Proceedings of World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA)
Vienna, June 30 - July 4, 2008

Supporting Instructors in Creating Standard Conformant Learning Designs: the Graphical Learning Modeler

Susanne Neumann
Centre for Teaching and Learning
University of Vienna
Austria
susanne.neumann-heyer@univie.ac.at

Petra Oberhuemer
Centre for Teaching and Learning
University of Vienna
Austria
petra.oberhuemer@univie.ac.at

Abstract: While the majority of e-learning standards focus on content to be described, the IMS Learning Design (IMS LD) specification represents the exception as it places the learning activity rather than the content in its focal point. Problematic is that the wide adoption of the IMS LD specification is hindered by its highly technical language. In this article, the modeling software Graphical Learning Modeler (GLM) is introduced that aids instructional designers and teaching practitioners in building standardized units of learning for online learning and teaching. The software closes the gap of practitioner language and technical language by translating the designed learning activity sequences from a graphical interface to an IMS LD conformant eXtensible Markup Language (XML) file that may be interpreted by various learning management systems.

Problem Scenario

Unlike face-to-face learning situations, which naturally employ learning (inter)actions between participants, e-learning often uses uninvolved single learner models of learning with its main focus on the display of information, also called content (Britain 2007). The development of e-learning standards mirrors this discrepancy as these standards commonly describe content. The first standards thus focused on technical rather than pedagogical aspects such as accessibility, adaptability, interoperability, and reusability of content (Advanced Distributed Learning (ADL) 2006). Prominent standards that drew on such attributes are the Sharable Content Object Reference Model (short: SCORM) (Advanced Distributed Learning (ADL) 2006) and Learning Object Metadata (short: LOM) (IEEE 2002).

An e-learning model that merely manages the sequencing of learning content cannot be considered pedagogically sound as content alone does not constitute a pedagogical situation (Klebl 2004; Sloep 2004). Excessive emphasis on content had in the past led to “rigid and unhelpful habits of instruction” (Beetham & Sharpe 2007, p. 2). The IMS Learning Design (IMS LD) specification (Koper, Olivier & Anderson 2003) answered this one-dimensionality by providing a technical language for describing multi-learner interactive learning design scripts, whose main focus is the learning activity rather than the learning content. The goal of this specification was to describe any pedagogical situation (Koper et al. 2003). Several advantages are seen in a standardized description of pedagogical settings, among them reusing pedagogic scenarios just as much as content, while at the same time allowing greater pedagogical flexibility for content reuse (Sloep 2004), and creating explicit representations of pedagogical knowledge (Beetham 2004) that visualize aspects previously taken for granted (Beetham & Sharpe 2007).

Although the authors of IMS LD have finally created the possibility to describe rich interactions between multiple learners and tutors in an e-learning setting, the language thus provided is far from the language teaching practitioners use in their daily practice (Beetham 2004; Griffiths & Blat 2005). Even though IMS LD employs metaphors derived from theater (Koper 2005) to create connections in terminologies, these metaphors are only partially helpful in visualizing the possibilities of online learning settings. As an illustration, consider that IMS LD distinguishes at the simplest of its three levels, i.e. *level A*, parallel scripts for learning activity sequences (so called *plays*), between which the roles of the unit of learning may switch back and forth at any point in time. Each of these plays then contains sequential *acts* that provide staccato disclosures of new activities in the following act. Teaching practitioners may not derive from the theater terminology the actual meaning of executing a unit of learning; the conceptual gap thus opened may be far greater than the theater metaphors are able to bridge.

Adequate use of the language provided by IMS LD is not trivial as an elaborate amount of technical and programming knowledge is needed (Heyer, Oberhuemer, Zander & Prenner 2007). The tools currently available to create IMS LD conformant units of learning require that learning designers know the syntax and semantics of the IMS LD specification, or they only allow modeling within a set of pre-defined templates. What is needed then is an IMS LD modeling tool that supports the instructional designer perspective. In the following sections, the Graphical Learning Modeler (GLM) is introduced. This software reduces the technical knowledge needed so that teaching practitioners and instructional designers are put in a position to create IMS LD conformant units of learning.

The Graphical Learning Modeler

Partners, Goals, and Problem Addressed

The GLM has been developed at the University of Vienna in conjunct work of pedagogic and software engineering experts. The participating departments at the University are the Centre for Teaching and Learning and the Institute for Distributed Multimedia Systems. According to the IMS LD specification, which phases implementation efforts to three levels A, B, and C, development work for level A has been completed, while work for level B is still underway.

The goal of the developments is to provide comprehensive and intuitive modeling software, which reduces the complexity of the IMS LD specification to a degree, where teaching practitioners are enabled to build IMS LD conformant units of learning. A subsidiary goal thus was to create translation mechanisms that interpret a graphical representation of a learning design and convert it to the required eXtensible Markup Language (XML)-format. These goals were achieved by viewing the *activities* of learners and instructors as the modeling core around which to build other aspects covered by the IMS LD specification. The activities are graphically displayed and may be freely defined and arranged by the learning designer.

Because of the GLM software, teaching practitioners are enabled to intuitively create units of learning to be played in learning management systems. A new educational opportunity is created as the barrier for access is lowered, and thus the number of instructors that produce IMS LD conformant units of learning may be enlarged; more units of learning may then be produced, exchanged, and evaluated as was one of the original goals of the IMS LD specification (Koper et al. 2003). Furthermore, learning designs are displayed in the GLM in a way that heightens the chance for designers to easily grasp and evaluate each others' pedagogical designs.

Previous Work

The GLM is based on the Reload Learning Design Editor, which was developed at the University of Bolton as part of a project that focused on the development of tools incorporating emerging learning technology interoperability specifications[1]. The Reload Editor was at the start of the GLM developments the only editor that mirrored the IMS LD specification 100%. The letdown of Reload was, however, that instructional designers and instructors must have complete technical knowledge of the IMS LD specification to use Reload. Yet, even if designers did have an understanding of IMS LD, the way the Reload Editor is set up with its separation of design elements it is hard for the designers to keep the overview.

[1] http://www.reload.ac.uk/background.html

For example, suppose we wanted to build a learning design, where a learner first brainstorms his or her prior knowledge and then categorizes the so attained results in preparation for a more complex problem solving activity. According to the IMS LD specification, we create the *role* (learner), the *activities* (brainstorm prior knowledge & categorize results), and then we arrange these components within a *method* to indicate the sequence (*acts*) of who performs what activities (i.e. *role-part*) at what point in time. In the Reload editor, you need three separate views to perform these steps: specifying roles (Fig. 1), specifying activities (Fig. 2), and arranging roles and activities within the method (Fig. 3). Switching back and forth between these views can become quite cumbersome once the design gets larger or more complicated, e.g. when the design features nested structures and has different learning resources attached to the activities.

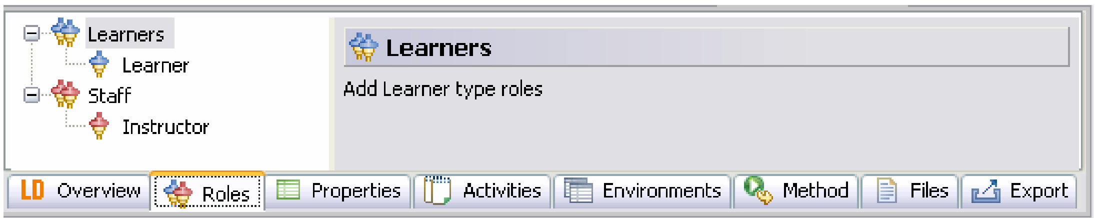

Figure 1: Detail of Reload with *roles* view.

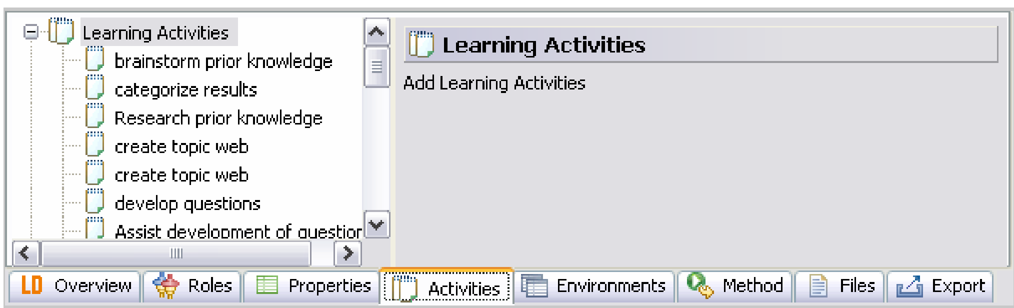

Figure 2: Detail of Reload with *activities* view.

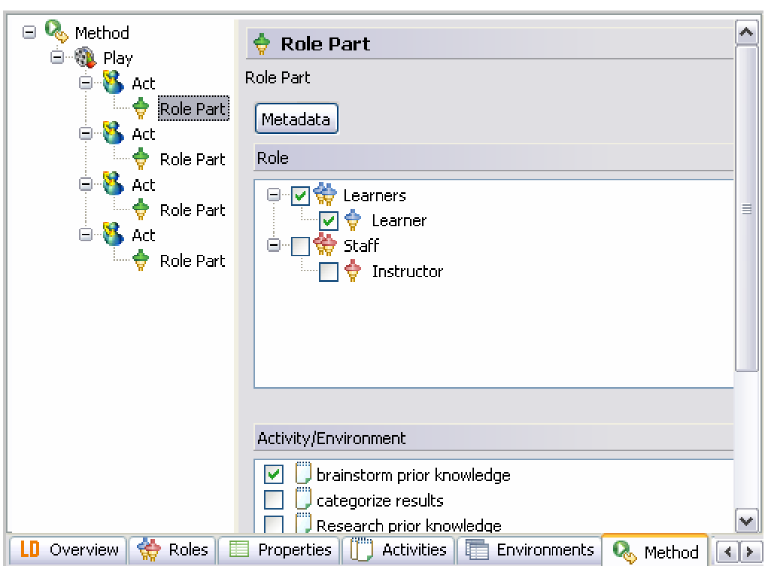

Figure 3: Detail of Reload with *method* view, where the role learner is assigned the activity “Brainstorm…”

Reload, despite its interface problems, formed the ideal base, onto which to build the GLM, since it was the only software available that supported IMS LD at all three levels A, B, and C at the start of our developments. The GLM uses the Reload Learning Design Editor as functional backbone and provides additional mechanisms for handling components and user interactions for the graphical representation of learning designs. This way the GLM can encapsulate the actual data components implemented in Reload and is able to complement them with additional information to build its own model, from which the visual representations are generated. The use of the Graphical Editing Framework (GEF)[2] allows for the uncomplicated extension and customization of this functional layer with additional functionalities. The GLM thus interprets the graphical representation of a learning design, and uses the Reload Editor business logic to write the IMS LD conformant imsmanifest.xml[3]-file.

Portrait of the Graphical Learning Modeler

Level A

The goal of the GLM is to hide as much of the IMS LD specific terminology as possible. This includes at level A the concealment of the *method*, *acts*, *role-parts*, and *activity structures* with its subtypes *sequence* and *selection*. The learning designer thus does not have to be concerned with the technical setups of these concepts.

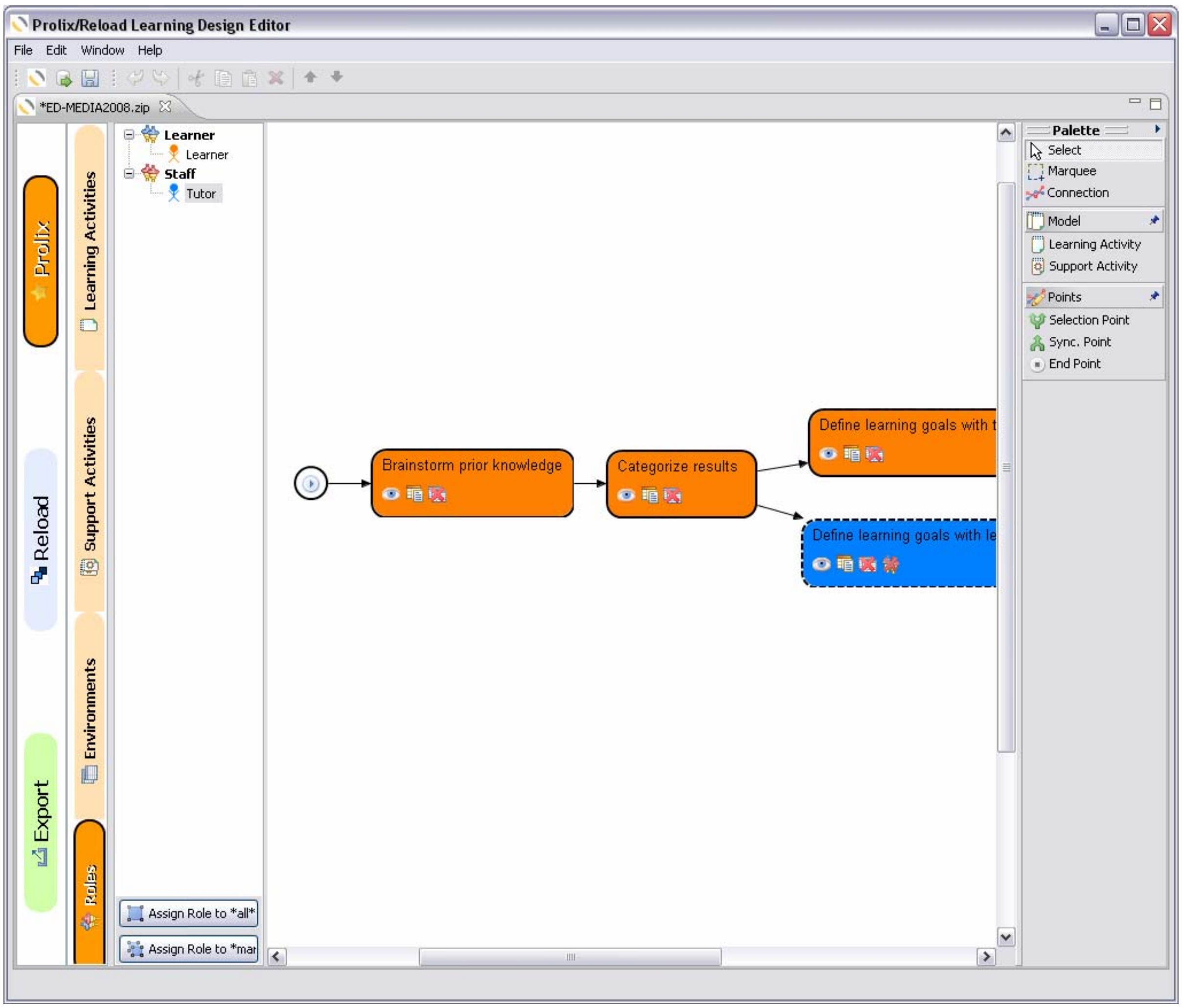

Figure 4: Total view of the Graphical Learning Modeler.

[2] http://www.eclipse.org/gef/
[3] The imsmanifest.xml is the part of the unit of learning content package containing the Learning Design.

The screenshot in Fig. 4 shows the total view of the GLM. On the left, you see two vertical navigation bars, which allow the designer to switch between the Prolix (i.e. GLM) and the original Reload view as well as between the subsidiary elements of each view like activities, roles, and environments[4]. To its right is displayed the instances tree of the currently selected view displaying the already created roles (the tree can be changed by using the navigation bar, e.g. showing all learning activities created). Roles and environments from the instances trees may be dragged & dropped onto activities in the workspace to assign them. The largest part of the GLM is the workspace in the middle, where the learning designer creates the sequence of activities. Activities are color-coded, where the color of the activity matches the color of the role performing that activity. Last but not least, on the far upper right, the design palette is seen that provides design and altering functions such as the creation of activities, or the drawing of connections between activities. Models and points from the palette may be dragged & dropped onto the workspace.

Compared with the Reload editor, the GLM displays the activities performed by each role as well as the sequence of activities in a single comprehensive view. Also, the association of roles to their corresponding activities (technically realized within a role-part) is heightened in the GLM as compared to Reload, since the learning designer does not have to look in multiple lists for this specific association (cp. Fig. 3, which visualizes the same setup of the learner performing the activity “Brainstorm prior knowledge” as is being displayed in Fig. 4).

Providing a technically reduced language means that translation mechanisms have to be established in order to adequately describe graphically designed scenarios in terms of IMS LD. The GLM performs this task by interpreting the graphical learning sequence according to a set of if-then-else rules we developed. The rules were created according to typical setups that emerge in the design of learning and teaching scenarios. The rules are currently constructed for levels A and B of the specification.

Level B Developments

The complexity of the IMS LD specification is exponentially greater at level B than at level A demanding an even higher technical understanding from instructional designers. At level B, the GLM conceals *properties* and *conditions*, which are technical elements used to store data and control data, respectively. Parts of the level B functionalities can be interpreted directly from the graphical sequence of activities. For these functionalities, the GLM uses the already mentioned if-then-else rules to orchestrate multiple parallel activity sequences of different roles (using *conditions*), e.g. when a role has to wait with the start of its activity until another has finished.

The if-then-else rules, however, only cover a portion of the complexity that level B allows. For the functions that cannot be detected automatically from the workspace, the learning designer has to determine them him- or herself. In this regard, the goal was again to stay as far away as possible from the technical specification and to approximate the instructional designers’ concepts and language. Therefore, a set of templates (interactions) and dialogs was developed that allows the instructional designer to make use of level B concepts without explicitly facing *properties* and *conditions*. Prerequisites to these dialog developments were extensive analyses of existing units of learning designed by IMS LD experts as well as correlations of these results with the requirements of typical pedagogic models. The results of the technical analysis are described in detail in Heyer et al. (2007).

Level B functionalities are drawn upon when persons in the unit of learning enter or change data during runtime, and these data are then possibly used to trigger other events to cause adjustments in the learning setting. For instance, if a learning design specifies that learners are to write a text where they state their prior knowledge, and this text is then displayed to someone else for assessment, level B functionalities must be employed to adequately express this setup. The screenshot in Fig. 5 shows the level B interactions that the GLM currently provides: writing/editing a text, working with external files, administering a text-based question & answer test, and administering multiple-choice tests. The symbols placed in front of interactions and actions are placeholders as this represents work in progress.

---

[4] Environments are containers for learning materials, i.e. learning objects, or learning services such as chat and forum that are needed to perform an activity.

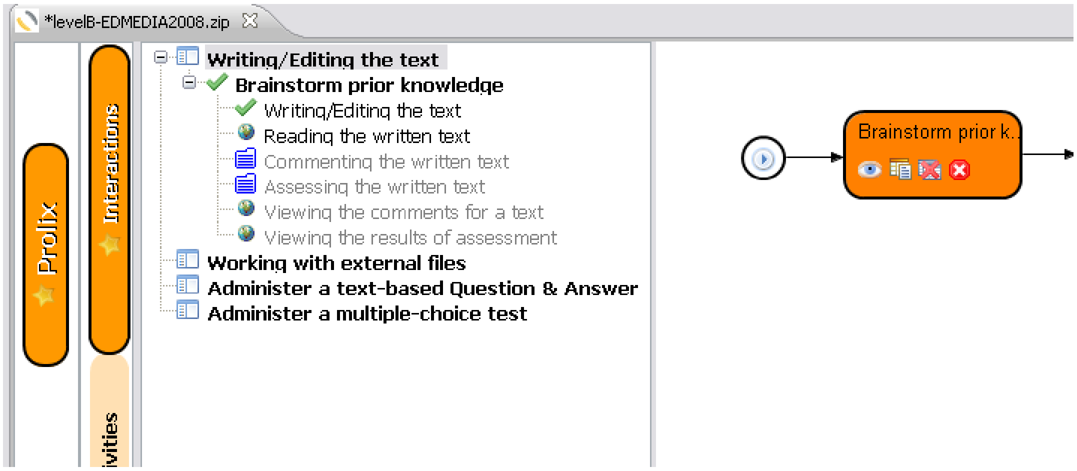

Figure 5: Partial view of the GLM with the level B extended interaction Writing/Editing text

The root of the tree shown in Fig. 5 specifies the type of interaction, e.g. “Writing/editing the text”. The learning designer may create new instances of this type of interaction by right-clicking it and selecting “new writing/editing text”. The learning designer then gives this new instance a name that identifies the text, e.g. in Fig. 5 the name is “Brainstorm prior knowledge”. Looking one level below the name, all the actions that may be performed with this text are listed (e.g. “Reading the written text”). The learning designer drags & drops the actions onto any activity within the workspace to indicate where the action is to be performed. An action may be dragged & dropped several times, for instance, if the written text is to be displayed more than once within the learning sequence. If an action is grayed out in the tree, e.g. in Fig. 5 “Assessing the written text”, then this action may not be dragged & dropped as a needed prerequisite action such as “writing/editing text” has not yet been assigned to an activity.

Validation

We have collected feedback for validation of the GLM with three different communities of users: pedagogical experts, PROLIX test bed partners, and IMS LD tool developers. A forth community to expect feedback from is the sourceforge community (www.sourceforge.net) as we have posted a GLM version for download on this open source community. All together, the feedback collected was positive. Especially the drag & drop functionalities and the quick setup of an entire learning design were well received. The different groups expressed different wishes for adjustments, which were used in the extended developments of the GLM. The feedback is summarized in the following sections.

Feedback from Pedagogical Experts

An early version of the GLM was tested with a group of pedagogical experts. None of them were knowledgeable of the IMS LD specification. The test users were given verbal descriptions of three differing learning and teaching scenarios, which they had to transfer into the GLM. Furthermore, the test users were shown an early animation of a stand-alone level B wizard, which they had to manipulate according to three more prescribed textual scenarios.

The usability test revealed that all test users could successfully and rather quickly complete the transfer of the three scenarios given to them. Their designs were accurate representations of the intended designs. The users applauded the easy use of the drag & drop functionalities. Critical feedback was given regarding the granularity of designs: pedagogical experts wanted to start from an overarching concept, for instance blended learning, and then go down into the specifics of the design, i.e. the concrete descriptions of individual sessions and then of individual learning activities. This included the wish for a zooming-in and zooming-out feature to switch between the overarching phases and the concrete design descriptions. The GLM, however, forces them to start at the granular level. One solution to work against this is the provision of overarching templates, which contain prepared sequences of activities, and which are specially marked in the workspace. We have already adopted this idea and will offer such templates below the GLM palette (cp. Fig. 4). Furthermore, the pedagogical experts were wishing for an annotating function within the GLM to make their design more vivid and to help explain their reasoning for the learning sequence.

The early animation of a level B wizard proved not ready for implementation as it was too strongly focused on the technical language used in the IMS LD specification. As a result of this feedback, an in-depth analysis of level B was executed (Heyer et al. 2007); the so obtained results led to the redesign of providing level B functionalities as described in the previous section.

Feedback from PROLIX Test Bed Partners

Within the PRocess-Oriented Learning and Information eXchange (PROLIX) project, where the GLM developments take place, the GLM was evaluated by its four industrial test bed partners according to the ISO 9241/110 (revised 9241/10) standard that covers seven dialogue principles (Faltin 2008):

- *Suitability for the task*: Does the software support the user in performing his or her tasks?
- *Self descriptiveness*: Is the software self-explaining and understandable?
- *Controllability*: Can you influence how to work with the software?
- *Conformity with user expectations*: Does the software fit your expectations and habits by using a consistent and coherent design?
- *Error tolerance*: How easily can the user recover from the errors?
- *Suitability for individualization*: Can you adapt the software to your personal needs without great effort?
- *Suitability for learning*: Does the design of the software provide an easy way of learning new functions?

The evaluation was performed by the Fraunhofer Institute in Stuttgart. Each dialogue principle is measured by a set of questions that the test bed partners had to answer. The overall rating for the GLM across all dialogue principles was a slightly positive value of 3,11. Fig. 6 shows the individual ratings for each dialogue principle, which are generally neutral (around the value of 3) or above neutral (above a value of 3).

Looking at the specific questions for each dialogue principle, the highest area for improving the GLM is to let users adjust the terminology in the software to their own needs, and to help the users recover from errors. The most positive values that the GLM received were for the setup and usability of the software such as the arrangement of interactive fields, the ability to interrupt actions, to switch between actions and to return to the main menu at any point in time.

Note that this evaluation is based on a previous prototype of the GLM. We expect the ratings to rise, since improvements have been made to the GLM such as the inclusion of cheat sheets and error messages as well as a differentiation between a simple and an advanced view when entering information within the GLM. Further improvements suggested by the test bed partners such as placing navigation buttons horizontally instead of vertically and having the design palette opened by default have already been looked after and debugged.

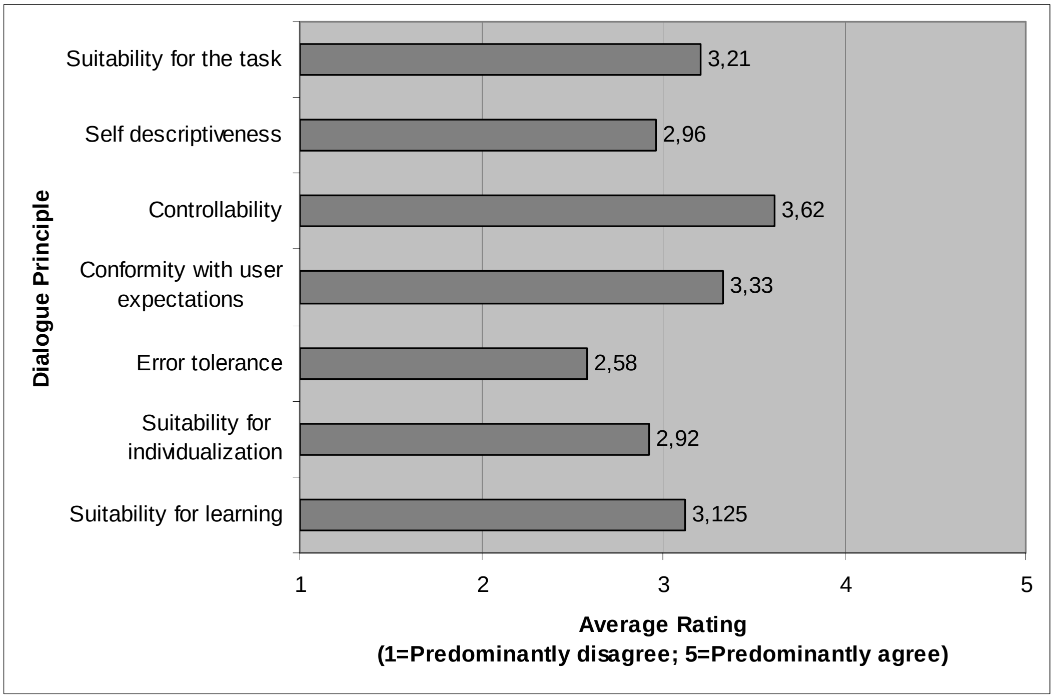

Figure 6: Average ratings for GLM from PROLIX test bed partners regarding ISO dialogue principles (adapted from Faltin 2008, p. 16)

Feedback from IMS LD Tool Developers and Experts

The following projects that also develop software related to IMS LD (either editors or players) were asked for their feedback: TENCompetence[5], UNFOLD[6], Collage[7] and Gradient/GRAIL[8]. Staff members, usually development staff, of these projects were given a formal feedback form to fill out. There were five responses to the feedback request (at least one from each project). The ratings for each statement are summarized in Fig. 7. The statements in the feedback form were positively stated, so that a low rating (below 3) indicates a positive attitude towards the GLM. For displaying purposes, Fig. 7 merely shows the essence of each statement rather than the complete statement as it appeared in the feedback form.

As Fig. 7 shows, the GLM developments were generally positively received as the majority of ratings is at or below value 3. One developer critiqued in the free feedback section that the pedagogical semantics are not visible in the way the GLM represents the activity sequence. We have already addressed this issue in regard to the pedagogical experts' feedback. Nevertheless, the IMS LD specification does not provide adequate terminology in its language to technically distinguish pedagogical approaches. Even if options were provided to make the pedagogical approach more apparent within the GLM, this additional information would only have value within the GLM environment, since it could not be translated into the IMS LD terminology and manifest. Integrating specific sets of metadata might be an option to circumvent this.

[5] http://www.tencompetence.org/
[6] http://www.unfold-project.net:8085/UNFOLD
[7] http://gsic.tel.uva.es/collage
[8] https://gradient.it.uc3m.es/xowiki/main_page

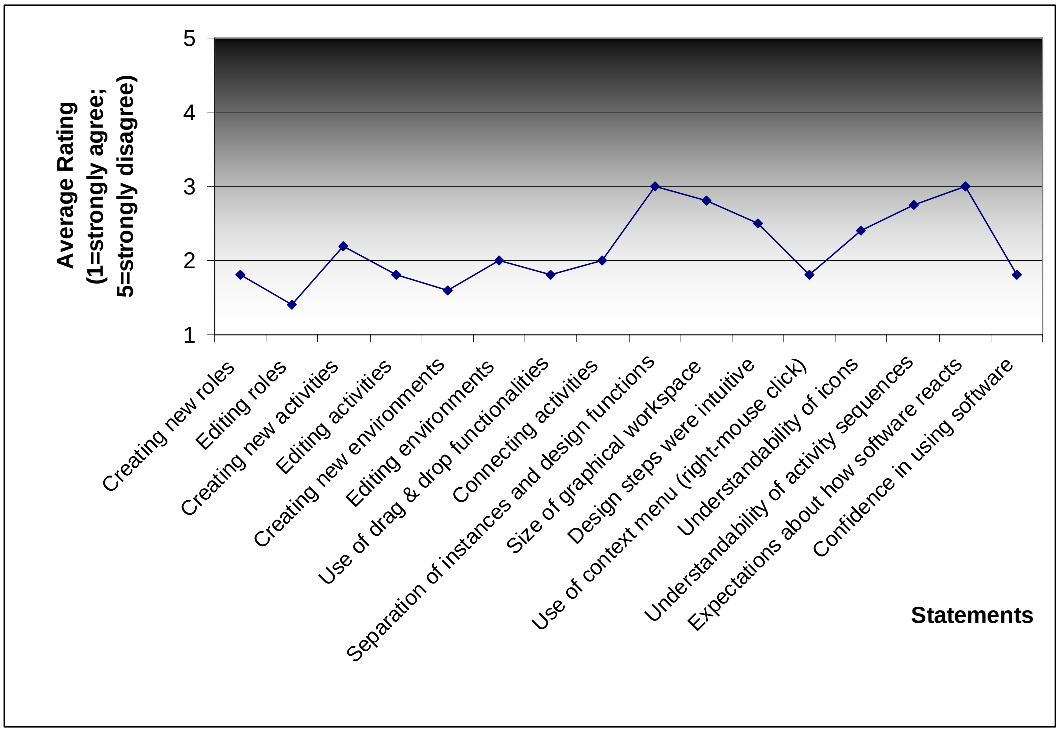

Figure 7: Average ratings for GLM from IMS LD tool developers and experts

Future Work and Implications for Others

Next steps regarding the development of the GLM include the further extension of level B functionalities. At this point, only a fraction of the possibilities that level B holds has been implemented. In the near future, focus will be placed on the definition of alternate learning paths, which are offered depending on events or learning outcomes that take place within the unit of learning, e.g. a learner finishes the multiple-choice test with a certain score and is shown different activities depending on that score.

As already mentioned, the GLM will offer readily designed learning design templates that incorporate specific pedagogical approaches. Preparations for the provision of these models have been completed resulting in a list of over thirty approaches. Next steps in this regard involve their integration within the GLM below the design palette (seen on the right in Fig. 4). Tests will be run to identify whether the usability and pedagogical usefulness of the GLM are improved.

The GLM has implications for developers of similar software, particularly because the graphical visualization of learning designs becomes increasingly important. For instance, the Learning Activity Management System (LAMS)[9] and Collaborative Learning Design Editor (Collage)[10] already implement graphical interfaces for learning design, while the Reload editor developers are working to integrate graphical interfaces (Barrett-Baxendale 2007). The GLM sets standards as other LD editors have yet to provide the flexibility of an open design environment outside of restrictive templates that simultaneously move away from the technical language of IMS LD. The GLM is therefore best suited for novices of IMS LD, who wish to design with a great degree of freedom.

---

[9] http://www.lamsinternational.com/
[10] http://sourceforge.net/projects/collage/

Acknowledgements

This article was written in the context of the research and development integrated project PROLIX, which is co-funded by the European Commission under 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 headed by Wolfgang Klas.

References

Advanced Distributed Learning (ADL). (2006). *Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Overview*. Alexandria, VA.

Barrett-Baxendale, M. (2007). New Version of Reload IMS LD editor. Retrieved December 18, 2007, from http://ld4p.blogspot.com/2007/10/new-version-of-reload-ims-ld-editor.html

Beetham, H. (2004). Review: developing e-Learning Models for the JISC Practitioner Communities, Version 2.1. Retrieved March 13, 2007, from http://www.jisc.ac.uk/uploaded_documents/Review%20models.doc

Beetham, H., & Sharpe, R. (2007). An introduction to rethinking pedagogy for a digital age. In H. Beetham & R. Sharpe (Eds.), *Rethinking Pedagogy for a Digital Age: Designing and delivering e-learning* (pp. 1-10). London, New York: Routledge.

Britain, S. (2007). Learning design systems: Current and future developments. In H. Beetham & R. Sharpe (Eds.), *Rethinking Pedagogy for a Digital Age: Designing and delivering e-learning* (pp. 103-114). London, New York: Routledge.

Faltin, N. (Ed.). (2008). *D9.3 PROLIX prototype version – pilot evaluation.* Saarbrücken: Process-oriented Learning and Information eXchange.

Griffiths, D., & Blat, J. (2005). The Role of Teachers in Editing and Authoring Units of Learning Using IMS Learning Design. *Advanced Technology for Learning, 2*(4).

Heyer, S., Oberhuemer, P., Zander, S., & Prenner, P. (2007). Making Sense of IMS Learning Design Level B: from specification to intuitive modeling software. In E. Duval, R. Klamma & M. Wolpers (Eds.), *Lecture Notes in Computer Science 4753* (pp. 86-100). Berlin, Heidelberg: Springer.

IEEE. (2002). Draft Standard for Learning Object Metadata. Retrieved December 14, 2007, from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

Klebl, M. (2004). Lehrprozesse planen, Lernprozesse strukturieren mit IMS Learning Design. Retrieved December 14, 2007, from http://dspace.ou.nl/handle/1820/234

Koper, R. (2005). An Introduction to Learning Design. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 3-20). Berlin, Heidelberg: Springer.

Koper, R., Olivier, B., & Anderson, T. (Eds.). (2003). *IMS Learning Design Information Model*: IMS Global Learning Consortium.

Sloep, P. (2004). The Language of Flexible Reuse; Reuse, Portability and Interoperability of Learning Content or Why an Educational Modelling Language. In R. McGreal (Ed.), *Online Education Using Learning Objects* (pp. 128-137). London: Routledge/Falmer.