Berlanga, A. & Garcia, F. (2005).
IMS LD reusable elements for adaptive learning designs.
Journal of Interactive Media in Education 2005(11).
[jime.open.ac.uk/2005/11].

17 August 2005

ISSN: 1365-893X

IMS LD reusable elements for adaptive learning designs

Adriana J. Berlanga, Francisco J. García

Abstract:

This paper presents an approach to designing adaptive learning environments based on IMS LD, which separates its elements (i.e. objectives, prerequisites, method, learning activities, adaptive rules, personalization properties, etc.) in order to use them in different Learning Designs and enforce their reusability and exchangeability. Moreover, it briefly presents an authoring tool under development to define adaptive learning designs compliant with IMS LD.

Keywords:

Adaptive Learning Design, IMS LD, Authoring tools for IMS LD

Commentaries:

All JIME articles are published with links to a commentaries area, which includes part of the article's original review debate. Readers are invited to make use of this resource, and to add their own commentaries. The authors, reviewers, and anyone else who has 'subscribed' to this article via the website will receive e-mail copies of your postings.

1 Introduction

In Chapter 12 Towle and Halm (2005) explain the modelling of Adaptive Learning using IMS LD (2003) by inserting the adaptive logic within the IMS LD element `<method>`. They exemplify how three kinds of adaptive strategies can be modelled using IMS LD. These strategies include synchronous vs. asynchronous interactions, rule-example vs. example-rule presentation of the content, and feedback adaptation.

Subsequently, the authors point out that one of the limitations of IMS LD for adaptive learning is its "manifest-centred" schema. That is to say, all the necessary information for interacting with a Unit of Learning (UoL) is inside the manifest of the UoL. For them, the problems of this representation are (p. 225):

(1) The difficulties inherent with rule interactions for multiple characteristics.

(2) Once delivered, manifests cannot be changed to take advantage of new adaptive strategies.

(3) The same strategy is encoded in multiple manifests, causing redundancy in authoring and storage.

(4) The knowledge about learning objects is often embedded in the manifest, and not accessible through metadata for use in new or arbitrary strategies.

The authors argue that a solution to tackle these problems is to move from "manifest-centred" schema, which forces static adaptivity, to a "server-centred" approach. This can be done by removing the adaptive logic from the manifest and using a LD player as a client (or agent) that communicates to the server what the learner has done. The server, then, will send back to the client the ID of the most appropriate next activity to follow.

However, problems 3 and 4 are not because of the specification, but the way the specification is used. If repositories are not used, or the creation of learning activities or methods has to be done for each Learning Design, then redundancy, inefficiency and lack of reusability are a fact.

In this paper we present a proposal we are developing to tackle these two issues. We claim that if IMS LD elements (i.e. learning objectives, prerequisites, learning activities, acts, plays, conditions, and so on) are defined as independent elements, they become reusable and exchangeable elements. In this contribution we present this approach, and outline related work.

2 Adaptive Learning Design (ALD)

We are investigating if a learning design with adaptive characteristics, or ALD, can be reusable and exchangeable among different courses, contexts, and applications.

An ALD is a UoL that contains personalized behaviour in order to provide each student with a learning flow adequate to her/him characteristics (Berlanga and García, 2005). In order to permit reutilization, ALDs are semantically structured and designed according to IMS LD. That is to say, ALDs elements are the same as IMS LD elements (with the exception of learning objects that are compliant with IMS LOM, 2001). However, elements are defined and stored as separate components that can be reused an exchanged among different ALDs, learning contexts, lessons, and courses.

The separation between learning activities and their learning resources is a key premise of IMS LD. A learning design can be repopulated with different contents and resources to use it in a new learning context (Richards, 2005), and/or a set of learning activities can be packed in different courses (McAndrew and Weller, 2005). Likewise, there are three kinds of reusability of an ALD:

- ALD as a template, where an "empty" ALD is provided in order to fill-in the desired elements (e.g. learning resources, properties, learning activities, conditions, etc.).
- Reusable ALDs, where an ALD is modified in order to suit new settings or contexts.
- Reusable elements of ALDs, where specific components of an ALD are exchanged among other ALDs.

2.1 The Lego metaphor and its elements

Since we claim that the separation of elements is crucial in order to support their reusability and exchangeability, the definition of an ALD follows the *Lego* metaphor. Figure 1 represents this approach. Notice that each type of element (e.g. rules, methods, plays, etc.) should be stored in different file folders that could be handled as repositories of IMS LD elements. For readability reasons, not all relationships among elements are presented; see Table 1 for a full list, including the ID of the element, its name, the elements in which it can be included, the elements it can include, and the elements where a learning object can be attached.

For instance, the learning object *LO-1* can be attached to learning activity *LA-1* (using the `<activity-description>` element). Then, *LA-1* can be included into activity structure *AS-1*, which could be incorporated in *ACT-1*, and so on. In the same way, *AS-1* could be included in *ACT*-2. In this manner different components can be reused and

exchanged among different applications and tools compliant with IMS LD, and the definition of a new method of instruction does not imply the creation of learning activities, roles, objectives, etc., that have been created before for other ALDs.

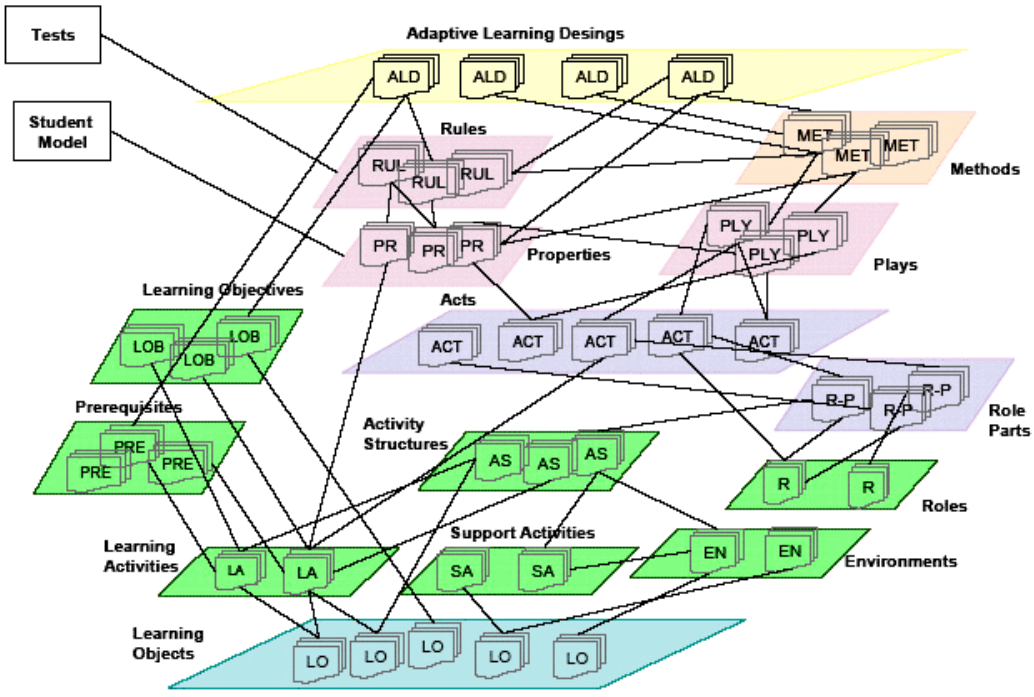

Figure 1: ALD *Lego* metaphor

ID	Name	It can be included in	It can include	Learning Object (LO) in IMS LD element(s)
LO	Learning Object	LO, OBJ, PRE, LA, SA, AS, EN, R, ACT, PLY, MET	LO	<item>
OBJ	Learning objective	LA, ALD	LO	<item>
PRE	Prerequisite	LA, ALD	LO	<item>
LA	Learning activity	AS, RP, RUL	LO, OBJ, PRE, EN, PP	<activity-description> <feedback-description>
SA	Support activity	AS, R, RP, RUL	LO, RP, EN, PP	<activity-description> <feedback-description>
AS	Activity structure	RP, AS, RUL	LO, LA, EN, SA, AS, ALD	<information>
EN	Environment	EN, LA, SA, AS, RP, RUL	LO, EN	<learning-object>
R	Role	RP, ACT	LO, SA	<information>
RP	Role-Part	ACT, SA	R, LA, SA, EN, AS, PP, ALD	
ACT	Act	PLY	LO, RP, PP, R, ALD	<feedback-description>
PLY	Play	MET, RUL	LO, ACT, PP	<feedback-description>
PP	Properties	RUL, LA, SA, RP, ACT, PLY, MET, ALD		
RUL	Rules (or conditions)	MET, ALD	PP, EN, LA, SA, AS, PLY, MET, ALD	
MET	Method	RUL, ALD	LO, PLY, RUL, PP	<feedback-description>
ALD	Adaptive Learning Design	RP, ACT, AS, RUL	OBJ, PRE, PP, RUL, MET	

Table 1: ALD Elements

Observe that in Figure 1 properties are connected to a student model (i.e. a repository that contains information about students as knowledge, preferences, etc., which is the base for performing adaptivity) in order to manage, update and retrieve users' information. Similarly, adaptive rules are connected to a test repository that contains assessments or

forms to evaluate properties. Ideally, the student model and the test repository should be compliant with specifications for learners' information and tests, such as IMS LIP (2003) and IMS QTI (2002), respectively. However, at present, this is out of the scope of an ALD definition.

It is important to point out that this *Lego* metaphor does not mean that any element is combinable with any other element, or that they can be assembled in any way –an approach that has been criticized by Wiley (2002). Quite the contrary, the combination of elements follows the information model of IMS LD.

Figure 2 depicts an ALD showing how its elements are assembled to build-up the UoL "Hypermedia Introduction", in the same manner used by Towle *et al.* (2005). It includes a personalization property (P-Initial-Knowledge) to store the student previous knowledge about the subject. This property is integrated in a condition that takes into account the value of this property to display a background learning activity (LA1) or an introduction about hypermedia (LA2).

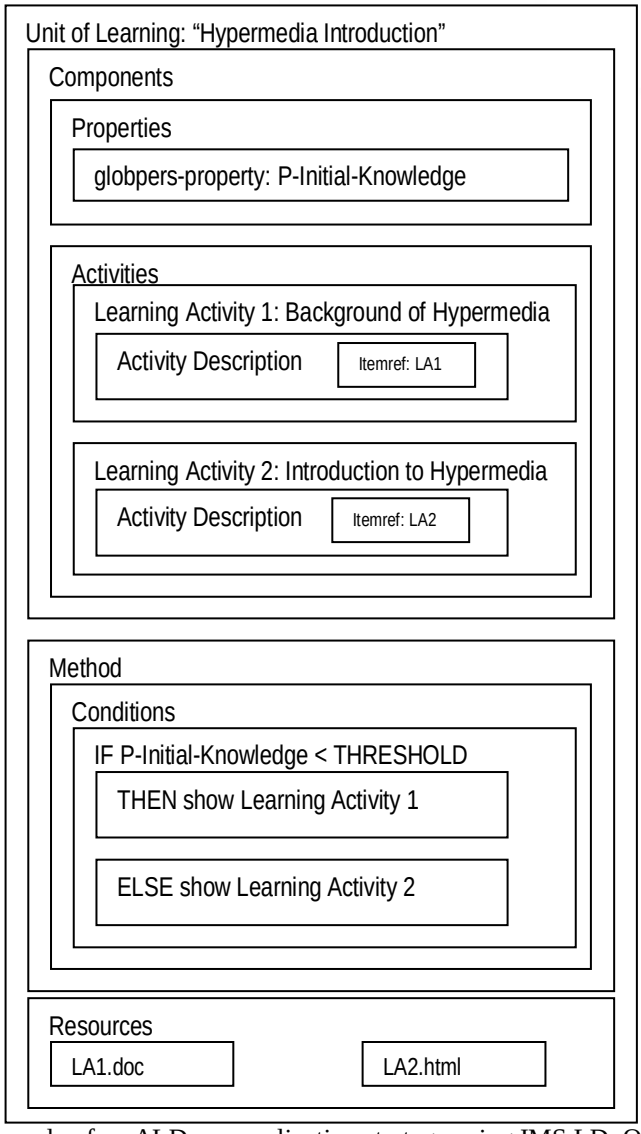

Figure 2: Example of an ALD personalization strategy using IMS LD. Only the parts of the ALD relevant to the adaptive strategy are shown

Reusability of these elements can take place if, for instance, learning activities LA1.doc and LA2.html are removed and authors repopulate them with the learning activities or resources

they want (i.e. ALD as a template). Moreover, the components of this ALD can be reused in other ALDs if, for example, the learning activity L1 (Background of Hypermedia) is included in a different ALD as an activity about the evolution of the World Wide Web, or the condition "IF P-Initial-Knowledge < THRESHOLD" is included in other ALD. Finally, this ALD can be modified for other settings if, for instance, a property that contains the final knowledge of the student (e.g. "P-Final-Knowledge") is included, and then used in the conditions section to show complementary learning activities if the "P-Final-Knowledge" value is less than the threshold value.

2.2 Definition of ALDs

In order to define ALDs we are developing an authoring tool. Our objective is to support authors in the creation of learning designs without prescribing any instructional approach, variables or conditions for adjusting learning to students' characteristics.

As a result, we are extending the functionality of a tool for creating hypermedia books called HyCo (Hypermedia Composer) (García and García, 2005) with the intention to use it as the ALD authoring tool editor.

HyCo is a multiplatform tool that supports the creation of learning materials. It has sets of galleries that permit authors to manage multimedia resources and bibliographical references, as well as to generate output files in formats such as HTML, PDF, XML or plain text. The current version of HyCo includes a LOM editor compliant with IMS LOM for defining and modifying the metadata of educational resources. HyCo stores these resources in a repository, in such way that, later on, they can be incorporated into elements of ALDs as prerequisites, objectives, components (i.e. learning activities, activity structures), and so on.

The ALD Editor follows the *Lego* metaphor explained before. Therefore, each element is defined independently from each other. Within the definition of each element, the interface presents a tab structure to group sets of attributes that might be described to annotate the element. Authors can attach to this definition resources created in HyCo (e.g. a chapter or a hypermedia book), or resources referenced by an URL. Figure 3 shows the HyCo-ALD Editor tab to define the description of a learning activity.

Figure 3: HyCo-ALD Editor. Definition of learning activities (description tab)

The editor presents, when possible, default values and combo-boxes. For instance, Figure 4 shows the tab for including learning objectives into learning activities. It provides a selection list that contains possible learning objectives (i.e. those that have been defined before and are in the learning objective repository). Moreover, it is connected to the HyCo-LOM Editor in order to provide authors with a tool for creating metadata.

Similar interfaces are provided for depicting other elements such as learning objectives, prerequisites, roles, learning activities, and so on.

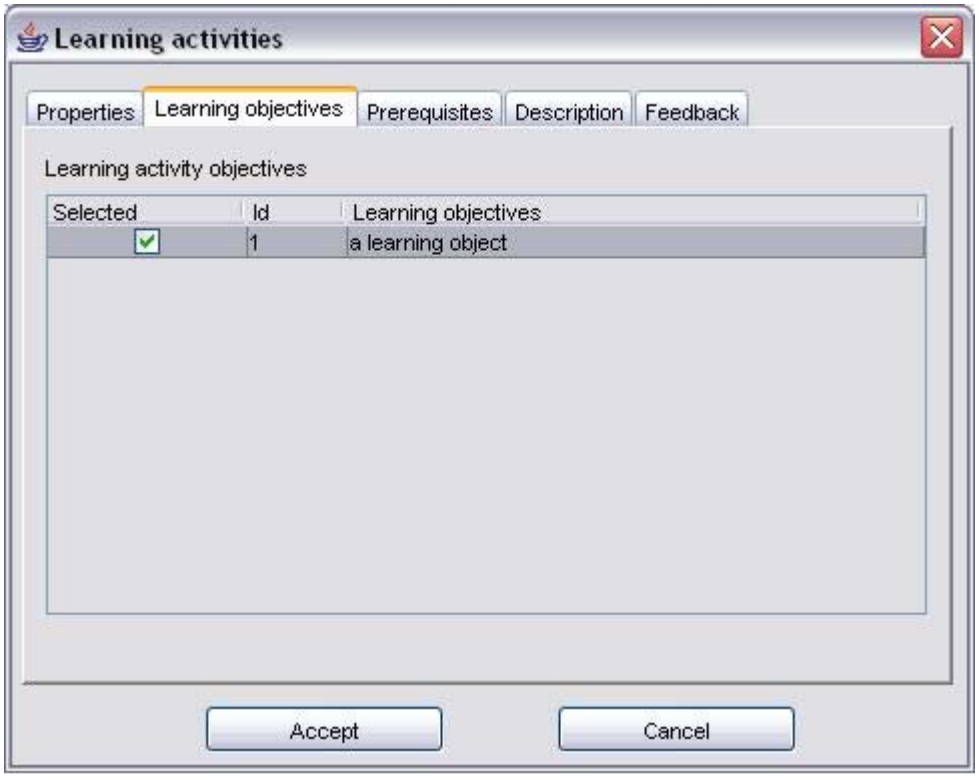

Figure 4: HyCo-ALD Editor. Definition of learning activities (learning objectives tab)

Currently, we are working on analyzing and designing the variables needed to define personalization properties and adaptive rules (i.e. IMS LD Level B). Personalization properties contain information about the users that can be included subsequently into adaptive rules, while adaptive rules are prescriptions defined by authors that will be taken into account to adjust the learning design, and that can be included into learning methods. For instance, returning to the example presented before, Figure 2 includes a personalization property named "P-Initial-Knowledge", and an adaptive rule named "IF P-Initial-Knowledge <THRESHOLD".

We are developing two approaches for defining these elements: one for novice users of IMS LD and other for expert users of the specification. In the former case, we are designing a wizard for defining adaptive techniques, and in the latter, we are developing an expression-builder-tool, based on an "if-condition-then-action" formalism (Berlanga and García, 2004), that permits authors to include learning design elements, personalization properties, and

logical and relational operators. In both approaches, authors will be able to save their adaptive rules and properties in repositories and reuse them in other ALDs.

3 Related work

Nowadays, IMS LD tools are in their early stages of development and testing. Reasons for this include the relative novelty of the specification and its high level of complexity. Until now, there are not available user-friendly authoring tools for teachers or non-specialists in the development of learning materials; existing authoring tools have not gone beyond research. Nonetheless, the high number of efforts attempting to develop authoring tools is significant [1].

At present, authoring tools for learning designs compliant with IMS LD can be categorized as:

- Basic editors, also known as close from specification: authoring tools which interface follows the specification to help users to create a UoL. Therefore, users should have enough knowledge of the specification. Editors that have been developed in this line include those with an interface that uses the tree metaphor for displaying and handling the specification, as CopperAuthor (2005), which uses the Coppercore (2005) engine to display a preview panel of the UoL, and the popular Reload Learning Design Editor (2005). Other basic editors include ASK-LDT (Karampiperis and Sampson, 2005), which has a graphical interface, and the aLFanet LD Editor (Van Rosmalen and Boticario, 2005) that, as part of the aLFanet learning management system, guides the designing process using windows for defining each element of the learning design.

- Advanced editors, also known as distant from specification: authoring tools that "hide" the specification to the final user. Editors developed in this line include MOT+ (Paquette, De la Teja, Léonard, Lundgren-Cayrol, and Marino, 2005), which has a graphical interface for creating courses according to the MISA method, and e-Live LD Suite (eLive GmbH, 2005), which is a commercial product under development that provides users with learning design templates for working with the specification.

Table 2 presents an overview of some IMS LD authoring tools that are leading the IMS LD implementations (Griffiths, Blat, Elferink, and Zondergeld, 2005). The table shows their type (basic/advanced editor), IMS LD level of compliance, characteristics of the interface, availability status, and authors.

Name	Type	IMS LD Level	Characteristics of the interface	Status	Authors
Reload LD Editor [2]	Basic	A B C	Tab-structured for separating IMS LD elements (properties, roles, environments, method, activities, etc.). Within each tab a tree metaphor is used for grouping elements.	Available / Open Source	JISC project (GB): University of Bolton, University of Strathclyde
aLFanet LD Editor [3]	Basic	A B C	Web-based interface. Uses windows to separate the elements of IMS LD. Wraps IMS LD concepts in substructures.	Available / Open Source	EU project: aLFanet OUNL (NL), UNED (ES), Software AG (ES)
CopperAuthor [4]	Basic	A	Tree based editor. Includes different views of the UoL (design, XML, play, manifest, etc.). Integrates the Coppercore engine in order to preview the UoL.	Available / Open Source	Open University of the Netherlands (NL)
ASK-LDT	Basic	A B	Graphical editor. Drag and Drop feature for connecting IMS LD elements. Authors can use either a standard notation or a self-customized notation to describe learning scenarios.	Underdevelopment	EU project ICLASS: Informatics and Telematics Institute. (CERTH) (GR)
MOT+ [5]	Advanced	A B	Graphical editor. Uses a generic model and specific symbols for each element following the MISA method.	Available / Proprietary (soon to be open sourced Griffiths *et al.*, 2005)	Centre de recherché LICEF. Télé-Université (CA)

LD Suite	Advanced	A B	Graphical editor. Contains patterns and micro methods. Similar approach to Business Process Modelling Tools for mapping methodical concepts and structures.	Underdevelopment / Proprietary	eLive GmbH (DE)
HyCo-ALD	Basic	A	Follows a Lego metaphor. Uses separate windows for defining each element of IMS LD. Elements defined previously can be included in new learning designs.	Underdevelopment / Proprietary	University of Salamanca (ES)

Table 2: IMS LD authoring tools

As shown in Table 2, the HyCo-ALD editor falls in the basic editors category. Like the aLFanet LD editor, it uses windows to present the specification, but in a stand alone mode. Moreover, HyCo-ALD is part of an authoring tool for creating hypermedia contents; this permits the inclusion of hypermedia learning resources created in HyCo (e.g. chapters, subchapters, etc.) into learning designs, and supports metadata for learning resources conform to the IMS LOM specification.

Nevertheless, the novelty of HyCo-ALD is its approach for reusing IMS LD components and adaptive techniques in different learning designs. Furthermore, learning activities created in HyCo-LD might be exportable components that will work across different learning systems, and vice versa, HyCo could import and take advantage of learning activities compliant with IMS LD. However, HyCo-LD is still under development and much work has to be done to test if reusability of IMS LD components is possible and to what extent.

4 Conclusions and further work

The separation of IMS LD elements in different repositories is an option to avoid the creation and annotation of the same elements (e.g. learning activities, activity sequences, etc.) for different ALDs.

One step further is that those repositories could be distributed in different servers, and UoLs could include URL references anchoring to adaptive conditions or, as Paquette, Marino, De la Teja, Léonard, and Lundgren-Cayrol (2005) suggest, take out Level B and Level C – and limiting IMS LD to Level A – in such way that adaptivity conditions can be stored outside the host system. These could be solutions to avoid static adaptation forced by IMS LD and complement the server-centred schema suggested by Towle and Halm in this chapter.

This paper has presented a proposal for defining ALDs using IMS LD. We are finishing

the HyCo-ALD editor and depicting a wizard that will guide non-expert users in the creation of adaptive techniques, as well as an expression-builder tool for supporting authors in the definition of adaptive rules. Subsequently, we will test if ALDs reusability is achievable in HyCo, and then examine their possibilities for exchange among systems or applications compliant with IMS LD.

Acknowledgements: We would like to thank Jorge Carabias for his contribution in the development of the HyCo-ALD editor. Adriana J. Berlanga thanks the support of the Mexican Council of Science and Technology (CONACyT).

5 References

Berlanga, A., and García, F. (2004). A Proposal to Define Adaptive Learning Designs. *Proceedings of Workshop on Applications of Semantic Web Technologies for Educational Adaptive Hypermedia (SW-EL 2004) in AH 2004*, 23 August, 2004, Eindhoven, The Netherlands. TUE Computer Science-Reports 04-19 AH2004: Workshop Proceedings Part II.

Berlanga, A., and García, F. (2005). Learning technology specifications: semantic objects for adaptive learning environments. *Int. J. Learning Technology*, 1, (4), 458–472.

CopperAuthor. (2005). SourceForge site. Available online at: http://sourceforge.net/projects/copperauthor

Coppercore. (2005). SourceForge site. Available online at: http://www.coppercore.org

eLive GmbH. (2005). e-Live LD Suite. Accessed online at: http://www.elive-ld.com

García, F. J., and García, J. (2005). Educational Hypermedia Resources Facilitator. *Computers & Education*, 3, 301-325.

Griffiths, D., Blat, J., Elferink, R., and Zondergeld, S. (2005). Open Source and IMS Learning Design: Building the Infrastructure for eLearning. *Proceedings of the First International Conference on Open Source Systems (OSS 2005)*, 11-15 July, 2005, Genova, Italy. Accessed online on July 2005 at: http://oss2005.case.unibz.it/Papers/OES/ES2.pdf

IMS LD. (2003). Learning Design specification v1. Available at: http://www.imsglobal.org/learningdesign

IMS LIP. (2003). Learner Information Package specification v1. Available online at: http://www.imsglobal.org/profiles

IMS LOM. (2001). Learning Resource Metadata specification v1.1.2. Available online at: http://www.imsglobal.org/metadata

IMS QTI. (2002). Questions and Test Interoperability Specification. v1.2.1. Available online at: http://www.imsglobal.org/question

Karampiperis, P., and Sampson, D. (2005). Designing Learning Services for Open Learning Systems Utilizing IMS Learning Design. *Proceedings of the 4th IASTED International Conference on Web-Based Education (WBE 2005)*, 21-23 February, 2005, Grindelwald, Switzerland. ACTA Press, 279-283.

McAndrew, P., and Weller, M. (2005). Applying Learning Design to Supported Open Learning. In R. Koper and C. Tattersall (Eds.), *Learning Design. A Handbook on Modelling and Delivering Networked Education and Training*. The Netherlands: Springer, 281-291.

Paquette, G., De la Teja, I., Léonard, M., Lundgren-Cayrol, K., and Marino, O. (2005). An Instructional Engineering Method and Tool for the Design of Units of Learning. In R. Koper and C. Tattersall (Eds.), *Learning Design. A Handbook on Modelling and Delivering Networked Education and Training*. The Netherlands: Springer, 160-184.

Paquette, G., Marino, O., De la Teja, I., Léonard, M., and Lundgren-Cayrol, K. (2005). Delivery of Learning Design: the Explor@ System's Case. In R. Koper and C. Tattersall (Eds.), *Learning Design. A Handbook on Modelling and Delivering Networked Education and Training*. The Netherlands: Springer, 311-325.

Reload Learning Design Editor. (2005). Reload Project website. Available online at: http://www.reload.ac.uk

Richards, G. (2005). Designing Educational Games. In R. Koper and C. Tattersall (Eds.), *Learning Design. A Handbook on Modelling and Delivering Networked Education and Training*. The Netherlands: Springer, 227-237.

Towle, B., and Halm, M. (2005). Design Adaptive Learning Environments with Learning Design. In R. Koper and C. Tattersall (Eds.), *Learning Design. A Handbook on Modelling and Delivering Networked Education and Training*. The Netherlands: Springer, 215-226.

Van Rosmalen, P., and Boticario, J. (2005). Using Learning Design to Support Design and Runtime Adaptation. In R. Koper and C. Tattersall (Eds.), *Learning Design. A Handbook on Modelling and Delivering Networked Education and Training*. The Netherlands: Springer, 291-301.

Wiley, D. (2002). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. *The Instructional Use of Learning Objects.* Accessed online on May 2005 at: http://reusability.org/read/chapters/wiley.doc

6 Footnotes

[1] See http://www.unfold-project.net/general_resources_folder/tools/currenttools for a list of learning design tools available or under development.

[2] Available at: http://www.reload.ac.uk/ldeditor.html

[3] Available at: https://sourceforge.net/projects/alfanetat

[4] Available at: http://sourceforge.net/projects/copperauthor

[5] Available at: http://www.unfold-project.net/UNFOLD/general_resources_folder/tools/mot