**Canadian Journal of Learning and Technology**
**Volume 31(2) Winter / hiver 2005**

# Implementation and Deployment of the IMS Learning Design Specification [1]

**Gilbert Paquette**

**Olga Marino**

**Ileana De la Teja**

**Karin Lundgren-Cayrol**

**Michel Lonard**

**Julien Contamines**

**Authors**

Gilbert Paquette, Director of the Research Chair on Instructional and Cognitive Engineering (CICE), LICEF-CIRTA Research Center,Tl-universit. Correspondence concerning this article can be sent to gpaquett@licef.teluq.uquebec.ca

Olga Marino, Associate director of CICE, LICEF-CIRTA Research Center, Tl-universit, omarino@licef.teluq.uquebec.ca

Ileana de la Teja, Associate researcher at CICE, LICEF-CIRTA Research Center,Tl-universit, idelatej@licef.teluq.uquebec.ca

Michel Lonard, Associate researcher at CICE, LICEF-CIRTA Research Center,Tl-universit, mleonard@licef.teluq.uquebec.ca

Karin Lundgren-Cayrol, Associate researcher at CICE, LICEF-CIRTA Research Center,Tl-universit, klundgrej@licef.teluq.uquebec.ca

Julien Contamines, PhD Student in Cognitive Computing at the LICEF research Center.

## Abstract

**Abstract:** Knowledge management in organizations, the learning objects paradigm, the advent of a new web generation, and the “Semantic Web” are major actual trends that reveal a potential for a renewed distance learning pedagogy. First and foremost is the use of educational modelling languages and instructional engineering methods to help decide how to aggregate learning objects in learning and knowledge management environments. This article proposes a set of tools under implementation, such as a graphic Learning Design Editor and a delivery system, using learning object repositories to create IMS-LD online environments. We also propose a strategy for the deployment of learning design tools and methods in learning organizations.

**Résumé:** La gestion du savoir dans les organisations, le paradigme des objets d'apprentissage, la venue d'une nouvelle génération Web et le « Web sémantique » sont des tendances importantes qui indiquent qu'il est possible d'améliorer la pédagogie en matière d'apprentissage à distance. D'abord et avant tout, l'utilisation des langages de modélisation éducatifs et des méthodes d'ingénierie pédagogique aideront à décider de la façon dont on doit regrouper les objets d'apprentissage dans les environnements de gestion des connaissances et de l'apprentissage. L'article propose une série d'outils présentement mis en oeuvre, tel un éditeur graphique de conception de scénarios d'apprentissage ainsi qu'un système d'exécution qui se sert de répertoires d'objets d'apprentissage pour créer des environnements en ligne de IMS-LD. On propose aussi une stratégie pour la mise en place d'outils et de modes de conception de l'apprentissage dans les organisations apprenantes.

## ***Introduction***

The deployment of new technology and methodology is probably the most underestimated phase of the general innovation process. However, this part is crucial if Research and development (R&D) results are to enable users to employ innovative products and services to increase the scope and quality of their activities. In the field of Learning Design Technologies, the products and services are central to Knowledge Management in companies and organisations and to the development of a knowledge economy and society, and their deployment should be a priority for any individual, group, organization or government.

The work on Educational Modelling Languages (Koper, 2001), and the subsequent integration into the IMS Learning Design specification (IMS-LD), is the most important initiative to date, in integrating instructional design issues into the eLearning standards' movement. In particular, it describes a formal way to represent the structure of a Unit of Learning and the concept of a pedagogical method specifying roles and activities that learners and support staff can play using learning objects and services in the environment.

An implementation process for IMS-LD includes authoring the LD document (for instance, a course), publishing the document into an LD instance or run (a session of the course) and delivering this run (learners and tutors performing in the course). A deployment process extends the implementation by dealing with issues such as methodological and technical support, formal and informal training, and facilitation of organisational change.

This article is divided into three sections. The first, *IMS-LD State of the Art,* provides an overview of work done in projects addressing the IMS-LD implementation issue. Although the specification is very recent, it has propelled extensive research and development in universities and research centres around the world and is starting to get publishers and other companies interested in adopting it. This section puts forward efforts undertaken in Canadian R&D centres, such as LICEF-CIRTA, and shows its place in the international scene.

The second section, *Implementation of a Suite of IMS-LD tools,* addresses the question concerning the lack of tools and presents our work on software architecture, design and development of a graphical LD editor and the adaptation of Explor@-2 LCMS to provide a delivery platform for IMS-LD.

The final section, *Towards the Deployment of IMS-LD in Canada,* provides elements of a strategy to address key issues concerning the implementation and deployment of IMS-LD, and proposes a road map for the development of a suite of tools and methods, formal and informal training activities, as well as considerations to promote organizational change through the deployment of IMS-LD.

### ***IMS-LD State of the Art***

The IMS-LD specification aims to facilitate the exchange and repurposing of units of learning regardless of the type of delivery systems. A final draft of the IMS-LD specification was approved in February 2003, and since then has been downloaded 10,444 times (Lisa Mattson from IMS, personal communication, March, 2004).

### ***The IMS-LD Specification***

The IMS-LD specification consists of three documents available from the IMS web site:

- *IMS Learning Design Information Model,* describing the conceptual model and data structures, as well as the behavioural model and runtime behaviour.
- *IMS Learning Design Information Binding,* providing detailed information on each of the elements in the specification’s XML binding.
- *IMS Learning Design Best Practice Guide,* describing how to implement an IMS-LD specification and providing both examples of structured learning scenario narratives and corresponding XML documents. It also provides an implementer’s guide.

The conceptual model specifies three embedded levels of implementation and compliance, each with its separate XML schema. Level A contains all core elements, roles, activities and environments (learning objects and services). Level B adds properties and conditions for user modelling and assistance to Level A elements. Level C adds notifications between actors.

IMS-LD embeds and generalizes other IMS specifications such as MD (metadata), SS (simple sequencing), CP (content packaging), RDCEO (competencies, objectives and prerequisites), QTI (questionnaires and tests), LIP (learner profile) and others.

SCORM, the Sharable Content Object Reusable Model proposed by the ADL Technical Team (2004), while sometimes seen as opposed to IMS-LD, is a single-user version of an activity structure during runtime. It is composed of a Content Aggregation Model (CAM), a Runtime Environment (RTE) definition and a Sequencing and Navigation (SN) scheme, providing a way to sequence a set of learners or system-initiated navigation events.

Although SCORM integrates some of the IMS specifications embedded in LD such as LOM, SS and CP, a SCORM delivery system cannot deliver a complete IMS-Learning Design. But conversely, an IMS-LD delivery system can deliver any SCORM compliant course or module, because LD is a more powerful scheme, expanding SCORM specifications in many ways:

- LD is based on educational modelling, a process for structuring the interactions between actors, ruling activities, and used/produced educational resources;
- LD favours instructional strategies like collaborative learning, problem solving, project-based learning, communities of practices, and multi-facilitators support as found in distance education universities;
- LD describes methods as workflow processes, that can provide alternative plays adapted to different target populations and delivery modes;
- LD integrates the description of collaboration services;
- LD integrates (at Level B and C) user modelling and notification between the different actor's environment grouping learning objects[2] and services adapted to each actor's roles.

***IMS-Learning Design Projects and Initiatives***

Internationally, there are many kinds of initiatives surrounding the IMS-LD implementation, including special-interest groups, research networks, projects focusing on or interested in IMS-LD, as well as companies beginning to implement one or several aspects of the IMS-LD specification.

Among others, the Center for Educational Technology Interoperability Standards (CETIS) and the Valkenburg Group, now grouped in the UNFOLD network are active special interest group promoting exchange and initiatives surrounding the implementation of the IMS-LD specification. AlfaNet and the Structuring Content for Online Publishing Environments (SCOPE) are two European research networks funding research teams interested in the implementation of IMS-LD. CopperCore (Open University of the Netherlands*),* Reload (CETIS), and LAMS (Macquarie E-Learning Center of Excellence*)* are examples of IMS-LD oriented project-developing tools and components to support IMS-LD.

At present, eLearning companies seem to lean towards the integration and support of the SCORM rather than the IMS-LD specification. However some companies, such as *GTK Press,*from Canada, *eLive,* a German company, and *Eduplone*, a group of companies using open-source tools (e.g., Zope, Plone), are showing interest in implementing the IMS-LD specification to develop new online learning technologies.

R&D initiatives in Canada include the *eduSource Network of Learning Object Repositories*, a pan-Canadian research project, funded by CANARIE working on software architecture and development on IMS-LD until May 2004; the *R2R* project, an eduSource spin-off project extending and enhancing the contributions and efforts in the areas of IMS-LD; and the*LORNET* project, a five-year research program led by our group and funded by NSERC and industrial partners throughout Canada, working on IMS compliant learning object repositories.

At LICEF-CIRTA, a network of researchers in Quebec, development efforts have focused on tools to facilitate authoring, storing and delivery of learning objects. Among these tools can be found an instructional design method (MISA), a knowledge modelling tool (MOT), a web-based design system (ADISA) and a learning content management system (Explor@). Briefly, MISA is a mature instructional design method produced and refined since 1992. It uses a graphic educational modelling tool (MOT and MOTplus) and is supported by a web-based instructional engineering workbench (ADISA). The Explor@-2 delivery system is intended to be a multi-actor production and delivery system, but some more efforts are needed for it to become fully IMS-LD compliant.

### ***Summarizing Progress in IMS-LD Implementation and Deployment***

Table 1 summarizes the previous discussion, and outlines the Canadian R&D contribution to international efforts on IMS-LD compliant software development/ deployment process.

Table 1. *Summary of R&D progress in the IMS-LD field*

| **Process Workflows** | **Internationally** | **Canada** |
|---|---|---|
| **Business Modelling** | *OUNL*: The Educational Modelling Language efforts are accepted as the IMS-LD specification. | *Télé-université*: Development of MISA, a mature knowledge modelling-based instructional engineering method used in many organizations. |
| **Requirements** | *IMS*: The IMS-LD documents provide a good set of indirect requirements for tools.<br>- Formal requirements are needed. | *eduSource Project*: Use cases and the IMS-LD prototyping sub-project of eduSource provide some formal requirements. |
| **Analysis and Design** | *Valkenburg Group*: A general architecture was produced for EML and extended recently.<br>- No systematic architecture process for the implementation of a suite of tools supporting IMS-LD. | *eduSource*: Use case documents, and extended documents offer a starting point to build the architecture for the implementation of a complete suite of tools supporting IMS-LD. |
| **Implementation** | *CETIS*: Development of an IMS-LD editor and packager. Most LCMS are limited to reading SCORM simple sequencing and single actor designs.<br>- An LD player not yet set up. | *Télé-université*: Development of MOT+, the first graphic editor to produce IMS-LD XML files (level A).<br>Work on an LD player, integrated on an open source LCMS has been started. |
| **Testing** | Small scale testing, with partial tools. | Small scale testing, with partial tools |
| **Deployment** | *OUNL*: Large scale deployment with previous EML version (limited to this distance university). | *eduSource*: Larger deployment now possible based on the interoperability of learning object repositories. . |
| **Change Management** | Few experiences at OUNL, concentrated in distance universities.<br>- To come. | Few experiences in the eduSource project with communities of practice in universities campus (CLOE) and within the R2R project.<br>- To come. |

From this overview, it can be inferred that the deployment of IMS-LD is integrated in a series of process workflows requiring different tools. So far, only a few IMS-LD tools have appeared internationally, and global deployment can only result from an international effort in which Canada shares its expertise. The contributions of different Canadian universities in the implementation efforts of IMS-LD tools are discussed next.

## ***Implementation of a Suite of IMS-LD Tools***

This section presents the starting points and a road map towards an IMS-LD implementation. The proposition is based on documentation and research, including IMS-LD Best Practice Guide (IMS, 2003a), empirical and theoretical studies on online course design and delivery using MISA and Explor@-2 (Paquette, 2003; Paquette, Lundgren-Cayrol, Miara, & Guérette, 2003), results from the eduSource architecture project (EduSource, 2003), the EML architecture (Kluijfhout, 2002), and work done within the Valkenburg Group (Paquette, De la Teja, Léonard, Lundgren-Cayrol, & Marino, 2005a, 2005b).

## ***A Three-Stage LD Document Lifecycle***

Which tools are to be implemented, and how, in order to effectively deploy IMS-LD? To answer this question it is important to understand the Learning Design lifecycle. Figure 1 shows the sequencing of the three stages of an LD document lifecycle, the products of each stage as well as the type of tools required (in grey).

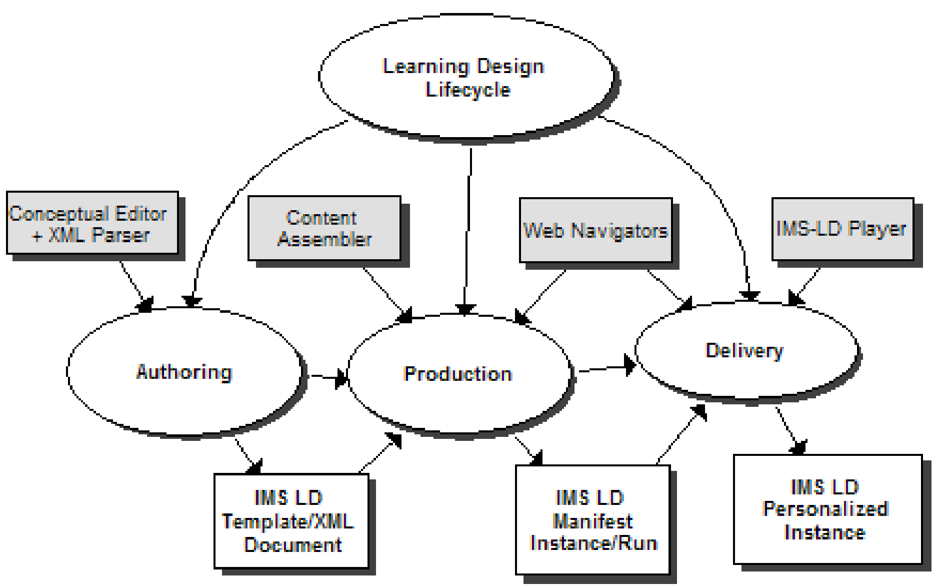


*Figure 1*. The Learning Design (LD) Life Cycle

## ***Authoring***

The product of the authoring process is an XML document which is IMS-LD compliant. This document describes a learning design in terms of its method (plays, acts, role-parts and conditions), as well as its components (roles, activities, environments and resources) and its properties (prerequisites, learning objectives, expressions, notifications and roles).

Authoring may also produce partial IMS-LD documents. It may produce a “content independent” pedagogical structure that is an XML document still compliant with IMS-LD but only with the pedagogical structure and possibly the roles, services and properties defined. The prerequisites and learning objectives do not need to be specified. We call this a content independent LD or an *LD template.* Its main interest concerns the possibility of integrating it into a repository of pedagogical methods which the user can choose from in order to create a new learning unit.

After retrieving an LD document, one may decide to build a content package in order to store and retrieve it using packaging and unpacking tools. An IMS-LD document integrated into an IMS content package is called a *Unit of learning*[3] (IMS, 2003b).

The tools needed during the *authoring stage* are a conceptual editor and an XML parser.

An abstract term used to refer to any delimited piece of education or training, such as a course, a module, a lesson, etc. A unit of learning represents more than just a collection of ordered resources to learn—it includes a variety of prescribed activities (e.g., problem solving activities, search activities, discussion activities, and peer assessment activities), assessments, services and support facilities provided by teachers, trainers and other staff members. A Unit of Learning includes a manifest, a learning design, resources, possible (sub-) manifests and physical files. An IMS content package is called a 'Unit of learning' if and only if it includes a valid IMS learning-design element in the organization part of the package's manifest.

***Production***

Both LD documents and templates may be completed or in progress. A completed LD template may be *well-formed* if its structure is validated (syntactical validation). If it also possesses *validreferences* and is consistent (semantically validated), it will be called a *valid IMS-LD document*. A unit of learning completeness may also be validated as a part of a standard content packaging validation.

A valid IMS-LD document may be used to produce a learning environment instantiation that can be used in different sessions. To do so, a process of describing the particular session is needed, establishing the community of users (teachers and students), defining the starting time, etc. In IMS-LD, this production process ensures the association between an LD document and a session or run (e.g., starting time, learning community). Such a session is called an *IMS-LD instance* or an *IMS-LD run*.

The *production stage* uses a content assembler and a web-based LD (course) navigator.

***Delivery***

An IMS-LD run is executed or delivered to the different members of the learning community (e.g., learners, teachers, tutors, experts) through a delivery platform. Each one of these actors has his/her own view of the run, which includes his/her personal properties, activity structure and trace. Each of these views is called an *IMS-LD personalized run*or*IMS-LD personalized instance*.

An IMS-LD document can be used to generate many IMS-LD runs which, in turn, are translated into as many IMS-LD personalized runs as there are users (student or a staff) associated with the course. Finally, these runs may be applied to all three levels (A, B and C) of the IMS-LD specification.

A web navigator and an IMS-LD player are used in the *delivery stage*.

***Software Architecture***

A complete architecture of the environment for implementing IMS-LD is outside the scope of this article, so this section only presents the main components that are needed in any architecture, their relationships, and the types of services they must provide to support all the actors' (machine or human) processes.

The kernel of the implementation environment is composed of four main components (Figure 2): LD Edition Environment, LD Templates and Documents Manager, LD Production Environment and LD Runtime Environment.

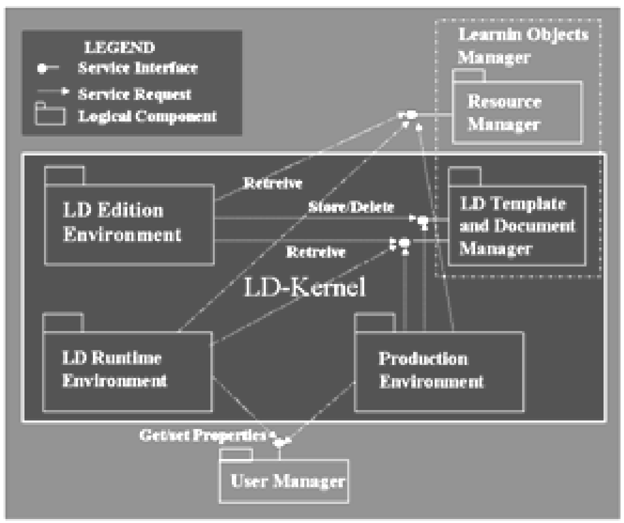


*Figure 2.* Implementation environment

The *Edition Environment* is in charge of supporting the whole authoring process. The central component of this environment is the LD Editor. The LD Editor interacts with the LD Manager to store and retrieve LD documents and templates and with the Resource Manager to aggregate objects to an LD document. Ideally the LD templates and documents should be treated as Learning Objects in which case the LD and Resources (LO and services) Management would be integrated in a single system and a content packager would take charge of the process of packaging and un-packaging the LD documents and templates.

The integrated *LD and Resource Manager* allows for storage, update, deletion and retrieval of three kinds of objects: learning materials (e.g., text and video), learning services (e.g., forum) and learning designs (LD documents, LD templates or LD runs).

The *Production Environment* is composed of a production system that interacts with the Resource Manager, LD Manager and User Manager. The Resource Manager helps instantiate the different services and tools needed for a LD run, while the User Manager provides the binding between institutional user information systems and the LD run. The LD Manager offers two services to the production system: it retrieves LD documents to be instantiated and it stores LD runs.

The *Runtime Environment* is basically composed of a *Runtime Engine* that interacts with the LD Manager as well as with the Resources Manager to recover a complete and valid LD run. The Runtime Engine creates each individual learning environment for an LD run and supports the interaction between the user and the system. While executing an LD run, user properties may change. The Runtime Engine asks the User Manager for services related to the storage and retrieval of personal and group properties. In the IMS-LD specification, activities may be synchronized by events activated when conditions including time constraints are satisfied. Time events are handled by a *timer component* that communicates with the Runtime Engine.

***From MOT+ to a Graphic LD Editor***

Let us now concentrate on the LD editor that supports the authoring stage. The architectures we have studied ( Griffiths, Blat, Garcia, & Vogten, 2005; Kluijfhout, 2002; Wilson, 2005) propose the following components for an LD Edition Environment:

- *LD Editor* : Provides services for the creation of an LD document from scratch. This means that the LD editor should allow for the creation of the LD global structure, environments, roles, objectives and prerequisites, as well as Level B and C properties, conditions and notifications. On one hand, we may have a global editor integrating all those services; on the other hand the system may be composed of several specialized editors: activity structure editor; rule, expressions and conditions editor; competency (objectives and prerequisite) editor, etc.
- *Content Packager:* Can be seen as the interface between the implementation environment and the learning objects or resource manager. It offers two services to the LD editor: the creation of an IMS-CP (content package) for an LD template or document, and the modification of an existing LD document or template content package.
- *Constraint Editor:* Its goal is to customize the edition process to satisfy organisational restrictions. Three modes are possible: 1) A pre-edition configuration service that enables the editor to customize itself to allow the edition of some constrained LD document, 2) a constraint assurance service that supervises the edition process and constrains edition actions, and 3) A post-edition constraint validation module that signals constraint violation in an LD document.
- *LD viewer/simulator:* This component might not be essential to the edition environment, but the high complexity involved in conceptualizing and designing multi-actor learning environments, calls for an LD design simulation tool. The viewer takes a valid IMS-LD document and generates a navigational simulated and controlled run in which the user can change roles, control interactions and notifications as well as manage time constraints.
- *LD validation system:* May be integrated into the simulator and/or the editor, or it may be viewed as an independent component. In any case, it provides a validation service, both for the editor and for the simulator to validate syntactical and semantic issues of an LD document.

Based on these components, in September 2004, LICEF-CIRTA Research Center released the first LD graphic editor, based on MOT+ graphical representation system. Because this system is generic and can be used for many kinds of models, such as domain ontology construction or business process modelling, it needed to be constrained in order to enable the modelling of IMS-LD compliant Units of Learning. Figure 3 displays some of the graphic symbolism used in the MOT+ graphic LD Editor.

Within MOT+, combinations of specific graphical symbols and links can be used to describe all the IMS-LD components as shown in figure 3. MOT+ exports an XML schema translated by a parser to produce an LD XML document. The parser also acts as a post-edition validation module signalling constraint violations in the LD document. A valid LD XML document can then be handled by a content packager for storage and retrieval.

The quality of the graphics environment reduces the need for a simulator. Moreover, the MOT+ system provides a way to simulate a learning unit by providing an OLE-based functionality enabling a designer to connect and display a learning object included in an environment or associated to a method component, as well as learning objectives and prerequisites. Still, the best simulation is to have a production and delivery environment run the content package produced from the LD document.

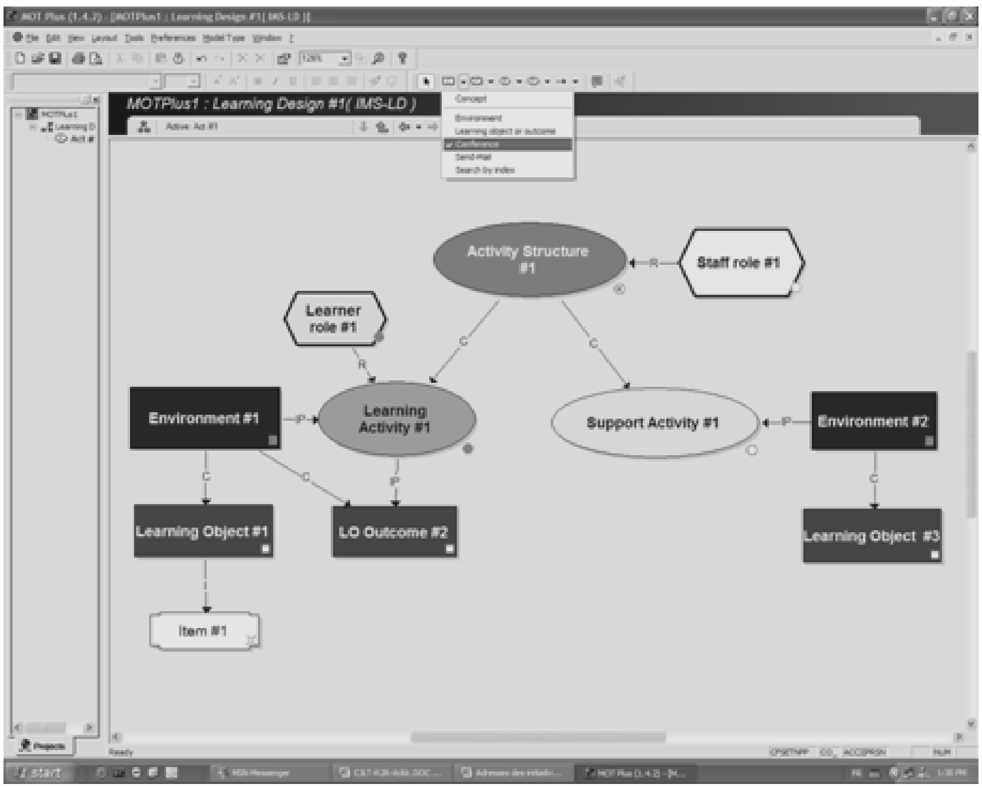


*Figure 3.* A MOT+LD model example

***From Explor@-2 to a LD Production & Delivery Environment***

The following features, already available in Explor@-2, can be used as a starting point to build an IMS-LD simulator that can manage a runtime session:

- *MOT+ LD Editor* : Can be enhanced with an importer to produce an LD activity tree structure, called Method. The tree structure serves as the basis to link tools and instructional materials stored in the Resource Manager to any level in the structure.
- Planning Editor: To insert timelines as well as evaluation and collaboration rules, which potentially can correspond to IMS-LD completion rules.
- *Advisory Editor* : To insert validation questions, contextual advice and a progression bar, that could be used to create Level B and C designs. This editor should respect the IMS QTI specification 2.0.
- *Student Assignment Editor and Manager* : To support a designer in establishing the IMS-LD for learning activity outcomes as well as for monitoring services.
- *Competency Editor* : For the designer to describe course competencies (entry and target levels), corresponding to prerequisites and learning objectives, in terms of a domain knowledge

description, skill and performance levels and then link each competency statement to the appropriate learning activity or activities. This editor produces a self-assessment tool allowing the student to estimate his/her level of understanding.

Using these features, it would be possible to build a representation of the IMS-LD Method's learning structure as well as a concrete instantiation of the activities in that structure, as displayed in figure 4.

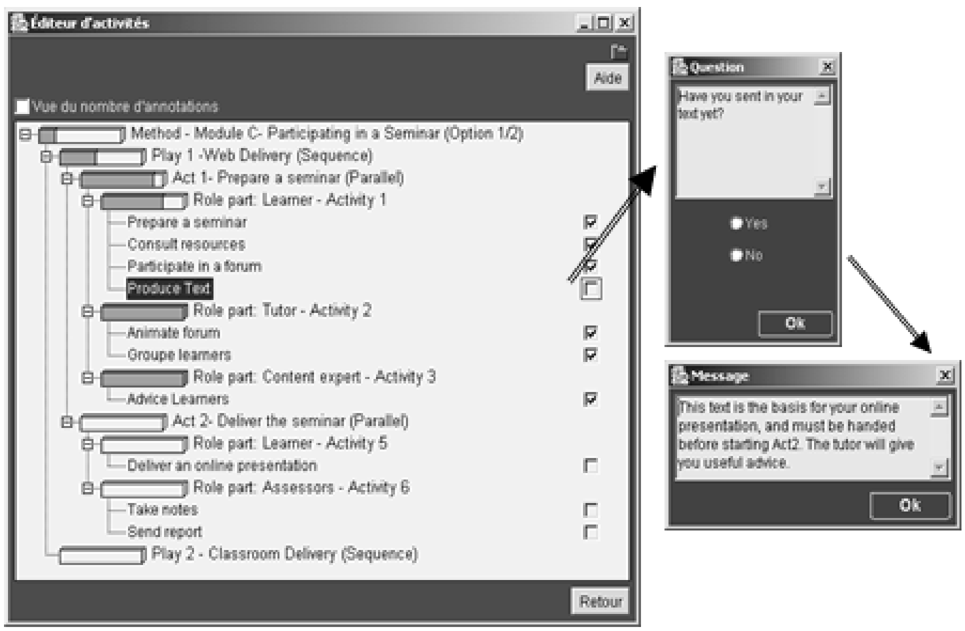

*Figure 4*. The Instructional Activity Structure during Runtime

In Figure 4, the *Method* corresponds to a learning unit called Module C, and the *Plays*present two alternative course delivery models from which a learner has to choose one: web delivery (play 1) or classroom delivery (play 2). Play 1 consists of two *Acts* in sequence. In the first Act, there are three role parts: 1) learners prepare a seminar by consulting resources, participate in a discussion forum and produce a presentation (which is actually selected); 2) tutors animate the forum and group learners; 3) experts provide advice to learners in and outside the forum. In the second Act, learners deliver an online presentation while assessors take notes to produce an evaluation report (this activity could also figure in a third Act).

Figure 4 shows that two of the three role parts in Act 1 have been completed; the learner has still to produce a text. If the learner clicks the check box of this activity, the system displays a validation question with two possible answers, each triggering advice on what to do next. Explor@-2 (Paquette, De la Teja, & Dufresne, 2000) has a built-in bottom-up propagation mechanism to assign a progression level to each node of the instructional structure calculated from its leaves, which can be used to provide feedback using completion requirements for Acts, Plays or Method as specified in IMS-LD.

In Explor@-2, each actor or role has its own activity structure (which is not multi-role) and its own resource environment, so additional functionalities will have to be built to exploit the full multi-actor capabilities of the IMS-LD specification. These include synchronization mechanisms when the completion of an Act requires verifying whether all or some other roles have also completed the Act. In the present case, this is needed to check if all the learners have completed Act 1 before Act 2 could start.

### ***Towards the Deployment of IMS-LD in Canada***

The deployment of IMS-LD in Canada is a great opportunity to increase the quality of learning, the development of skills for the new economy and, more generally, for the development of a knowledge society. Structuring learning activities and resources respecting the IMS-LD specification may support the introduction of new ways of learning and designing (Hummel, Manderveld, Tattersal, & Koper, 2004). According to the UNFOLD community of practices, this specification offers new opportunities to create eLearning activities "in which learners and teachers work together, taking on a variety of roles and working with learning resources in a wide range of ways" (UNFOLD, n.d.). Presently, the implication of this specification remains unknown since, as discussed in section one, the editors and delivery tools are just appearing and large-scale testing of integrated tools is still to be carried out.

The greatest impact of educational modelling lies in the improvement of the quality of learning, mainly because the use of the IMS-LD specification should lead to more emphasis on pedagogy and the reuse of good designs and innovative methods such as problem-based and project-based learning, as well as collaborative learning employing communities of practice strategies. Reusing good learning design would also entail a decrease in efforts and costs that would encourage educators to put more attention on effective learning design than fancy multimedia or web sites.

The main obstacle to the wider use of IMS-LD comes from the inherent complexity of educational modelling, the need for a paradigm shift in organizations, and from the lack of appropriate tools offered by eLearning companies. Ways to overcome these obstacles require the promotion of a better understanding of the role of educational modelling, a wider availability of support tools, case studies and methods, as well as the development of explicit learning unit examples and templates on educational modelling and IMS-LD implementations. Furthermore, the demand from well-informed educators could initiate change in their institutions as well as triggering innovations from the eLearning industry. The following section briefly describes key issues to be considered in a strategy to address implementation and deployment of IMS-LD.

### ***Development of a Complete IMS-LD Tool Set***

To build and deploy a set of IMS-LD tools on a large scale, we first need to define a software architecture that will provide the conceptual integration bonding agent required to building one or more web-based design portals and expanding research and development of user-friendly multi-actors delivery systems. Such architecture would:

- Define a complete set of tools, from design, to development and delivery;
- Provide alternative LD tools, from "easy to use" tools for relatively simple designs and editing, to sophisticated tools for the expert designer;
- Provide simple LD tools for the individual (e.g., the average campus professor, who is

responsible for the whole course design: from creating resources to facilitating it);
- Provide complex LD workflows for organizations to produce big-scale eLearning (like the open universities);
- Plan the set of LD tools as open source components and/or sustainable web services, supported by a stable Canadian community of developers.

One or more IMS-LD editors should support the entire instructional design process. There are many ways to provide a user-friendly tool for LD authoring in the format of a web-based form editor, a tree editor or a graphical editor. All three possibilities are interesting in that they provide a selection for each designer to choose the one that is best suited for his/her project or skills, the output being the same in all three cases: an IMS-LD XML manifest file. The graphical MOT+ editor provides a robust IMS-LD graphical editor as an interesting option.

The major point of concern lies in connecting design editors and content packagers with learning content management systems that can read IMS-LD XML files for the development and delivery of courses and learning events. Actual LMS or LCMS are sometimes SCORM compliant, but they are not ready for LD, essentially because they provide single actor environments. Analysis has shown that the Explor@-2 system built at Télé-université can be adapted to IMS-LD by importing an IMS-LD activity structure, generate role environments, display activity structures for each type of actor, and provide contextual alternate views to help an actor situate its activities within a play or an act in relation to other actors' activities.

***Repositories of Learning Design Documents and Templates***

To support reusability of good learning designs, it is essential that libraries of learning designs can be made available to learners and designers as learning objects in one or more repositories. These repositories could be added to existing repositories in the eduSource network, or built anew. They could then be searched, even from other repositories outside Canada. Conversely, learning designs developed elsewhere could become available to Canadian designers.

***Development of an IMS-LD Methodology***

Creating a good LD requires a systematic application of a process ensuring the pedagogical quality of the product. IMS-LD provides us with a wonderful opportunity to find out what improvements need to be done to existing instructional design methodologies because it is adaptable and reusable, by structuring, implementing and testing them according to an agreed set of Canadian evaluation criteria (Krauss & Ally, 2005; Nesbit, Belfer, & Leacock, 2002). This sort of test bed would be a great outcome, and probably a strong asset for the emergence of an IMS-LD methodology.

In Canada, initial methodological principles have been proposed in different projects exploring the use of learning design compliant with IMS-LD at several universities: Télé-université (project led by Gilbert Paquette), University of Alberta (project led by Kathy Campbell), University of Waterloo (project led by Kevin Harrigan) and University of Toronto (project led by Jutta Treviranus). The proposed methodological principles are in a seminal stage and need to be expanded.

***Formal and Informal Training to apply IMS-LD***

Integrating IMS-LD in instructional design practices at a national level is a complex endeavour. Let's face reality: there is an inherent reluctance—and sometimes even resistance among teachers and content experts—to fully design courses that conform to a certain method or prescription. An integration process is complicated by the fact that most of the academics involved in higher education training programs in educational technology, and education in general, are not yet informed of the possibilities of the IMS LD specification. This calls for special attention to provide IMS-LD training opportunities.

To develop IMS-LD training materials and units, the following elements need to be created: competency profiles for designers, tutors and training managers; knowledge models on IMS-LD associated to a set of units of learning and training resources; training scenarios adapted to each competency profile and training context; delivery models for blended as well as self-training; and communities of practice providing information and discussion forums.

As a result of a scientific elaboration process, the International Board of Standard for Training, Performance and Instruction (ibstpi), has published competency profiles for instructional designers, training managers and instructors that can provide a starting point to build a set of IMS-LD formal and informal training activities. Formal training could be offered by a university or a recognized organization awarding official certification. This kind of training is common within standards, specifications, and norm communities (e.g., ISO). Informal training provides a more flexible approach. Combined with a web-based training portal, it could be more easily integrated with actual design work, in the context of communities of practice.

***Support for Organisational Change***

IMS-LD is central to knowledge management in organisations, companies, public administration as well as universities, college and schools alike. Knowledge management goes well beyond information in databases, putting more emphasis on the knowledge and competencies of persons involved with the organization. Here, instructional design plays a crucial role providing methods for knowledge extraction from expert and knowledge dissemination to personnel and clients through formal or informal training. The introduction of IMS-LD in an organization should then be seen as a catalyst for knowledge management and effective use of learning units in the training context. It requires organizational change but it also helps to accelerate, promoting the evolution towards learning organizations.

Moreover, an open, flexible knowledge management support system is needed to cope with the requirements of a learning organization. Such a system has been partly developed in projects like Explor@-2, POOL and eduSource (McGreal et al., 2004).

A 5-year pan-Canadian R&D effort, the LORNET research network, has started to develop a multi-actor delivery platform facilitating knowledge and competency management, as well as linking a network of learning object repositories that will not only be used for enhancing learning, but also as a test bed for research on organizational change.

## Notes

1.This article summarizes a report on the implementation and deployment of IMS-LD. It is one of the products of the R2R project, financed by Industry Canada. It also integrates results from a software integration effort led by our team in the eduSource Canada project funded by Canarie.

2.Any resource or asset that can be used to support learning. A resource typically becomes thought of as a learning object when it is assigned learning object metadata, is discoverable through a digital repository, and can be displayed using an eLearning application.

3.An abstract term used to refer to any delimited piece of education or training, such as a course, a module, a lesson, etc. A unit of learning represents more than just a collection of ordered resources to learnit includes a variety of prescribed activities (e.g., problem solving activities, search activities, discussion activities, and peer assessment activities), assessments, services and support facilities provided by teachers, trainers and other staff members. A Unit of Learning includes a manifest, a learning design, resources, possible (sub-) manifests and physical files. An IMS content package is called a 'Unit of learning' if and only if it includes a valid IMS learning-design element in the organization part of the package's manifest.

## References


ADL Technical Team (2004). *Sharable Content Object Reference Model (SCORM)*, 2nd Edition. Retrieved March 22, 2004 from http://www. dlnet.org/ ndex.cfm?fuseaction=rcdetails&libid=648

EduSource (2003) *Edusource Software Architecture Document*, version 0.9.01, technical report, p. 57.

Griffiths , D., Blat, J., Garcia, R., & Vogten, H. (2004). Learning Design Tools. In R. Koper & C. Tattersall (Eds.). *Learning Design—A Handbook on Modelling and Delivering Networked Education and Training* (pp. 109-136). Berlin-Heidelberg: Springer Verlag.

IMS Global Learning Consortium (2003a). *IMS Learning Design Best Practice*, Version 1.0 Final Specification.

IMS Global Learning Consortium (2003b). *IMS Learning Design Information*, Version 1.0 Final Specification.

Hummel, H., Manderveld, J., Tattersall, C., & Koper, R. (2004). Educational Modelling Language and Learning Design: New Opportunities for Instructional Reusability and Personalised Learning. *International Journal in Learning Technology, 1*(1), 111-126.

Koper, R. (2001). *Modelling units of learning from a pedagogical perspective: The pedagogical meta-model behind EML*. Retrieved January 26, 2004, fromhttp://www.learningnetworks.org/downloads/ped-metamodel.pdf

Kluijfhout, E. (2002) *Work Document for the development of an open-source component-based reference architecture for EML Design-time tooling*, technical report, Valkenburg Group, p. 41.

Krauss, F. & Ally, M. (2005). A Study of the Desing and Evaluaiton of a Learning Object and Implications for Content Development. *Interdisciplinary Journal of Knowledge and Learning Objects, Volume 1*. Retrieved April 20, 2004, from http://ijklo.org/Volume1/v1p001-022Krauss.pdf

McGreal, R., Anderson, T., Babin, G., Downes, S., Friesen, N., Harrigan, K., Hatala, M., MacLeod, D., Mattson, M., Paquette, G., Richards, G., Roberts, R., & Schafer, S. (2004). EduSource: Canada's Learning Object Repository Network. *International Journal of Instructional Technology & Distance Learning.* March 2004. Retrieved April 20, 2004, fromhttp://www.itdl.org/Journal/Mar_04/article01.htm

Nesbit, J., Belfer. K., & Leacock, T. (2003). *Learning object review instrument (LORI)*. Version 1.5, E-Learning Research and Assessment Network. Retrieved January 20, 2004, from http://elera.net/eLera/Home/Articles/LORI%201.5.pdf

Paquette, G. (2003). *Instructional Engineering for Network-based Learning.* San Francisco: Wiley-Pfeiffer.

Paquette, G., Lundgren-Cayrol, K., Miara, A., & Guérette, L. (2003) The Explor@-2 Learning Object Manager. In R. McGreal (ed.), *Online education using learning objects*. London: Routledge/Falmer.

Paquette, G., De la Teja, I., Léonard, M., Lundgren-Cayrol, K., & Marino, O. (2005a). How to use an Instructional Engineering Method and a Modelling Tool. In R. Koper & C. Tattersall (Eds.). *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training* (pp.161-184). Berlin-Heidelberg: Springer Verlag.

Paquette, G., De la Teja, I., & Dufresne, A. (2000). *Explora: An Open Virtual Campus.* In Proceeding of ED Media 2000 conference, June 26 - July 1, Montreal, Canada.

Paquette,G., Marino, O., De la Teja, I., Léonard, M., & Lundgren-Cayrol, K., (2005b). Delivery of Learning Design: the Explor@ System's Case. In R. Koper & C. Tattersall (Eds.).*Learning Design—A Handbook on Modelling and Delivering Networked Education and Training* (pp. 311-326). Berlin-Heidelberg: Springer Verlag.

UNFOLD project (n.d.). Website project, Retrieved January 12, 2004, fromhttps://www.unfold-project.net:8082/UNFOLD

Wilson , S. (2005). Architectures to support authoring and content management with

Learning. In R. Koper & C. Tattersall (Eds.). *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training* (pp. 41-62). Berlin-Heidelberg: Springer Verlag.

## Organizational References

AlfaNet: http://alfanet.ia.uned.es/index

CETIS: http://www.cetis.ac.uk/

CopperCore: http://www.coppercore.org/

Eduplone: http://eduplone.net/index_html?cl=en

eduSource: http://www.edusource.ca/

eLive: http://www.elive-ld.com/content/e65/e114/index_ger.html

GTK Press: http://www.gtkpress.com/

Ibstpi: http://www.ibstpi.org/

LAMS: http://www.lamsinternational.com/

Reload: http://www.reload.ac.uk/

Structuring Content for Online Publishing Environments:http://www.hltcentral.org/

UNFOLD: http://www.unfold-project.net:8085/UNFOLD

---



**ISSN: 1499-6685**