HAL open science

EMLs : case study in distance learning education

Vincent Barré

▶ **To cite this version:**

Vincent Barré. EMLs : case study in distance learning education. International Conference on Computer Aided Learning in Engineering education CALIE04, 2004, Grenoble, France. hal-00190236

HAL Id: hal-00190236
https://telearn.hal.science/hal-00190236

Submitted on 23 Nov 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

EMLs : CASE STUDY IN DISTANCE LEARNING EDUCATION

Vincent Barré (Vincent.Barre@univ-lemans.fr)
LIUM, IUT de Laval
BP 2045, 53020 LAVAL Cedex 09 - FRANCE

KEYWORDS: Norms, Educational modeling language

Abstract

After a short characterization of what an educational modeling language should be, we will deal with the respective interest of two EML proposals : one proposed by OUNL (OUNL/EML and IMS/LD which is derived from it) and one proposed by UNED (PALO). We will then identify reasons that lead us to choose OUNL/EML for experimentation in distance learning education in order to model our scenario. Finally we will present some limitations or extensions we have identified.

INTRODUCTION

Course design for learning management systems (LMS) partly consists in handling digital resources in order to produce other digital resources. If we want to comprehend this process better, we can differentiate four granularity levels describing these pedagogical resources (Arnaud *et al.* 2001) : (0) *assets* (raw content), (1) *content unit* (aggregated assets that form an inseparable pedagogical object that must have a meaning independent from its context), (2) *composite units* (amalgamation of objects from levels 1 or 2 coming generally with taxonomy and navigation rules) and (3) *course* (the course on its own, including possible prerequisites validation and an evaluating system). LMS allows, among other things, the diffusion at all levels of pedagogical resources (of any level) to learners. In order to respond to an increasing need for reusability and interoperability, these systems generally rely on known specifications such as those published by AICC or ADL. One of the more widely used specifications is proposed by ADL[i] (Advanced Distributed Learning, an initiative launched in 1997 by The Department of Defense with a number of American universities and firms). The main goal of this specification (SCORM 1.2) is to provide a neutral pedagogical standard that allows course designers to link pedagogical resources together in order to build a curriculum. This specification differentiates three levels of pedagogical resources (matching the first three preceding levels): *Asset* (level 0*)*, *Sharable Content Object* (level 1) and *Content Aggregation* (level 2). We will focus more particularly on Content Aggregation, which describes the way we can navigate between SCO. This navigation is defined in a structure that uses prerequisites in order to identify SCO (or CA) scheduling. Compared to preceding systems, such a vision has the advantage of defining navigation information separately from the content. Nevertheless, in the conceptual description of these systems (based on AICC or SCORM specifications), there is a need for another abstraction level that will allow the description of system components in a pedagogical way (instead of the traditional computer vision as present in SCORM Content Aggregation). More precisely, this pedagogical vision is not really absent but included in the LMS or hard coded by the designer. However, the next evolutions of SCORM specification (SCORM 1.3) will consider this question and provide some *ad hoc* solutions.

A significant effort has been made by the international community to model and describe pedagogical objects (see for example IEEE/LTSC[ii] work on LOM and more recently work done by ISO[iii] on MLR). But a “real” pedagogical content cannot be reduced to simple object gathering, it is more likely to be a set of pedagogical components including learning processes definition and activities, and an explicit pedagogical design. An *Educational Modeling Language* (EML) is a *modus operandi* that describes "units of learning" in a pedagogical way. Once described in this *modus operandi*, a scenario will be interpreted by appropriate software component (as HTML is interpreted by a browser for example).

Several educational modeling languages, such as OUNL/EML[iv] from Open University of the Netherlands (OUNL) (Koper 2001) or PALO[v] from Universidad Nacional de Educación a Distencia de Madrid (Rodriguez-Artacho 2002) are available. Recently, a number of attempts to produce to a standard modeling language have begun. In the United States, the IMS Global Learning[vi] consortium (whose goal is to promote open specifications to make learning technologies interoperable) have just published their *Learning Design* specification (Koper *et al.* 2003) that integrates OUNL work on their EML to the preceding IMS specifications (like Content Packaging, Simple Sequencing…). On a European level, the CEN/ISSS Workshop on Learning Technologies[vii] has formed a workgroup on EML to define a generic model of what an EML should be (Rawlings *et al.* 2002).

So, what are these languages really and what can they bring to LMS ? In the framework of the MOCA project (Barré *et al.* 2003), we have launched an experiment whose main goals were: (1) to determine the distance between learner activities and preconceived scenarios, (2) and to gather information on interactions between human actors (intervention strategies and content).

In order to investigate the first point, we have decided to use an EML to model pedagogical activities. During this experimentation, used units of learning were related to network services courses for students and were composed by the following six activities: (1) *introduction* (a video that presents all concepts needed to understand the unit of learning), (2) *text study* (some reference texts on web protocols), (3) *slide show* (two slide shows, one on the functioning of a web server and the other one describing Java code of a web server), (4) *case study* (some exercises about web server programming), (5) *check up* (two tests) and (6) *learning by doing* (in which learners use all their knowledge on network services to realize, step by step, a working web server). In our experiment, we have a predicted scenario but we choose to keep learners free to navigate as they want. We want to model both of these scenarios with an EML. Moreover we keep in mind that in a subsequent experiment of this system, we will transform activity (6) into a collaborating activity. Below, we will briefly characterize what an educational modeling language should be and deal with the respective interest of the two EML proposals: one by OUNL (OUNL/EML and IMS/LD which is derived from it) and the other by UNED (PALO). Reasons for experimenting with OUNL/EML to model our distance learning scenario will be given, before finally presenting some limitations or extensions we have identified.

EDUCATIONAL MODELING LANGUAGES

A first outline of a generic conceptual model for an *Educational Modeling Language* was proposed by the CEN/ISSS WS LT (Rawlings *et al.* 2002). This model is built on the concept of units of learning; such a unit describes scenarios, resources and services required to reach one or more learning objectives. From this viewpoint, a unit of learning cannot be broken into smaller parts without redefining its main learning objectives, but a unit of learning can contain other units of learning, with different learning objectives or with objectives contributing to the main objectives of the parent unit. The essential idea, on which the CEN/ISSS WS LT model and many existing EMLs are based, is that whatever pedagogical approach is used (for example a behaviorist approach or a cognitive one), any scenario can be modeled as a method specifying activities to certain actors and in a certain order. Activities allow actors to link with the pedagogical objects and services needed to accomplish pedagogical activities. Actors can of course be learners, but also tutors or course designers, while the order in which activities (for learners, tutors…) are to be done may be described by the course designer (but not necessarily).

In order to allow the description of individual learning paths, learner properties will have to be considered along with possible prescribed scheduling modifications (with conditions). Those scenarios can involve one or more participants (of the same type or not) that can work alone or collaborate with other participants. An EML should allow for description of “hybrid” scenarios containing digital resources, course attendance or “concrete” resources. Actually, the CEN/ISSS WS LT report identify six EML systems, but only two of them satisfy their definition of an EML: OUNL/EML from the OUNL (and by extension IMS/LD) and PALO from UNED. The next two sections will briefly present them.

IMS LEARNING DESIGN

The main objective of the IMS “Learning Design” specification (Kopper *et al.* 2003) was “the development of a framework that supports pedagogical diversity and innovation, while promoting the exchange and interoperability of e-learning materials”. This specification falls into the preceding framework and we will first review the conceptual model of IMS/LD specification (dotted lines correspond to the ‘SCORM’ vision of learning activities):

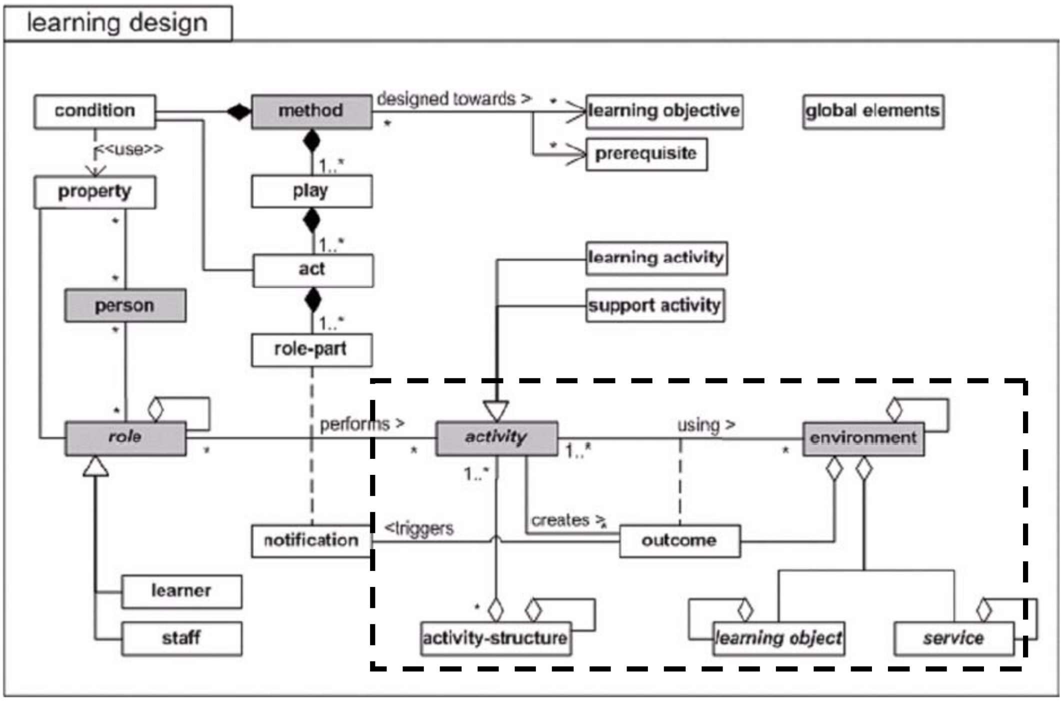

Fig. 1: IMS Learning Design 1.0 conceptual model

As specified by the generic model of an EML, the main idea is that in any pedagogical approach, one person has a role in the learning process. In this role, he works to produce outcomes by carrying out learning or support activities in a given environment. These activities play a central role into IMS/LD specification since they form the link that associates roles, learning objects and environments. Environments are formed by learning objects and services to be used during activities. To know who does what (which role) and at which moment is determined by the method (which, strictly speaking, can be considered as the scenario). A method is build to satisfy learning objectives (*i.e.* what has to be done by learners) and assumes some prerequisites; moreover, it is constituted by one or more competing plays. Each play consists of a sequence of one or more acts associated to one or more role-parts (that associate a role to an activity or an activity-structure). A method can also contain conditions to control the visibility of the different activities for the people and their roles using Boolean conditions based on properties (linked to a person or a role). As an example, we can say that a scenario must involve some basic components of a unit of learning (roles, resources, activities and methods):

```
<learning-design identifier="SRC2" uri="URI" level="A">
<components>
  <roles>
    <learner identifier="R_learner"/>
  </roles>
</components>
<method>
  <play identifier="P_SRC2_SR" isvisible="true">
    <act identifier="ACT_individualized_learning">
      <role-part identifier="RP_individualized_learning">
        <role-ref ref="R_learner"/>
        <activity-structure-ref ref="AS_exo1 "/>
      </role-part>
    </act>
  </play>
</method>
</learning-design>
```

In this example, activities are defined in another unit of learning. If we had decided to define them here, we would have done it in a ‘component’ section. Thus, it becomes possible to define different activities intended for learners and/or tutors. One way to control activity sequencing is to structure them using an activity-structure. For example, one can state that the learner has to complete two activities in any order or one activity before the other. Activities or activity-structures defined in this way can take place within specific environments (resources that can be used to complete an activity). A unit of learning or an activity can contribute to one or many learning objectives and it is thus interesting to be able to determine completion of an activity. More generally, it can be interesting to be able to determine the completion of any step of the scenario (act, play or method) and to react in consequence:

```
<method>
  <play identifier="P_SRC2_SR" isvisible="true">
    <act identifier="ACT_individualized_learning">
      <role-part identifier="RP_individualized_learning">
        <role-ref ref="R_learner"/>
        <activity-structure-ref ref="AS_exo1 "/>
      </role-part>
      <complete-act>
        <when-role-part-completed ref="RP_individualized_learning"/>
      </complete-act>
    </act>
    <complete-play>
      <when-last-act-completed/>
    </complete-play>
  </play>
</method>
```

PALO

PALO (Rodriguez-Artacho 2002) is an EML developed by UNED since1988 (as defined by the CEN/ISSS WS LT) that allows the description of courses structured in modules. Each module contains (i) structure definition, (ii) tutors and learner activities, (iii) content and activity scheduling. Designers of PALO language have consider an approach resulting in a strict distinction between the content of learning objects and the effective use that can be made with these objects (such as learning strategies or description of learning paths). PALO restricts itself to the second aspect. In PALO, activity (and thus module) scheduling is done using language attributes. These attributes define deadlines and dependencies between activities (or modules) based on different prerequisite types. PALO defines learning strategies with *instructional templates* which are some scenario models with particular pedagogical properties (for example a constructive approach or a behavioral one). Thus PALO can be extended each time there is a need for a certain pedagogical functionality. It is important to note that these instructional templates are not different languages but rather subsets of the PALO language itself, defined by XML or SGML DTDs, that express a particular pedagogy. Thus, when we want to create a new scenario, we first need to select the right DTD (*i.e.* instructional template), and then create the PALO description using this particular DTD. Elements constituting the PALO language can be split into five groups:

Layer 1 (Pedagogical content): description of the pedagogical content (of any type, but without information about the context or associated processes). This content can be retrieved from external databases (like ARIADNE).

Layer 2 (Activities): In PALO, task description not only consists of work description strictly speaking, but also of other aspects such as available resources or supplied tools.

Layer 3 (Structure): description of hierarchical decomposition of the learning environment (for example, stating that a course is composed of one or more units of learning composed as well of one or more parts or themes). It is also possible to link some pedagogical information (such as prerequisites, usability or user interface) to each element of the hierarchy.

Layer 4 (Scheduling): description of temporal constraints (such as deadlines) as well as pedagogical dependencies (such as prerequisites) between elements of the structure (or between activities, but in that case it is, in part, a layer 3 description).

Layer 5 (Management): management of learning environments, definition of the location of data, resources and tools previously mentioned as well as metadata. We also find elements describing models used during learning sessions (for example, models and databases used to monitor system activity).

For a given instructional template, each of these 5 layers will contain some of the elements present in the general PALO information model. It is important to notice that actually cooperative activities cannot be defined in PALO, but a forthcoming version will support them.

CONCLUSION AND PROSPECTS

In the framework of our experimentation, we have chosen OUNL language (OUNL/EML) to model course designer prescribed scenarios. This choice is bound to some interesting aspects of this language:

- OUNL/EML is clearly involved in a standardization process (as shown by the active participation of many OUNL members in standardization committees and by the IMS adoption of OUNL/EML for their *Learning design* specification) and is compatible with various important international standards.
- It is an “open” specification and software development in progress must lead to open plug-ins that can insert themselves in e-learning platforms. Moreover authoring tools will be available soon.
- There exists an XML *binding*.
- It supports any king of pedagogy, including cooperatives activities.

In this perspective, IMS/LD (or rather OUNL/EML at the time) seemed to be able to answer (at least partially) to our modeling needs. Since an EML has been conceived to describe pedagogical scenarios independently from handled or given physical resources, all EMLs that conform to the CEN/ISSS WS LT definitions should have met our first objective. Nevertheless, some of our needs were only partially satisfied. Compared to what was previously done (for example in AICC or SCORM specifications), EML brings into consideration the different roles and so expresses interventions from all actors of the system in the scenario. However, tools offered by OUNL/EML do not allow us to take into account the diversity of tutoring strategies that we want to express in a simple and satisfying way. For example, it is difficult in OUNL/EML (but not in IMS/LD with a notification mechanism) to model opportunist tutoring strategies (where the tutor reacts to a situation in which some help appears to be interesting).

In addition to this, we also think that OUNL/EML can be very useful to model computer tracks collected during a learning session. From this viewpoint, it will be interesting if course designers can add some information to the learning design to guide the collection of feedback by the system. This information, together with computer tracks, will help us in a subsequent phase, in which we need to build scenarios with gathered information in order to compare them with preconceived scenarios. This is what we are doing actually, and for this particular aspect of the use of an EML, IMS/LD seems to be the most indicated among EML proposals we have studied.

REFERENCES

ARNAUD M., NARDIN R., PERRIAULT J. & SAILLANT J.-M. (2001), Rapport d'étape sur les pratiques en matière de normes et standards pour l'apprentissage en ligne, Université Paris X Nanterre / GEMME 2001.

BARRE V., CHOQUET Ch., CORBIERE A., COTTIER Ph., DUBOURG X. & GOUNON P. (2003), MOCA, une approche expérimentale de l'ingénierie des EIAH, Environnements Informatiques pour l'Apprentissage Humain, Strasbourg 2003

KOPER R. (2001), Modeling units of study from a pedagogical perspective : the pedagogical meta-model behind EML, 06/2001.

KOPER R., OLIVIER B. & ANDERSON T. eds. (2003), IMS Learning Design Information Model, IMS Global Learning Consortium, Inc., version 1.0, 20/01/2003.

RAWLINGS A., ROSMALEN P., KOPER R., RODRIGUEZ-ARTACHO M. & LEFRERE P. (2002), Survey of Educational Modelling Languages (EMLs), CEN/ISSS WS/LT Learning Technologies Workshop, version 1, 19/09/2002.

RODRIGEZ-ARTACHO M. (2002), PALO Language Overview, Universidad Nacional de Education a Distancia, Technical Report TR-2002-01.

---

[i] http://www.adlnet.org/
[ii] http://ltsc.ieee.org/
[iii] http://jtc1sc36.org/
[iv] http://learningnetworks.org/
[v] http://sensei.lsi.uned.es/palo/
[vi] http://www.imsglobal.org/
[vii] http://www.cenorm.be/isss/