

Seamless production of interoperable e-Learning units: stakes and pitfalls

Nadia Spang Bovey and Nicolas Dunand

University of Lausanne, Bâtiment Provence, 1015 Lausanne, Switzerland
Nadia.SpangBovey@unil.ch, Nicolas.Dunand@unil.ch

***Abstract.** The modeling of a 28-week course in Information Theory using the IMS Learning Design level B specification proved its efficiency for describing complex learning scenarios. This article briefly summarizes the method used to create the real-life unit of learning. The experiment showed that, although various editing software and rendering engines are available, the resulting production process relies too much on computer specialists to be adopted as a strategy at institutional level and that their lack of integration with Virtual Learning Environments prevents large-scale deployment.*

Keywords: learning scenarios, interoperability, IMS Learning Design

Introduction

As far as content is concerned, IMS Content Packaging or SCORM compliant VLEs provide basic export/import tools that allow a relatively easy redeployment. Learning scenarios, which convey the teachers' personal views and know-how, are on the contrary stuck in the tool where they were initially developed. From a pedagogical point of view, learning scenarios have an indispensable added value and cannot conceivably be manually rewritten in case of a change of VLE. Therefore, the publication of the IMS Learning Design (hereafter "IMS-LD") specification seemed to be a good omen.

The experience and point of view presented here are those of a Center providing pedagogical support to the teaching staff. Its main duties are non-technical. Designing learning activities and scenarios is a core practice, with or without the support of technology, and the involvement of coding specialists to ensure their sustainability cannot be regarded as a viable possibility. We definitively need to establish a design and production process that integrates the issue of reusability and interoperability of learning scenarios from the very start and – unlike the industrial approach adopted by large institutes for distance education [1] – relies on generic com-

puter skills. We decided to experiment the potential and shortcomings of the methodology proposed by IMS. Though being aware that no compliant VLE was available, we hoped – and still hold the view – that the next generation of authoring and teaching tools would provide us with a suitable long-term solution.

The work presented here has been using the IMS Learning Design version 1.0 technical specification to create a rich learning unit. The latter can account for all aspect of the learning scenario, allowing rendering the finest interactions between all actors, thus providing the largest possible didactic liberty to the teachers and developers [2] [3]. The following text aims at detailing each step needed to translate some real-life academic course into an interoperable learning unit; it also will provide a walkthrough pointing out the difficulties arising from the whole process, showing why the seamless production of such learning units is not yet at hand.

Approach

The approach used to design the learning unit presented here was a three-step process involving three different people. To fulfill our expectations, this process should however count two

or even one single step, performed by the teacher him- or herself. Firstly, discussions took place between the teacher and a learning engineer, in order to create a model of the course. When a common agreement was reached, an UML-like activity-diagram of the course scenario was made, taking into consideration all roles involved in the learning/teaching process, mapping activities to them, and showing links between activities. In a subsequent step, the concepts of the UML diagram were translated into a learning unit (an IMS-LD compliant file) using various software and some manual coding. The next parts of this section describe these steps in more detail and signal where simplifying the process would be a decisive advantage.

Course Description and Pedagogical Modeling

The course being modeled is a two-semester online course in Information Theory at Bachelor level. Students are allowed to choose between several learning paths and types of exams. Depending on their choice, they are provided with various study helps, such as control questions, exercises, videos and online demonstrations. Their work is supported by an online tutor, who answers questions and monitors the learning journal students are required to write. At the beginning of the second semester, at a fixed date, students willing to submit a personal work are required to make their proposal public to the class, so that the professor can validate the projects and the whole group get a picture of various possible applications of Information Theory.

According to the IMS methodology, pedagogical modeling should begin with the creation of a textual and visual representation of the course [4]. Since no UML editor can generate automatically an IMS-LD compliant XML file, the modeling of the 28 weeks of teaching and learning activities had to be divided in two separate operations. The activity diagram was designed as a visual representation of the roles, activities, decision points and timeline of the course.

Though necessary to get a full picture of the sequence of activities, this step is only a preliminary modeling of the teaching and learning activities. In an ideal world, this would not be

an isolated operation and would provide the modeler with a skeleton of the actual course in an IMS-LD compliant format, to which the learning environment, as well as the variables and conditions, could easily be added. MOTplus [5], though not an UML editor, heads in this direction and allows the generation of IMS-LD level A-compliant XML files. However, the modeling of a complex sequence of activities with this tool remains too disconnected a task from daily practice of the average teacher, if only because the modeler needs to master the rules and constraints used for pedagogical modeling in the software, and a fair knowledge of the specification itself.

Learning Unit Design and Conception

Once the UML diagram of the pedagogical process established, the learning unit itself had to be created. This was mainly achieved using the Reload Learning Design Editor software, which provides a comprehensive and intuitive graphical UI allowing the creation of a learning unit complying with the Learning Design specification [6].

The so-produced learning unit consists of a ZIP package containing an XML manifest file and all files needed to render the course properly: instructions, content, etc. The learning unit produced adopts the level B of the IMS-LD specification; this means that, for scenario flexibility purposes, the use of variables (properties) and of conditional events (conditions) is possible.

The Reload Learning Design Editor allows the user to build a course scenario based on the IMS-LD concepts. These concepts include an approach based on roles and activities; each participant being related to a role (i.e. teacher, tutor, student, etc.) performs different activities, based on his role and preferences. Such activities can then be associated with various environments, which can provide facilities such as communication tools (e-mail, discussion forums, and so on), tracking and indexing functions, or simply a means to feed the user with additional content.

Building a course scenario based on this role/activities description from an UML workflow diagram requires identifying roles and activities first; this is the easy part. The next step

demands some reformulation, as a description of the various relations between the different roles and activities is needed. This reformulation mainly consists in converting human semantics into the limited number of concepts available from the IMS-LD specification.

The IMS-LD specification uses roles and activities to define role-parts, which are the building bricks of the learning scenario: each role-part associates one role to one activity. Several of these role-parts can then be grouped into acts: an act is a set of role-parts which takes place (i.e. begins and ends) at the same time for all actors of the scenario, so providing synchronization abilities. In the unit developed here, one act groups all activities of one academic semester (which of course takes place at the same time for everyone), whilst some other

synchronization features are achieved using properties and conditions. It is indeed possible to use conditions on the values of variables to make visible or invisible elements of the learning scenario, such as tools, content, or activities.

Firstly, all activities are built; each one is then mapped to one role, thus defining role-parts. The use of conditions allows making various elements visible or invisible depending on any constraint. Furthermore, the use of variables allows the precise tracking of students by the monitoring of their properties (i.e. variables), which for instance may keep track of how (or if) some activity has been performed or of their learning preferences. To allow the setting and visualization of variables by users playing the learning scenario, XHTML files have to be written from scratch.

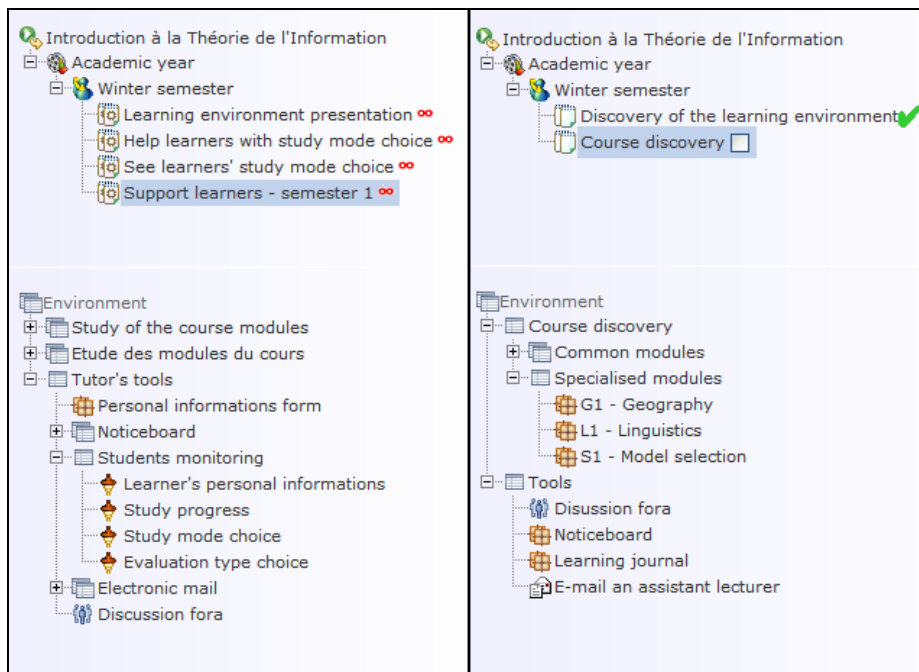


Fig. 1. Different activities and environments are presented to different roles. On the left part are shown the activities and corresponding environment presented at some point of the scenario to a tutor, allowing him to see the various students' choices made so far among other things. On the right part of the figure are shown the activities available at the same time to the student

Running the Learning Unit

In our institutional context, the primary purpose of modeling units of learning would be to ensure their portability from one VLE to the other. Although sharing learning objects and scenarios in repositories might be an additional incentive,

our main worry is the issue of durability and this is where the experiment turns out to be inconclusive. Getting the learning unit running necessitates a Learning Design rendering engine, as at this moment no course management system allows the importation of such a unit. The CopperCore engine [7] was chosen to test

the produced learning unit because it was able to render most elements of the learning scenario. It however only provides a basic rendering layer; it was indeed not aimed at providing a virtual learning environment, but a low-level Learning Design engine (which could be incorporated into a real learning environment).

The first step to get the learning scenario running is then the instantiation step. That is, one needs to map actual users to the roles of the scenario, thus creating one scenario instance. Users have to be manually added to the different learning scenario roles. Once all needed user added, the CopperCore engine allows the user to pick up a role and to run the learning scenario. The Fig. 1 shows the various activities proposed at the same moment to two different actors having different roles within the learning unit.

A need for integration

In our real-life settings however, both the production and the use of the learning unit would have to be different to be accepted as a viable strategy. Writing the IMS-LD compliant file with Reload Learning Design Editor and rendering it with CopperCore is certainly suitable for learning and testing the specification itself¹. Except for a small number of projects, the courses would have to be designed and produced by the professors or their teaching assistants. Therefore, the generation of IMS-LD files should be embedded into simple design tools, preferably within the VLE, in a way similar to that implemented in LAMS (Learning Activities Management System) [8]. Though not based on IMS-LD, LAMS illustrates a concept that might bridge the gap in a context where teachers are the main producers of technology-enhanced courses. While the framework, tools and sequencing of the course is provided by the VLE² the teacher models one learning activity after the other, dragging icons representing the tools that are needed to proceed with the activity on a design screen where instructions, resources and

conditions can be added in a very natural way³. The relevant product would be an IMS-LD compliant VLE with learning activities design functionalities. Those would provide visual and intuitive means to create sets of instructions linked to the relevant resources and tools, and be able to automate the generation of the XML files needed to redeploy the course in another compliant VLE. Specialized help could thus be restricted to a few highly sophisticated courses and the specification be adopted on a large scale.

Conclusion

At the University of Lausanne, the production of technology-enhanced courses is mainly done by the teachers themselves. Therefore, the issues of sustainability and interoperability of the learning scenarios, though fully appreciated by the eLearning support staff, must be kept behind the scene. Regular teaching staff members would very easily be discouraged by additional technical constraints imposed upon their work. In such an institutional context, the modeling of the Online Course in Information Theory according to the IMS Learning Design specification served two major objectives: testing the adequacy of the specification to describe real-life courses that were not designed on purpose, and identifying the conditions needed for the adoption of the specification to ensure the portability of the online courses on a large scale.

The result of the experiment is promising but also clearly shows that the natural integration of the specification with actual practice is not yet at hand. Surely enough, IMS-LD proved being adequate to successfully model the 28 weeks of learning activities and all the related tools and interactions. The complete process required a three-people team composed of a professor, a learning engineer and a skilled computer staff willing to dig into the specification, who produced the level B-compliant XML file using the Reload Learning Design Editor. The resulting learning unit can be run using a rendering engine such as CopperCore, with each role correctly performing the intended actions with the adequate tools.

¹ A method for a full evaluation benchmark of expressiveness and suitability of IMS-LD is proposed by Caeiro-Rodriguez et al [9]. The authors designed a comprehensive methodology based on pattern recognition.

² Moodle, Blackboard, Sakai and WebCT in a near future.

³ For a technical discussion of LAMS and IMS-LD, see the article of Berggren et al [10].

However, although IMS-LD seems to provide a potential solution to a problem encountered by many Higher Education institutions, its practical use is hampered by a much too complex flow of production. The UML modeling and the translation of the activity diagram into IMS-LD concepts of activities, activity-structures and properties are out of reach of the typical staff in an educational context. Unless both the visual modeling of the learning activities and the generation of the compliant XML files can be integrated into the usual pedagogical design practice of the teachers, the large-scale use of the specification will remain an unviable option in our institution.

While probably a technical and commercial challenge, the missing integrating product can easily be described: an IMS-LD compliant Learning Management System, that would provide the course framework and set of tools, equipped with a LAMS-like visual learning activities design tool allowing the teacher to simply sequence activities and type in instructions and resources references. In addition, the VLE would be able to generate a proper IMS-LD file with all the necessary resources and properties, ready for importation into any other compliant VLE. A natural and intuitive production process could then be implemented, ensuring that teachers' work and creativity are not at risk of being lost.

References

1. Schlusmans, K., R. Koper, et al. (2004): Work processes for the development of integrated e-learning courses. *Integrated e-Learning: Implications for Pedagogy, technology and Organisation*. In W. Jochems, J. v. Merriënboer and R. Koper. London and New York, RoutledgeFalmer: pp. 126-138
2. IMS-LD-IM (2003): *IMS Learning Design Information Model. Version 1.0 Final Specification, revision 20*. IMS Global Consortium
3. Koper, R., & Tattersall, C. (2005): *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*. Springer, Heidelberg, The Netherlands
4. IMS (2003): *IMS Learning Design Best Practice and Implementation Guide. Version 1.0 Final Specification* IMS Global Consortium: pp. 20ss
5. Mot and MotPlus are available at : <http://www.licef.teluq.quebec.ca/eng/index.htm>
6. The Reload Learning Design Editor is available at: <http://www.reload.ac.uk/ldeditor.html>
7. The CopperCore engine is available at: <http://coppercore.sourceforge.net/>
8. Dalziel, J. (2003): *Implementing Learning Design: The Learning Activity Management System (LAMS)*. LAMS is available at: <http://www.lamsinternational.com/index.html>
9. Manuel Caeiro-Rodríguez, Martín Llamas-Nistal and Luis Anido-Rifón (2005): *Towards a Benchmark for the Evaluation of LD Expressiveness and Suitability*. *Journal of Interactive Media in Education*, 2005/04
10. Anders Berggren, Daniel Burgos, Josep M. Fontana, Don Hinkelman, Vu Hung, Anthony Hursh and Ger Tielemans (2005): *Practical and Pedagogical Issues for Teacher Adoption of IMS Learning Design Standards in Moodle LMS*. *Journal of Interactive Media in Education*, 2005/02