

MICHAEL DERNTL

## 11. OPENGLM

### *Integrating Open Educational Resources in IMS Learning Design Authoring*

#### INTRODUCTION

The Open Graphical Learning Modeller (OpenGLM) is a learning design authoring toolkit that supports the authoring of IMS Learning Design (LD) (IMS Global, 2003) units of learning. IMS LD is a complex specification that allows learning designers to define the flow of teaching and learning activities in a unit of learning along with required services and learning objects. A unit of learning described using IMS LD authoring software can subsequently be deployed and used over and over again in any IMS LD compliant learning environment.

The main goal of developing OpenGLM was to provide comprehensive and intuitive IMS LD modelling 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 XML format as specified in the IMS LD information model. These goals were achieved by viewing the activities of learners and instructors as the modelling 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.

Using OpenGLM, teaching practitioners are enabled to intuitively create units of learning to be played in IMS LD enabled learning management systems. A new educational opportunity is created as the barrier for access is lowered, and thus the number of learning designers that produce IMS LD conformant units of learning may be increased; more units of learning may then be produced, exchanged, and evaluated as was one of the original goals of the IMS LD specification (Koper & Olivier, 2004). To support the reader in understanding of how typical course planning and learning activity management steps are supported during authoring in OpenGLM, Table 1 matches typical course planning steps to the corresponding IMS LD authoring steps in OpenGLM. The typical course planning steps were obtained in a study with teachers (Derntl, Neumann, Griffiths, & Oberhuemer, 2011, p. 19–23) and the top ten steps were used for constructing the table. More details on

the concepts used in IMS LD and in OpenGLM are given in the subsequent sections of this Chapter.

The Chapter is structured as follows. We first outline the technical background and development history of OpenGLM. We then go on to describe how learning design concepts and processes are represented as features in the authoring tool, with a focus on integration of open educational resources in the designs. We describe how this facilitates sharing and reuse of resources and designed units of learning, and eventually review how the tool and its features were evaluated in several previous end-user evaluations.

*Table 1. Relating typical course planning steps to authoring steps in OpenGLM*

<i>Course Planning Step</i>	<i>OpenGLM Authoring Step</i>
Design/select materials	Learning object creation, import, and management
Define content-oriented course structure	Content structure is defined via activities
Learning outcome definition; needs analysis	Learning outcome definition is possible at the learning design and activity levels. Prerequisites can also be included. Needs analysis is performed outside of OpenGLM.
Design teaching method/learning activities/task	Define learning and support activities, create environments (learning objects and services), and assign roles to activities
Set up learning management system	Export IMS LD compliant package and create a run in IMS LD player
Define time structure of course	Not possible in terms of calendar dates. Time limits can be assigned to activities
Design assessment method/assessment resources	This is achieved as part of regular activities and learning environments in OpenGLM
Choose course topic(s)	Not applicable; outside OpenGLM
Look at course description in curriculum	Not applicable; outside OpenGLM
Provide administrative course data (institution's course data base etc.)	Not applicable; outside OpenGLM

#### DEVELOPMENT CONTEXT AND TECHNICAL BACKGROUND

OpenGLM was developed in the context of the ICOPER project<sup>1</sup>. It is open source software, available for download from SourceForge<sup>2</sup>. There are platform specific binaries available for Windows, Mac OS X, and Linux. It is a cross-platform Java

application based on Graphical Learning Modeller (GLM; Neumann & Oberhuemer, 2009), which was developed on top of the Reload Learning Design Editor's Java code base in the EU project Prolix<sup>3</sup>. Reload LD Editor (Griffiths et al., 2007) 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. OpenGLM thus builds on a stack of existing code developed in previous R&D projects. OpenGLM's main add-ons to the original GLM include enhancements for supporting communities of practice in sharing IMS LD units of learning along with standardised learning outcome definitions by providing built-in features for search, import from and export to a large online repository—the Open ICOPER Content Space (OICS).<sup>4</sup> The OICS is a repository for different types of educational resources containing about 80,000 openly accessible objects contributed by content providers from all over the world, including OU's OpenLearn, OER Commons, MIT OpenCourseWare, to name a few.

#### LEARNING DESIGN CONCEPTS AND FEATURES IN OPENGLM

OpenGLM supports IMS LD levels A and B. Level A provides the core elements of the specification allowing the definition of activity sequences and their learning environments. Level B adds the concepts of properties and conditions which are useful for modelling the input to and output of learning and support activities and controlling more complex activity flows using conditional expressions, respectively. Level C adds notifications, enabling to dynamically control the assignment of roles to activities based on events. Since most learning designs can be modelled with levels A and B, level C was not implemented in OpenGLM. Of course, supporting the authoring of level C elements can be added to OpenGLM without limiting the existing functionality. For the current version of OpenGLM, the main goal was to provide a visual modelling metaphor that hides the complex aspects of IMS LD. For instance, OpenGLM does not confront the user with IMS LD concepts like plays, acts, and properties. The OpenGLM main window is presented in Figure 1.

The main window is organised into three panes: The left navigation pane contains shortcuts to all OpenGLM features; the centre pane contains the actual content in the context of the selection in the left pane; and the right pane contains the modelling palette and a set of ready-to-reuse teaching methods. The right pane is only visible in the modelling mode (i.e. the orange-coloured part of the navigation pane).

OpenGLM's visual modelling metaphor for IMS LD was conceived as described in the remainder of this section, whereby initial appearances of terms referring to IMS LD elements are printed in italics.

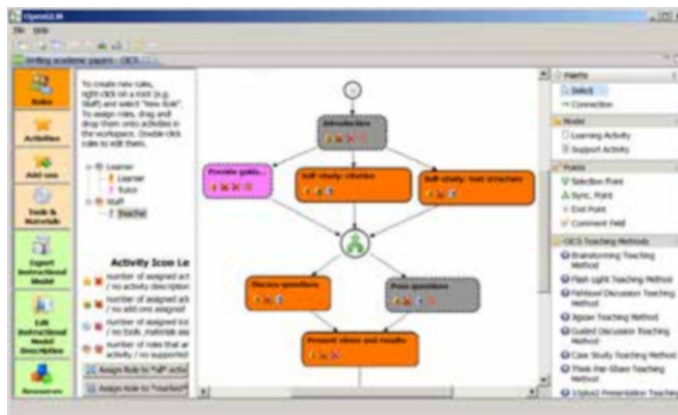


Figure 1. OpenGLM main screen

### *Learning and Support Activities*

Activities are represented as rectangular symbols carrying the title of the activity and small icons referring to the activity's contents such as linked *activity descriptions*, *learning objects*, etc. The activity's fill colour reflects the colour of the role that is associated with it. This makes it easier for the designer not only to recognise at a glance which activities have roles assigned (this is a requirement of IMS LD), but also which particular roles are assigned. The colour code also facilitates a quick recognition of the overall distribution of roles over activities in the modelling workspace. A *learning activity* has a solid bounding box, while a *support activity* has a dashed bounding box.

Details of the activity can be edited by double clicking the activity symbol. In the edit dialogue (see Figure 2), it is possible to provide activity descriptions and other settings, as well as adding learning objects and *services* that are used by the activity. The terminology from the IMS LD specification was adapted for some of the concepts with more intuitive terms like add-ons, tools, and materials. The fact that IMS LD requires learning objects and services to be contained in *environments* is hidden from the user; the environment is created automatically (without any visual representation) when learning objects or services are added. Frequently used interactive activities like uploading files, writing a piece of text, etc., are offered without mentioning the *property* concept, which is used in IMS LD to capture role or person related data at runtime.

Learning objects can be added as *resources* either from the web via their hyperlink or as local, physical files. OpenGLM additionally allows the user to search for learning objects and other resources on the OICS and to add those resources to the *unit of learning* (see Figure 3).

Roles are represented as stick figures (see Figure 1); each role has a title and a colour, which can be defined by the modeller. Roles can be assigned to activities simply by dragging the stick figure and dropping it onto the activity symbol.

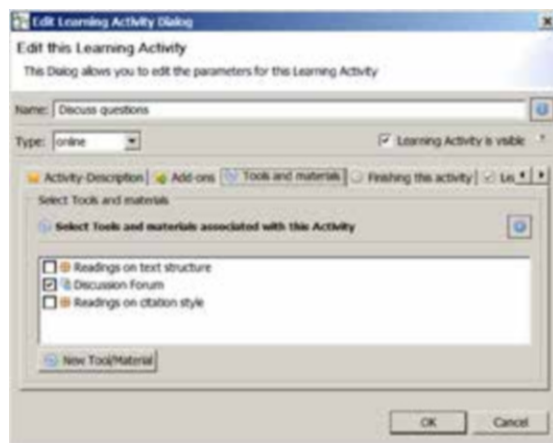


Figure 2. Learning activity editing dialogue

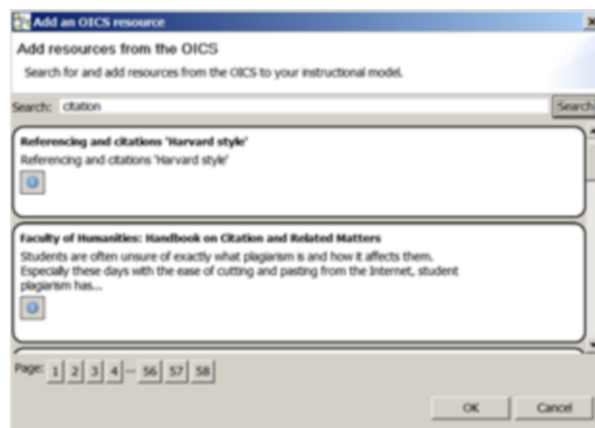


Figure 3. Adding a learning object from the Open ICOPER Content Space

### Activity Flow

The flow of learning and support activities can be specified by connecting the activities with one of the routing symbols in the palette pane on the right-hand side (see Figure 1). This can be a connection (displayed as an arrow connecting the source

activity with the target activity) as well as selection (fork), synchronisation (join), and end points. From the sequenced elements used in the unit of learning OpenGLM automatically creates the *play*, the required *acts*, and the *activity structures* for the forks and joins. However, these IMS LD elements are not presented to the learning designer in the user interface.

To facilitate the learning designer in building the unit of learning based on good-practice teaching methods, OpenGLM enables dragging one of the pre-defined teaching methods from the right pane (see Figure 1) and dropping it onto the modelling pane. These good-practice teaching methods are stored in the OICS in the form of IMS LD packages. Upon being dragged and dropped to the modelling pane, OpenGLM downloads the teaching method and places the contained activity sequence into the current unit of learning. The user can then adapt the pre-defined sequence and/or integrate it with the current set of activities.

#### *Metadata and Learning Outcomes*

Of course it is possible to edit the general descriptive metadata for the unit of learning, including the title, version, description, rights, prerequisites, learning outcomes, and other elements defined in IMS LD. However, OpenGLM goes one step further when it comes to defining the intended learning outcomes (see Figure 4; in IMS LD learning outcomes are called learning objectives) by providing the following features:

- Searching the OICS learning outcome repositories for existing learning outcome definitions, which can then be added as intended learning outcomes for the current unit of learning (see Figure 5).
- When the user creates a new learning outcome (see Figure 6), the newly created learning outcome definition is not only added to the current unit of learning as an intended outcome, it is also sent to a learning outcomes repository on the OICS, allowing it to be reused by other learning designers in other units of learning.
- While in IMS LD a learning objective can be provided as any kind of resource (e.g. plain text, binary document, etc.), OpenGLM adopted the IEEE Reusable Competency Definition (RCD) (IEEE, 2008) specification by describing each learning outcome with title, description, and type. This decision was made after a review of existing learning outcome specifications (Najjar & Klobucar, 2009) in the eContentplus project ICOPER, which supported the development of OpenGLM. Moreover, OpenGLM allows the learning designer to define for each learning outcome the proficiency level of the outcome according to the numeric scheme introduced by the European Qualification Framework for lifelong learning (EQF) (European Commission, 2008).

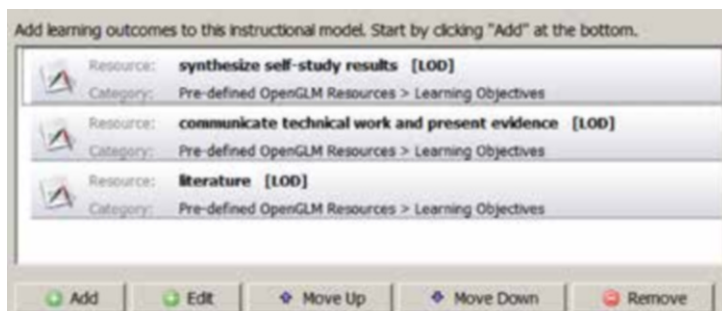


Figure 4. Overview of intended learning outcomes

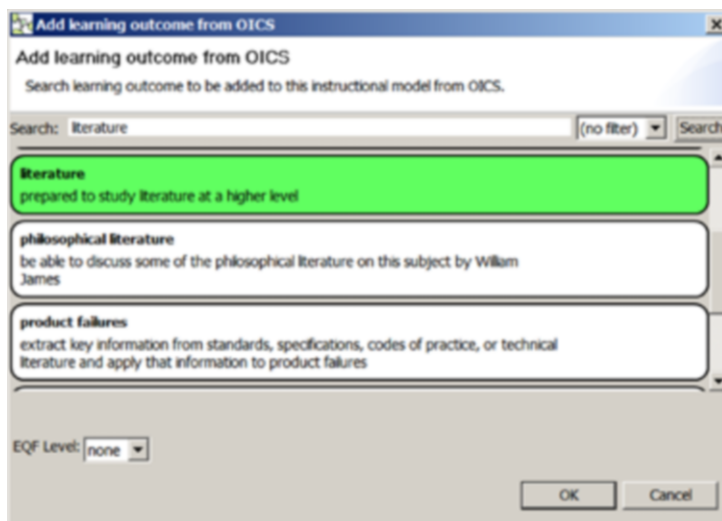


Figure 5. Searching for an existing learning outcome definition in the learning outcome repository

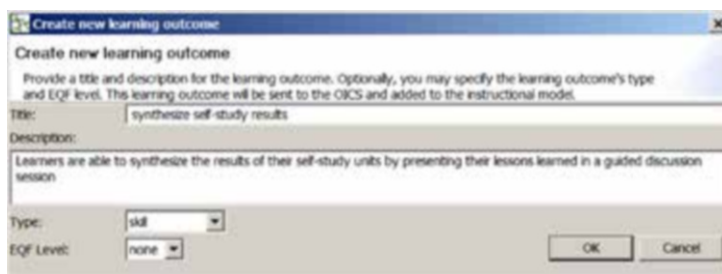


Figure 6. Defining a new intended learning outcome

### *Exporting Learning Designs*

At any time, the learning designer may save the unit of learning and export it into an IMS LD compliant ZIP package. If there are any errors in the unit of learning (e.g. activities that are not connected to any other activity), OpenGLM will issue an error message with simple non-technical explanations.

If there are no errors, the unit of learning can be exported as an IMS LD package either on the local computer's hard drive, or to a remote unit of learning repository (see Figure 7). In the latter case, the unit of learning can be uploaded to any collection on the remote repository where the current user has write privileges. All other users with read privileges will subsequently be able to find and import this unit of learning into their own OpenGLM environment. By supporting this kind of online repository-based sharing, learning design communities of practice at individual and organisational level are provided with a powerful toolkit to manage their shared units of learning.



Figure 7. Exporting the unit of learning to the Open ICOPER Content Space

### *Searching and Importing*

As mentioned earlier, OpenGLM offers features for searching in and importing from the remote unit of learning repository. The search dialogue window (see Figure 8) offers keyword based search in the full Learning Object Metadata (LOM; IEEE, 2002) record of the unit of learning. For instance, this can be used to search for units of learning that are licensed under a particular license, which is captured in the "Rights" category of the LOM standard. The search dialogue also allows to define special search filters so the keywords are matched in specific parts of the units of learning metadata, e.g. the intended learning outcomes, implemented teaching methods, or simply in the title and description. The units of learning that match the query are displayed in the result box. By clicking on the information icon, OpenGLM displays all information on the selected unit of learning, like full description, learning outcomes, end user language, licensing, and so forth. By clicking on the import button, the selected unit of learning is downloaded from the repository and visualised in the modelling pane. This feature enables learning



design communities to build on each other's units of learning instead of starting from scratch every time.

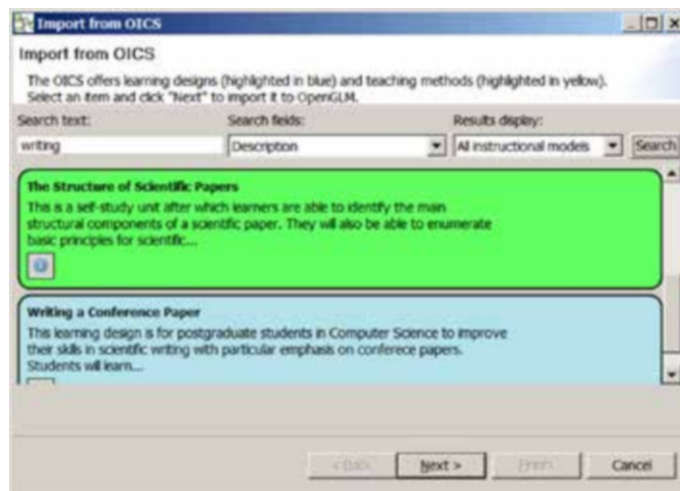


Figure 8. Searching and importing an existing unit of learning from the Open ICOPER Content Space

#### OPENGLM EVALUATIONS

OpenGLM and its predecessor GLM have been used by and for test-bed partners in large European projects. These include PROLIX, an integrated project in the Sixth Framework Programme on aligning professional learning with business processes and business requirements where GLM was initially developed. The OpenGLM add-ons were designed and implemented in the eContentplus project ICOPER, a best practice network on standards and specifications in learning outcome based education and educational content. These projects were the context for extensive evaluations of the authoring tool and its integration with educational practice.

To understand these evaluations, it is necessary to recall that OpenGLM was built on top of GLM, i.e. the visual IMS LD modelling metaphor was implemented in GLM, while the sharing use cases and the remote repository and metadata-related features were added in OpenGLM. Since the evaluations have been published previously elsewhere, in this Chapter we will provide an overview of the results and findings of these studies and point to the original papers for full details.

First studies evaluating GLM (Neumann & Oberhuemer, 2008; 2009) addressed different stakeholders of the tool including pedagogical experts, test-bed partners, IMS LD developers, and instructors as end-users. The studies revealed that the pedagogical experts without any knowledge of IMS LD were able to model given scenarios using OpenGLM; they were particularly fond of the drag and drop

functionalities. The industrial test-bed partners in PROLIX evaluated the tool according to ISO standards for user dialogue principles. This evaluation produced mixed results, presumably due to an early incomplete version of the tool having been evaluated. IMS LD tool developers and experts had a highly positive opinion of the many aspects of IMS LD realization in GLM. GLM was also evaluated with a sample of 21 higher education instructors, who had to create units of learning using with the tool. While the instructors were generally successful in creating the units of learning, they had several suggestions for improvement, some of which were subsequently implemented, for instance the provision of templates or activity design patterns to reuse.

An end user evaluation focusing on opportunities and challenges of formal instructional modelling by example of OpenGLM was presented in Derntl, Neumann and Oberhuemer (2011b). The study used a given instructional design task structure for participants and then administered structured interviews with open-ended questions. The results revealed that users generally perceived a smooth authoring process in OpenGLM, however they did report some issues related to the terminology used in the tool, the reuse of existing resources and the integration with the Open ICOPER Content Space for import, export and reuse of teaching methods. In addition to these technical difficulties, users seem to have problems with the terminology used in OpenGLM, or with learning design concepts and terminology in general. The study also showed that participants were also questioning whether such formal design activities have any impact or relevance on their life as a teacher, since there is typically no appropriate reward by higher education institutions for spending effort on improving teaching.

The technical realization of standards-based sharing and reuse were discussed in detail in Derntl, Neumann and Oberhuemer (2011a), also focusing on the role of metadata and the key artefacts involved in such sharing processes. In this study several use cases and scenarios for communities of learning design practice are presented, such as searching for instructional models, annotating instructional models, and some learning outcome related scenarios. The study concludes that while such sharing scenarios are relevant to support teacher-designer communities, current academic practice is far from adoption, and that it would require commitment of practitioners and their institutions to move forward.

## CONCLUSION

In this paper we introduced the Open Graphical Learning Modeller (OpenGLM), a learning design authoring tool that supports IMS LD levels A and B. OpenGLM intends to support the learning designer with an intuitive visual modelling metaphor that conceals the complex elements of IMS LD in the user interface, while still supporting these concepts “under the hood”. The tool is open source and available for all major operating systems.

One of the main advantages of OpenGLM is its support for communities of practice in the spirit of Web 2.0: the most important artefacts used and produced during unit of learning authoring can be searched, retrieved, and published in the Open ICOPER Content Space (OICS), an open, federated repository for educational resources. Collections within OICS can have a fine-grained hierarchy and privilege model, to support individual and organisational use cases.

The reviewed studies about the use of OpenGLM involving practitioners in real-world contexts reveal that these design-sharing tools do have a lot of potential in facilitating and improving the design process, and ultimately also the quality of teaching. However, it was also found that academic institutions and teaching practitioners are still reluctant to adopt such innovations and practices. Partially this is because the tools and processes are not mature enough yet, and not tailored to specific usage contexts and scenarios. Another reason that was identified is the general lack of a reward system in academic institutions for individuals who strive to propel excellence in teaching and learning design.

#### ACKNOWLEDGMENTS

I greatly appreciate the work that was jointly put into developing OpenGLM with my colleagues Susanne Neumann, Petra Oberhuemer and Philipp Prenner at the University of Vienna. The development of OpenGLM was mainly supported by the EU *eContentplus* best practice network ICOPER (ECP 2007 EDU 417007; <http://icoper.org>), then by the FP7 ICT support action TEL-Map (FP7-257822; <http://telmap.org>), and most recently by the EACEA Lifelong Learning Programme multilateral project METIS (531262-LLP-2012-ES-KA3-KA3MP; <http://metis-project.org>). This work reflects the views only of the author, and the European Commission cannot be held responsible for any use which may be made of the information contained therein.

#### NOTES

- <sup>1</sup> <http://icoper.org>
- <sup>2</sup> <http://sourceforge.net/projects/openglm>
- <sup>3</sup> <http://prolix-project.eu>
- <sup>4</sup> <http://icoper.org/oics>

#### REFERENCES

- Derntl, M., Neumann, S., & Oberhuemer, P. (2011a). Community support for authoring, sharing, and reusing instructional models: The Open Graphical Learning Modeler (OpenGLM). Proceedings of IEEE ICALT 2011, Athens, Georgia, USA: IEEE.
- Derntl, M., Neumann, S., & Oberhuemer, P. (2011b). Opportunities and challenges of formal instructional modeling for web-based learning. Proceedings of ICWL 2011, Hong Kong: Springer-Verlag.
- Derntl, M., Neumann, S., Griffiths, D., & Oberhuemer, P. (2011). Report on usage of and recommendations for instructional modeling specifications. ICOPER Deliverable D3.2. Retrieved from <http://is.gd/icoperd32>

M. DERNTL

- European Commission (Ed.) (2008). *The European qualifications framework for lifelong learning*. Retrieved from <http://is.gd/eqf2008>
- Griffiths, D., Beauvoir, P., Baxendale, M., Hazlewood, P., & Oddie, A. (2007). *Development and evaluation of the reload learning design editor*: Paper presented at TENCompetence Open Workshop on Current research on IMS Learning Design and Lifelong Competence Development Infrastructures, June 21–22, Barcelona, Spain.
- IEEE. (2002). IEEE standard for learning object metadata (1484.12.1-2002).
- IEEE. (2008). IEEE standard for learning technology – Data model for reusable competency definitions (1484.20.1-2007).
- IMS Global: IMS Learning Design Information Model. (2003). Retrieved from. <http://is.gd/imslidspec>
- Koper, R., & Olivier, B. (2004). Representing the learning design of units of learning. *Educational Technology & Society*, 7(3), 97–111.
- Najjar, J., & Klobucar, T. (Eds.). (2009). Report of the study of learning needs building blocks and the standards to be used. *ICOPER Deliverable D2.1*. Retrieved from [http://icoper.org/deliverables/ICOPER\\_D2.1.pdf](http://icoper.org/deliverables/ICOPER_D2.1.pdf)
- Neumann, S., & Oberhuemer, P. (2008). *Supporting instructors in creating standard conformant learning designs: The graphical learning modeller*. In 20th World conference on educational multimedia, hypermedia & telecommunications, Vienna, Austria.
- Neumann, S., & Oberhuemer, P. (2009). User evaluation of a graphical modeling tool for IMS learning design. In M. Spaniol, Q. Li, R. Klamma, & R. W. H. Lau (Eds.), *Advances in Web Based Learning – ICWL 2009, 8th International Conference, Aachen, Germany, August 19–21. Proceedings. Lecture Notes in Computer Science 5686* (pp. 287–296).

*Michael Derntl*  
*Advanced Community Information Systems (ACIS)*  
*RWTH Aachen University, Aachen, Germany*