool developers. Generally, the feedback was positive; especially the drag & drop functionalities and the quick setup of an entire learning design were well received. However, the different communities also expressed varying wishes for extensions to the GLM.

Pedagogical experts wanted an overarching view on the learning design: The level of design, which in the GLM is represented by activities, was too small for them to plan a semester-long course. This problem was attempted to be partially solved by

[2] Process-Oriented Learning and Information Exchange (PROLIX), http://prolixproject.org/

including ready-to-use learning design templates that cover a longer frame of time so that instructors do not have to start at the activity level. Furthermore, an annotation function is planned for future implementation that will allow the learning designer to comment the design and make connections between different parts of the learning design.

PROLIX test bed partners continuously give feedback for all software components used within the project, to which the GLM developments also belong. The international standards ISO 9241-110[3] for guidance principles on the ergonomics of human-system interaction and ISO 9241-11[4] for usability guidance on visual display terminals build the foundation of this evaluation. The feedback showed that controllability of the GLM software, conformity with user expectations, suitability for the task and suitability for learning were positively viewed. Improvements were especially recommended for error tolerance, i.e. supporting the learning designer to recover from errors. Results for the late 2007 and late 2008 evaluation can be seen in Fig. 4.

IMS LD tool developers were the third community that gave feedback during early GLM development. Again, the feedback was generally positive as indicated by the positive values the developers rated on the structured feedback form. The critical comments regarded specific IMS LD features that the tool developers were missing in the GLM. This feedback has to be viewed in the light that novices are the target audience of the GLM. Therefore, not all functionalities of IMS LD are offered in the GLM, even some of the main concepts like *plays* or *acts* are hidden from users. It is therefore hard to integrate functions for elements that are not even visible within the GLM, for instance, controlling how an act is ended. Some criticized that even though the GLM attempts to hide the IMS LD concepts, it still contains too much visibility of these elements in the software. In further developments, we integrated this feedback.

More details on previous evaluations can be found in [5].

2.2 Method

A principal evaluation was then planned, especially with the purpose to collect feedback from the audience that had originally been the target group of the GLM development: teaching practitioners, who are not knowledgeable of the IMS LD specification. The goal of the evaluation was to test the usability of the GLM version 0.4.6, to test instructors' ability to create units of learning and their understanding of the IMS LD concepts as represented in the GLM. The applicability of the GLM in the instructors' own teaching context was also of interest during the evaluation.

Test persons were chosen to partake in the evaluation if they were instructors (professor, lecturer, teaching assistant, or trainer) at the University of Vienna. Their age was not considered relevant, although a focus was placed on instructors who have had experience with technology-supported teaching.

The test population eventually comprised 21 test users. They represented a wide range of subject areas (e.g. physics, business administration, theology, geography,

[3] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38009
[4] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=16883

social sciences, computer sciences etc.) and judged themselves to have considerable knowledge in the practice of e-learning (on a 1 – 5 continuum, with 5 representing the highest knowledge, test users rated their knowledge at an average of 4.3). The range of test users' general experience of teaching at a higher education institution was between one and twenty years.

There were three separate dates of testing. The difference between the test days was that test users on the first and second day received a ten-minute live demonstration about the main functions of the GLM, while test users on the third day watched a three-minute flash video of the main GLM functions. Following a quick round of question-and-answer after the presentation, the test persons had to create two learning designs using the GLM: One design was prescribed, and one design they had to derive from their own teaching context. The prescribed design was presented as a printed narrative less than a page long.

The entire test population was split so that half of the test users started off with the prescribed learning design, while the other half started off with a learning design that was derived from their own teaching context.

Test users kept a protocol while performing the evaluation tasks. In the protocol they recorded the time they started and ended each learning design, any questions that arose while they worked on each learning design, and a yes/no remark whether they were able to resolve each question on their own. When finished creating the learning designs, test users filled in one structured questionnaire using a four-point Likert scale with usability questions and finally answered three open-ended questions.

2.3 Results

The evaluation showed that test users were successful in building the learning designs. Of the 40 learning designs[5] that test users built, 37 were complete or nearly complete. *Complete* means that the learning design had all necessary components to be exported as an IMS LD unit of learning; *nearly complete* means that only a minimal element, like a connection between two activities, was missing to completion. This can be considered a success as the time to introduce test users to the functions of the GLM was minimal, yet sufficient to put them in a position to build complete learning designs. This is true for instructors that watched the live demonstration as well as for instructors, who watched the shorter flash film.

The test users made extensive use of the functionalities offered within the GLM. They used both types of activities offered (learning activity and support activity), set up and assigned multiple roles and even child roles, created environments[6], used level B functions, and built complex learning paths with varying modes and simultaneous activities of different roles.

[5] Note that two test users did not attempt to build the *prescribed design*; they just built a design from their own teaching context. These are the two missing learning designs to a total of 42.

[6] Environments are containers for learning objects and learning services such as chat or forum that are used during activities. Once created, an environment can be referenced multiple times within different activities.

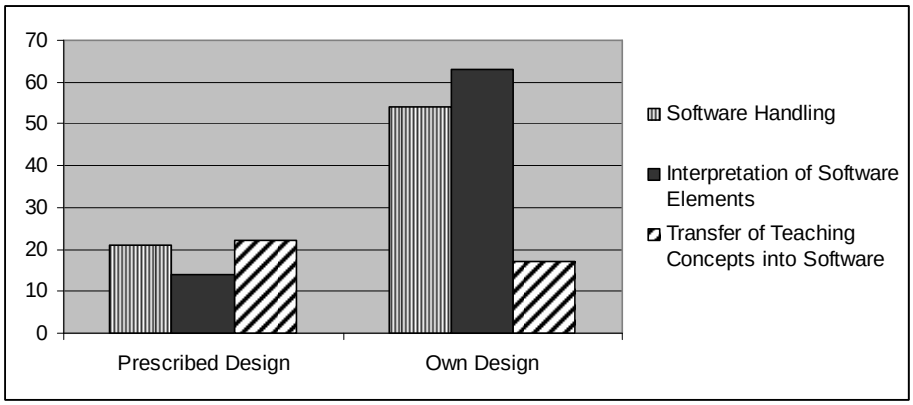

Fig. 2. Number and types of questions recorded by test users during the preparation of prescribed and own learning designs (N=21)

Despite this success, test users still encountered problems. Fig. 2 shows groups of questions recorded by the test users in their protocols. *Software handling* summarizes questions of the kind, "how do I delete a connection?" or, "how can I copy activities?" *Interpretation of software elements* summarizes questions like, "what is the difference between environments and interactions?" or, "I don't know what the error messages mean when I try to export the unit of learning." Examples for *Transfer of teaching concepts into software* include, "how do I enter session times and deadlines?" or, "how do I define a group for learners' group work?"

As Fig. 2 shows, test users recorded overall nearly twice as many questions when creating a learning design that was derived from their own teaching context as compared to when creating the prescribed design. Since the test user group was split (half the users started with the prescribed and half started with their own design), this difference in questions recorded cannot be attributed to users starting first with one or the other design. It seems that transferring a teaching situation from one's own context into the GLM is a farther step to take and thus generates more questions with instructors, than when transferring a purpose-built narrative into the GLM. Nevertheless, this did not impede users' ability to complete learning designs as the test users were just as successful in completing or nearly completing their own designs as they were with the prescribed design. Of the questions recorded, test users reported that they were able to solve 23% of questions recorded when working on their own designs, and 30% of the questions recorded during the prescribed learning design.

When giving feedback using the structured questionnaire when they had finished the two learning designs, test users attributed mostly positive values for the GLM, meaning that they found the GLM quite usable. The only critical element, whose average value had a clear negative tendency was, "I understood the concept of the special design elements Interactions." *Interactions* is the overarching term for all available IMS LD level B functions that the GLM offers. Although the GLM packages the complicated technical setups for level B properties and conditions, and

presents them to users in a teaching practitioner-oriented terminology (e.g. text work, uploading files), test users clearly indicated that they had trouble using this feature. Major reconstruction for this interface element is thus necessary.

To wrap up the GLM evaluation, test users answered three open-ended questions:

1. To what extent can you imagine an integration of the GLM in your teaching practice?
2. What problems do you see in using the GLM, and why do these problems occur?
3. What suggestions do you have to improve the GLM?

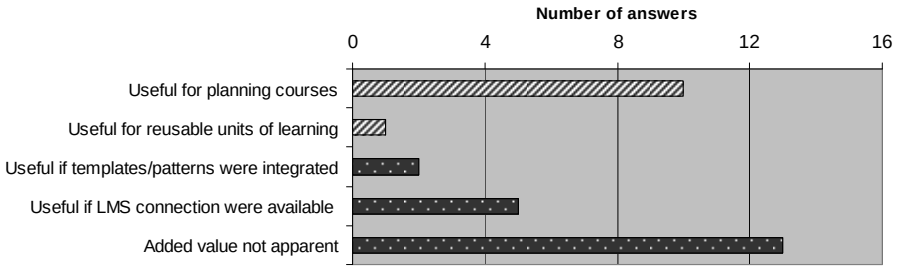

Fig. 3. Free answers given by test users to the question, "To what extent can you imagine an integration of the GLM in your teaching practice?" (N=21)

Fig. 3 shows the most common answers given in response to the first open-ended question. About half the test users answered freely that they find the GLM useful for planning courses. However, more than half of the test users answered that the added value of the GLM was not apparent to them. Others noted that if additional functions such as ready-to-use learning design templates or a connection to a learning management system (LMS) were available, then they would see a possibility to integrate the GLM in their teaching practice.

Questions 2 and 3 aimed to identify problems and areas for improvement. Test users reported here that they would like to see the Interactions improved (which had already been shown in the quantitative analysis of the usability questionnaire). It is praiseworthy that test users were using level B concepts in their learning designs. This may be the first record of instructors without knowledge of IMS LD autonomously using level B functions. Previous reports mentioned that instructors could only create level B units of learning when someone else besides the instructor built the necessary level B components [4]. The instructors taking part in this evaluation had no help and no introduction to level B concepts as these functions were not even part of the minimal introductory demonstration; yet, more than two thirds of them included level B specific functions in their learning designs. After using the level B functions, test users reported that the interface was not intuitive to them, e.g. dragging and dropping artificial sub-steps of a text-work interactive element onto activities in the workspace. They wanted to drag the text work itself, not the sub-step. However, if the setup of level B functionalities as it was represented in the GLM 0.4.6 (the version being tested) was sufficient for test users to understand the concept of interactive elements, then this represents a comfortable starting point for improvement.

Test users further reported that they would like to see improvements of the environments. The concept *environment* as a container that holds materials and services used during activities bewildered the test users. First of all, they confused the environment with the runtime environment, i.e. the LMS. This may stem from the term "environment" usually being used to describe the setting where the learning will eventually take place. Furthermore, the incessant assignment of titles when working with environments (e.g. title for the environment, title for learning object inside the environment, title for the resource that represents the learning object etc.) agitated test users. They did not understand why it is necessary to assign so many titles. This may be due to the fact that the instructors had no visualization of the runtime environment, where it may be more apparent that titles are needed to reference environments, learning objects, and included resources (just like activities) in navigation menus. Finally, some instructors questioned what the difference is between activities and an environment; To these instructors, the two seemed to be integral.

Last but not least, test users noted that a connection from the GLM to a learning management system needs to be made so that they can visualize what they are building in the modeling tool.

2.4 Changes to the GLM as a Result of the Evaluation

The improvement of Interactions has been tackled by redesigning this portion of the GLM. The previous drag and drop functionality was simplified in version 0.4.7 as requested by the test users, and additionally, level B functions were offered within each dialog for editing an activity. This more closely corresponds to the way a learning design with level B functions is built, as these special functions are often directly associated with activities. A wizard that guides learning designers step-by-step through the needed elements for setting up level B functions was integrated for both access options (instance tree and activity dialog). Further, level B functions were renamed from *Interactions* to *Add-Ons* to make clearer to the learning designers that the offered level B functions belong to and thus extend activities through interactive elements.

Reacting to the problems test users encountered with environments, the container that represents the environment was hidden from the user's view in GLM 0.4.7. Instead, users now directly work with learning objects and services as they had requested during testing. This means that learning designers do not have to create the environment container anymore, but every time a learning object or service is being created and assigned to an activity, the GLM automatically creates an environment element, automatically assigns a name to the environment, and references the assigned items within the environment. Again, this design function was renamed to make clearer to users what is meant, since they had confused "environment" with the LMS, where the learning design will be executed. Therefore, environments are now called *tools & materials*.

The wanted integration of the LMS into the GLM cannot be taken care of as part of the current developments as the amount of resources needed for this undertaking is considerable larger than the entire GLM development. It is recognized, though, that the direct connection to a runtime environment or LMS that visualizes the learning

design even while being designed is an important feature that would provide tremendous aid to learning designers.

Another request test users mentioned in answers to the open-ended questions was to provide ready-to-use templates. In GLM 0.4.7, a set of eleven templates capturing common teaching methods like brainstorming or discussion exercises have been implemented as a new feature. Now, learning designers can via drag & drop create an entire sequence of activities including all roles and all activity descriptions as well as associated materials.

The changes described above have been implemented in GLM version 0.4.7. PROLIX test bed partners, who regularly test the software, evaluated this advanced version. The results are shown in Fig. 4, depicting two ratings collected at two different points in time, namely in late 2007 (GLM0.4.4) and in late 2008 (GLM0.4.7). The foundation for the evaluation built the ISO human-system interaction dialogue principles mentioned above. As can be seen, significant improvements have been made for all the principles except suitability for individualization. This decline may have appeared because PROLIX test users have become more cognizant of suitability issues during the time of the project, and with this raise in expectations, they did not see a match for them reflected in the GLM.

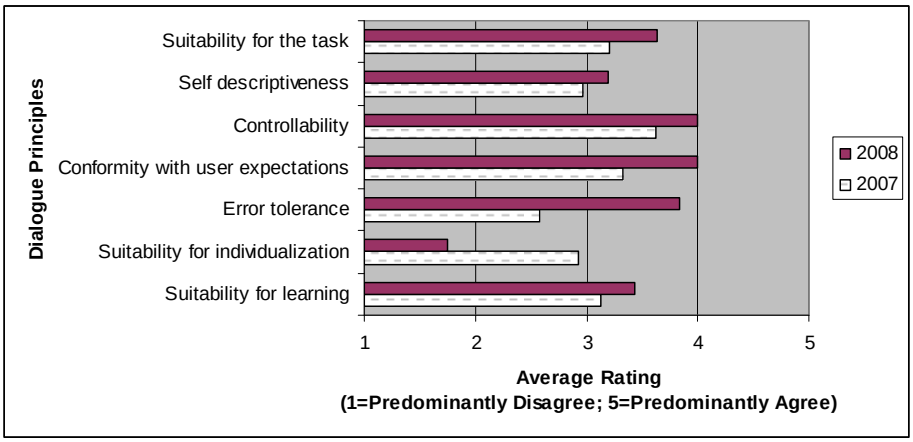

Fig. 4. Average Ratings for GLM versions 0.4.4 and 0.4.7 by PROLIX test bed partners according to ISO human-system interface dialogue principles in late 2007 and late 2008

3 Conclusion

This article introduced the GLM, editing software for IMS LD units of learning. The barriers that IMS LD presents due to its technical focus were aimed to be overcome by designing a graphical modeling tool that allows easy editing of units of learning, even for novices of IMS LD. This goal has to some extent been achieved as the instructors participating in the herein described user evaluation successfully created complete learning designs that could be exported as units of learning. They did so

after having watched a minimal demonstration of the main features. There has not been a report for such a short learning time in regard to other IMS LD editors. Another success is that more than 70% of the instructors included level B functions in their learning designs, even though they were not introduced to this feature.

Results of the GLM evaluation showed that there are still improvements necessary. Some of the improvements have already been accomplished like the improved access to level B functions, the hiding of the environment, and the inclusion of ready-to-use learning design templates. Despite test users reporting good usability in their feedback, the added value of the GLM was not necessarily apparent to them. A major reason for this is probably the lack of a connection between the GLM and a runtime environment, which cannot be realized within the current development.

The lack of connection between editor and runtime environment points out a major problem that all IMS LD tool developments essentially have: Creating a unit of learning outside the runtime environment, where the unit of learning will be implemented with learners and instructors, represents a great barrier to the eventual success of IMS LD. Instructors have a hard time creating a learning environment in an abstract manner, away from the place where the unit of learning will eventually be visible to learners. The evaluation showed that instructors would have favored to create their unit of learning directly in the system where it will be implemented. If the concept of keeping editors separate from runtime environments remains, then a smooth bridge between the two has to be built that leaves the learning designer believing that the two software systems (editing and runtime) are homogeneous.

Acknowledgments. This article was written in the context of the research and development project PROLIX, which is co-funded by the European Commission in the Sixth Framework Programme “Information Society Technologies”. The Graphical Learning Modeler that is being discussed in this paper is a development of Philipp Prenner and Stefan Zander at the Multimedia Information Systems Group at the University of Vienna headed by Wolfgang Klas.

References

1. IMS Global Learning Consortium: IMS Learning Design Information Model, http://www.imsproject.org/learningdesign/ (2003)
2. Griffiths, D., Blat, J.: The Role of Teachers in Editing and Authoring Units of Learning Using IMS Learning Design. Advanced Technology for Learning 2(4) (2005)
3. Neumann, S., Oberhuemer, P.: Bridging the Divide in Language and Approach between Pedagogy and Programming: The Case of IMS Learning Design. In: 15th International Conference of the Association for Learning Technology, Leeds (2008)
4. Barrett-Baxendale, M.: A practitioner’s view of Learning Design. Presentation at TEN-Competence Open Workshop on Current Research on IMS Learning Design and Lifelong Competence Development, Barcelona (2007)
5. Neumann, S., Oberhuemer, P.: Supporting Instructors in Creating Standard Conformant Learning Designs: The Graphical Learning Modeller. In: 20th World Conference on Educational Multimedia, Hypermedia & Telecommunications, Vienna (2008)