Preface to the special issue: “Comparing Educational Modelling Languages on the “Planet Game” Case Study”

Peter B. Sloep

Programme Director Research & Development Technology Enhanced Learning

Centre for Learning Sciences and
Technologies (CELSTEC), formely OTEC
The Open University of the Netherlands
Valkenburgerweg 177

6419 AT Heerlen, The Netherlands
http://www.ou.nl

People with a background in computer science tend to refer to technical specifications for designing learning as metadata standards. And strictly speaking, they are right. Such educational modelling languages, as specification languages for designing learning opportunities are generically called, take a collection of learning opportunities as its starting point and then 'embellishes' them by allotting to them students and staff in various roles, as well as pertinent background information. These embellishments really are data about the learning opportunities, they explicate who is involved with the opportunities, and what information and services are relevant for them. Indeed, they are data about data, hence metadata.

However, such a characterisation amounts to a tremendous underestimation of the power of such educational modelling languages. The first thing that comes to mind when referring to metadata in learning is the Learning Object Metadata specification (LOM). This specification also provides data about data, in this case about learning objects. For the LOM, learning is about content objects, strung together or arranged to meet the learner's demands. For this process of stringing them together to work smoothly, one of course needs to know what each of the beads in the string is about. This is where the LOM comes to the aid, as it describes precisely this, in nine categories, ranging from a learning object's name and evolutionary history via its technical and educational requirements to its intellectual property rights and position in existing classificatory systems. Although not explicitly thus stated, for the LOM learning thus is very much a process of solitary content consumption, for which the specification provides the cutlery.

This conception of learning stands in sharp contrast with the richness of educational modelling languages. Such languages look at learning as a process of social interaction among learners and between learners and teachers (staff). Admittedly, this interaction is set against a background of content, which indeed is made available in the form of learning objects, and content-related services such as for instance mail and search facilities. But most significantly, interaction between learners, staff and their environment is added to this mix. And this wealth of possible kinds and instances of interaction needs to be scripted as in a theatrical play: who interacts with whom and what, in what order, perhaps even given the fulfilment of what conditions? Educational modelling languages shine in providing the descriptive apparatus to allow one to write such scripts (cf. Sloep, 2004).

So, apparently, there are two different views of designing learning, one described by a bead-stringing metaphor, one by a play-scripting metaphor. Whoever is familiar with both LOM-like metadata specifications and educational modelling languages, will readily admit that the former may not be simple but certainly are simpler than educational modelling languages. So why then still invest in research in such languages? The answer surely must be that educational modelling languages are much more powerful and hence, ultimately, educationally much more satisfying than simple metadata specifications. Indeed, one may argue, that only educational modelling languages are up to delivering the rich kind of learning experiences that modern education demands. Where does this difference in conceptions come from? In my view, it harks back to the backgrounds of the people involved in their development.

Educational modelling languages were developed by people who were knowledgeable about computer science but also and more importantly by people versed in educational theory and practice. The present issue bears witness to this. Modelling tools such as MOT and MOT+ are underpinned by insights from education and applied cognitive science (Paquette & Léonard, 2008). LAMS was built to answer a request for a tool with which to design learning opportunities (Dalziel, 2008). The Learning Design Language LDL discussed by Ferraris, Martel & Vignollet in this issue had 'to give teachers and instructional designers the conceptual and the technological means to create and manage learning activities'. And, finally, to the modelling language captured by the IMS Learning Design specification, the same argument applies. Since I am most familiar with this specification, I'll dwell upon its history a bit longer as it quite well illustrates my point.

In the late nineties of the previous century one of the first educational modelling language was developed by the Open University of the Netherlands - aptly called EML for Educational Modelling Language - that sought to alleviate a variety of problems that the OUNL as a distance teaching university was facing: increasing maintenance costs of course materials, rising printing costs, but most significantly the obsolescing of its then standard pedagogical model. This model had been built in the early eighties, still pre-Internet times, and sought to remedy the lack of instantly available teachers so characteristic of distance teaching. Hence much time was spent by course teams writing readily understandable materials, in which they tried to anticipate most of the problems students might run into. Extensively testing out courses on willing volunteer students even improved the anticipatory qualities of courses. Not only was this a costly procedure, the necessity of interaction moments between students and staff or among students themselves was obviated. With the advent of the Internet, interactions became possible again, desirable they had always been. So this offered distance teaching universities such as the OUNL the opportunity to turn a vice (lack of interaction) into a virtue (designed interactions). Of course, designs thus became more complex, but also opened the door to a variety of different pedagogical models (problem-based learning, for example). Although these motives may be specific to the OUNL's situation, I am sure that the desire to have more complex designs, to be able to alternate between pedagogical models and to orchestrate student-student and student-staff interactions played a large part in the development of educational modelling languages tout court.

However, for all their pedagogical virtues, one should not forget that the complexity of educational modelling languages poses serious challenges too, for instance in terms of the severity of the problems that designers of authoring and runtime environments have to overcome. For a 'simple' metadata specification such as the LOM, an authoring environment does an eminent job if it provides a sophisticated filing card, perhaps with restricted values for some fields and default values for others. An authoring environment for an educational modelling language is much more demanding. It should provide a vehicle for expressing pedagogic convictions into learning opportunities. As those convictions are not only held by professional designers, but also by teachers, the interaction design should be intuitive enough for teachers to be able to work with this. Perhaps, working with design templates that are specific to particular pedagogic models may simplify things (Sloep et al., 2005). The papers in the present issue provide examples of interaction designs that attempt to do this. Also, a runtime environment for a 'simple' metadata specification is satisfactory if it portrays the available information in a coherent and synoptic way. A runtime environment for educational modelling languages is much more demanding. It should be able to translate a variety of designs according to a variety of pedagogic models into a variety of equally credible user experiences. In addition to that, this user experience should be time-dependent and role-bound. A tall order!

Contrasting the Learning Object Metadata specification with the Learning Design specification in the way I have done should in no way detract from the timeliness and usefulness of the LOM. Indeed, the LOM serves an important function inside, for example, IMS LD. The comparison, however, does serve to point out what educational modelling languages seek to achieve and that we should be willing wholeheartedly to pay a high price for their implementation, in terms of workflow (not discussed but important all the same), authoring environment and runtime environment. This, in turn, also makes clear why we should not become impatient if things do not progress as swiftly as we may have hoped. The ultimate will be worth the wait. The papers in this issue show how far we've come already since the OUNL's Educational Modelling Language was published in December 1999. A first inventory was made in 2005, with a specific though not exclusive focus on IMS LD (Koper & Tattersall, 2005). The present collection of papers signifies the next step forward. Discussing a central use case of game-based learning, the various educational modelling languages all show their respective strengths at dealing with it. For their efforts to bring together such a collection of papers, based on an ICALT symposium held in Kerkrade, the Netherlands in 2006, the editors are to be complemented. I am sure the reader interested in modelling educational opportunities will reap much benefit from this special issue.

1 References

Dalziel, J. (2008) Using LAMS Version 2 for a game-based Learning Design. This issue.

Ferraris, C., Martel, C., Vignollet, L. (2008) Modelling the “Planet Game” Case Study with LDL and Implementing it with LDI. This issue.

Koper, E. J. R., & Tattersall, C. (2005). Learning Design - A Handbook on Modelling and Delivering Networked Education and Training. Berlin-Heidelberg: Springer-Verlag.

Paquette, G, & Léonard, M. (2008) A Visual Ontology-Driven LD Editor and Player: Application to the Planet Game Case Study. This issue.

Sloep, P. B. (2004). 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.

Sloep, P. B., Hummel, H., & Manderveld, J. (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, 367-385). Heidelberg: Springer Verlag.