Using the IMS Learning Design notation for the modelling and delivery of education

DOI:
10.4018/978-1-59904-729-4.ch015

Document status and date:
Published: 01/01/2008

Document Version:
Peer reviewed version

Preprint – please contact authors before citing

Using the IMS Learning Design notation for the modelling and delivery of education

Colin Tattersall, Tim Sodhi, Daniel Burgos, Rob Koper
The Open University of the Netherlands, Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands

Abstract. IMS Learning Design (IMSLD) is a notation system for learning and instruction. It supports the description of learning processes using a set of standardised concepts, including roles, activities, acts, objectives and prerequisites. With the availability of such a notation, descriptions of learning processes can be shared, critiqued, modified, rated, compared and evaluated. Moreover, the machine-interpretable nature of the notation means that designs can be executed by software to support the dynamic orchestration of multi-learner, multi-role learning processes. This chapter introduces IMSLD and describes experience with its use, supported by the first generation of tooling. We then combine these experiences with observations on the tools in the light of new developments in e-learning in order to derive a set of requirements for IMSLD-enabled visual design environments.

1. Introduction

In a recent paper, Merrill (2006) highlights that training is often created by designers-by-assignment without the use of a systematic process, and that instructional products are often designed without sufficient consideration of the applicable instructional design theory. Other research indicates that even when designers are aware of theories, there appears to be a difference between their practice and instructional design models (Eseryel, Schuver-van Blanken, & Spector, 2001; Kenny, Zhang, Schwier, & Campbell, 2005; Kirschner, Carr, Van Merriënboer, & Sloep, 2002). Part of this problem is the absence of a tradition of the use of notations (Gibbons & Brewer, 2005; Koper, 2005; Waters & Gibbons, 2004). A widely adopted notation allows educational processes to be modelled for subsequent sharing, critiquing, modification, execution, rating, comparison and evaluation.

In order to address this issue, several initiatives have been pursued to derive a modelling language for education (Koper & Manderveld, 2004; Rodríguez-Artacho & Verdejo Maíllo, 2004; Süß & Freitag, 2002). The results of these initiatives, notations for describing educational processes, have been input to standardisation processes (Rawlings, Van Rosmalen, Koper, Rodríguez-Artacho, & Lefrere, 2002) and, in 2003, an open technical specification known as IMS Learning Design (IMSLD, 2003), was approved by a consortium of universities, system vendors, providers and other e-learning stakeholders.

In Waters and Gibbons' (2004) terms, IMSLD can be positioned as a notation system. The notation is characterised in (Botturi, Derntl, Boot, & Figl, 2006) as a layered, formal, textual specification offering a single pespective. This chapter describes the IMSLD notation system and reviews experience with its use. We then identify a number of requirements for IMSLD-aware design environments with broad utility, a high degree of usability and support for interoperability.

2. IMS Learning Design: a notation system for education

IMSLD focuses on the creation of a formal description of educational processes known as a Unit of Learning (UoL). A wide variety of pedagogical approaches can be represented by IMS LD, such as problem based learning, competence based learning and game based learning. Prior to turning to the details of the specification, it is helpful to review the requirements the specification was written to meet:

1. Completeness: describe the teaching-learning process in a unit of learning, including references to the digital and non-digital learning objects and services needed during the process.
2. Pedagogical Flexibility: describe different kinds of pedagogies without prescribing any specific pedagogical approach.
3. Personalization: describe personalization aspects within a learning design, so that the content and activities within a unit of learning can be adapted based on the preferences, portfolio, educational needs, and situational circumstances of users. In addition, the control over the adaptation process must be given, as desired, to the student, a staff member, the computer, and/or the designer.
4. Formalization: describe a learning design in the context of a unit of learning in a formal way, so that automatic processing is possible.
5. Reproducibility: describe the learning design abstracted in such a way that repeated execution in different settings with different persons is possible.
6. Interoperability: support interoperability of learning designs.
7. Compatibility: use available standards and specifications where possible.
8. Reusability: identify, isolate, de-contextualize and exchange useful learning artefacts, and to re-use these in other contexts.

The specification meets these requirements by defining a modelling language. The specification defines the various concepts in the IMSLD language in an Information Model – descriptive text together with diagrams of the relationships between the concepts expressed in the Unified Modeling Language (ISO, 2005). Advice and explanation on putting the language to use is includes in the Best Practice and Implementation Guide, and a machine interpretable representation is given as an XML Binding.

The specification prescribes a standardised, flexible language for representing learning scenarios for multiple or individual learners, able to be executed by software responsible for coordinating learners, teachers, learning resources and activities as the learning process progresses. The specification reflects a general model which underlies many different behaviourist, cognitive, and (social) constructivist approaches to learning and instruction: People act in different roles in a teaching-learning process. In these roles, they work towards certain outcomes by performing learning and/or support activities within an environment, consisting of learning objects and services to be used during the performance of the activities. The approach separates learning objects and services from the educational method used in the unit of learning. Put succinctly, IMSLD allows designers to specify who should do what, when and with which support facilities in order to reach learning objectives.

Which concepts, then, does the specification prescribe? Central to the modelling language are the *activities* to be carried out by learners and staff (or other *roles* involved in the learning process). An activity is associated with learning objectives, prerequisites, a description and an *environment.* Environments include the material, tools and facilities needed by learners and staff in order to carry out their activities, and include learning objects (such as documents, explanatory videos and animations) and learning services (such as forums and chat facilities). These core components (roles, activities and environments) are orchestrated in a *method*. Methods use the metaphor of a theatrical play to describe the temporal flow of events, whereby sequential acts are described, with the play ending with the completion of the last act. The transition from one act to another serves as a synchronisation point for the multiple participants, ensuring that they can all start the next act at the same time.

In addition to the basic language constructs, referred to as level A, the specification provides additional concepts to cater for more sophisticated process descriptions. IMSLD levels B and C (Koper & Burgos, 2005) allow the expression of conditions, so that designers may describe circumstances under which specified actions should follow. For example, a designer may which to arrange for the learning process to accommodate different paths through learning activities depending on the results of a self-assessment by learners, or given learners' preferred approach to learning. Additional constructs allow information to be stored during a learning process and used subsequently to influence the flow of events. In this way, designers can arrange for peer review of documents, whereby one learner's contribution is rated and critiqued by peers before being returned to the original author for reflection. The ability to notify roles that an event has happened or that intervention is required is also afforded by the IMSLD modelling language.

The language constructs offered by IMSLD allow a wide range of educational processes to be modelled in a standardised way (Van Es & Koper, 2006). Use of the language provides the basis to rejuvenate e-learning systems, increasing the 'richness' of learning activities. New, more effective, efficient & attractive learning models can be specified (e.g. active learning, problem based learning) giving specific attention to support of the teaching/learning process to decrease workload (particularly that of teachers). Furthermore, the advantages of a standard notation can be realised: reflection, communication, sharing, reuse, research, similarity studies and evaluation.

These benefits are, however, predicated on more factors than those addressed by the open technical specification itself. Just as the HTML specification (W3C, 1999) describes the constructs offered by the HTML language without specifying the nature of the software which interprets it, the IMSLD specification does not address how to record or create the notation, how to adapt or edit it, how to aggregate several use of the notation and other factors involved when putting IMSLD to use.

3. Putting IMS Learning Design to use: processes and tools

Seen in the abstract, e-learning production processes (Koper, 2003) have sub-processes in which Units of Learning are developed and stored, populated with specific learners and teaching staff, and, to continue the theatrical analogy, performed, or 'run'. Since UoLs do not relate to specific individual learners and teaching staff, they can be created once and delivered many times (Tattersall et al., 2005).

The IMSLD specification prescribes the form and structure of Units of Learning so that software applications may be created for their interpretation. As noted above, XML is used as the machine interpretable language in which Units of Learning must be described to be IMSLD compliant. A fragment of the XML representation of a UoL is shown in Figure 1.

```
<imsld:play identifier="P-1" isvisible="true">
   <imsld:title>A unit of learning on the European Constitution</imsld:title>
   <imsld:act identifier="A-1">
            <imsld:title>Views on the European Constitution</imsld:title>
            <imsld:role-part identifier="RP-Learner-1">
                 <imsld:title>Learner RP</imsld:title>
                 <imsld:role-ref ref="Learner"/>
                 <imsld:activity-structure-ref ref="AS-first-step"/>
            </imsld:role-part>
            <imsld:role-part identifier="RP-Facilitator-1">
                 <imsld:title>Facilitator RP</imsld:title>
                 <imsld:role-ref ref="Facilitator"/>
                 <imsld:support-activity-ref ref="SA-first-step"/>
            </imsld:role-part>
        <imsld:complete-act>
                 <imsld:when-role-part-completed ref="RP-Facilitator-1"/>
        </imsld:complete-act>
   </imsld:act>
   <imsld:complete-play>
            <imsld:when-last-act-completed/>
   </imsld:complete-play>
</imsld:play>
```

Fig. 1. A fragment of the XMl representation of a Unit of Learning

The software which interprets this XML notation is referred to as an IMSLD engine (Martens & Vogten, 2005; Vogten, Koper, Martens, & Tattersall, 2005), a software service used by an IMSLD player (McAndrew, Nadolski, & Little, 2005), the interface with which learners and staff interact. The engine makes the appropriate activities and environments available to people playing the various roles, through the player, coordinating and synchronizing the dynamics of a learning process as multiple learners work through a learning process. The distinction between engine and player allows a variety of approaches to the look and feel of interaction in learning processes (different players) to be supported by a single orchestration service (engine).

Figure 2 shows a member of staff interacting with an IMSLD player.

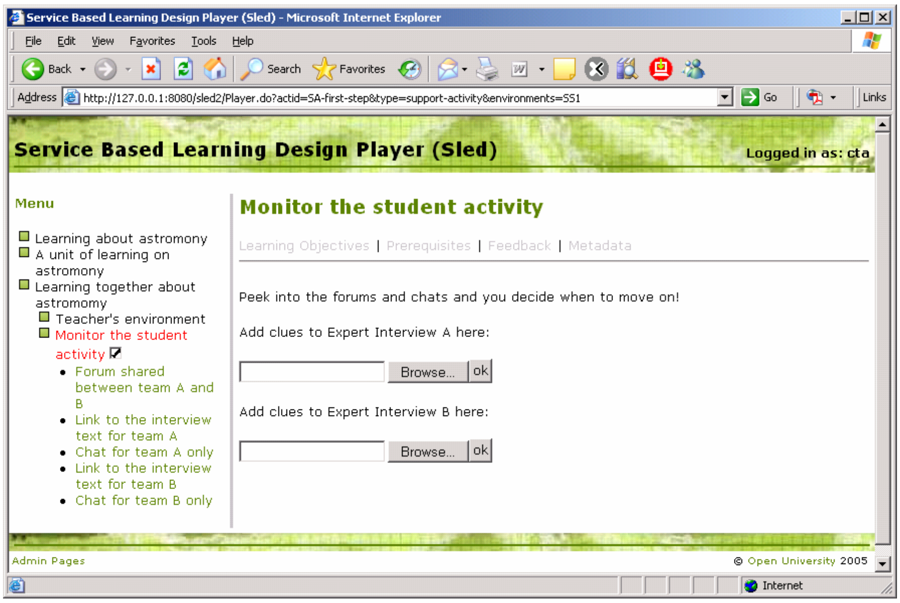

Fig. 2. An IMSLD player interpreting a Unit of Learning

The player shown in Figure 2 is a web server-based application. Once the user has surfed to the appropriate web address and been authenticated, learning or support activities are made available. Note that what is seen by a particular user (learner or staff member) is a personalised, specific set of activities and associated environments according to the design recorded in the IMSLD notation and the current state-of-play of the learning process. The content of the web page is generated from the ongoing processing of the notation by the engine. Drop-down boxes, forms and buttons are all derived from concepts in the language. Whether and when the learner is presented with a given activity depends on the conditions specified by the designer of the learning process.

So, along which processes and using which tools do designers create Units of Learning? The IMSLD Best Practice and Implementation Guide offers a suggestion for a three step approach to the creation of Units of Learning. Individuals or teams involved in the design of e-learning start with a narrative description of the proposed learning process. Unified Modeling Language Activity Diagrams are used to make the flow of events within and between roles explicit, before a final phase in which the XML-based Unit of Learning is created. Sloep, Hummel and Manderveld (2005) offer a related procedure for UoL design, and Janssen and Hermans (2005) provide experience with the approach in a distance learning context.

Applying such an approach to designing within a Problem-Based Learning context, a narrative description would be:

- The coordinator for the course makes a problem description available to the group.
- Each of the students in the group reads the problem, as does the facilitator.

- The students decide on a chairperson, the spokesperson for the group, responsible for recording key group decisions, and the chosen representative is appointed as such by the facilitator.
- The group then communicate amongst themselves to clarify the problem, using each other and the facilitator to discuss and clarify terminology and any open issues, eventually arriving at their own succinct statement of the problem at hand.
- The chairperson states this problem description in a document and the group continues by identifying possible solutions or explanations for the problem.
- These possible explanations are clustered into a small number to be explored further by the students.
- The explanations to be pursued are listed in a document.
- The group then identifies the learning goals of the problem and individuals embark on the required research.
- Eventually, the group meet up to discuss their findings, again assisted by the facilitator.
- The chairperson summarizes the findings in a document.
- Subsequently, an Evaluator and the Facilitator discuss the performance of the group and the Evaluator provides an Evaluation of the group.

A transformation of the above narrative to an activity diagram is shown in Figure 3.

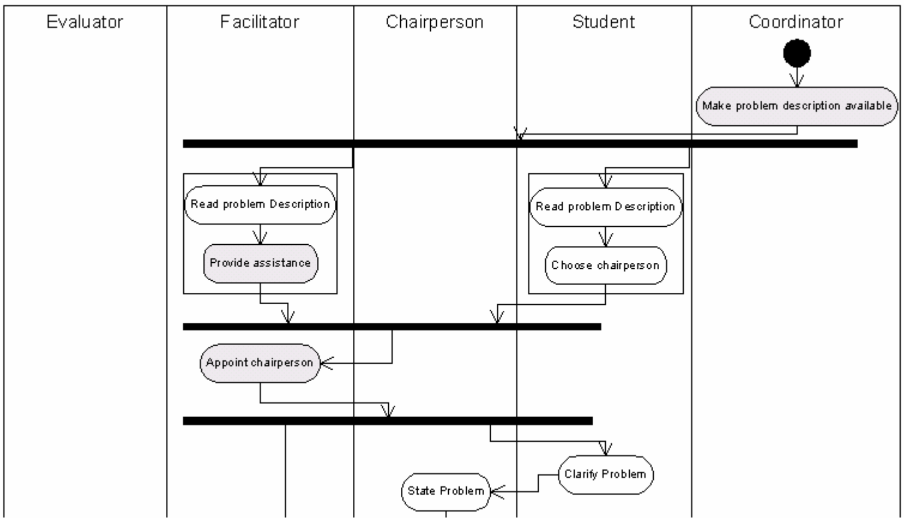

Fig. 3. A partial activity diagram for Problem Based Learning

Turning to tooling, the XML illustrated in Figure 1 can be created with text based editors such as Windows NotePad. This is, of course, a time-consuming and error-prone activity. Syntax problems (for example, missing angled brackets and string quotes), the need for authors to maintain a mental list of XML element names and long “create-test-debug” cycles reminiscent of the early days of computer programming languages, all indicate the need for a more supportive environment for designers. The use of XML editors alleviates the difficulties to only a small degree. Template-based approaches such as that described by Janssen and Hermans (2005) either require the development of significant tooling skills by learning designers or a team-based approach which includes notational specialists.

A number of initiatives have been carried out to create a higher degree of support for the creation of Units of Learning. Reload (Milligan, Beauvoir, & Sharples, 2005) is a tree and form-based editor which has seen significant use in the IMSLD community. Figure 4 shows the editor in use.

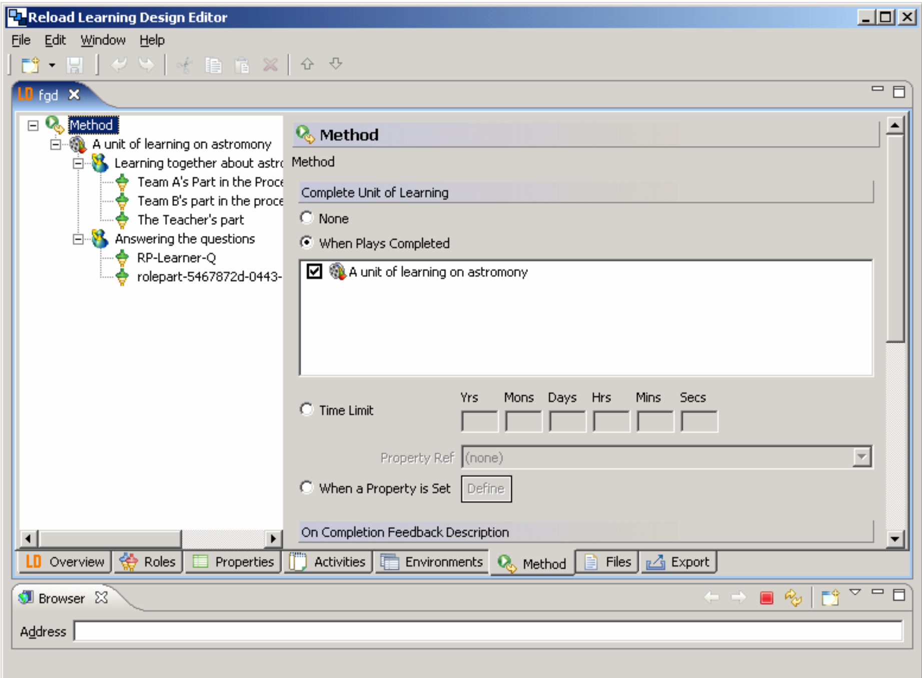

Fig. 4. The Reload editor

Reload takes the learning designer away from the intricacies of the XML binding, organising the user interface around core IMSLD concepts (the tabs for overview, roles, properties and so on) using forms to gather data through particular controls (check boxes, drop down lists etc). Designers' work can be checked for completeness with respect to the requirements on a UoL, and can be exported as a UoL content package. Once in this format, the UoL can be uploaded to an IMSLD player environment and tested.

Experience with using Reload (Barrett Baxendale, 2005) highlights a steep learning curve and a need to resort to combinations of editors (eg Reload plus Notepad) to cope with IMSLD level B. Fundamentally, since Reload was written as reference implementation of an IMSLD editor, everything which is possible to be encoded in IMSLD notation (whether using Reload or another IMSLD editor) must be able to be imported, adapted and exported. This leads the tool to be close to the specification (Griffiths, Blat, Garcia, Vogten, & Kwong, 2005), requiring the user to be familiar with the notation. A variation on this style of support can be seen in the CoSMoS editor (Miao, 2005), shown in Figure 5, and in CopperAuthor (Van der Vegt & Koper, 2006), shown in Figure 6.

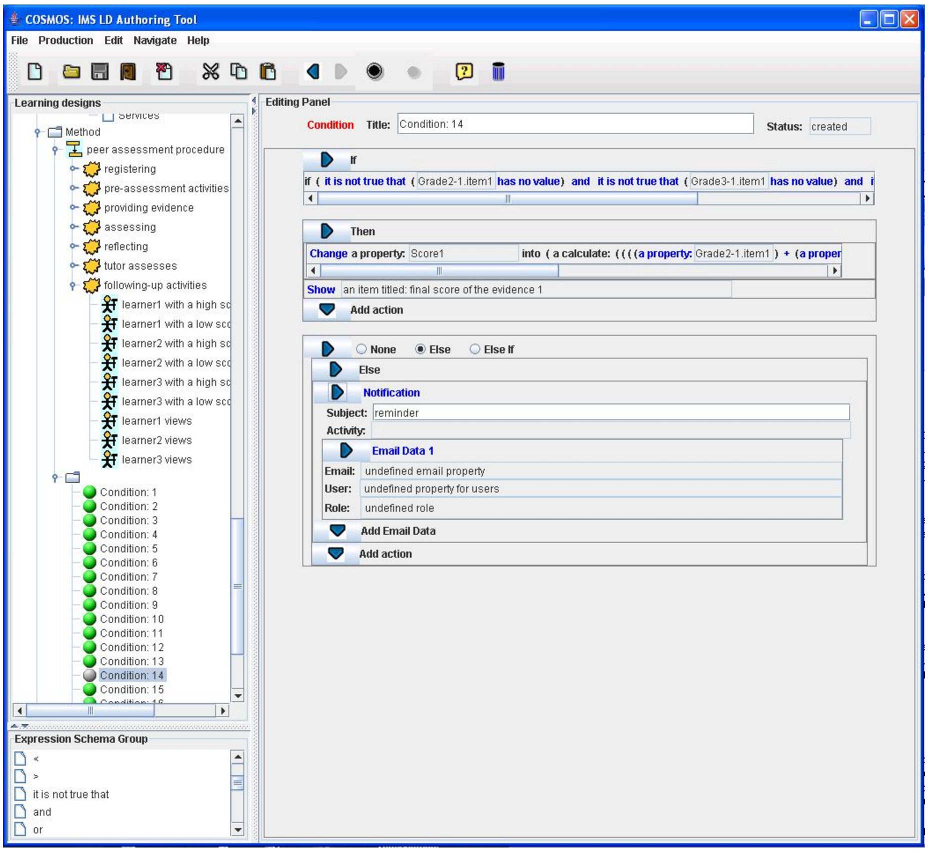

Fig. 5. The CoSMoS editor

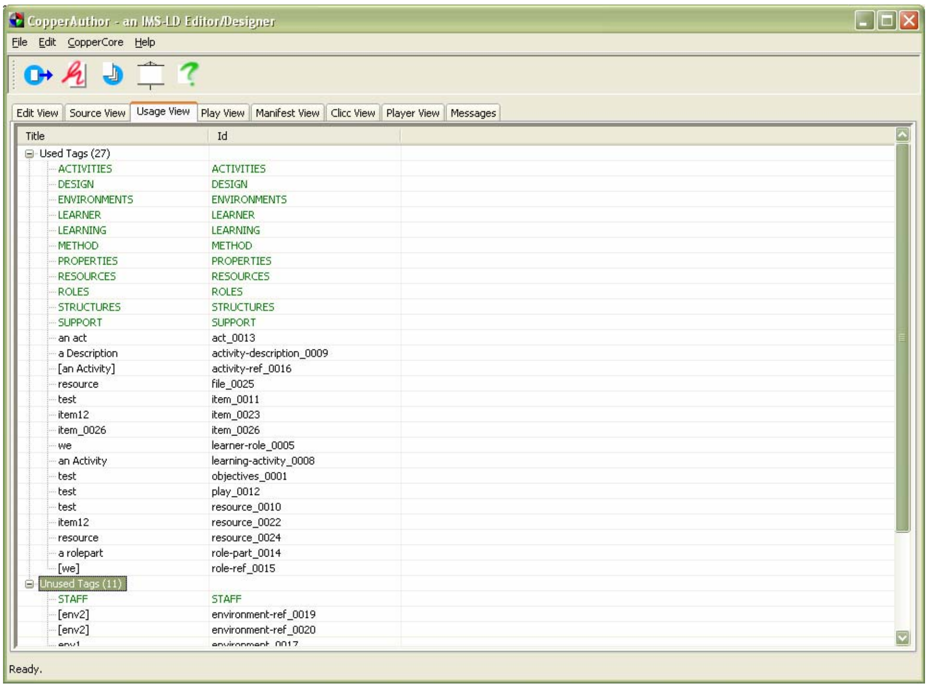

Fig. 6. CopperAuthor

Yu, Zhang and Chen (2006) highlight the need to support those who may have little or no knowledge of the IMSLD notation yet are involved in the design of e-learning, giving teachers as an example target group. The authors cite MOT+ (De la Teja, Lundgren-Cayrol, & Paquette, 2005) and ASK-LDT (Sampson, Karampiperis, & Zervas, 2005) as more appropriate for this user group. Figures 7 and 8 show the graphical approaches used by MOT+ and ASK-LDT.

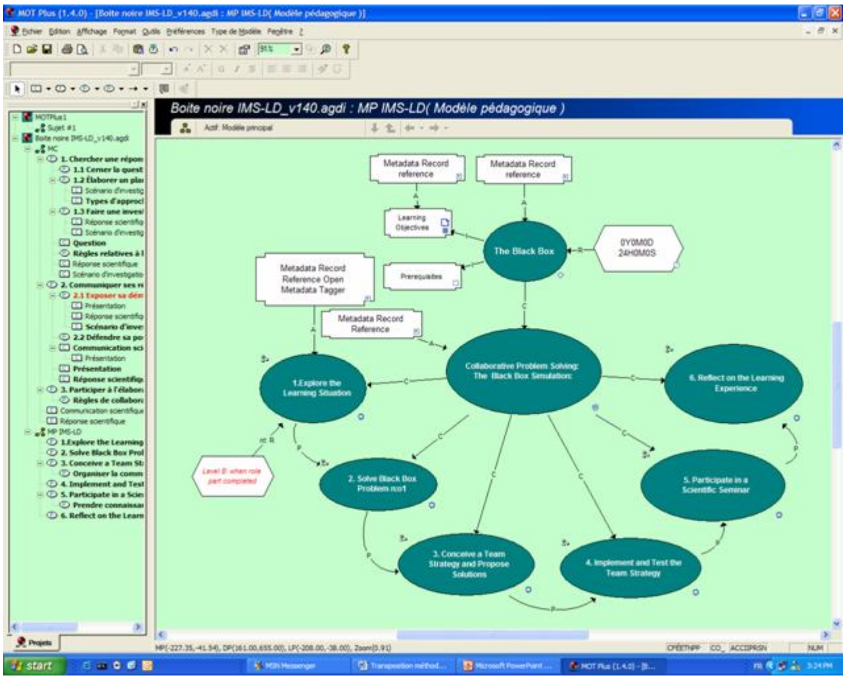

Fig. 7. The MOT+ environment

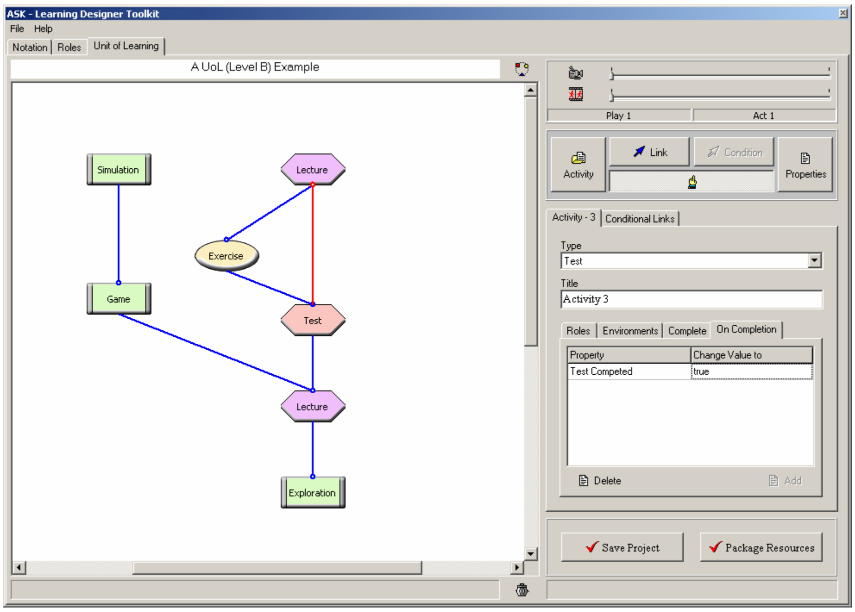

Fig. 8. The ASK-LDT environment

Although certainly offering a more protective environment in which designers can work, caution is needed before concluding that use of graphical interfaces will remove barriers to UoL production. Spang Bovey and Dunand (2006) highlight that modelling a complex sequence of activities with MOT+ is "too disconnected from the daily practice of the average teacher". The issue of the part played by teachers' in production of UoLs is explored in some detail by Griffiths and Blat (2005). The authors share the opinion of Spang Bovey and Dunand that, while the underlying concepts of the LD modelling language are not complex, they differ from the concepts that a teacher uses to plan educational activities. Two challenges are identified in the article: supporting teachers during preparatory stages and helping teachers author and edit Units of Learning. A number of approaches to meeting these challenges are explored by the authors including the use of patterns, templates, primitives and placing constraints on the available design options. In line with these suggestions, the Collage system (Hernández-Leo et al., 2006) provides a set of Computer Supported Collaborative Learning patterns (eg Jigsaw, Brainstorming) layered above the Reload tool. Users are led through the selection and adaptation of patterns towards a point at which a UoL can be generated for subsequent interpretation by an IMSLD engine. Initial evaluations of teachers using Collage have been positive, though adaptation of the pattern structures represented in IMSLD remained problematic. Figure 9 shows the Collage editor.

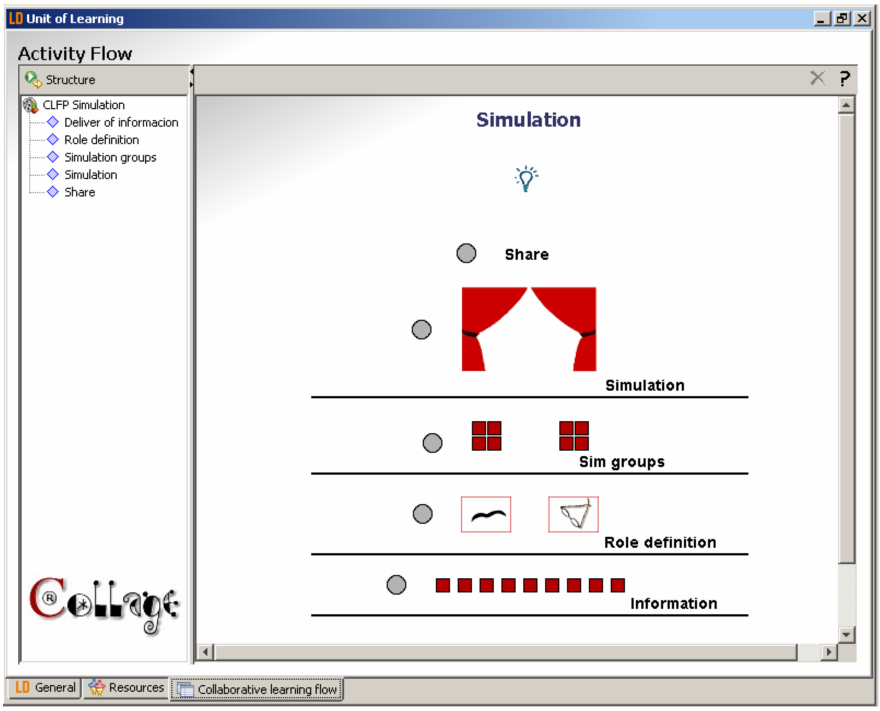

Fig. 9. The Collage editor

Putting IMSLD to use today, then, is done along multi-step unit of learning production processes supported by a number of different editing environments, engines and players.

4. Some observations

Reflecting on the various initiatives in the area of IMSLD-aware tooling, a number of observations can be made.

The first generation of IMSLD-aware tool supports only part of the design process. The early stages, in which ideas are sketched, initial narratives developed and storyboards worked out, tend to fall outside the tools' scope. Similarly, the tools tend to offer partial support for the development of Units of Learning, relying on separate tools for the creation of textual and graphical content, assessments, simulations etc. While pragmatic in terms of reusing existing tools and other initiatives, this approach tends to lead to a rather fragmented design experience. Once created, these components must be integrated into an orchestrated learning process described by the learning design of a Unit of Learning. Moreover, the issue is not only one of linking to passive resources on the web, but integrating and orchestrating active components into a learning process (eg posts to a blog, new entries in a wiki). What happens in an assessment, or in a simulation, should be able to be used to influence the rest of the learning flow.

Although most of the tools have seen evaluation, these have often been small scale studies. A body of evidence on how designers use today's tools is still being accumulated. But who are the users of these tools? Some tools explicitly address technically-savvy users who are interested in using the tools to learn more about the specification rather than for use in educational production processes (e.g. Reload). Others state the target group as being teachers. This group is, however, extremely diverse and may rely on very different production processes and need a variety of tools. Teachers in primary education have different approaches to e-learning than those in vocational education, which differ again to those in higher education. Even within a particular type of education, different approaches are seen, such as the faculty-driven model and design-team driven model in higher education, described in (Rungtusanatham, Ellram, Siferd, & Salik, 2004). While the use of the word teacher suggests a traditional academic setting, e-learning is heavily applied in corporate e-learning and human resource development environments (Dagada & Jakovljevic, 2004; Trombley & Lee, 2002). Designers in these environments may well require different support environments. Moreover, the group "designers of e-learning" is broadening as those outside traditional educational and training contexts use networking technologies help others learn. This help takes the form both of suggestions or recommendations of how to *approach* the attainment of a particular competence, as well as the full "recreational design" of learning processes (such as designs for learning languages, sports, musical instruments, project management, plumbing, architecture and so on).

Part of the challenge when designing e-learning (whether by schooled instructional designers, designers-by-assignment or recreational designers) revolves around how to take advantage of theoretical insights and best practices. Merrill and Wilson argue for embedding principles of effective and efficient learning design so that tools "provide intellectual leverage to designers who may not know the required instructional design theory" (Merrill & Wilson, 2006). Although such principles can be encoded in IMSLD, exposing designers to the raw notation does not give the required leverage. In this context, Waters and Gibbons (2004) refer to the intuitive versus non-intuitive dimension of notation systems, and the trade-off which needs to be stuck between human and computer use of a notation. Could IMSLD, then, be married to a standardised graphical notation which would appeal to the intuition of designers? If so, mappings could be defined between the graphical notation and IMSLD allowing translations between the two, and visual design environments could be created to support the notation. It seems however, that faced with a large diversity of users, a single intuitive visual notation may not suffice. Experience with UML (Arlow, Emmerich, & Quinn, 1999) suggests its models are no panacea for the many and varied stakeholders involved in system development. We note, however, that diversity of graphical notations, as is the situation today with ASK-LDT, MOT+ and Collage, makes the interoperability situation more complex – today's tools all export IMSLD able to be interpreted by compliant engines, yet round-tripping between the tools is not possible. Supporting variety in visual notations while preserving interoperability is a key research topic for the next generation of visual design environments.

5. Conclusion: requirements for a new generation of interoperable e-learning design environments

With these observations in mind, we list with a series of requirements which form the basis for our ongoing research in the area of visual design environments for IMSLD.

- There is a need for end-to-end support of design processes from idea formation through to complete UoL (see Botturi (2006) for results in this area).
- Environments should support holistic e-learning design, incorporating (formative) assessment, simulations, multimedia content and other parts of an e-learning experience. This is not merely a question of linking to objects or services on the web, but requires designers to be a position to specify which information should be passed to, and taken back from, learners' interactions with content and services.
- E-learning designers should be shielded from the intricacies of notational bindings. Wizards, templates, alternatives metaphors and techniques from the world of visual design environments, can all help in meeting this need. Moreover, research in End User Development (Sutcliffe & Mehanddjiev, 2004) offers a number of pointers to address this problem. A closer involvement of groups of designers in the development of environments (rather than solely in their evaluation) would seem appropriate.
- Environments should accommodate a high degree of variation in designers' knowledge and experience with pedagogies, both traditional and those focused on e-learning. Although some groups of designers may require extensive handholding, Hoogveld (2003) notes that teachers do not like prescriptive methods. Having the

flexibility for designers to experiment with, tune and indeed, create templates, patterns and primitives might help strike the correct balance between too restrictive an environment and an unsupportive one.

- Finally, we emphasise the need for design environments to be created to be interoperable. Without the capability to both import and export standardised notations such as IMSLD, tools users become shackled to a particular design tool, fragmenting the community and creating competition where cooperation would offer more benefits.

Creating environments to meet these needs will help reach the goals of sharing, critiquing, modifying, executing, rating, comparing and evaluating learning designs across the broad spectrum of e-learning designers.

6. References

Arlow, J., Emmerich, W., & Quinn, J. (1999). Literate Modelling Capturing Business Knowledge with the UML. In *Proceedings of The First International Workshop on the Unified Modeling Language, UML'98 - Beyond the Notation*. Mulhouse, France.

Barrett Baxendale, M. (2005). E-Learning Technical Framework and Tools – Demonstrator Projects: SLeD Integration Demonstrator Final Report. Retrieved October 11th, 2006, from http://www.hope.ac.uk/slide/documents/slide_final.doc

Botturi, L. (2006). E2ML. A Visual Language for the Design of Instruction. *Educational Technologies Research & Development, 54*(3), 265-293.

Botturi, L., Derntl, M., Boot, E., & Figl, K. (2006). A Classification Framework for Educational Modeling Languages in Instructional Design. In Kinshuk, R. Koper, P. Kommers, P. Kirschner, D. G. Sampson & E. , Kerkrade, Netherlands. (Eds.), *Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT2006).* Los Alamitos, CA: IEEE.

Dagada, R., & Jakovljevic, M. (2004). 'Where have all the trainers gone?' E-learning strategies and tools in the corporate training environment. In *Proceedings of the 2004 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries table of contents*. Stellenbosch, Western Cape, South Africa ACM.

De la Teja, I., Lundgren-Cayrol, K., & Paquette, G. (2005). Transposing MISA Learning Scenarios into IMS Units of Learning. *Journal of Interactive Media in Education (Advances in Learning Design Special Issue), 2005/14*.

Eseryel, D., Schuver-van Blanken, M., & Spector, J. M. (2001). Current practice in designing training for complex skills: Implications for design and evaluation of ADAPT-IT. In C. Montgomerie & J. Vitelli (Eds.), *Proceedings of ED-MEDIA 2001: World Conference on Educational Multimedia, Hypermedia, & Telecommunications* (pp. 474-479). Tampere , Finland Association for Advancement of Computing in Education.

Gibbons, A. S., & Brewer, E. K. (2005). Elementary Principles of Design Languages and Design Notation Systems for Instructional Design. In J. M. Spector, C. Ohrazda, A. Van Schaak & D. A. Wiley (Eds.), *Innovations in Instructional Technology: Essays in honor of M. David Merrill* (pp. 111-130). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Griffiths, D., & Blat, J. (2005). The role of teachers in editing and authoring units of learning using IMS Learning Design. *Advanced Technology for Learning 2005*(4).

Griffiths, D., Blat, J., Garcia, R., Vogten, H., & Kwong, K. (2005). Learning Design Tools. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 109-136). Berlin-Heidelberg: Springer Verlag.

Hernández-Leo, D., Villasclaras-Fernández, E. D., Asensio-Pérez, J. I., Dimitriadis, Y., Jorrín-Abellán, I. M., Ruiz-Requies, I., et al. (2006). COLLAGE: A collaborative Learning Design editor based on patterns. *Educational Technology & Society, 9*(1), 58-71.

Hoogveld, A. W. M. (2003). *The Teacher as Designer of Competency-Based Education.* Open Universiteit Nederland, Heerlen.

IMSLD. (2003). IMS Learning Design Specification. Retrieved February 27th, 2004, from http://www.imsglobal.org/learningdesign/index.cfm

ISO. (2005). *ISO/IEC 19501:2005 Information technology -- Open Distributed Processing -- Unified Modeling Language (UML) Version 1.4.2* Geneva, Switzerland International Organization for Standardization.

Janssen, J., & Hermans, H. (2005). How to Integrate Learning Design into Existing Practice. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 253-266). Berlin-Heidelberg: Springer Verlag.

Kenny, R. F., Zhang, Z., Schwier, R. A., & Campbell, K. (2005). A Review of What Instructional Designers Do: Questions Answered and Questions Not Asked *Canadian Journal of Learning and Technology, 31*(1).

Kirschner, P., Carr, C., Van Merriënboer, J. J. G., & Sloep, P. (2002). How Expert Designers Design. *Performance Improvement Quarterly, 15*(4), 86-104.

Koper, R. (2003). Learning technologies: an integrated domain model. In W. Jochems, J. Van Merrienboer & R. Koper (Eds.), *Integrated eLearning* (pp. 64-79). London: RoutledgeFalmer.

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). Heidelberg: Springer Verlag.

Koper, R., & Burgos, D. (2005). Developing advanced units of learning using IMS Learning Design level B. *International Journal on Advanced Technology for Learning, 2*(4), 252-259.

Koper, R., & Manderveld, J. M. (2004). Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning. *British Journal of Educational Technology, 35*(5), 537-551.

Martens, H., & Vogten, H. (2005). A Reference Implementation of a Learning Design Engine. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook* *on Modelling and Delivering Networked Education and Training* (pp. 91-108). Berlin-Heidelberg: Springer Verlag.

McAndrew, P., Nadolski, R., & Little, A. (2005). Developing an approach for Learning Design Players. *Journal of Interactive Media in Education (Advances in Learning Design Special Issue), 2005/14*.

Merrill, M. D., & Wilson, B. (2006). The Future of Instructional Design (Point/Counterpoint). In R. Reiser & J. V. Dempsey (Eds.), *Trends and Issues in Instructional Design and Technology, Second Edition* (pp. 335-351). New Jersey, USA: Prentice Hall.

Miao, Y. (2005). CoSMoS: Facilitating Learning Designers to Author Units of Learning Using IMS LD In C. K. Looi, D. H. Jonassen & I. Mitsuru (Eds.), *Proceediongs of the International Conference on Computers in Education* (pp. 275-282). Singapore: IOS Press.

Milligan, C. D., Beauvoir, P., & Sharples, P. (2005). The Reload Learning Design Tools. *Journal of Interactive Media in Education (Advances in Learning Design Special Issue), 2005/14*.

Rawlings, A., Van Rosmalen, P., Koper, R., Rodríguez-Artacho, M., & Lefrere, P. (2002). *CEN/ISSS WS/LT Learning Technologies Workshop: "Survey of Educational Modelling Languages (EMLs)"*.

Rodríguez-Artacho, M., & Verdejo Maíllo, M. F. (2004). Modeling Educational Content: The Cognitive Approach of the PALO Language. *Educational Technology & Society, 7*(3), 124-137.

Rungtusanatham, M., Ellram, L. M., Siferd, S. P., & Salik, S. (2004). Toward a Typology of Business Education in the Internet Age *Decision Sciences The Journal of Innovative Education, 2*(2), 101-120.

Sampson, D. G., Karampiperis, P., & Zervas, P. (2005). ASK-LDT: A Web-Based Learning Scenarios Authoring Environment Based on IMS Learning Design *Advanced Technology for Learning 2005*(4).

Sloep, P., Hummel, H., & Manderveld, J. M. (2005). Basic Design Procedures for E-learning Courses. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 139-160). Berlin-Heidelberg: Springer Verlag.

Spang Bovey, N., & Dunand, N. (2006). Seamless production of interoperable e-Learning units: stakes and pitfalls. In R. Koper & K. Stefanov (Eds.), *Proceedings of the Workshop on Learning Networks for Lifelong Competence Development,* . Sofia, Bulgaria: INCOMA Ltd.

Süß, C., & Freitag, B. (2002). *LMML - The Learning Material Markup Language Framework.* Paper presented at the International Workshop: Interactive Computer Aided Learning, Villach, Austria,.

Sutcliffe, A., & Mehanddjiev, N. (2004). Special Issue: End-user development - tools that empower users to create their own software solutions. *Communications of the ACM, 47*(9).

Tattersall, C., Vogten, H., Brouns, F., Koper, R., Van Rosmalen, P., Sloep, P., et al. (2005). How to create flexible runtime delivery of distance learning courses. *Educational Technology & Society, 8*(3), 226-236.

Trombley, B. K., & Lee, D. (2002). Web-based Learning in Corporations: who is using it and why, who is not and why not? . *Journal of Educational Media, 27*(3), 137-146.

Van der Vegt, W., & Koper, R. (2006). CopperAuthor 1.6. Retrieved October 11th, 2006, from www.copperauthor.org

Van Es, R., & Koper, R. (2006). Testing the pedagogical expressiveness of IMS LD. *Educational Technology & Society, 9*(1), 229-249.

Vogten, H., Koper, R., Martens, H., & Tattersall, C. (2005). An Architecture for Learning Design Engines. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 75-90). Berlin-Heidelberg: Springer Verlag.

W3C. (1999). HTML 4.01 Specification, W3C Recommendation. Retrieved October 11th, 2006, from http://www.w3.org/TR/html401/

Waters, S. H., & Gibbons, A. S. (2004). Design Languages, Notation Systems and Instrictional Technology: A Case Study. *Education Technology Research & Development, 52*(2), 57-63.

Yu, D., Zhang, W., & Chen, X. (2006). New Generation of E-Learning Technologies In *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences - Volume 2 (IMSCCS'06)* (pp. 455-459).