*Journal of Universal Computer Science, vol. 13, no. 7 (2007), 938-947*
*submitted: 7/3/07, accepted: 25/7/07, appeared: 28/7/07* © *J.UCS*

# Supporting the Authoring and Operationalization of Educational Modelling Languages

**Iván Martínez-Ortiz**
(Dept. of Software Engineering and Artificial Intelligence
Complutense University of Madrid, Spain
imartinez@fdi.ucm.es)

**Pablo Moreno-Ger**
(Dept. of Software Engineering and Artificial Intelligence
Complutense University of Madrid, Spain
pablom@fdi.ucm.es)

**José Luis Sierra-Rodríguez**
(Dept. of Software Engineering and Artificial Intelligence
Complutense University of Madrid, Spain
jlsierra@fdi.ucm.es)

**Baltasar Fernández-Manjón**
(Dept. of Software Engineering and Artificial Intelligence
Complutense University of Madrid, Spain
balta@fdi.ucm.es)

**Abstract:** The modelling of educational processes and their operational support is a key aspect in the construction of more effective e-learning applications. Instructional models are usually described by means of an *educational modelling language* (EML). The EML used can be one of the available standards (e.g. IMS Learning Design), the customization of a standard to meet a specific application profile, or even a domain-specific EML specifically designed to better fit the very particular needs of a learning scenario. In this paper we present <e-LD>, a general authoring and operationalization architecture capable of dealing with all these possibilities in a highly modular and flexible way. We also outline a specific implementation of <e-LD> based on standard XML technologies and workflow management systems, and we describe how this implementation can be used to support IMS Learning Design.

**Keywords:** IMS LD, learning design, graphical notation, units of learning, authoring
**Categories:** K.3.1, K.3.2

## 1 Introduction

In the last years, the representation of learning content in the form of potentially reusable learning objects that can be adapted and assembled in many different e-learning scenarios [Downes 2001, Polsani 2003] has received a lot of attention. However this is only a piece of the whole puzzle. To obtain a full learning experience, the learning process usually involves other activities like solving some problems,

making some experiments in a lab, discussing with the teacher about a specific topic, etc.

Thus, an optimal e-learning application should coherently integrate resources and participants such as students or instructors across a well-defined set of learning activities that are structured carefully and delivered in a learning flow to promote more effective learning. In other words, e-learning applications must be based on a well-founded activity-based educational process.

To achieve this objective, these learning processes must be modelled with the level of detail required in the development of computer applications able to execute them. For this purpose, a suitable *educational modelling language* (EML) can be used [Koper 2001]. An EML is a domain-specific language specially suited for describing instructional processes, which allows for the creation of a formalized version of the learning process that is usually called a *unit of learning* (UoL). A good example of EML is IMS Learning Design (IMS LD) [IMS 2003, Koper and Olivier 2004]. For a comparative study of different EMLs see also [Martínez-Ortiz et al. 2007].

There are several factors that must be addressed regarding the applicability of the EMLs:

- On one hand, the instructional models should be provided by educators. Although EMLs are domain-specific, and therefore they integrate concepts close to the domain of expertise of these educational experts, the additional objective of making these languages directly executable by computers hinders their usability. For instance, to have educators who directly represent their educational designs in the XML encoding of IMS LD is not a realistic assumption, even with the help of XML editing tools (e.g. XML Spy). For this purpose, authoring tools must be provided.
- On the other hand EMLs are equipped with operational semantics, and therefore they can be interpreted by execution platforms in order to automatically produce e-learning applications. Moreover, there is no universal EML that can be applied to all learning scenarios. In fact, there are different types of EMLs that can be catalogued in terms of their complexity and objectives. In addition, while the use of a standard EML (e.g. IMS LD) makes it possible to reuse educational designs (i.e. UoLs) and tools (e.g. execution platforms and players), the EML itself can evolve into a new version, or its operational semantics, and even the language itself, can be specialized to meet the needs of particular application profiles. Finally, some learning domains can require more specialized EMLs (if there is a single lesson learned in Computer Science, it is that there is no such thing as a universal solution). Thus the authoring and the execution environments must be able to rapidly react to these changes.

In this paper we propose <e-LD>, an authoring and execution architecture equipped with the flexibility required by the effective use of EMLs in realistic learning scenarios. It is a revised and extended version of the previous work described in [Martínez-Ortiz et al. 2006].

The rest of the paper is organized as follows. In section 2 we describe our <e-LD> architecture from a conceptual point of view. In section 3 we propose a concrete implementation of <e-LD> based on standard XML technologies and BPEL4WS, a workflow management and web service orchestration language. In

section 4 we outline how this architecture is used to support a specific EML (i.e. IMS LD). Section 5 outlines the related work. Finally, in section 6 we give the conclusions and the lines of future work.

## 2 The <e-LD> Conceptual Architecture

The conceptual architecture for <e-LD> is outlined in [Figure 1]. This architecture, which is conceived to accommodate the different sources of variability identified in the introduction, distinguishes the following components:

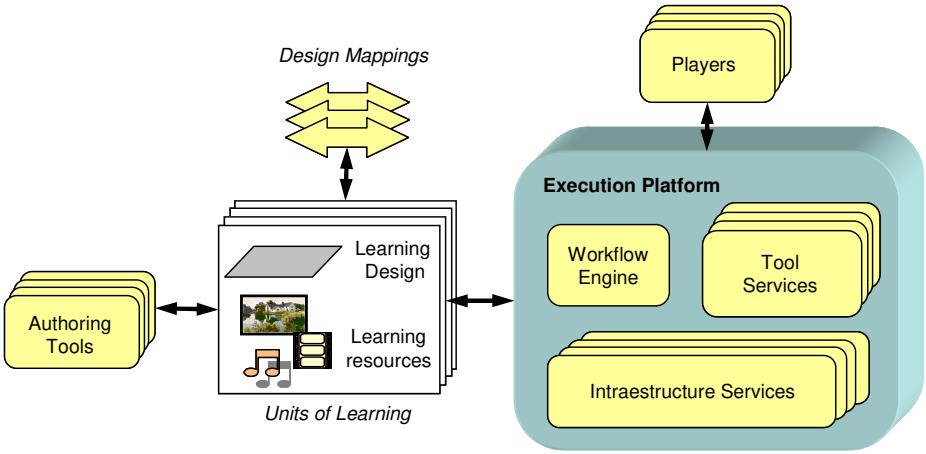


*Figure 1: The <e-LD> conceptual architecture*

- *Units of learning*. The main goal of the architecture is to produce e-learning applications based on units of leaning. A unit of learning is formed by a learning design that is a formal description of a particular educational process, and a set of learning resources, which are other information items required for the model to work (e.g. contents, services, etc.). Notice that, although the terminology used is to some extent borrowed from IMS LD, here we are using it in a conceptual and a representation-agnostic sense.
- *Authoring tools*. These tools are used by instructional designers to formalize the learning designs that are integrated in the units of learning. Besides the educational concepts, EMLs usually include technical concepts or concepts related to reusability and standardization efforts (e.g. LOM Metadata, IMS Content Packaging). However, these elements mostly distract the attention of the designer from her main task: producing good learning designs. Hence, one of the objectives of authoring tools must be to isolate learning designers from those aspects that are not directly relevant for their needs. A good example of an authoring tool is RELOAD Editor for IMS LD (web site: www.reload.ac.uk).
- *Execution Platform*. This component is intended to help during the execution of the learning process, coordinating the people involved in this process and

also providing tools and learning contents. This component plays a role analogous to engines for IMS LD such as Coppercore (web site: www.coppercore.org). As recognized by some researchers [Vantroys and Peter 2003, Paquette, Marino et al. 2005] and in the IMS LD specification itself, learning designs can be viewed as particular cases of workflows, as defined in the context of business process integration [Dumas, Aalst et al. 2005]. Therefore, <e-LD> adopts a workflow management system as the basic execution environment for the learning designs. The system provides a suitable workflow language, which is used to describe workflows that will be interpreted by a workflow engine. The chosen workflow language will be to <e-LD> as an assembling language is to a compiler of a high-level programming language. Moreover this component includes different support services (e.g. a dossier service to manage user model properties) that are internal to the execution platform, usually used for management purposes. In addition, this component includes definitions of the different tools that will be used during the running of the unit of learning. The implementation of these services can be packaged inside the execution platform or can be implemented outside (e.g. adapting an existing application to the tool service contract).

- *Players.* These components are the graphical user interfaces that are used by students and teachers to interact with a unit of learning during its execution. Therefore they have a similar role to IMS LD players such as IMS LD Reload player or SLeD (a player for Coppercore; web site: sled.open.ac.uk). Players also interact with tools and other services used, providing a unique user interface for all of these tools.
- *Design mappings*. These components are used to translate a learning design from a source representation to a target one. Design mappings are used to connect the other components together. This way, authoring tools can be designed in an EML-neutral way and later on translated to a suitable specific EML. In addition, those mappings are used to translate a UoL into a specific EML, or into a general-purpose execution-oriented format (e.g. a workflow-oriented language script). It is interesting to notice that a design mapping architecture does not require these mappings to be entirely automatic. Instead, they can comprise intervention by the user (e.g. in order to translate complex parts). This possibility also promotes a rational separation of roles in the development process (e.g. collaboration between instructors and EML experts during authoring, and collaboration between learning designers and developers during operationalization).

## 3 Implementing the <e-LD> Architecture

We are implementing the <e-LD> architecture in the context of our experimental XML-based <e-Aula> Learning Management System [Sierra, Martínez-Ortiz et al. 2005, Sierra, Moreno-Ger et al. 2007].

To facilitate the authoring of units of learning, a graphical notation for the concepts typically found in an activity-oriented EML is proposed. This graphical

notation is intended to be used not only during the authoring process but also to facilitate the reuse of already available units of learning. In particular, the UML-like notation [OMG 2005, OMG 2006] has been selected.

Regarding the operationalization of EMLs, we have chosen BPEL4WS [Andrews, Curbera et al. 2003] as the workflow language. The resulting implementation is outlined in [Figure 2]. BPEL4WS plays the role of conductor, coordinating learning activities and tools that will be used during the learning process.

In this implementation, learning designs must be represented using suitable XML-based domain-specific languages. Notice that this is not a severe constraint, since XML is currently being adopted as a standard interchange format between tools, and existing non XML-compliant authoring tools can be wrapped to produce an XML-based representation. Besides, it is also common practice for the existing EMLs to provide an XML binding, and it also facilitates the provision of EMLs tailored to specialized domains following a document-oriented approach, such as those described in [Sierra, Fernández-Valmayor et al. 2004, Sierra, Fernández-Manjón et al. 2005, Sierra, Fernández-Valmayor et al. 2006]. Finally, BPEL4WS itself also has an XML-based language.

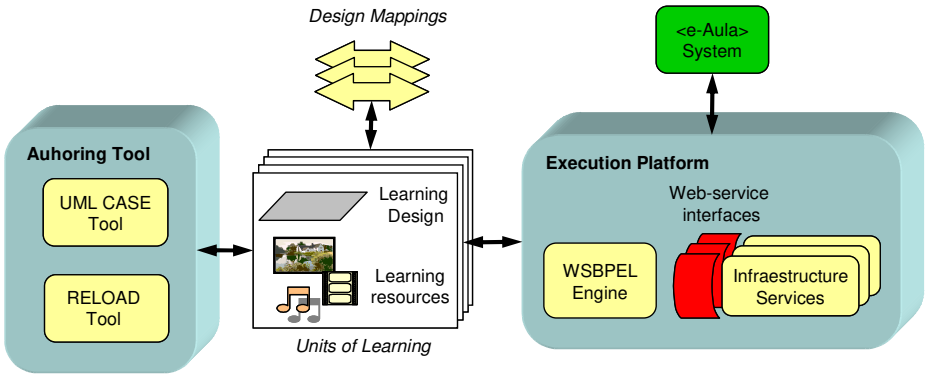


*Figure 2: <e-LD> BPEL4WS-based implementation*

As a consequence of the use of XML, the design mappings are transformations between XML-based markup languages. Notice that the design mappings can be provided using standard XML processing technologies and many of these mappings can be given using transformation languages such as XSLT. For more complex mappings, it may be possible to use standard XML processing frameworks (e.g. DOM or SAX), and even more sophisticated solutions for the incremental operationalization of XML-based domain-specific languages [Sierra, Navarro et. al. 2005]. Finally, since basic activities in BPEL4WS are represented as web services, the implementation of the basic activities in the architecture must be provided by a web service programmatic interface. Thus, the resulting applications will exhibit a service-oriented architecture.

## 4 Supporting IMS LD in <e-LD>

IMS LD has been readily integrated in <e-LD>. This section describes this integration, which has been implemented by reusing existing tools and technologies in order to decrease the overall cost as further explained in [Martínez-Ortiz, Moreno-Ger et al. 2005].

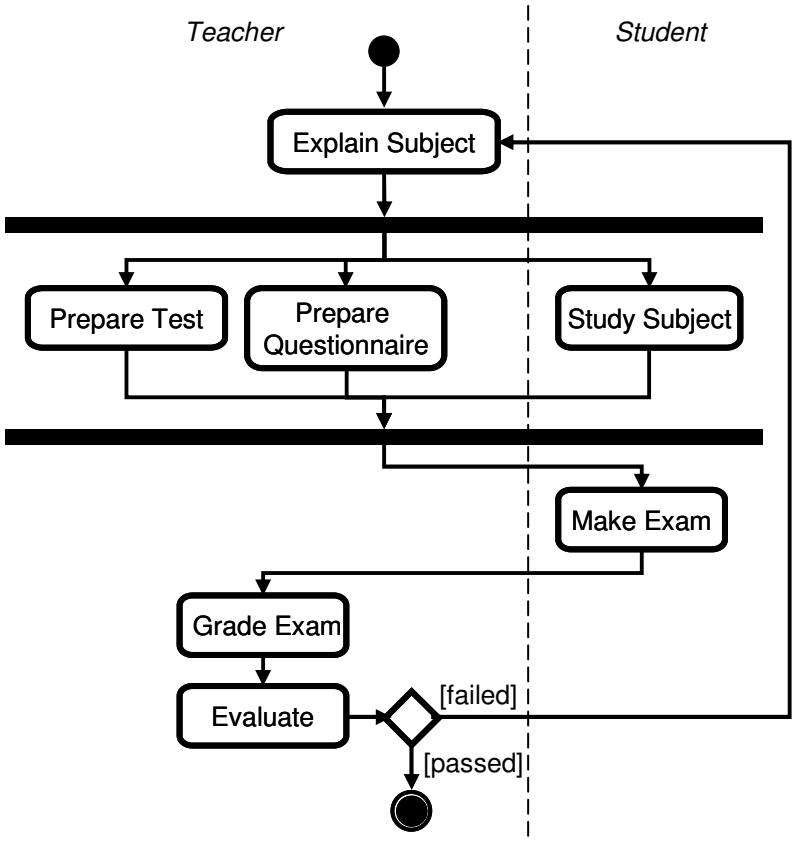


*Figure 3: UML Activity Diagram representing a simple educational scenario*

The authoring of IMS LD descriptions is mostly done using UML activity diagrams ([Figure 3]), establishing a parallelism between IMS LD activities and UML activities. Sequencing is expressed using transitions, synchronization bars and diamond shapes. Activity structures are represented using simple sequencing of the corresponding UML activities, when the activities in the structure must be executed in a sequential way (e.g. Grade Exam and Evaluate in [Figure 3]), or by using a number of transitions when they must be chosen in random order (e.g. Prepare Test and Prepare Questionary in [Figure 3]). Finally, assignments of roles to activities are represented using *swimlanes* in the activity diagrams (as Teacher and Student in

[Figure 3]). In addition to these dynamic aspects, in our integration of IMS LD the static aspects (e.g. environments, conditions, and the definition of the prerequisites/objectives and learning contents to be presented in the atomic activities) are edited using the RELOAD editor.

Regarding operationalization, the dynamic parts of IMS LD (method, activity structures) have a correspondence with the flow control structures of BPEL4WS and the static parts of the learning design are used as configuration parameters for the web services to be orchestrated by the BPEL4WS process. Some of these web services will be reusable across many different learning scenarios, while others will be specific to a concrete application. In addition, many of these web services will be fed with the appropriate learning resources during the initialization stages.

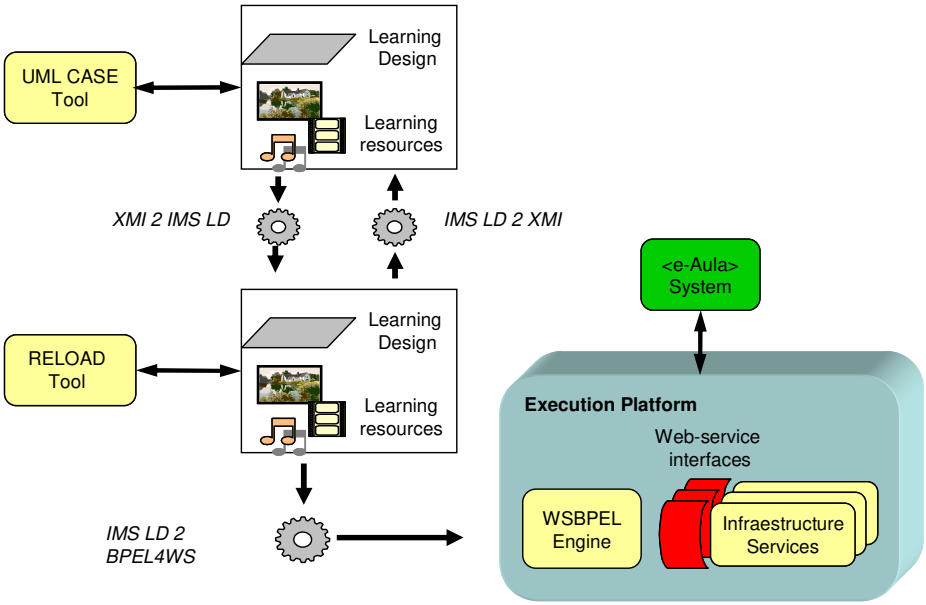


*Figure 4: Integration of IMS LD in <e-LD>*

The architectural details of the integration are depicted in [Figure 4]:

- The authoring of the activity diagrams can be performed with any tool supporting XMI (an XML-based specification which allows interchange of UML models between tools as XML documents) as interchanging format. Besides, the representation conventions described below allow for the provision of a design mapping for generating IMS LD-aware representations from XMI documents containing activity diagrams, as well as another mapping for performing the inverse step (i.e. for representing any IMS LD-aware learning design as an XMI document that can be edited with the UML editing tool). These two mappings are largely based on XSLT stylesheets, although tackling a few issues requires more conventional programming (e.g. dealing with the IMS Package Interchange Format used in packaging units of learning in this implementation).

- Operationalization is carried out using a design mapping for translating IMS LD onto BPEL4WS. This mapping, which follows the operationalization conventions described above, is again implemented using an adequate set of XSLT stylesheets with conventional programming support.

## 5 Related Work

The application of business process management techniques as operational support for e-learning solutions and the application of workflow management systems is not new. They have been applied in the creation of LMSs [Gibson 2003], also using BPEL4WS as the workflow language [Anane, Bordbar et al. 2005], or generalized workflows to support SCORM language [Kim, Yoo et al. 2005]. Another interesting work was the ASSIS project (web site: www.hull.ac.uk/esig/assis.html) in which BPEL4WS was used to orchestrate and integrate an IMS QTI engine and an IMS Simple Sequencing Engine. The main contribution of the present work with respect to these approaches is the promotion of a general architecture where educational modelling languages are conceived as the main artefacts for the production and operationalization of e-learning applications. Since these languages can be conceived as particular cases of workflow modelling languages, it is natural to adopt a workflow management system as a common execution platform.

The <e-LD> implementation is service oriented. Service Oriented Computing and Web Services are becoming standard technologies for the development of software applications. LMSs and other learning tools have been influenced by this emerging technology. Projects like the ELF Framework (web site: www.elframework.org) propose a set of the services that make up an e-learning platform. Another initiative based on web services is the OKI project (web site: www.okiproject.org) which defines an e-learning architecture by providing a set of specifications of common services and collections of programming bindings for such services. In addition to these abstract efforts, there are also commercial and open source projects that are based on service oriented computing such as the SAKAI project (web site: www.sakaiproject.org). Finally a more related work to IMS LD and services is the SLeD project, which tries to improve the functionalities of the Coppercore engine through the integration with services using service-oriented computing. In <e-LD>, however, the use of web services is a natural consequence of the choice of BPEL4WS as workflow representation language. The result is an architecture that can be tailored to each specific situation by introducing the appropriate set of supporting services.

## 6 Conclusions and Future Work

We think that the modelling of educational processes and their operational support is a key aspect in the construction of more effective e-learning applications. We conceive EMLs as domain-specific languages that can be used directly by educational experts in the modelling of educational processes. In <e-LD> these languages acquire an operational flavour, allowing educational experts to lead the development of this kind of applications. The architecture allows the integration of authoring tools with a workflow-oriented execution platform. The connection between these components is

carried out using suitable design mappings. Therefore, the architecture can support a wide variety of EMLs and authoring styles. In addition, since the execution platform can be enriched with an appropriate set of basic activities, the execution capabilities can be tailored to each specific domain. The main drawback is the effort required to set up the production environment. However, this initial investment should pay off during the production and maintenance stages and it will allow the reuse of workflow tools previously developed for the business domain.

Currently we are finishing the integration of IMS LD in <e-LD> following the approach described in this paper. The next step in the project is to carry out an evaluation of the results involving educators in several domains. We are also working on the development of more integrated authoring support for IMS LD. As future work we are planning to test other workflow management systems as execution support, as well as to integrate other, more specialized, EMLs.

**Acknowledgements**

The Spanish Committee of Science and Technology (projects TIN2004-08367-C02-02 and TIN2005-08788-C04-01) has partially supported this work, as well as the Regional Government of Madrid (grant 4155/2005) and the Complutense University of Madrid (research group 910494).

## References

[Anane, Bordbar et al. 2005] Anane, R., Bordbar, B., Deng F., Hendley, R. J.: "A Web services approach to learning path composition"; Proc. ICALT 2005, IEEE Computer Society, Kaohsiung, Taiwan (2005), 98-102.

[Andrews, Curbera et al. 2003] Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: "Business Process Execution Language for Web Services Version 1.1"; IBM, BEA Systems, Microsoft, SAP AG, Siebel Systems Specification (2003).

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

[Dumas, Aalst et al. 2005] Dumas, M., Aalst, W. M. P. v. d., Hofstede, A. H. M. t.: "Process-Aware Information Systems: Bridging People and Software through Process Technology"; Wiley-Interscience, (2005).

[Gibson 2003] Gibson, F. P.: "Supporting Learning in Evolving Dynamic Environments"; Computational & Mathematical Organization Theory, 9, 4 (2003), 305-326.

[IMS 2003] IMS: "IMS Learning Design Information Model Version 1.0 Final Specification"; IMS Global Learning Consortium Specification, (2003).

[Kim, Yoo et al. 2005] Kim, K.-H., Yoo, H.-J., Kim, H.-S.: "A process-driven e-learning content organization model"; Proc. ICIS 2005, IEEE Computer Society, Jeju, Island (2005), 328-333.

[Koper 2001] Koper, R.: "Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML"; Educational Technology Expertise Centre (OTEC) Working Paper, Open University of the Netherlands, http://hdl.handle.net/1820/36, (2001).

[Koper and Olivier 2004] Koper, R., Olivier, B.: "Representing the Learning Design of Units of Learning"; Educational Technology & Society, 7, 3 (2004), 97-111.

[Martínez-Ortiz et al. 2006] Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J.L., Fernández-Manjón, B.: "A General Architecture for the Authoring and the Operationalization of e-Learning Applications with Educational Modelling Languages"; Proc SIIE06, León, Spain (2006), 280-288.

[Martínez-Ortiz et al. 2007] Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J.L., Fernández-Manjón, B.: "Educational Modeling Languages: A Conceptual Introduction and a High-Level Classification"; In Fernández-Manjón, B, Sánchez, J.M et. al (eds.) Computers and Education: E-learning – from theory to practice. Springer (2007), 27-40.

[OMG 2005] OMG: "Unified Modelling Language version 2.0: Superstructure"; Object Management Group Specification, (2005).

[OMG 2006] OMG: "Unified Modelling Language version 2.0: Infraestructure"; Object Management Group Specification, (2005).

[Paquette, Marino et al. 2005] Paquette, G., Marino, O., De la Teja, I., Lundgren-Cayrol, K., Léonard, M., Contamines, J.: "Implementation and Deployment of the IMS Learning Design Specification"; Canadian Journal of Learning and Technology 31, 2 (2005).

[Polsani 2003] Polsani, P.: "Use and Abuse of Reusable Learning Objects"; Journal of Digital Information 3, 4: Article No. 164, (2003).

[Sierra, Fernández-Manjón et al. 2005] Sierra, J. L., Fernández-Manjón, B., Fernández-Valmayor, A., Navarro, A.: "Document-Oriented Construction of Content-Intensive Applications"; International Journal of Software Engineering and Knowledge Engineering, 15, 6 (2005), 975-993.

[Sierra, Fernández-Valmayor et al. 2004] Sierra, J.L., Fernández-Valmayor, A., Fernández-Manjón, B., Navarro, A.: "ADDS: A Document-Oriented Approach for Application Development"; Journal of Universal Computer Science, 10, 9 (2004), 1302-1324.

[Sierra, Moreno-Ger et al. 2007] Sierra, J.L., Moreno-Ger, P., Martínez-Ortiz, I., Fernández-Manjón, B.: "A Highly Modular and Extensible Architecture for an Integrated IMS based Authoring System: The <e-Aula> Experience"; Software-Practice & Experience 37, 4 (2007), 441-461.

[Sierra, Fernández-Valmayor et al. 2006] Sierra-Rodriguez, J. L., Fernández-Valmayor, A., Fernández-Manjón, B.: "A Document Oriented Paradigm for the Construction of Content-Intensive Applications"; The Computer Journal, 49, 5 (2006), 562-584.

[Sierra, Martínez-Ortiz et al. 2005] Sierra-Rodriguez, J. L., Martínez-Ortiz, I., Moreno-Ger, P., Lopez-Moratalla, J., Fernández-Manjón, B.: "Building Learning Management Systems Using IMS Standards: Architecture of a Manifest Driven Approach"; Lecture Notes in Computer Science 3583 (2005), 144-156.

[Sierra, Navarro et. al. 2005] Sierra-Rodriguez, J. L., Navarro, A., Fernández-Valmayor, A., Fernández-Manjón, B.: "Incremental Definition and Operationalization of Domain-Specific Markup Languages in ADDS"; ACM SIGPLAN Notices 40, 12 (2005), 28-37.

[Vantroys and Peter 2003] Vantroys, T., Peter, Y.: "COW, a Flexible Platform for Enactment of Learning Scenarios"; Proc. CRIWG 2003, Springer Verlag, Autrans, France (2003), 168-182.