# Enhancing Reusability of IMS LD Units of Learning: The e-LD Approach

Iván Martínez-Ortiz, José Luis Sierra, Baltasar Fernández-Manjón

*Dpto. Ingeniería del Software e Inteligencia Artificial.*
*Fac. Informática. Universidad Complutense de Madrid. 28040, Madrid (Spain).*
*{imartinez, jlsierra, balta}@fdi.ucm.es*

## Abstract

*In this paper we describe the e-LD approach for the design and repurposing of Units of Learning (UoLs). This approach is centered in domain-specific Educational Modeling Languages (EMLs) built with the close collaboration between instructors and developers. The products of this collaboration are: (i) the definition of a suitable authoring EML, and (ii) the construction of software tools to allow the importation and authoring of UoLs. The domain-specific authoring EMLs and tools simplify the production and repurposing of UoLs.*

***Keywords:*** *Educational Modeling Languages, Authoring, Repurposing, Importation, Exportation, Language-driven approach*

## 1. Introduction

Educational Modeling Languages (EMLs) provide instructors with a formal notation to describe their teaching methods. The resulting formalized methods, together with the other elements required for their execution, are commonly known as Units of Learning (UoLs). In addition, EMLs usually offer operational semantics which allow the interpretation of the UoLs, that is, the automation of the mentioned teaching methods.

Currently there are available a few EMLs with wide difference in their expressiveness and generality, ADL SCORM Sequencing and Navigation [1] (SCORM-SN), IMS Simple Sequencing [6] (IMS SS) and IMS Learning Design [5] (IMS LD) being some of the most popular ones. Although all these EMLs have been designed to support some use cases, their expressiveness makes their use very complex for the average instructor hindering the reuse of UoLs.

To address this complexity-expressiveness balance, our approach distinguishes conceptually two kinds of EMLs: *exchange EMLs* and *authoring EMLs*. Exchange EMLs have a large expressiveness including too low level characteristics that are not very relevant for the instructors. Therefore, these EMLs make up an abstraction layer between e-learning applications, virtually allowing instructors to customize an e-learning platform to their specific needs. Whereas, authoring EMLs have a more restricted expressiveness but also supporting the needs of instructors and providing a high level of abstraction. Because authoring EMLs are specifically adapted to the instructors' bodies of expertise, authoring and repurposing tasks are dramatically eased. That way IMS LD should be classified as an exchange EML.

Our approach, called e-LD [7], proposes a collaborative process model for designing domain-specific authoring EMLs. This process involves *instructors*, as well as *developers*, experts in computer science who provide support to instructors during authoring and repurposing. The outcome of this process is an EML specially oriented to simplify authoring. The aspects of interoperability with standard EMLs are addressed by importation/exportation facilities.

The rest of this paper is structured as follows: Section 2 introduces the e-LD approach, section 3 exemplifies the approach applied to IMS LD and finally section 4 provides some conclusions and lines of future work.

## 2. The e-LD approach

In this section we analyze the e-LD process model from three perspectives: Products and tasks, scheduling of tasks, and participants and their roles.

### 2.1. Products and tasks in e-LD

The products and tasks of the e-LD process model are depicted in Figure 1.

- The *Design of Authoring EML* task comprises the definition of an authoring EML. This authoring EML will prime usability on expressivity. In addition, authoring EMLs are designed in two different levels: as abstract conceptual models, and as concrete bindings introducing different authoring notations.

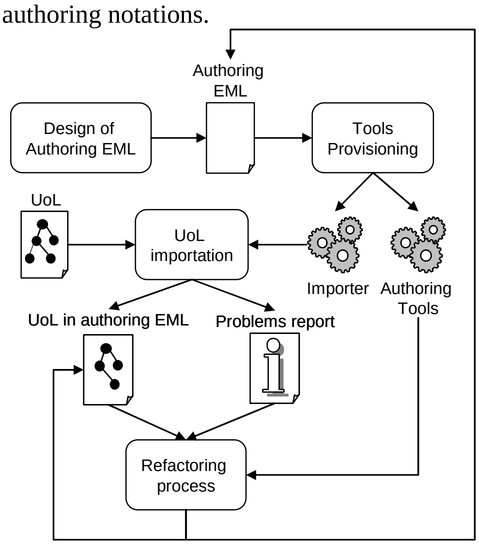

Figure 1 Products and tasks in e-LD

- The *Tools Provisioning* task involves the development of a suitable authoring tool, which will be developed based on the authoring EML model using a particular authoring notation. Also an importer is developed acting as a bridge between the authoring and the exchange EML.
- During the *UoL importation* task two products are generated: a UoL described using the authoring EML model and a report with the problems found. The problems' report contains a set of annotations regarding which parts of the source UoL couldn't be translated automatically.
- Finally during the *Refactoring process* task, the UoL generated in the *UoL importation* task is redesigned accordingly to the needs of instructors but also taking into account the problems' report.

## 2.2. Scheduling of tasks in e-LD

Figure 2 shows the scheduling of tasks in the e-LD process model. The execution of these tasks is interleaved in time. According to this incremental and collaborative process, the authoring EML and the associated tools can evolve whenever new expressivity requirements are identified by instructors.

The e-LD process model introduces two types of iterations in the construction and repurposing of UoLs (highlighted in Figure 2): *corrective iterations* and *evolutionary iterations*. *Corrective iterations* are mainly related to the fine tuning of the importer and the authoring tools. *Evolutionary iterations* are started after the identification of lack of concepts to express the teaching methods of instructors in the authoring EML.

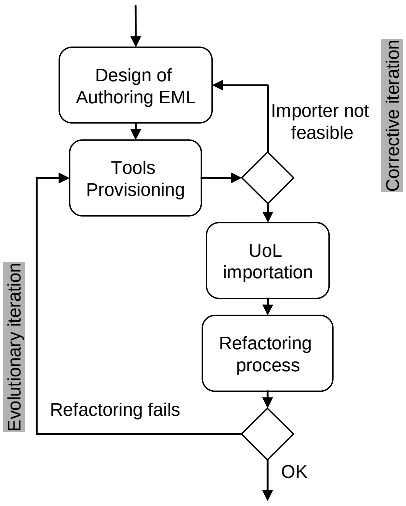

Figure 2 Scheduling of tasks in e-LD

## 2.3. Participants and their roles in e-LD

Instructors and developers are the two main actors that participate in the e-LD process model.

- During the *Design of Authoring EML* task, instructors describe their teaching methods to developers who will be in charge of the formalization of the authoring EML.
- During the *Tools Provisioning* task developers develop the importer and authoring tools driven by the authoring EML model. Instructors assess developers regarding the usability of tools.
- In the *UoL Importation* task instructors import the previously created UoL and interact with the tools to fix all the identified problems in the importation process. Developers assess and provide technical assistance to instructors.
- Finally, during the *Refactoring process* instructors use the authoring tool to refactor the imported UoL. Developers help instructors with the interpretation of the generated problems' report.

# 3. An example of e-LD approach applied to IMS LD

This section exemplifies how the e-LD approach is applied for supporting the reuse of already created UoLs using IMS LD. We have successfully applied the e-LD approach to the reuse of different IMS LD UoL examples available in the IMS LD specification and community repositories. The process model has been tested in a postgraduate e-learning course with PhD students, and with instructors of our Computer Science Studies.

### 3.1. Design of the Authoring EML

The main points identified during the design of the authoring EML are the mechanisms for defining learning activities and their sequencing.

As computer scientists are familiar with the description of algorithms and protocols using flowchart diagrams and process diagrams, in this case a flow oriented authoring metaphor has been chosen for representing the sequencing of learning activities in the authoring EML.

### 3.2. Tools provisioning

To facilitate the management and evolution of the authoring EML designed, we have adopted a model-driven development approach [8]. For this purpose, we have characterized the abstract conceptual model of the EML as a meta-model. Based on this meta-model the authoring tool has been developed using the Eclipse Platform [4].

In addition to the authoring EML meta-model, an EMF meta-model for IMS LD has been defined too. Based on the authoring EML and IMS LD meta-models the importation tool has been developed as a translator between meta-models.

### 3.3. UoL Importation

The importer has been tested against IMS LD UoL of levels A, B and C. Support of IMS LD Level B has been the most complex process due to the behavior of properties and conditions. IMS LD conditions are similar to artificial intelligence rule-based systems [2]. Therefore, the mapping of this rule-based behavior to a flow-oriented formalism is a difficult problem. Indeed, not all the conditions can be smoothly translated to conditions in the flow definition, becoming an annotation in the problems' report for further analysis during the refactoring process.

### 3.4. Refactoring process

The authoring tool is being used during the refactoring process. During the performance of this activity, instructors have to address the problems' report to tweak and adapt the imported UoL. After all problems are fixed, instructors can use the flow oriented metaphor to adapt the UoL to their needs.

## 4. Conclusions and Future Work

The e-LD approach described in this paper lets instructors carry out the authoring and repurposing of UoLs. For this purpose domain-specific authoring EMLs are formulated and importation and authoring tools are developed. These tools are used by instructors during the creation and re-authoring of UoLs. We have realized that this approach is very valuable to facilitate the application of EMLs in the educational community, but also facilitates the reusability of preexisting UoLs. In addition, the application of the model driven development process speeded up the development of the tools used in the e-LD approach letting participants perform iterations over the e-LD process model in a very agile way.

Our experiences applying the e-LD approach has been focused on its application to IMS LD. However we believe that a valuable future line of work will be the application of this approach to other EMLs like IMS SS and SCORM SN.

## 5. Acknowledgements

The Spanish Committee of Science and Technology (projects TIN2005-08788-C04-01, FIT-360000-2007-23 and TIN2007-68125-C02-01) has partially supported this work, as well as the Complutense University of Madrid (research group 921340), and the Santander/UCM project PR34/07-15865.

## 6. References

[1] Advanced Distributed Learning (ADL), *Shareable Content Object Reference Model (SCORM) 2004 3rd Edition Sequencing and Navigation Version 1.0*, 2006.

[2] Brownston, L., Farell, R., Kant, E., Martin, N., *Programming Experts Systems in OPS5: An Introduction to Rule-based Programming*, Adisson-Wesley, 1985.

[4] Clayberg, E., Rubel, D., *Eclipse: Building Commercial-Quality Plug-ins, 2nd Edition*, Addison Wesley Professional, 2006.

[5] IMS Specification, *IMS Learning Design 1.0*, IMS Consortium, http://www.imsglobal.org/learningdesign, 2003

[6] IMS Specification, *IMS Simple Sequencing Specification 1.0*, IMS

[7] Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J.L., Fernández-Manjón, B., Supporting the Authoring and Operationalization of Educational Modelling Languages, *Journal of Universal Computer Science* 13(7), 938-947. 2007

[8] Stahl, T., Voelter, M., Czarnecki, K., *Model-Driven Software Development: Technology, Engineering, Management*, Wiley. 2006