Chapter 3

EDUCATIONAL MODELING LANGUAGES

A Conceptual Introduction and a High-Level Classification

Martinez-Ortiz, I.[1], Moreno-Ger, P.[2]; Sierra, J.L.[2], and Fernandez-Manjon, B.[2]

[1]Centro de Estudios Superiores Felipe II, C/ Capitan, 39, 28300 Madrid, Spain; [2]Facultad de Informática, Universidad Complutense de Madrid, C/ Prof. Jose Garcia Santesmases, s/n, 28040 Madrid Spain

Abstract: Creating good quality learning resources is not sufficient for an optimum learning experience. Equally important is having a more enabling learning process involving not only the delivery of learning materials but also other activities that the learner must carry out to meet the learning objectives proposed by the instructor (tutoring, tests, reading books, etc.). Educational Modeling Languages (EMLs) are the cornerstone of e-learning because they provide a language that can be used by the instructors to formalize their own teaching process so that it can also be interpreted by computers. In this chapter we provide a conceptual introduction and a high-level classification of some of the proposed EMLs.

Key words: Educational Modeling Language (EML), Learning Design (LD), learning process, activity, learning object.

1. INTRODUCTION

In the past few years the popularity of the Internet has facilitated new ways of learning, numerous educational tools and applications have appeared and e-learning has come into being. In this context, the idea of reusable resources (i.e Learning Objects, LO (Downes 2001; Koper 2003)) appeared, leading to the development of several specifications and standards to represent learning content (IEEE-LOM 2002), as well as educational resources, and methodologies to facilitate the development of learning materials (Fernandez-Manjon and Sancho 2002; Martinez-Ortiz, Moreno-Ger et al. 2005). The aim of these initiatives was to decrease the total cost of producing and maintaining good quality LOs, thereby promoting their

B. Fernández-Manjón et al. (eds.), Computers and Education: E-learning, From Theory to Practice, 27–40.

reutilization among companies and institutions. To allow this interchange, different initiatives have arisen (e.g. Instructional Management Systems -IMS-, Advanced Distributed Learning -ADL-, Aviation Industry CBT Committee -AICC-, etc). In the past and also in the present there has been an ongoing and active research into how to get the most out of LOs and regarding how to create LOs that can be adapted to different learner needs (Martinez-Ortiz, Moreno-Ger et al. 2005), or how to enhance motivation and engagement among authors is underway. For instance, in our research group we are betting on game based learning (Moreno-Ger, Martinez-Ortiz et al. 2005; Martinez-Ortiz, Moreno-Ger et al. 2006) providing teachers with a set of tools and a methodology to develop their own small games that can replace an LO inside a unit of learning.

However, most recent works (Koper 2000; Weitl, Süß et al. 2002; Paquette, de la Teja et al. 2005) show that creating and reusing good learning materials, although important, is only one aspect of the whole story. In the words of Prof. Koper (Koper 2000): *"providing adequate knowledge is not enough: it has to be learned"*, meaning that the learning process is not only a simple transfer of knowledge. When a course is being designed, it is necessary to decide not only what learning material will be used, but also in which order this material will be shown, and which other *activities* are needed (i.e. self-assessment, problem resolution, tutoring, class discussions, etc.) during the *learning process*. Therefore, one of the most prominent trends in the development of e-learning software is to provide means for describing these learning processes. As introduced in this chapter, this can be done by using suitable Educational Modeling Languages (EMLs).

The rest of the chapter is structured as follows: Section 2 provides a general overview of the concept of EML. Section 3 proposes a three-category high-level classification for EMLs, and summarizes the most relevant ones. Section 4 surveys the *de-facto* EML standard (IMS Learning Design -IMS-LD-). Finally, section 5 closes this chapter.

2. EDUCATIONAL MODELING LANGUAGES

The generalization of the term *Educational Modeling Language* (EML) in e-learning comes from the work developed at the *Open University of the Netherlands* (OUNL). They analyzed the diversity of Learning Management Systems (LMSs) in use and tried to address the shortcomings of e-learning systems derived from the lack of application of instructional and pedagogical theories. As a result they designed and put into practice the language called EML-OU.

In a study of Educational Modeling Languages made by the CEN/ISSS WS/LT Learning Technology Workshop, (Rawlings, van Rosmalen et al. 2002), the concept of EML was defined as a: *"semantic information model and binding, describing the content and process within a 'unit of learning' from a pedagogical perspective in order to support reuse and interoperability"*.

From this definition the following concerns stand out:

- **Semantic information model.** Like an ontology or a schema, it is a meta-model (conceptualization) of a particular domain of discourse. In this case it is a meta-model of the teaching/learning process.
- **Information model and binding**. The "binding" of an EML is a linguistic formalization of the semantic model. Usually, this formalization is done using a Domain Specific Language (DSL) based on XML technologies, and therefore this binding is machine readable.
- **Units of learning**. This concept is the key point of an EML. As defined in (Koper 2001) a *unit of learning* (also known as a *unit of study*) "*is the smallest unit providing learning events for learners, satisfying one or more interrelated learning objectives*". Therefore, a unit of learning can not be divided without losing its semantics and its effectiveness towards the attainment of learning objectives. A *unit of learnin*g can be a course, a workshop, a practice, a complete study program, etc. The unit of learning defines the model of training/teaching, and the *environment* where this activity is done. Such an environment is characterized by the resource material and the services (e.g. forum, chat, videoconference, e-mail) that will be used during the performance of the unit of learning.
- **Pedagogical perspective.** An EML should be relatively independent of teaching theories, so the teacher or the learning designer could decide which of these theories he/she applies.
- **Reuse and interoperability**. Just as with LOs, the idea behind an EML its not only to allow computer applications to interpret an EML script, but also to promote the reuse of successful units of learning descriptions and also to allow the exchange of these units between different e-learning applications without taking into account how the target information system implements the semantics of the EML.

In (Koper 2000) several desirable characteristics for an EML are identified. The main ones can be summarized as:

- An EML must be defined formally and be machine-readable, so the script created with the EML can be interpreted by a computer application.
- An EML must be pedagogically neutral so different trends of teaching can be applied in a unit of learning using the same EML.
- An EML must allow designers to create complete units of study that include the *activities* to be done by the learner (what to do), the people

involved in the activity (with whom), and the environment where the activities will be performed (which learning materials are needed, which software tools, etc.).

- The units of learning created using an EML should be resilient to technical changes, evolution, and platforms, since their purpose is to facilitate reusability between systems and tools.

To sum up, EMLs are used to describe units of learning that in turn describe the learning process. But they also provide a mechanism of communication between technical and non-technical staff inside an organization during the operationalization of such units of learning.

3. A THREE-CATEGORY CLASSIFICATION OF EDUCATIONAL MODELING LANGUAGES

From the different initiatives developed based on the principles of the EMLs described in the section above and the EMLs described in (Rawlings, van Rosmalen et al. 2002), it is possible to classify the EMLs to be studied (this classification can also be found in (Vantroys 2003)):

- **Evaluation Languages**. This category is formed by languages allowing designers to describe the stages of the learning process in which problem-solving or question-answering are involved in an abstract way.
- **Content Structuring Languages**. This category is formed by languages allowing designers to arrange the learning resources in sequence, always taking into account the learner's needs and performance in order to improve the learning experience.
- **Activity Languages**. This category is made up of languages focused on the activities in general (using computers or not) during the learning process.

Table 3-1. High-level classification of some Educational Modeling Languages

Type of Language	EML
Evaluation Languages	Tutorial Markup Language (TML) http://www.ilrt.bris.ac.uk/netquest
	IMS Question & Test Interoperability (IMS-QTI) http://www.imsglobal.org/question/index.html
Content Structuring Languages	TArgeted Reuse and GEneration of TEAching Materials (Targeteam) http://www.targeteam.net
	Learning Material Mark-up Language (LMML) http://www.lmml.de

Type of Language	EML
	ARIADNE Course (Curriculum) Description Format (A-CDF)[1] http://www.ariadne-eu.org/en/publications/references/index.html
	AICC Course Data Model http://www.aicc.org/
	IMS Simple Sequencing (IMS-SS) http://www.imsglobal.org/simplesequencing/index.html
	ADL Sharable Content Object Reference Model 2004 (ADL SCORM 2004) http://www.adlnet.gov/scorm/index.cfm
Activity Languages	Educational Modeling Language – Open University of the Netherlands (EML-OU) http://eml.ou.nl
	IMS-Learning Design (IMS-LD) http://www.imsglobal.org/learningdesign/index.html
	PALO http://sensei.lsi.uned.es
	Educational Environment Modeling Language (EEML) http://www.istituti.usilu.net/botturil/web/e2ml/index.htm
	Méthode d'ingénierie d'un système d'apprentissage (MISA) http://www.licef.teluq.uquebec.ca/gp/
	XEDU

In Table *3-1* several EMLs are classified according to these categories. The next points give several details about these languages.

3.1 Evaluation Languages

TML/Netquest (Brickley 1995) uses the Tutorial Mark-up Language (TML), which is an extension of HTML intended to produce questions. TML was designed to separate the semantic content of the layout from the question content itself. The TML files are in text format, and can be generated from other formats or other questions in a database.

IMS Question & Test Interoperability (IMS-QTI) is an ongoing effort of the IMS initiative to produce question banks and test banks (IMS-QTI-ASI_INFO 2002; IMS_QTI2-INFO 2006). The principal aim of IMS QTI is to allow the interchange of test and test data between LMSs. An IMS-QTI test decouples the questions themselves (what is being asked) from how to display the questions and how the questions are graded. IMS-QTI allows interactive tests to be created, which allows hints to be included inside questions. It is also possible to create test templates that will be instantiated when students take the test, creating different tests from the same templates.

---

[1] The A-CDF specification is not available at the moment (July 2006), but can be retrieved using the Web Archive service at http://www.webarchive.org

3.2 Content Structuring Languages

Targeteam enables the production and maintenance (use and reuse) of learning material (Koch 2002). This EML allows the use of material in different learning situations and pedagogical domains (primary, secondary and higher education). Using Targeteam, it is possible to create course notes and other contents such as explanations, motivation, and examples. It is focused on the use of an XML-based language, TeachML, and uses the concept of *issue* as a unit of study.

LMML is based on a meta-model that can fit into different application domains. LMML relies on XML for the description of e-learning material (Slavin 1995), and comprises various learning material modules, each one containing other sub-modules. Focused on a conceptual, modular and hierarchical structure of e-learning content, LMML can be adapted to different learning situations and students. It uses the concept of *course* as a unit of study.

ARIADNE Course Description Format (A-CDF) is an EML for the description of learning objects (Verbert and Duval 2004). A course in A-CDF consists of XML documents along with a course generator LMS (Durm, Duval et al. 2001). It places special emphasis on the content and its aggregation, but it is expressive enough to describe the learning process in accordance with a pedagogic model. The didactic material that can be managed through CDF is restricted to text format. It uses a combination of tools developed by the ARIADNE consortium (curriculum editors, LMS, KPS (Duval, Forte et al. 2001)) and establishes the concept of *course* as a unit of study.

IMS Simple Sequencing (IMS SS) (IMS-SS 2003) defines a method for representing the intended behavior of an authored learning experience so that any learning technology system (LTS) can sequence discrete learning activities in a consistent way. A learning designer or content developer declares the relative order in which elements of content are to be presented to the learner and the conditions under which a piece of content is selected, delivered, or skipped during presentation.

AICC Course Data Model (AICC/CMI_CMI001 2004) contains all of the information needed to describe a course. This format may be passed from one LMS system to another through a course import/export process and has Assignable Units (AUs) as its components. Data in this format is also stored internally by the LMS system and is used by the CMI in determining values of the communication data model elements sent to AUs in the course at runtime. The sequencing within a course is controlled using *prerequisites*, which are requirements that must be satisfied by a student before entering a new AU. It uses the concept of *course* as a unit of study.

ADL SCORM (SCORM_OVW 2004) represents a coordinating model intended to give a collection of standard practices that can be generally accepted and widely implemented. Indeed, ADL SCORM can be considered as an application profile of these practices. The SCORM initiative puts into practice different technological developments from groups such as IMS (IMS), AICC (AICC), ARIADNE (ARIADNE), and the IEEE LTSC (IEEE_LTSC); all within a single reference model to specify consistent implementations that can be used throughout the e-learning community. SCORM defines the technical foundations of a web-based LMS, describing:

- A "Content Aggregation Model" (CAM) that describes the components used in a learning experience, how to package those components for exchange, and how to describe those components to enable search and discovery (i.e. metadata). It also defines requirements for building content aggregations (e.g., course, lessons, modules, etc).
- A "Runtime Environment" (RTE) for Learning Objects (LO) to support adaptive instruction based on learning objectives. It describes the Learning Management System (LMS) requirements for managing the run-time environment (i.e communication between content and LMSs). The RTE covers the requirements of Sharable Content Objects (SCOs) (smarts LO) and their use of the API and the SCORM RTE Data Model. The purpose of the SCORM RTE is to provide a means for interoperability between SCOs and LMSs. SCORM provides a means for learning content to be interoperable across multiple LMSs regardless of the tools used to create the content.

In addition, in SCORM 2004 (formerly SCORM 1.3) a "Sequencing and Navigation" model was introduced to allow the dynamic presentation of learning content taking into account learning needs. It is based on IMS Simple Sequencing and describes how SCORM conformant content may be sequenced through a set of learner-initiated or system-initiated navigation events. The branching and flow of that content may be described by a predefined set of activities, typically defined at design time. Also described is how a SCORM conformant LMSs interprets the sequencing rules expressed by a content developer along with the set of learner-initiated or system-initiated navigation events and their effects on the run-time environment.

3.3 Activity Languages

PALO is a modeling language that has been developed by the UNED (Universidad Nacional de Enseñanza a Distancia, Spain) (Rodrıguez-Artacho, Verdejo et al. 1999). PALO describes courses organized into modules that contain learning activities, content, and an associated teaching plan. Using PALO the designer can create templates to define types of learning scenarios. Using the features of the language, it is possible to sequence the learning tasks and modules. In addition course constraints can be created, defining deadlines and dependencies between modules and tasks. It uses the concept of *Module* as a unit of study.

Educational Environment Modeling Language (E^2ML) (Botturi 2006) is proposed as a visual modeling language for the design of educational environments in Higher Education. E^2ML is a visual modeling language, which allows an explicit definition of the learning process and of the educational activities.. In particular, it addresses the following issues:

- It facilitates the communication between the different stake-holders involved in the process (unit of learning designers, technical staff, teachers, etc.) by having a visual representation of the design like the "blueprints" of a building that is going to be built.
- The design of a UoL can be used as basis for another UoL, not only by the same designer, but also by the community.

XEDU (Buendía-García and Díaz-Perez 2003) is oriented towards offering instructional designers a framework for the specification of any instructional application from both instructional design theories and software engineering points of view. The main entities defined in XEDU are: *learner profile*, which stores all relevant information about the learner, including the outcome of the learning process; the *learning scenario* that comprises the activities and the conditions in a specific learning context; and finally the *didactic structure* that organizes educational content with a specific didactic purpose. The concept of *didactic structure* represents a unit of learning in XEDU.

MISA (Paquette, Crevier et al. 1997; Paquette 2004) is a new approach called *instructional engineering* (IE) (Paquette 2001). IE is based on *instructional design* (ID) theories (Reigeluth 1983; Merrill 1994; Dick, Carey et al. 2000) plus software and cognitive engineering. The IE provides a methodology to support the planning, analysis, design and delivery of a learning system, sharing the principles of the EMLs. MISA enables the design of a learning system through 35 tasks, producing 35 main deliverables called documentation elements (DE). The creation of these DE is divided into six well-defined phases. The concept of *learning scenario* represents a unit of learning in MISA.

Finally, among the different proposed specifications, IMS-LD (IMS-LD-MOD 2003) has emerged as the *de-facto* standard for the representation of any unit of learning applying any pedagogical theory. This language is detailed in the next section.

4. IMS LEARNING DESIGN

Based on OUNL-EML, the IMS Learning Design (IMS-LD) specification was drawn up by the IMS/LDWG work group extracting its main concepts and adapting those parts that overlap with other IMS specifications like the packaging for interchange. In this way, an IMS-LD design is embed inside a content package distributed following the IMS-Content Packaging specification. Additionally, some parts of OUNL-EML have not been reused, like the XML syntax (a DocBook (Walsh and Muellner 1999) dialect) for developing the educational content itself.

In addition, one of the aims of IMS LD has been to integrate and work together with other IMS specifications. For example, IMS Simple Sequencing content can be used as a LO inside an activity of IMS LD. Moreover IMS QTI can also be part of an IMS LD activity so that the learner's grades on the test can be used to select the learning activities that will be delivered to the learner.

The high-level structure of a learning design according to IMS LD is sketched in figure *3-1*. Also, to facilitate understanding and the implementation of the specification, it is divided into 3 levels (A, B and C) where each level is built on top of the model and the semantics defined at the previous level:

- Level A contains the core model components of IMS-LD. When a learning design is created, two distinguished parts need to be created: a static and a dynamic part (the dynamic part will be described later). Inside the static part we find: *roles* (that define the type of participants in the learning process), *activities* (what should be done during the learning process), and the *environment* where each activity will be carried out (providing the learning content, LOs, and service tools needed). Also as a header we find: prerequisites (previous knowledge required), learning objectives (what is intended to be learned), some related metadata, and finally the title of the unit of learning.
- Level B introduces *properties* and *conditions*, where properties define a user data model and conditions are used to personalize the presentation of the unit-of-learning, based on if-then-else rules that usually query the values of the learner properties.
- Level C adds notifications behavior. A notification happens after an event and should be emitted by the runtime environment. Notifications can be triggered as an activity is completed or an expression proves to be true.

The dynamic behavior of IMS-LD, where the learning process is defined, can be seen as a theatrical metaphor. A *method* consists of one or several *plays* that are performed in parallel. Each *play* (see figure *3-2* (a)) is made up of one or more *acts* that will be performed in sequence. The acts serve as the synchronization point between the people (roles) involved in the learning design. Inside an *act*, different *role-parts* are found, each one indicating an *activity* and a *role*. These components are references to entities that should have been defined in the static section. During runtime, each user that has the referenced *role* will perform the *activity*. A *role-part* can be seen as a swim lane of a UML Activity Diagram which can only contain simple activities or structured activities (see figure *3-2* (b)).

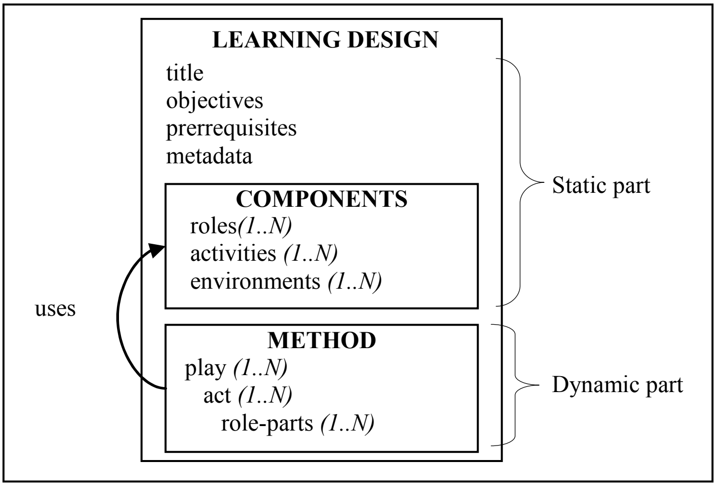

Figure 3-1. IMS-LD Structure

Using Levels B and C more complex dynamic sequencing can be created, in particular dependencies between activities. For example, a tutor could be required to grade an essay after it has been submitted by a learner.

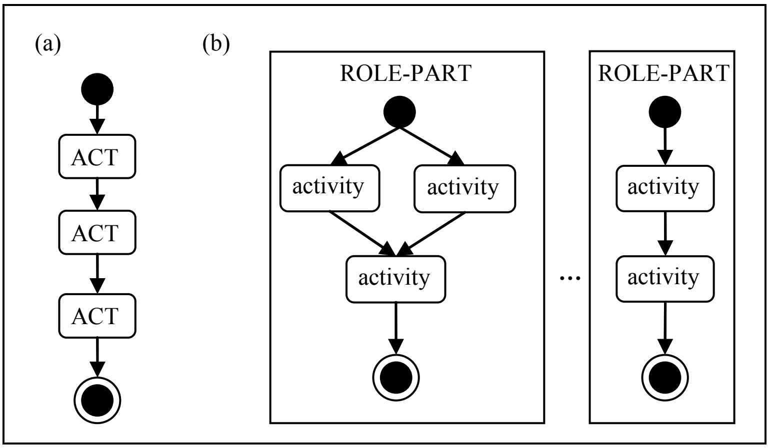

Figure 3-2. UML Activity diagram: (a) A play made up of three acts, (b) An act made up of two role-parts, where the activities are selected as user choice (left) or in a prefixed sequence (right)

5. CHAPTER CONCLUSION

Educational Modeling Languages enable instructors and educators to build and exchange courses based on the general concept of units of learning and taking into account the whole learning process. These learning designs can be part of a repository for their reuse in similar situations. For this purpose, the concept of *learning patterns* applied to LMSs (Avgeriou, Papasalouros et al. 2003) can be useful. The origin of this kind of pattern is from the concept of Object Oriented Programming design patterns (Gamma, Helm et al. 1994): "*Design patterns describe simple and elegant solutions to specific problems [...]. Design patterns capture solutions that have developed and evolved over time. [...]They reflect untold redesign and recoding as developers have struggled for greater reuse and flexibility*". Therefore these patterns can also be exchanged, used and validated by the community, reflecting proven pedagogical approaches for certain contexts.

In addition, an EML serves as a communication language between technical and educational staff. The Educational staff is responsible for the description of the learning experience while the Technical staff is responsible for the implementation of the engine that interprets the EML and the development or integration of the tools used inside the learning process. Therefore, a LMS that supports an EML can be seen as a meta-LMS, enabling non-technical staff to define the behavior of their own LMS without any programming knowledge.

Although IMS-LD is becoming the *de-facto* standard EML, there are a few matters that IMS-LD does not resolve such as the support for groups of learners, who are usually needed in peer-to-peer education, and collaborative learning in general. Also, the specification does not allow the dynamic role changing again needed in peer-to-peer education, where sometimes the learner becomes the teacher/coach of other learners in the group or the class. In addition, it is not usually possible to design an entire course completely and sometimes planning may also have to be adapted to fit the learner level during the course. It must be said that IMS-LD specification deals almost exclusively with static design and does not care about dynamic aspects, so the e-learning community may have to wait for a specification that particularizes IMS-LD like ADL-SCORM does for IMS-Content Packaging. For now IMS-LD only includes four types of service tools that can be used in an environment (mail, conference, monitor, index search), however there are usually more services available in a LMS that might a be useful to launch inside a learning design as well. IMS is concerned with this issue and there are subsequent specifications such as IMS Enterprise Services, IMS General Web Services and IMS Tools Interoperability Guidelines that will hopefully be included in the next versions of IMS-LD.

Finally one of the issues that teachers who want to use an EML come up against is that there are few software developments on EMLs and also few LMS that support EML. IMS-LD may be the exception thanks to the development of Coppercore (a IMS-LD engine) (Martens and Vogten 2005; OUNL 2005) and partially in LAMS (an LMS and authoring environment based on activities) (Dalziel 2003; LAMS 2005). However, some problems have been found (Barrett-Baxendale, Hazlewood et al. 2005): (1) The integration of all the software tools involved (LMS, IMS-LD engine, authoring tools) needs to be facilitated, (2) the usability of these tools must be improved to disguise the fact that at the present time IMS-LD and XML are behind the scenes, (3) the reliability of the software tools should be improved.

These considerations raise potential lines for future research into enhancing the current EML arena. These lines include the development of graphical tools for IMS-LD-based authoring (UML graphical notation is an interesting starting point) and the application of existing mature technologies and tools regarding workflow systems (Dumas, Aalst et al. 2005) in the context of running dynamic content. At the moment our group at the Universidad Compluentse de Madrid is engaged in these lines of research.

ACKNOWLEDGES

The Spanish Committee of Science and Technology (TIN2004-08367-C02-02 and TIN2005-08788-C04-01) and the Regional Government of Madrid (4155/2005) have partially supported this work.

REFERENCES

AICC. “Aviation Industry CBT Committee.” Retrieved July 1, 2006, 2006, from http://www.aicc.org.

AICC/CMI_CMI001 (2004). CMI Guidelines for Interoperability AICC revision 4.0, Aviation Industry CBT Committee - AICC Subcommittee: 241.

ARIADNE. “Alliance of Remote Instructional Authoring and Distribution Netoworks for Europe (ARIADNE).” Retrieved July 1, 2006, 2006, from http://www.ariadne-eu.org.

Avgeriou, P., A. Papasalouros, et al. (2003). “Towards a Pattern Language for Learning Management Systems.” Educational Technology & Society 6(2): 11-24.

Barrett-Baxendale, M., P. Hazlewood, et al. (2005). SliDe, SLeD Demonstrator - Final Report v 1.0. Hope Park (United Kingdom), Liverpool Hope University: 13.

Botturi, L. (2006). “E2ML: A Visual Language for the Design of Instruction.” Educational Technology Research and Development 54(3): 265-293.

Brickley, D. (1995). “Towards an open question-interchange framework.” Retrieved June 1, 2006, from http://www.ilrt.bris.ac.uk/netquest/about/lang/motivation.html.

Buendía-García, F. and P. Díaz-Perez (2003). "A Framework for the Management of Digital Educational Contents Conjugating Instructional and Technical Issues." Educational Technology & Society 6(4): 48-59.

Dalziel, J. (2003). Implementing Learning Design: The Learning Activity Management System (LAMS). 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE 2003), Adelaide, Australia.

Dick, W. O., L. Carey, et al. (2000). The Systematic Design of Instruction (5th Edition), Allyn & Bacon.

Downes, S. (2001). "Learning Objects: Resources For Distance Education Worldwide." The International Review of Research in Open and Distance Learning 2(1).

Dumas, M., W. M. P. v. d. Aalst, et al. (2005). Process-Aware Information Systems: Bridging People and Software through Process Technology, Wiley-Interscience.

Durm, R. V., E. Duval, et al. (2001). Extending The ARIADNE Web-Based Learning Environment. World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2001), Tampere, Finland.

Duval, E., E. N. Forte, et al. (2001). "The ARIADNE knowledge pool system." Communications of the ACM 44(5): 72-78.

Fernandez-Manjon, B. and P. Sancho (2002). "Creating cost-effective adaptive educational hypermedia based on markup technologies and e-learning standards." Interactive Educational Multimedia 4: 1-11.

Gamma, E., R. Helm, et al. (1994). Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley.

IEEE-LOM (2002). IEEE Standard for Learning Object Metadata: 39.

IEEE_LTSC. "Institute of Electrical and Electronics Engineers (IEEE) Learning Technology Standards Committee (LTSC)." Retrieved July 1, 2006, from http://ltsc.ieee.org/.

IMS-LD-MOD (2003). IMS Learning Design Information Model Version 1.0 Final Specification. IMS Specifications, IMS Global Learning Consortium.

IMS-QTI-ASI_INFO (2002). IMS Question & Test Interoperability: ASI Information Model Specification, IMS Global Learning Consortium.

IMS-SS (2003). IMS Content Simple Sequencing Information and Behavior Model Version 1.0 Final Specification, IMS Global Learning Consortium.

IMS. "IMS Global Learning Consortium." Retrieved July 1, 2006, 2006, from http:// www.imsproject.org.

IMS_QTI2-INFO (2006). IMS QTI and Test Interoperability Assessment Test, Section, and Item Information Model, IMS Global Learning Consortium.

Koch, M. (2002). "Interoperable Community Platforms and Identity Management in the University Domain." International Journal on Media Management 4(1): 21-30.

Koper, R. (2000). From Change to Renewal: Educational Technology Foundations of Electronic Learning Environments, Open Univerity of the Netherlands: 41.

Koper, R. (2001). Modeling units of study from a pedagogical perspective : the pedagogical meta-model behind EML., Educational Technology Expetise Centre (OTEC, Open University of the Netherlands: 40.

Koper, R. (2003). Combining re-usable learning resources and services to pedagogical purposeful units of learning. Reusing Online Resources: A Sustainable Approach to eLearning. A. Littlejohn. London, Kogan Page: 46-59.

LAMS. (2005). "LAMS v 1.0.2." Retrieved July 1, 2006, from http://www.lamsfoundation.org/.

Martens, H. and H. Vogten (2005). Reference Implementation of a Learning Design Engine. Learning Desing: a Handbook on Modelling and Delivering Networked Education and Training. R. Koper and C. Tattersall, Springer: 91-108.

Martinez-Ortiz, I., P. Moreno-Ger, et al. (2005). Using DocBook To Aid in the Creation of Learning Content. New Trends and Technologies in Computer-Aided Learning for Computer-Aided Design. A. Rettberg and C. Bobda, Springer: 147.

Martinez-Ortiz, I., P. Moreno-Ger, et al. (2006). Production and Deployment of Educational Videogames as Assessable Learning Objects. First European Conference on Technology Enhanced Learning (ECTEL 2006), Crete, Greece, LNCS.

Merrill, M. D. (1994). Instructional Design Theory, Englewood Cliffs.

Moreno-Ger, P., I. Martinez-Ortiz, et al. (2005). The <e-Game> Project: Facilitating the Development of Educational Adventure Games. International Conference on Cognition and Exploratory Learning in the Digital Age (CELDA 2005), Porto, Portugal.

OUNL. (2005). "Coppercore v 3.0." Retrieved July 1, 2006, from http://www.coppercore.org.

Paquette, G. (2001). "TeleLearning Systems Engineering - Towards a new ISD model." Journal of Structural Learning 14: 1-35.

Paquette, G. (2004). Educational Modeling Languages, from an Instructional Engineering Perspective. Online education using learning objects. R. McGreal. London, Routledge/Falmer: 331-246.

Paquette, G., F. Crevier, et al. (1997). "Méthode d'ingénierie d'un système d'apprentissage (MISA)." REview Informations In Cognito 8.

Paquette, G., I. de la Teja, et al. (2005). Using an Instructional Engineering Method and a Modeling Tool. Learning Desing: Modelling and Implementing Network-based Eduation & Training. R. Koper and C. Tattersall, Springer-Verlag.

Rawlings, A., P. van Rosmalen, et al. (2002). Survey of Educational Modelling Languages (EML), CEN/ISSS WS/LT Learning Technologies Workshop: 79.

Reigeluth, C. M. (1983). Instructional Design Theories and Models: An Overview of Their Current Status, Lawrence Erlbaum Associates.

Rodrıguez-Artacho, M., M. F. Verdejo, et al. (1999). Using High-Level Language to Describe and Create Web-Based Learning Scenarios. IEEE Frontiers In Education FIE '99, San Juan, Puerto Rico.

SCORM_OVW (2004). SCORM 2004 Overview 2nd Edition, Advanced Distributed Learning: 57.

Slavin, R. E. (1995). Cooperative Learning: Theory, Research and Practice (2nd Edition), Allyn & Bacon.

Vantroys, T. (2003). Du langage métier au langage technique, une plate-forme flexible d'exécution de scénarios pédagogiques. Laboratoire TRIGONE - Équipe NOCE, Université des Sciences et Technologies de Lille: 182.

Verbert, K. and E. Duval (2004). Towards a Global Component Architecture for Learning Objects: A Comparative Analysis of Learning Object Content Models. World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004, Chesapeake, VA, AACE.

Walsh, N. and L. Muellner (1999). DocBook: The Definitive Guide. Sebastopol, CA, USA, O'Reilly.

Weitl, F., C. Süß, et al. (2002). Presenting Complex e-eLearning Content on the WEB: A Didactical Reference Model. Proceedings of E-Learn 2002 world conference on E-Learning in Corporate, Government, Healthcare, & Higher Education, Montreal, Canada.