Learning Design Update
Bill Olivier, Director CETIS, July ‘04

The IMS Learning Design 1.0 specification was released in February 2003. This briefing provides an update on developments since then.

**Aims**

The original aims of the Learning Design specification were set out in the Learning Design Scope 2.0 document:

**Scope**
This specification will provide a means for defining diverse learning approaches. It will allow any given unit of learning to describe the learning approach, processes, activities, descriptions and resources that it uses and the learning outcome that it is intended to deliver. These specifications will support the interpretation and repurposing of units of learning, as well as the process of searching for particular kinds of units of learning and those that have particular, intended learning outcomes.

The IMS Learning Design workgroup’s (LDWG) goal is to work towards establishing specifications for describing the elements and structure of a unit of learning, including:
- resources
- instructions for learning activities
- templates for structured interactions
- conceptual models (e.g. problem-based learning)
- learning goals, objectives and outcomes
- assessment tools and strategies

The specifications, which describe this framework, need to:
- describe and implement different kinds of learning approaches
- enable repeatable, effective, and efficient units of learning
- provide access to, and interchange of, units of learning between learning systems
- support multiple delivery models
- support reuse and re-purposing of units of learning or their component elements
- leverage existing specifications and standards
- be culturally inclusive and accessible (internationalization)
- support reporting and performance analysis

The goal is to enable many kinds of educational designs to be created, using a consistent notation, which can be implemented uniformly in multiple courses or learning programmes.

This had a number of implications. One important question is the relationship between Learning Design and other IMS specifications. Up to that point specifications such as Content Packaging, QTI and SCORM were content focused and assumed a model of a single learner engaging with this content.

With respect to these specifications, the final Learning Design specification is positioned as a higher level layer that can include instances of these specifications, but adds new features that considerably extend the range of pedagogies supported. It does this by providing the constructs needed to support:
- the description of activities and processes as well as of content in such a way that, when populated with learners and teachers, they can be managed and coordinated by an appropriate runtime engine.
- multiple as well as single learner models
• activities for teachers or other types of support staff, coordinated with those of learners
• learning services as well as learning content, organised, sequenced and allocated, with appropriate permissions, to participants according to the role they play
• system readable temporary and permanent information about learners and about the unit of learning
• personalisation of learning, based on preset and generated learner information and changes to it during their engagement with a unit of learning
• a conditional event model that, on the occurrence of specified events such as a given time period completing, the completion of activities, the setting of definable property values, amongst others, it allows information to be set or changed, resources to be made available or hidden, messages to be sent or further events to be triggered.

It thus wraps the content layer with a new layer that provides a rich range of capabilities that enabled the learning designer both to specify the activities in which the content was embedded and the coordination of multiple players, expressed as roles, where each may engage in different activities simultaneously. Precisely how these specifications should work together is considered an implementation issue, but some new small ‘speclets’ may be needed to effect this, where for example SCORM or QTI engines are provided as independent services.

As the Scope was ambitious, it was recognised that a specification that met the requirements could only be achieved if it built on existing work. Thus the Learning Design Scope Document also set out the intent to adopt, as a basis for the proposed specification, the Educational Modelling Language 1.0, produced by the Open University of Netherlands (OUNL) in December 2000, after a three year internally funded R & D effort. This provided a formal XML-based ‘meta-language’ for describing learning activities based on any of the 100+ pedagogical approach they studied in detail. It had the benefit of going through three specification -> implementation -> refinement cycles.

**Specification Adoption Factors**

The adoption of IMS specifications has proved a lengthy process, with the exception of the Metadata and Content Packaging specifications. This is no small part due to their adoption by ADL as components of its SCORM (Sharable Content Object Reference Model) – and even then, with a high level of support, it has taken three releases of SCORM and as many years to reach a reasonable level of adoption.

As well as producing tighter profiles of the specifications that it incorporates, ADL also represents a large market. But, equally significant, it has provided a support function for adopters, producing an open source reference implementation, providing a forum for developers to discuss problems and issues, running interoperability tests (‘plugfests’) and more recently providing conformance tests.

From this it is possible to extract a number of factors that facilitate the broader adoption of a specification:

• The identification of real needs
• An open source reference implementation – more generally 2 open source reference implementations, one for each side of the interoperability exchange.
• The hosting of interoperability testing events (‘plugfests’)
• The provision of conformance testing software
A forum for interested parties to raise and resolve issues
Maintenance of the specification through removal of bugs
Enhancement of additional features to meet further key needs of users
An information and dissemination service that prepares press releases, particularly when implementations from multiple sources can be demonstrated to work together i.e. interoperability has been achieved in practice
A market pull, which can be highly effective when adoption is a mandatory condition attached to a large purchasing budget!

These are important factors in speeding up the adoption process and we shall see that similar factors, and a support agency, are now being put in place for Learning Design.

**Interest in the UK**

CETIS and others have made presentations run a number of workshops on Learning Design in the UK and it can be reported that the level of interest among practitioners and others is both extensive and deep. In response to the growing interest in Learning Design, CETIS on several occasions invited James Dalziel to demonstrate the LAMS (Learning Activity Management System), an early ‘Learning Design inspired’ system that he developed first at Macquarie University in Australia then with the commercial company WebMCQ. This was started when the Learning Design specification was still under development. So, while LAMS does not import or export units of learning in IMS Learning Design format, it implements the basic concepts and structures of Learning Design in a runtime system and this enables practitioners to get a good idea of what Learning Design conformant systems will be able to do. In particular it provides a simplified and easy to use authoring system. The graphical, drag and drop interface is in itself a contribution to the development of Learning Design and emphasizes the importance that the quality of authoring tools will have on the adoption of Learning Design. There are now a number of evaluation projects in the UK using LAMS with live students at School, Further and Higher Education levels.

The potential benefits of Learning Design are now becoming widely appreciated by practitioners in the UK eLearning community, and it has also been written into the Department for Education and Skills’ recent Strategy for eLearning.

**… and elsewhere**

Awareness elsewhere may not be as broad as in the UK but where LD is presented it is met with enthusiasm, and there is growing interest in Europe, Australia, and Canada, though less it would seem in the US, possibly because there are fewer eLearning support agencies in the US than elsewhere. But I shall make a bold prediction: it is only a matter of time before Learning Design is included in the SCORM reference model, or both are included in a larger ADL reference model. The reason is that, although they have special needs that require SCORM, US Federal Agencies, to which ADL responds, have the same needs for the kind of enhancements that Learning Design provides as any other training and educational body.

What is notable is the speed with which a demand can build up, once awareness of Learning Design is raised, and in the UK many are now eagerly awaiting delivery of the first LD compliant tools.
Architectures: the Basis for a Learning Activity Reference Model

Before looking at the current efforts at implementing tools and systems for Learning Design, it is worth looking at the larger picture in which they will sit. Two learning Design related architectures have been developed, one for the authoring and one for the runtime environment. When the interfaces (the lollipops in the diagrams) are fully specified, then, together with the architecture, this will evolve into a full reference model.

The Learning Design Authoring and Repository Architecture

An architecture diagram for a distributed LD authoring environment, as well as a generic model for an authoring tool, was produced at the first meeting of the Valkenburg Group in March 2002. The Valkenburg Group was set up by the OUNL in response to some 20-30 requests for help in implementing EML received from all over the world. The OUNL did not have the resources to help individual efforts, but felt it could support a group of developers. The first meeting focused on authoring systems, as the OUNL was already engaged with Perot Systems in developing a commercial EML runtime player.
Those attending the Valkenburg Group meeting were asked to provide prior to the meeting their main use cases. The resulting requirements, which drove this architecture, included the need for a team workflow where different people edited different parts of unit of learning, for example an interaction specialist might create activity structures while a domain expert might draw up the required content.

Others again might be creating the presentation of that content using typical Web editing and authoring tools. Others participants wanted generic templates, perhaps embodying certain pedagogical approaches, which could be adapted to meet a variety of specific learning objectives and outcomes.

This resulted in a further outline architecture for a flexible authoring tool with ‘pluggable UIs’ over a common LD instance DOM. Later we shall see that the RELOAD authoring environment is a realisation of just such an architecture.

These needs suggested, at the system level, a separate Learning Design Editor, Materials Editor and Metadata Editor for creating the materials describing them and assembling and organising them in learning design activities. Both content and Learning Design authoring tools would need corresponding ‘Learning Design Repositories’ and ‘Material Repositories’ which would have different characteristics. As with SCORM and Content Packaging, learning designs include embedded metadata, so a Metadata editor can be used to describe all types of learning content, although a new vocabulary (or vocabularies) is needed to describe the different pedagogical approaches.

As with SCORM and Content Packaging, a repository supporting Learning Design needs to understand the internal structure of a Learning Design package in order to extract the embedded metadata and index the unit. Once this is done, it should then support searches to be made on components elements of the stored units of learning and finally, to fulfill the potential of the specification, the repository must also be able to extract and deliver these components when they meet the search criteria.

To support both the materials and the learning design editors and repositories it was therefore proposed that there should be ‘search toolkits’ to facilitate their use.

When creating learning designs, an author will typically want to be able to preview the unit of learning to see how it performs. An LD editor would therefore need access to a specialised runtime environment (the ‘Reference Runtime’ in the diagram) that can be easily set up with pseudo users allowing authors to preview their unit at its current stage.

It is also important to be able to validate learning designs, so a ‘constraint editor (and checker)’ was also included.

Finally, an LD unit of learning, being expressed in XML often involves the use of stylesheets either at runtime or in a pre-runtime phase that ‘compiles’ it into a form that is more efficient or better suited to the runtime engine. Creating these would be done by an XSLT ‘Stylesheet Editor’, while their storage and retrieval, linked to appropriate units of learning, would be an additional task for a specialised LD repository.

In this scenario the repository forms the link with the runtime environment which is where the resources get to be used by learners.
The Runtime Architecture is illustrated in the following two diagrams. For simplify the first diagram shows only the set up stage, while the second diagram adds the components and interfaces needed for live learning sessions. These diagrams were produced by the author and Scott Wilson at a Valkenburg event earlier this year, but has not had scrutiny and approval of the Valkenburg Group, so it should be considered a work-in-progress.

Learning Design Runtime Architecture: Set Up Stage

The Learning Design Runtime architecture intersects with the Learning Design Authoring architecture at the Learning Design Repository which is represented in both (shown in grey at the top left of the above diagram).

It is envisaged that the Learning Design Runtime Engine is surrounded by a set of ‘Manager’ Systems (shown in green), although in practice these may be incorporated directly into the services that they are shown here as managing. The manager systems would call on or have well defined interfaces. In the scenario where the services directly include the manager functions, they would have to support these interfaces directly.

To set up an LD Unit of Learning (UoL), an administrator first needs to be able to retrieve it a from a repository and prepare it prior to passing it to and starting it on an LD Runtime Engine. This preparation and setting up typically involves a number of steps:

1. The Unit of Learning will specify one or more roles, including, and derived from, the base LD roles of Learner and Staff. People need to be appropriately
assigned to these roles. Their data will have to be drawn from appropriate User Management System, drawing on Human Resource, Student Information or ePortfolio Systems. This is likely to use the forthcoming IMS Enterprise 2 Web services interface.

2. Optionally, if there are any global (persistent) person properties used in the UoL, these have to be drawn from a Dossier repository ready to pass to the runtime system. Alternatively this could be done directly by the Runtime Engine after the UoL has been passed to it as it has the mechanism for reading and writing learner properties. As Learning Design properties have essentially the same structure as IMS LIP activity outcomes, LIP could be used for this purpose.

3. Once Roles of have been populated with actual users, it then becomes possible to set up any services required by the UoL. Each service defined in the UoL contains a mapping of LD roles to the ‘roles’ or permissions in the service (e.g. The Learner role may be mapped to ‘participant’, and the Staff role may be mapped to ‘moderator’ in a conferencing system). Each service then needs to be set up with the IDs of specific persons mapped via the LD roles to the service roles. This is done through the Service Manager package in the diagram which has interfaces to the local service instances. When the service instance has been set up (e.g. a dedicated threaded discussion space has been set up on a conference system), the service has to return a handle (e.g. a URL) to that instance which the Setup and Scheduling will pass on to the Runtime Engine so that it can provide access to it at the appropriate point or points in the UoL.

4. Optionally, any Global Properties that are provided with initial values in the UoL must also be extracted from a persistent property service/store, and passed to the Runtime Engine. Alternatively this could be done by the Runtime Engine itself, after the UoL has been passed to it, by directly making read calls on the dossier interface.

5. As an optional step, the UoL may be compiled, in whole or in part, into a format that enables efficient execution on the Runtime Engine. This is not shown in the diagram as this might be done as part of the Set up and Scheduling, or by the Runtime Engine after the UoL has been passed to it.

6. At this point the Setup and Scheduling System has the information it needs to pass the populated UoL to the Runtime Engine, and it calls the Runtime Engines initialise interface with this data and the UoL.

7. Once the Runtime Engine has completed its internal setting up, it is ready to be started. For this, a start event has to be sent to the Runtime Engine’s start interface, either or through a timed scheduler or through a human action.
Learning Design Runtime Architecture: Live Session Stage

When a UoL is ‘live’ on the runtime system, users (e.g. learners & teachers) will be given appropriate access to the and activities and resources defined in the UoL, its first task will be to render content to the users. In the diagram this is done through a Content Rendering Manager, here separated out of the Runtime Engine as it would be a generically useful service. The learning objects will typically be accessed by the user through a Web server and these objects may be of several different types, each requiring a corresponding type of Content Rendering service. The Content Rendering Manager’s task is to examine the type of the content and call on the appropriate Content Rendering service or engine. The diagram shows the following special types of a Content Rendering service:

1. Web Content – standard Web pages and media types that can be rendered in standards Web browsers.

The remaining types are those that can be embedded within a Learning Design package and require specific processing. Here they are shown as type-specific services. These presuppose an underlying model in which, when a learner reaches the point where one of these types needs to be presented, the LD runtime then hands over control to the appropriate dedicated service. In the diagram this is done through the Content Rendering Manager. The selected service then
renders its objects, which the learner interacts with, generating data in the process. To communicate with the runtime engine, the service calls an interface on it to get and set properties values, and hands control to the runtime engine back when the learner/user completes.

2. IMS Content Packages have an internal structure which is often rendered as a navigation tree. Content Packaging typically has a small server side app that translates from the Content Package into standard Web format.

3. SCORM packages require a SCORM specific engine which in turn plays them to the learner through a Web server. SCORM content can get and set values in a SCORM defined (CMI) data model. This SCORM data model could be integrated into LD by mapping these values to LD learner properties. The SCORM runtime service would then need an interface to get and set LD properties on the LD runtime engine.

4. QTI compliant tests can also be provided through a QTI engine. If the results of tests are to be taken account of in the UoL, the QTI results have to be transferred back to the LD runtime. Fortunately the QTI tests results have the same structure as LD properties and advantage can be taken of this by enabling a QTI rendering engine to set properties on the LD engine.

5. Finally, there is a special form of ‘LD content’, consisting of XHTML pages with a specific type of XML, defined as global-elements’

6. in the LD specification, which enables users to directly access LD properties from a browser, enabling them, with appropriate permissions to get and set LD properties manually.

When it was set up, a Learning Service provided a handle that allows the Runtime Engine to present the user a Web link to the service. Clicking on the link takes the user to the appropriate service, passing to the service the user ID that was originally passed when the service instance was set up. The service can then grant the user appropriate access to its services.

When a learner session starts, the runtime engine needs to get the current set of learner properties in a ‘dossier’, learner profile or ePortfolio system. As properties associated with learners change, the runtime engine needs to update the associated property records. These would be done through the dossier interface in the diagram which might in turn be mapped a human resource, student information or ePortfolio system.

The architecture diagrams indicate a set of interfaces that will be needed to support a fully operational Learning Design environment. It will be necessary to agree on the adoption, and where the necessary the development of specifications for these. As these get put in place, the architectures will effectively evolve into LD Authoring and LD Runtime Reference Models.

The JISC in the UK, in collaboration with others, most likely through the UNFOLD Project (see later), is planning to contribute to the development of various Web service definitions for these interfaces. After prototyping, these will be offered to IMS or other standards bodies for formal adoption.

The next sections describe some of the work on implementing Learning Design and developing interfaces that is already underway.
**Open source implementations**

We now look at the work underway on developing tools and systems needed to support Learning Design. As Learning Design is a large specification, any such implementation is a non-trivial task. We begin with open source implementations, not because they are ‘better’ in any sense but because they lead the way and will take the risk out of commercial implementations that follow.

**CopperCore**

The OUNL, as part of the EC funded Alfanet Project, is producing CopperCore, an open source reference implementation for a Learning Design runtime engine.

This will prove a vital ingredient for the adoption of Learning Design, both by providing the first player able to run LD units of learning, but more significantly in the long run, by providing other implementers a definitive interpretation of the behavioural features of the Learning Design specification. Apart from incorporating the engine directly, the code can act as a guide to other developers carrying out their own implementations. It enables them to see how particular features of the specification are played out both from a user’s point of view and through tracing the code as it runs. Other developers may well implement it using different languages and different approaches, but it enables them to create implementations that present the same behaviours when presented with the same designs.

This consistency of behaviour is an essential to the wider adoption of LD. When creating a unit of learning for use across multiple LD players, learning designers need to know that their UoL will be treated in a behaviourally consistent way. Players may change the appearance (though some would argue that this too should be a designer’s choice through attached stylesheets) and they will most likely change the detailed algorithms, but the behaviours with respect to the coordination of participants, activities, learning resources and learning services should be the same.

CopperCore is being developed as a set of J2EE Enterprise Java Beans. This allows them to be incorporated in easily in any EJB supporting systems. However CopperCore itself is just the engine: it doesn’t provide UIs (beyond a command line interface) and it doesn’t provide the integration needed with other supporting systems. What it does provide is a course management capability which supports the setting up of a unit of learning on the engine, and the runtime engine itself.

The OUNL have also developed a validation suite for units of learning. This goes beyond the standard XML parser validation, checking for a number of errors which a parser doesn’t catch but which can cause the unit of learning to fail or run incorrectly at runtime. As an example, an Act can be set to finish when a the user/s in a specified role-part complete their activities. If in error, the role-part referenced is in the following act, which won’t start until the current act finishes, then the unit of learning will hang indefinitely. An XML parser will confirm only that a reference does indeed point to a unique ID within the instance. It doesn’t check either that the element is of the correct type, or that if it does, the element is in the correct location, which, as in this example can be critical. It is these kinds of errors that the TELCERT Project (see later) is addressing in developing test suites for eLearning Specifications.

The target audience for CopperCore is therefore system developers. CopperCore provides three API’s and a Test Suite. Here are some characteristics:
• has three API's covering publication, administration and delivery of IMS Learning Design
• provides a validation library
• includes a command line interface to most of the API calls
• includes an example of a publication interface
• includes an example of a web delivery interface (MS Internet Explorer only)
• supports level A of the IMS Learning Design specification
• platform independent
• has support for two relational databases (MS SQL Server/MSDE or PostgreSQL)
• is ready for use with JBoss 3.2.x, but should be able to run on other application servers as well
• licensed under the GNU GPL

The ultimate goal is to upgrade CopperCore so it will be capable to handle level C of the IMS Learning Design specification.

http://www.coppercore.org/
Source code available at:
http://sourceforge.net/projects/coppercore

An interesting technical article on the implementation of a Learning Design runtime engine as a collection of finite state machines, by Hubert Vogten (senior developer of CopperCore) et al. at the OUNL, is available at:

**OU UK collaboration with OUNL adding Web Services to CopperCore**

As part of the JISC eLearning Frameworks Programme, the Open University in the UK is collaborating with the OUNL to provide a working runtime environment around the CopperCore engine.

To enable this they are planning to provide Web Service interfaces to the CopperCore engine so that it can be integrated into a large institutional environment. This takes an important step towards creating the new interfaces needed to define a learning activity reference model.

The OU UK team is also proposing to develop a simplified template based editor for Learning Design, possibly based on the RELOAD editor (see next).
See:

For anyone planning on integrating CopperCore into an EJB environment, I recommend looking first at the valuable LD Diary maintained by Alex Little at the OU UK, as he presents the problems encountered along the way and the solutions found.
http://iet.open.ac.uk/pp/a.little/index.cfm?page=LDdiary

It would be good if more developers published such a diary on the Web – particularly when working with open source tools where others might encounter similar problems.

There is also an interesting paper on this work by Patrick McAndrew et al that was recently presented at AusWeb 04
RELOAD

The RELOAD is another JISC funded project that is producing an open source XML authoring framework and authoring tools for eLearning specifications that is being developed at the Bolton Institute.

RELOAD has produced editors and supporting players for IMS Content Packaging and SCORM 1.2, together with a LOM Metadata editor. As well as a low level ‘tree-table’ editor, these provide higher level, ‘drag-and-drop’ interfaces that hide all the XML from the user by automatically generating this and creating the packages. These are already successful with downloads now at over 1,000/month and rising.

RELOAD XML Schema-driven Editor Architecture

The RELOAD designers participated in and contributed to the first Valkenburg meeting and it is no coincidence that the RELOAD architecture is an implementation of the detailed authoring tool architecture also produced at that meeting (not shown above).

It provides a generic Schema-driven core which generates both the instance DOM to be edited, as well as the Tree-table and Form-view editor interfaces directly from the provided XML Schema. However the kind of UI that can be automatically generated is limited, and to go beyond this, UIs that depend on human understanding of the processes involved need to be developed by hand. This is what has been done for the Content packaging and SCORM editors and is now being done for the Learning Design editors.

The RELOAD framework and authoring tools have been adopted by a number of other developers as a basis on which to develop their tools, notably by ADL as the basis on which to build its forthcoming SCORM 2004 editor, and also by Harvest Road a company that produces Beehive, a distributed repository for eLearning content.

This year, through to July 2005, it is focusing on Learning Design authoring tools, the first versions of which should be available by September. These should provide a basic
drag-and-drop capability for all aspects of Learning Design Level A, with Level B and Level C editing to follow. (see screen shots in Appendix)

Once this is completed, it is intended to enable Learning Design Activity Components to be developed and packaged independently, together with an icon. These will be made available to a simplified higher level editor as drag and drop icons which can be used to wire together activities and LD plays. These components will be configurable through properties. Authors will also be able to drop back down to the lower level editor if they want to modify aspects of the components.

The editor will also have a version of the CopperCore engine integrated into it so that an author can carry out edit-view cycles on a UoL they are working on via virtual users, presented as different browser windows which the author will be able to switch between.

As Learning Design can support many different pedagogical approaches, it can be expected that different editors will be developed to provide support for specific approaches. With much of the low level work carried out by the RELOAD XML, filing configuration and help engines, it is a much simpler task to create alternative editor interfaces to meet different needs, or indeed editors for other specifications and Schemas.

RELOAD is currently being internationalized which will be available in the next release.

The editors can be downloaded from:
http://www.reload.ac.uk/
The source code is available at:
http://sourceforge.net/projects/x4l-reload/

**eduplone**

http://eduplone.net/

EduPlone is an adaptation of the well liked Plone Content Management System, which is in turn based on the open source Zope Web application server. This should be an interesting development to watch as a number of learning management systems have been developed using Plone/Zope, for example by a number of medical schools whose varied and specialised needs rule out the more standard offerings.

A strong feature of Zope/Plone is the ease of setting up and modifying the system. This has been carried over into EduPlone.

**Individualised Content**

Being content based there is a focus on selecting and adapting content ot individual needs. Learning content is tagged with what the EduPlone developers refer to as ‘didactical’ metadata thus making the LMS aware of the ‘didactical functions’ inherent in the materials. With this information, the LMS can then resequence and reorganize learning materials and assignments at runtime according to the individual needs of each learner.

EduPlone also supports self-organised and self-directed learning and the learner's active engagement of knowledge and cooperation with other learners and tutors.
Learning Sequences
EduPlone also supports interactive Learning Sequences. These support both f2f courses or for completely online virtual learning groups. Again the emphasis is on making it easy to define and manage tutor-learner interactions.

The relevance here is that EduPlone Learning Sequences use the IMS Learning Design Level A specification as their standard format for the export and exchange of its learning units.

http://eduplone.net/products/learningsequence/index_html?cl=en

Gilbert Paquette
Centre de recherche CIRTA-LICEF, Télé-université

GP had developed an approach to eLearning termed Instructional Engineering which had a number of similarities with EML/LD. Having a number of existing tools to support this approach, his team is now working on adapting them to support input and output of units of learning in LD compliant XML.

In particular, he has developed MOT+, a graphical editor for learning activities which is being adapting for LD, possibly by transforming its XML output into LD XML. He has also developed a runtime engine, Explor@-2 which is being considered as the basis for an LD runtime player.

Contact Glibert Plaquette for further information on the LD version of MOT+ (gpaquett@licef.teluq.uquebec.ca)

The current version of MOT+ is available for download from: http://www.licef.teluq.uquebec.ca/francais/real/demot.htm
(Please note that the page is French but both French and English versions of the software are available for download.)

SCOPE Project
As part of the EU SCOPE project UPF developed an open source set of libraries implementing the main elements of LD Level A as a library of Java classes, jLDA.

This is currently available from the SCOPE web site in a beta version. http://www.tecn.upf.es/gti/leteos/newnavs/libsdeveloped.html

Commercial Implementations

Perot Systems Edubox
This runtime system, developed by Perot in collaboration with the OU Netherlands, is primarily an implementation of the LD predecessor EML. However there is an import facility which enables it to accept LD Units of Learning.

The main point to note about this system is that it was developed by Perot Systems for their training services and it is, unfortunately for the rest of us, not being made available as a commercial product in its own right. Outside of Perot, it is in use at the OU NL.
LAMS (Learning Activity Management System)

This is an interesting product as it could sit under both the commercial and open source categories. It is currently a closed commercial product although an announcement that all or part of it will be made open source is expected shortly.

A commercial company, LAMS International, has been set up to take LAMS forward. LAMS is built using standard components and highly scalable back end systems, which enable it to run on most of the popular server platforms. LAMS International offer to integrate LAMS into existing Learning Management Systems, or to provide it as a stand alone system.

LAMS International has an affiliated not-for-profit organisation, the LAMS Foundation,

The LAMS Foundation is a not-for-profit organisation that manages research and development into LAMS and the concepts of Learning Design. The Foundation is based at Macquarie University, Australia as part of the Macquarie E-learning Centre of Excellence (MELCOE), under the leadership of Professor James Dalziel (Director of MELCOE and inventor of LAMS). The LAMS Foundation collaborates with LAMS International to foster the adoption and implementation of LAMS across all education sectors. It also seems to be the intention that they work together in IMS to extend the IMS Learning Design standards to support the capabilities built into LAMS, when they would then be able to implement the LD specification.

As mentioned earlier, this system was in the main developed before completion of the IMS Learning Specification. Hence it is described as ‘Learning Design inspired’, meaning that it implements many of the concepts and ideas of Learning Design and adds anumber of its own, together with a range of additional specialised learning services. The consequence is that it is not an implementation of the IMS Learning Design specification, meaning that it cannot import or export IMS Learning Design compliant units of learning.

The main features of LAMS are its easy-to-use authoring tools which make it usable by teachers without a technical background. It is an integrated system with a run time engine, good monitoring and administrative capabilities and LAMS sequences can be stored in repositories such as Intrallect’s IntraLibrary which supports metadata and searching.

LAMS International has an affiliated Not for Profit organisation LAMS Foundation based at Macquarie University which is likely to handle the open source side

For more see the separate Alt-i-Lab briefing paper by James Dalziel.

http://www.lamsinternational.com/
http://www.lamsfoundation.org/
james@lamsfoundation.org.

eLive LD-Suite

An interesting and advanced graphical editing tool for Learning Design is being produced by the commercial company eLive. eLive Learning Design, based in Germany, is working in co-operation with cogito GmbH, to develop an integrated editing toolset, called "eLive LD-Suite". This will support the design, documentation and optimisation
of learning scenarios, including e-Learning and blended learning designs, and will also support scenarios focusing exclusively on conventional face to face settings.

See appendix for screen shots.

The LD-Suite will be able to export web-based and hybrid Learning scenarios as IMS LD documents in IMS LD / IMS CP format. Version 1.0 of the suite supports IMS Specification Levels A and B. Support of Level C is scheduled for a later version. LD-Suite’s Graphical User Interface hides all the XML-Code from the user while still supporting all three levels of LD. The elive LD Suite version 1.0 is scheduled to be released in September 2004

Additionally the LD-Suite provides features for administration, re-use and adaptation of existing learning designs. For this purpose, it provides pre-modelled structures, templates and pedagogical patterns. The user will be able to extend the existing repository and interchange effective patterns and scenarios.

Contact development@LD-Suite.com
LD editor announcement
http://www.elive-ld.com

GTK Press
The following announcement has been made by GTK Press:

“GTK Press is partnering with Online-Learning.com to develop an "Authoring and Adaptation Environment" - a standards-based authoring environment for the development of online courseware. The Project uses the Educational Modeling Language (EML), and its successor, the IMS Learning Design Specification (IMS-LD) as the foundation for the authoring environment.”

For further information check:
http://www.gtkpress.com/index.php?currentNumber=3&currentIsExpanded=0

Further sources of information

OU Netherlands Learning Networks site has information available about learning Design in general and in particular, maintains a list of Valkenburg Group developments of LD tools and systems. Refer to this site for updated information.
http://learningnetworks.org/forums/showthread.php?s=&postid=377#post3

Support Activities

Valkenburg Group
As mentioned earlier, the Valkenburg Group was set up in March 2002 by the OU Netherlands to meet the requests for help in implementing its Educational Modeling Language (EML) and Learning Design. The first three day was productive and successful and series of one day meetings have been held since then, usually in association with IMS meetings. At the meeting linked to the IMS Technical Board in
Sestri, Italy in May 2003, it was proposed that the Group should collaborate and produce a book on Learning Design. This has been a significant focus of the Group, with a week of intensive writing held in January this year in Dagstuhl castle in Germany. Since then the book has been accepted by the publisher Springer-Verlag and the final editing is now completed. The book should be available before the end of this year.

http://www.valkenburggroup.org/

UNFOLD
As the Valkenburg Group was an unfunded effort supported by the OUNL, and hence a closed membership, the UNFOLD Project sought, and got, funding from the European Commission to set up three interlinked communities of practice for:

- Software developers
- Learning designers
- Learning providers

This was based on an adoption model for eLearning standards that models the stages, tasks and practitioners that need to be involved to take a specifications form its public release through to widespread adoption.

The first step is to get the specification implemented in appropriate tools and systems, for which early open source reference implementations are important facilitators. Once implementations are produced then it becomes important to test interoperability between them. This typically involves a few cycles where developers adjust their understanding and their systems and any problems identified in the specifications get fed back to the specification body for the next ‘point’ release. UNFOLD expects to run such events once a sufficient number of systems that implement Learning Design become available, probably during the first half of 2005.

Once authoring tools become available, designers can begin to use them to create units of learning. For these to be used with learners, runtime systems need to be available for learning providers to set the units of learning up, make them available and work through them with learners.

At each stage, the exchange of experience and the development of good practices are essential to moving forward the state-of-the-art.

It is also essential to have good communications between the different types of practitioner. Thus if a unit of learning is not working well with learning, the providers need to identify the cause and communicate to the learning designers. They will either be able to fix it and return the amended version or they will find that their authoring tool does not support the desired changes. They in turn need to go back to the tool developer and ask for a change. The tool developer will similarly either be able to make the changes or they will find that the current specification does not support the required change. They then have to go back to the specification developer with the end users use case and request changes to support it.

At present this communication process happens in an informal way and tends to be patchy and slow. UNFOLD is seeking to provide the channels needed to enable these communications take place easily and swiftly, and thereby accelerate the roll out and adoption of Learning Design.
UNFOLD is currently just completing it’s setting up stage, putting in place the systems needed to support the communities of practice.

The first three day meeting, one day for each Community of Practice with overlap sessions for cross community communication, will be in Barcelona from 8-10 September 2004. For full details see: http://www.unfold-project.net

R2R "Repository to Reality"
University of Calgary

The purpose of the Industry Canada funded R2R Project is to extend and enhance Canadian work in the area of IMS Learning Design. It seeks to direct the findings, in terms of best practices, recommendations, and tools created, back to those involved in developing specifications, on the one hand, and use them to inform those implementing and applying Learning Design on the other.

Supporting the pilot programs with an effective communications and community-building plan and contributing to existing strategies for training, adoption, and evolution of e-learning are key components of this project. As part of the project, it provides a Weblog as a tool for capturing, synthesizing and disseminating research and current activities related to Learning Design. In order for a project of such scope to be successful, contributions from a wide variety of people are essential. It is only through collaborative efforts that innovation and adoption flourish. The weblog is also intended to facilitate discussion and debate and serve as a repository for capturing and disseminating information. They encourage participation from all interested parties.

This project is a joint effort involving several Canadian universities. Mike Mattson, University of Calgary is managing the project. Dr. Katy Campbell, University of Alberta, and who was also co-chair of the IMS Learning Design Working Group, is investigating pedagogical and technical issues related to the implementation of the IMS Learning Design specification in a post-secondary environment. Dr. Tom Carey, University of Waterloo is developing a controlled vocabulary and investigating the technical issues surrounding Learning Design. Dr. Gilbert Pacquette, TeleUniversity will be developing an implementation report which will address the challenges of implementing the IMS specification. http://commons.ucalgary.ca/weblogs/learningdesign/

TELCERT

The European Commission has funded the TELCERT Project which is developing Conformance Test Suites for eLearning Specifications, together with a toolset to support the definition of application profiles and a Content Reengineering tool, based on RELOAD.

Recently the OU Netherlands has become a partner and there is a reasonable expectation that conformance tests for Learning Design will be produced in the second phase of the project, in 2005-6. http://www.opengroup.org/telcert/
**Issues**

As a specification moves out into the world, a number of issues inevitably arise. The following indicate a few of these.

**Supporting specifications (architecture/reference model)**

For Learning Design to be effectively implemented in a working environment, a number of systems need to be developed along with interfaces to support these, resulting in a supporting reference architecture. These have been discussed earlier.

**Services (we want more!)**

When the Learning Design specification was being implemented, the number of services it supported was cut down to the lowest common denominator to ensure maximum interoperability, the main ones being email and conferencing. Quite clearly many more collaborative and other learning services are needed and the question is how to support these in an open and interoperable way.

At the specification level, the LD service specifications essentially map LD roles into the ‘roles’ (or permissions) of the service (e.g. in the LD conferencing service, these are embedded as participant, observer, moderator and administrator elements). If, instead of hard wiring these types into the specification, they became an enumerated list \{participant|observer|moderator|administrator\} to be imported using the IMS VDEX (Vocabulary Definition and Exchange) specification, then a generic ‘service’ type could be developed which held place holders for the roles to be filled in according to the service.

However while this might make the specification more flexible, it would not address the question of interoperability between systems. Two possible approaches are suggested:

1. Use Web Services to make available required services in a way that is independent of the LD runtime engine. Once an open service is available on the Web it can be used by many different implementations at different locations. These could be either freely available services or based on a commercial model.

2. To include an internet address for a site where an appropriate service can be downloaded for on site use. This would be a step needed at set up time. If a unit of learning specifies a service that is not available locally, then, if it is desired to continue with that unit, the URL would enable the required service to be downloaded and installed. Again the service could be made available on an open source or commercial basis. Ideally, there would be set of URLs pointing to several alternatives that can provide the service.

**Designer control of layout of elements**

Another issue that has come up with groups, is that of control by learning designers of the layout of the Learning Design elements, in particular, Activity descriptions, and objectives. The LD specification leaves such decisions to the runtime system, but a designer may specifically want to have the description of the activity available to learners at the same time as they view the resources referenced. Or they may want to have the learning objectives available on a tab so that the learner can easily refer back to them at any point.

Given that the activities are expressed in XML, the possibility of including an XSLT reference in a learning activity, to be used when rendering it, could be explored.
Control of groups and more sophisticated learning flows

One issue, raised when implementing LAMS, is that of groups, and in particular, rotating the roles played by members in groups through successive iterations. LD specifies Roles and these can have upper and lower limits set on the numbers of players, effectively forcing multiple smaller groups playing the same Role, when there are more learners assigned to a Role than its upper limit allows. But there is no way in which these can be automatically rotated in successive activities.

In project based learning it is also desirable to decide who is going to play which role in discussion during the session. Typically a unit of learning pre-allocates players to roles at the outset, making it not possible to change them during the session, but this is not a requirement of the specification. However there is no way for a designer to specify that this is what is wanted and could be a useful future extension.

Currently in LD, the main Acts of Play have to run in a sequence. While within an Act, each Role runs in parallel and different Activities can be allocated to each Role to engage with in parallel, it is not possible to have a subset of Roles complete and start another Activity set, independently of another Role or group of Roles carrying out a longer lasting Activity. This is a fairly sophisticated sequencing demand but it may come up once the uses of Learning Design systems mature.

Steven Downes

To end on a controversial note, a quote from Stephen Downes in reply to the many responses he had to a feature of his that was critical of LD:

“Once learning objects are widely available and widely used, the traditional thinking surrounding the organization of learning will be increasingly questioned. People will begin to ask why learning resources must be organized by hand by a designer before they can be used by students. Systems will emerge that allow students to be their own designers. Instead of viewing learning design as some sort of script in which students are actors, following directions, we will begin to see a model where students are players, following no script at all.”

“But we're not there yet, nor will we be for a good number of years. So it is appropriate, for now, to revel in what we have created. And that, it seems to me, is what this month's issue is all about.”

Is it a question of granularity: some one has to design something for learners to assemble. How large or small should these be? The smaller they are, the more work the learner has to do assembling them.

What about collaborative learning? How do they co-design collaborative learning, rather than individually assembling knowledge items? An interesting story from James Dalziel describes how a teacher used the LAMS authoring tool with the learners to co-design the learning activity they were to engage with. This is already a step towards what Stephen had in mind. Is it just a question of having sufficiently easy authoring tools for learners to be their own designers, engaging in writing their own scripts, on their own, in groups, or with a teacher? But still an authoring tool and a runtime system are needed to express their designs and to coordinate their activities.
Conclusions

It takes time to move a specification out into the world, and Learning Design, being a large specification, could be expected to take longer than most. But an encouraging number of developments are underway that suggests that this timescale can be shortened:

- There are open source reference implementations being developed
- There are further commercial and open source developments also under way
- There are LD focused support groups and communities of practice being put in place, supporting learning designer and learning providers as well as software developers
- Support is in place for plugfests when sufficient systems are ready
- Conformance testing is likely to be developed in 2005
- …and there is great interest in LD out in the community.

These elements coming together suggest that in a year from now, systems will be available and a thriving and activity Learning Design community will be in place.
Appendix – Screen shots

eLive Learning Design Suite

eLive LD-Suite: general screenshot
An important feature of the LD-Suite is that it supports reuse of modules and patterns.

eLive LD-Suite: Screen shot with Module menu showing reuse capability
eLive LD-Suite: Integration of Training Portfolios
eLive LD-Suite: Detail showing the definition of a Learning Activity
LAMS: Learning Activity Authoring Web interface
RELOAD Learning Design Editor

RELOAD - Role Editing
RELOAD - Environment Editing

Environments

- Environment 1
  - Aims of the Versailles Experience
  - Introduction to the Preparatory phase
  - GB Objectives for the Treaty of Versailles
  - USA Objectives for the Treaty of Versailles
  - Polish Objectives for the Treaty of Versailles
  - French Objectives for the Treaty of Versailles
  - Italian Objectives for the Treaty of Versailles
  - Serbian Objectives for the Treaty of Versailles
  - Negotiation Recorder
  - Environment 2
  - GB Team Forum
  - USA Team Forum
  - Polish Team Forum
  - French Team Forum
  - Serbian Team Forum
  - Environment 3
    - Send Mail
    - Conference
    - Learning Object
    - Index Search

Italian Objectives for the Treaty of Versailles

This section provides information about the Learning Object. You can add a reference to a Resource.

Resource: [Input Field]
RELOAD - Method Editing