A Review of Learning Design:

Concept, Specifications and Tools

A report for the JISC E-learning Pedagogy Programme

Sandy Britain

May 2004

1.1 Introduction

The principle aim of this report is to review and evaluate currently available software tools related to learning design. Whilst the IMS Learning Design Specification (IMS-LD), which we will consider in some detail here, provides a very thorough framework for evaluating the capabilities of software tools within the learning design space, it is also possible to argue that it presents only one of many possible realisations of the concept of learning design and that there are other possible ways to model the concept of learning design that do not implement IMS-LD.

Consequently, it has been suggested (Dalziel, in press) that it is useful to make the following distinctions between:

1. Learning design as a broad concept
2. The instantiation of the concept in the IMS-LD specification[1]
3. The realisation of both in software tools to support the process of creating and managing learning designs.

In the remainder of Section 1 of this report, we examine in more detail what is involved in first the broad concept of learning design and secondly the IMS-LD specification. Following that we identify the range of software tools that are related to learning design in both senses and construct an appropriate framework for evaluation of these tools.

In Section 2 of the report we review a small selection of tools that are currently attracting a great deal of interest and finally we draw out our conclusions and recommendations based on this work.

Since we are differentiating throughout this report between the general concept of learning design and the way the concept is implemented in the IMS-LD Specification we shall attempt to make the distinction clear by adopting a convention of using 'learning design' (small 'l', small 'd') when we are talking about the general concept and 'Learning Design' (Capital 'L" and 'D") when referring to the concept as implemented in the IMS specification. References to the IMS Learning Design Specification will be abbreviated to IMS-LD.

1.2 The concept of learning design and models of (e)- learning

Despite its relatively recent appearance in connection with e-learning, this concept of 'designing for learning' is far from being a new idea. In a traditional face-to-face context, many teachers may consciously and reflectively engage in the process of learning design in this general sense as part of everyday lesson planning, whilst other teachers or lecturers may never have given it much thought, but nonetheless make subconscious learning design decisions every time they prepare a teaching session.

Yet, whilst they are hardly groundbreaking new ideas in education, the central ideas behind learning design represent new possibilities for increasing the quality and variety of teaching and learning within an e-learning context:

- The first general idea behind learning design is that people learn better when actively involved in doing something (i.e. are engaged in a *learning activity*).
- The second idea is that learning activities may be sequenced or otherwise structured carefully and deliberately in a *learning workflow* to promote more effective learning.
- The third idea is that it would be useful to be able to record 'learning designs' for sharing and re-use in the future.

[1] Dalziel (in press) notes that the IMS-LD specification is not the only relevant standards effort in this area

1.2.1 Learning Activities

Whilst learning is an effortful and active process of knowledge construction that humans perform quite naturally, not all learners are equally capable of effective and efficient learning on their own. Indeed, most if not all, benefit from some level of guidance and support. Successful teaching involves a variety of strategies and techniques for engaging, motivating and energising students over and above merely presenting them with well-designed learning materials. There are a number of pedagogical techniques that focus on providing activities for learners to perform either in groups or as individuals that help to create deeper, swifter and more effective learning. These may be in the form of discussions, simulations, mimicry, problem-solving exercises, role-plays and quizzes or meta-learning tasks such as construction of mnemonics and mind-maps.

The recent trend within e-learning has been to focus on quite a narrow set of learning activities that can be easily managed within a browser-based VLE: ‘read this content’, ‘do this multi-choice quiz’ etc. Part of the aim of learning design is to help broaden the set of activities that are used to support learning in an e-learning context

1.2.2 Orchestrating Activities – Creating a Learning Workflow

A second feature of successful teaching is not just the creation of thoughtful and engaging activities for students to undertake, but also giving thought to the sequential order and timing of the various activities and the presentation of the resources needed to support them. This orchestration may form a simple sequential flow, and in most cases it will, but there may sometimes be call for a learning design that involves branching of workflow into parallel activities undertaken by sub-groups before coming back together. Or a design may be constructed that allows different routes to be taken based on achievement at a testing stage within a sequence. Thus a second key aspect of tools to support the concept of learning design will be the notion of workflow.

From the teacher’s perspective there are two main advantages associated with consciously thinking about the process of designing learning activities. The first is that it provides a framework for teachers to reflect in a deeper and more creative way about how they design and structure activities for different learners or groups of learners and the second is that designs that prove to be effective may then be communicated and shared between teachers or archived for re-use on future occasions.

1.2.3 Sharing and Re-using Learning Designs

However there is a problem in that it is not so easy to describe a given ‘learning design’ in a consistent and transferable way that will allow easy re-use. The ‘design’, ‘pattern’ or ‘recipe’ needs to be described at a sufficient level of abstraction that it can be generalised beyond the single teaching and learning context for which it is created, but not at such an abstract level that the pedagogical value and richness is lost. This problem is exacerbated when we begin to think about creating, transporting and re-using learning designs in electronic learning environments. It is this problem that IMS-LD is intended to solve. A learning design that conforms to the IMS-LD specification

Whilst the benefits of engaging in the process of learning design exist regardless of the mode of delivery (electronic or face to face), they are particularly relevant to e-learning, which, unlike traditional face-to-face learning, has tended to focus on content and services at the expense of learning (inter)actions. Whereas instructional design in e-learning has focussed predominantly on *learning objects* as the core entity within a course or other programme of learning, learning design, as we have seen, is centred on *learning activities*. The underlying reason for this shift in emphasis is the feeling amongst many educators that the learning objects approach places too much emphasis on content delivery rather than looking more carefully at what learners do. Also software environments for elearning (VLEs) have been designed to cater for this rather simplistic content-delivery model at the expense of a variety of pedagogical models that are built around collaborative activity on the part of learners. Whilst good and well structured content is undeniably important in creating a quality course, also important are the tasks, activities and dynamic interactions that occur between people (learners and teachers) and the software environment There is no means to encode these features of learning and teaching within the prevailing content-based model.

1.2.4 IMS Learning Design Specification

The aim of the IMS Learning Design Specification is to provide a model within which to describe the structure of tasks and activities, their assignment to roles, and the workflow of a unit of learning as a 'learning design', and also to provide a platform-independent notational convention to allow sharing and re-use of these designs.

What we have outlined above are in effect two related but independent ideas that affect the creation of software tools to support learning design. The first is the general concept of learning design (activities, collaboration, workflow etc.) and the second is the particular instantiation of that concept in the IMS Learning Design specification (IMS-LD). It is important to understand that the two ideas need to be treated separately in reviewing software tools in this field, since some of them are aimed at implementing the IMS-LD specification, whilst others may not implement the specification yet they do embody their own model of learning design, albeit one that is not necessarily transferable between systems. Also, since the IMS-LD specification is still evolving, some software designers are reluctant to adapt their software to conform to the specification in its current form, preferring to adopt a 'wait and see' stance.

1.2 Background to Learning Design and EML

The development of EML by Rob Koper and colleagues at the OUNL grew out of dissatisfaction with the prevailing content-oriented *learning objects* focus within e-learning. The team felt that instructional design within e-learning had been hijacked by those pushing a very narrow model based on digital content and virtual learning environments that ignores the richness of interactions between teacher, learner, resources and environment.

EML is a notational system intended to provide a way of describing teaching and learning interactions at a level of abstraction above the specific instance of the context in which it was created. The resulting model acts as a design pattern for that teaching and learning instance. At the heart of EML is the idea that Learners perform Activities in an Environment with Resources. According to Koper (2001), this general statement about the core entities and relationships involved in learning expresses a key axiom that is common to all major educational approaches.

Unsurprisingly, the OUNL team that had been involved in the development of EML became major contributors to the IMS Learning Design Working Group – the Valkenburg group (named after the location of the first meeting of the group). The work of this group led to the recent production of the first version of the IMS Learning Design Specification. The aim of this specification is to provide a digital format for encoding, transporting and playing learning designs. There are a number of differences between EML and Learning Design. Most importantly it is essential to understand that whilst EML was intended to encapsulate all teaching and learning interactions, LD is designed to work in combination with other IMS specifications such as Metadata, Content Packaging etc. Hence LD does not attempt to include all aspects of the educational process. For example assessment is handled by the QTI specification and so is not handled by IMS-LD

IMS-LD is a complex specification and the best practice implementation guide produced by IMS to assist educators in understanding how to use the specification is a difficult document to read and understand. Consequently there is currently some degree of confusion about how the specification relates to the overall concept of learning design described above. One of the aims of this report is to clarify the distinctions involved as well as the underlying motivations and practical application of the specification.

1.4 Overview of the IMS – LD Specification

The main reason for implementing a standard for Learning Design is to make digital information encoding learning designs consistent and thus both transportable and re-usable in different software packages.

The IMS Learning Design Specification (IMS-LD) consists of three interrelated documents in common with all IMS specifications:

- XML Binding Document
- Information Model
- Best Practice Guide

The XML Binding Document is a technical document detailing how learning design elements are represented in xml and does not need to be addressed here. It is perhaps useful, however, to reiterate at this point that the aim of IMS-LD is to provide a specification of the elements and structure of Units of Learning as conceived in EML. This specification is provided in XML format, which is a platform-independent web-standard notation for describing arbitrary structured data. This means that the 'Learning Design', encoded in XML, can be read by any runtime environment that can read the XML description.

A second point that it is important to be clear about is that IMS Learning Design is designed to work together with IMS content packaging as Learning Design does not itself specify information about content. The way the relationship is expressed in the Information Model for Learning Design is that the aim of Learning Design is, as we have said, to model Units of Learning, so:

A Unit of Learning = IMS Content Package + IMS Learning Design.

Technically this is achieved by including Learning Design elements within the manifest of a content package.

What we are interested in here is outlining how the IMS-LD specification is intended to add value to teaching and learning practice in e-learning, this information is provided using the Best Practice Guide and the Information Model as our source.

1.4.1 The Anatomy of Learning Design

The core components of Learning Design identified by the IMS-LD working group, derived from the earlier analysis performed by Koper and colleagues in their work on EML are based around the conceptual entity of a *Unit of Learning* or *Unit of Study.* This is the smallest unit that satisfies one or more learning objectives. In practice this may be a course, a module, a lesson or single activity such as a discussion.

For any given unit of learning some or all of the following elements need to be described in an IMS Learning Design:

Learning Objectives. One or more learning objectives

Roles. There are two kinds of Roles used to represent people: learner or staff. Specific individuals are not a generalisable component, but Roles are, so the role is specified in the design rather than a person.

Activities. These can be of two types, either *learning activities* or *support activities*

Activity-structures. Activities can be aggregated using activity structures. Activity-structures can also reference other activity-structures and external units of learning.

Environment. The environment element contains two basic types:

- *Learning Objects* which would typically be a URL to external content, tools or tests with optional metadata

- *Services.* This refers to a service provided within the environment that is available at runtime but cannot be specified at design-time. Examples of services may be discussion forums, chat rooms, monitoring tools and other features typically provided by VLEs.

Some of the generalisable design components and the learning objectives described above need to be bound to specific instances at either at design, instantiation or run-time depending on the context. This binding is achieved using elements called Resources:

Resources. Resources can be of five different types: web content, imsld content, person, service facility or dossier

Finally the learning design needs to specify how the learning and support activities performed by different roles using the various learning objects and services are organised into a coherent workflow. This facility is provided by the *Method* element.

Method. The method consists of a *Play* (or concurrent *Plays*), which contains a sequence of *Acts*. Each *Act* contains one or more *Role-Parts*. Each *Role-Part* associates one *Role* with one *Activity* or *Activity-Structure*

As can be seen from the above the Learning Design specification uses the metaphor of a theatrical play to describe the workflow involved in a learning and teaching scenario. The workflow is fundamentally sequential as the acts are sequential, but there may be more complex behaviour than a single sequence through the provision of concurrent Role-Parts which means that branching and simultaneous activity by sub-groups is possible.

1.4.2 The Levels of Learning Design in IMS-LD

There are currently three levels of Learning Design that have been formulated by the IMS-LD Working Group:

Level A: This includes all the elements outlined in the previous section. The main added value to e-learning of Level A learning design is that it defines Activities and Roles as reusable components that can be designed into a workflow using the Method element. It also allows Services such as email and conferencing to be specified at design time as placeholders within the design that will be instantiated by the run-time system. These features are a qualitative difference from IMS content-packaging and SCORM, which include no concept of activities or roles and only work with content so that when they are included in a VLE are completely divorced from discussions or other collaborative tasks.

Level B: This allows the inclusion of properties and conditions. Two types of property have been proposed: *Internal* and *External.* The addition of external properties is important for adaptation to the design based on properties of the individual learner such as may be provided by the Accessibility and IMS-LIP specifications. This means that activities and activity sequences could potentially be adapted to suit the needs and preferences of individual learners

Level C: This provides a notification capability that allows messaging between system components and means the flow of events could be adapted at run-time based on event triggers such as completion of earlier tasks. This paves the way for adaptive sequencing capabilities as well as role-play and event-driven simulations.

1.4.3 Building a Learning Design

The Best Practice Guide describes a sequence of steps that characterise the development of a learning design for a unit of learning:

1. The first task is to analyse a specific educational problem as a use case and then turn it into a scenario describing the learning objectives and tasks or activities establishing the basic order of events that can be captured in a narrative form.

2. It is suggested in the Best Practice Guide that the narrative is then cast into UML activity diagram. This UML diagram then forms the basis for creating the XML document that implements the IMS-LD spec

3. Then the actual content (resources) can be created and finally a content package can be created that incorporates the learning design.

Table 1 below distinguishes the various activities involved in the process of learning design and how each of those stages is handled within IMS-LD.

Learning Design Process	**IMS-LD Process**
Define Learning Objectives	Specify Learning Objectives
Develop narrative description of learning and teaching scenario	Not defined within current scope
Create learning activity workflow from Narrative description	Create a Method using Play, Acts and Role-Parts
Assign resources, tools and people to activities	Specify Roles, Resources Environment and Services
Running (real-time)	Use a Learning Design aware player
Learner support and on-the-fly adaptation	Not Defined
Reflecting (including sharing outputs for peer reflection)	Not Defined

Table 1. Key Activities in Learning Design

This outline of the process of creating a learning design illustrates very well the need for software tools to be developed in order for ordinary teachers to engage with this process. Even if teachers were used to developing scenarios in narrative form (as many are not), very few would contemplate turning these into UML diagrams and then IMS-LD conformant XML. Software tools are needed that will support the authoring of learning designs and tools are needed to play learning designs in a run-time environment.

1.5 An Evaluation Framework for Learning Design Software Tools

In order to differentiate tools in this area we have developed a set of evaluation questions. These questions are designed to provide a simple and easy-to-use evaluation framework.

The questions are divided into three groups:

1. Questions about the intended purpose of the software
2. Questions about the design characteristics and functionality of the software
3. Questions about the technical aspects of the software

Intended Purpose	
1: Description of intended purpose	
	A brief overview of the intended purpose of the software. This can usually be gleaned from the software or project website
2: Who is the system for? Who else is involved?	
	Some of the tools are intended for use by ‘end’ users i.e. teachers and learners, others are intended for use by software developers or instructional designers with a high-level of technical expertise. It is important to be aware of the difference
3: Perspective or world-view of the designers	
	It is useful to know something of the background of the software designers to gain a deeper understanding of what their software is about and the pedagogical orientation of the software. If they come from a commercial or training background the software may have a different emphasis to software designed specifically for further or higher education.
4: Scope	
	Some of the tools in this area are authoring or editing environments, some are runtime environments also known as players. Yet others do both. Some tools are intended to support single-learner electronic delivery, others are intended for handling multiple-learner and blended learning situations
6: Integration	
	Some of the software tools are designed to be used independently of any other application; others are designed to work as part of a wider suite of tools or environment. Can the software integrate with other tools?
7: Does the software implement IMS-LD? (if so, what level: A, B or C?)	
	A simple question. If it doesn’t (and none do at the present time) then is it intended that it should in the future?

Design Characteristics	
1: What are the main concepts or entities built into the software?	
	A great deal of information can derived about a piece of software by looking at the business model inherent in the software. If the business model is not explicit, a useful guide is what are the main entities, objects and concepts built into the software. For example, for learning design we might wish to know whether it is built around model of 'Activity' and 'Workflow' objects or whether it is built around 'learning object' and 'content' objects. Of course this is only a guide and deeper analysis may be required as software designers might label a software entity as an 'activity', with operations such as 'create new activity' and 'save activity' when in fact the activity is merely a piece of content.
2: What is the model of 'Activity' built into the software	
	For Learning Design 'Activity' is one of the important concepts so it may be useful to find out more about what an activity involves in any given piece of software
3: What is the Workflow model built into the software	
	Similarly workflow is an important concept. So it is important to find out about the model of workflow. Is it simple sequencing or are more complex workflows possible?
4: What are the UI characteristics of the software	
	A key aspect of the usability of the software by different groups is the nature of the user interface.
5: For run-time environments - how interactive is the design once it is underway?	
	What can learners actually do within the environment? What can teachers do?

Technical Characteristics	
1: What form is the software in (web-based, stand-alone app etc?)	
	What sort of software is it? Will it run on a variety of platforms etc?
2: What are the technical requirements to run the software?	
	Any other technical requirements or additional software required to run the application

Using this question set as a guide, it is possible to gain a sufficiently rich picture of a software tool to evaluate the nature of its contribution to supporting learning design.

As part of the research for this report, we have conducted a review of the software applications identified in section 1.5 as being related to both the concept of learning design and the IMS-LD specification. Those tools are: LAMS, Reload, Coppercore, EduPlone and EduBox. We have also looked at the content authoring environment – Lobster.

2. A Review of Software Tools to Support Learning Design

At the present time the most commonly used software for supporting teaching and learning interactions in e-learning is the VLE or LMS. VLEs provide a means for building courses, managing roles and groups and building in the various 'services' as defined in IMS-LD such as conferencing, chat etc. However, as we mentioned earlier the course structuring functionality in VLEs typically means structuring content; the activity management capabilities in VLEs to date have been very limited, although some of the major vendors are beginning to respond to the community's demand for better activity management capabilities, (Britain and Liber, 2004).

During this period, a plethora of dedicated content-authoring systems along with SCORM and the IMS content-packaging and metadata specifications have appeared. By comparison the activity management side has been very slow to get off the ground and for the obvious reason that manipulating digital content in a web environment is a much simpler technical proposition than modelling activities and workflow.

Possibly the most significant software development to date in the area of activity management within elearning and the one that has helped kick-start the current high level of interest in learning design is the Learning Activity Management System (LAMS) created by James Dalziel of MacQuarie University in Sydney and WebMCQ Ltd. Although LAMS does not implement the IMS-LD specification, it does embody the core ideas behind the specification in terms of a focus on creating sequences of activities, rather than content. LAMS also acts as the run-time environment for LAMS activity sequences. One of the highly attractive features of LAMS is that it provides a simple and highly intuitive user interface that allows the course designer to drag and drop LAMS activity tools into the workspace and use connecting arrows to organise the activities into a sequential workflow.

One problem with LAMS with regard to the goals of Learning Design is that LAMS sequences cannot be exported for use or re-use in other environments. LAMS sequences can only be run within LAMS. On the positive side, the fact that LAMS acts as both the authoring environment for activity sequences and the runtime player means that the LAMS software is capable of more sophisticated run-time functionality (e.g. real-time monitoring of sequences by the teacher) than would currently be possible if the activity sequence was transferred to another environment.

The eventual aim of the IMS-LD spec is that it should be possible to achieve powerful runtime behaviour based on a transferable xml description of a learning design but both the specification and the tools to support it are still at a relatively immature stage of development.

As of the time of writing this report there are no tools available for end-users that support both the creation and running of an IMS Learning Design at any level. However there are a number of significant recent developments in the field.

The OUNL (the developers of EML) created a tool called EDUBOX, which was originally designed as an EML player but has since been adapted to act as a run-time environment for IMS-LD. There has been a recent announcement by Blackboard of a strategic alliance with Edubox to allow Learning Designs to run in the Blackboard VLE

The OUNL has also recently announced the release of CopperCore, http://coppercore.org which is a runtime engine that is designed to allow software developers to incorporate IMS-LD into their applications. Coppercore provides 3 API's which cover the publication, administration and delivery of IMS Learning Design. This is potentially a highly useful development that will allow VLE vendors to build learning design into their products although it does not immediately benefit users. There are no known examples of Coppercore implementations as yet.

The Reload project www.reload.ac.uk sponsored by the JISC X4L programme is currently building both a Learning Design editor and a run-time environment for learning designs.

Reload have already produced a successful content packaging and metadata editing environment. The learning design editor will become part of this suite of tools. Reload are planning to trial Coppercore as the engine underlying their run-time environment.

In the next section of this report we provide a more detailed evaluation and review of the progress of development and the functionality of the tools mentioned above.

In addition to LAMS there a variety of other software tools that fit broadly into the field of ‘designing for learning’ but do not implement IMS-LD. Many of these tools are intended to support the creation and management of specific learning activities such as concept-mapping (e.g. Mind-Map) or role-play scenarios (e.g. Kartouche) and so on. Finally there is a question over whether content-authoring environments are relevant to a discussion of the concept of learning design. As part of this review, we consider Lobster which is a content-authoring tool currently being offered free to colleges of further education.

ReLoad

www.reload.ac.uk

Reload is a project funded by JISC as part of the X4L programme developing tools to support the learning technology interoperability specifications such as IMS and SCORM. The Reload project has already produced a successful metadata and content-packaging editor tool. Development is currently underway to add the Learning Design specification to the Reload editor tool. This will require research into an appropriate user interface model to allow easy creation of learning designs. In addition the Reload developers are working on a Run-time environment for Learning Design. They are currently evaluating Coppercore (also reviewed here) as the engine to underpin the run-time environment.

The Reload software is in an early stage of development with respect to implementation of Learning Design. However the work that has already gone into the development of the content-packaging editor means that progress should be relatively rapid.

Intended Purpose	
1: Description of intended purpose	
	An extension to the existing content packaging and metadata editor that will allow the creation and editing of learning designs in IMS-LD format. In addition a run-time environment is planned that will allow the playing of IMS-LD Learning Designs
2: Who is the system for? Who else is involved?	
	The current editor is for content developers who have some level of technical knowledge. E.g. they must know what a 'content-package' is. Efforts are being made to make the Learning Design editor interface as friendly as possible but it will still probably require some level of technical knowledge to edit the various fields correctly.
3: Perspective or world-view of the designers	
	The Reload team are closely linked to CETIS and to the IMS Learning Design Working Group, so their background and perspective is closely allied to that of IMS-LD. The aim of reload is to produce tools that make the specifications easily usable to support a wide range of pedagogies and to support multi-learner blended learning situations. The software will be available free on an open-source license from Sourceforge.
4: Scope	
	Development of both an editor and player is planned.
6: Integration and outputs	
	The reload editor is a stand-alone Java application. It will output data in XML format that conforms to IMS-LD. Details of the player are as yet unknown but it maybe based on Coppercore
7: Does the software implement IMS-LD? (if so what level: A, B or C?)	
	Not yet, but Level A implementation is planned for a release in July 2004 and Levels B and C following that

Design Characteristics	
1: What are the main concepts or entities built into the software?	
	The Reload editor will contain all the main entities of the IMS-LD specification: Learning objectives, activities, activity-structures, roles, resources, method, play, acts, role-parts etc.
2: What is the model of 'Activity' built into the software	
	The activity model in Reload will be the IMS-LD model
3: What is the Workflow model built into the software	

	The workflow model will be the IMS-LD model of Plays, Acts and activities which are associated with roles
4: What are the UI characteristics of the software	
	The User Interface is yet to be decided but is likely to involve creating a new learning design and then editing the various data entry fields that correspond to the required data for creating an IMS-LD learning design
5: For run-time environments - how interactive is the design once it is underway?	
	Unknown as yet.

Technical Characteristics	
1: What form is the software in (web-based, stand-alone app etc?)	
	The editor is a java application.
2: What are the technical requirements to run the software?	
	Java run-time environment.

Coppercore

http://coppercore.org

Coppercore is a J2EE runtime engine that has been developed by the OUNL as part of the EU Alfanet project (Kraan, 2004a). Coppercore is the first software to implement the IMS-LD specification. Unfortunately no tools have yet been built that use Coppercore, apart from the test environment that comes with it, although that will not remain the case for very long.

Intended Purpose	
1: Description of intended purpose	
	A J2EE runtime engine to allow developers to incorporate IMS Learning Design into their applications. Coppercore provides three API's which cover publication, administration and delivery of IMS Learning Design.
2: Who is the system for? Who else is involved?	
	The main target audience is software developers who will plug the runtime engine into their software and in the future, elearning systems integrators will be able to take advantage of coppoercore to add Learning Design capability to their VLE or other learning platform through its planned webservices (SOAP) interface
3: Perspective or world-view of the designers	
	The OUNL contributed greatly to the design of IMS-LD through their experience in developing EML. Since Coppercore has come from the same stable then it is likely to become an IMS-LD reference implementation, if it proves to be useful. Coppercore has been developed using component technology (Enterprise Java Beans) aand is open source, so this makes it an attractive development in an elearning context which is moving increasingly towards component-based frameworks.
4: Scope	
	Coppercore consists of 3 sets of API's, some UI's for testing purposes and a validation library. The main API is LDEngine which covers run-time behaviour. There is also a CourseManager API that covers users, roles etc and a Timer API.
6: Integration and outputs	
	Coppercore is designed to integrate with other software as a component. As a run-time engine Coppercore will take a Learning Design encoded as an IMS-LD XML document and then ensure that the environment it is running in behaves according to the specification of the design.
7: Does the software implement IMS-LD? (if so what level: A, B or C?)	
	Coppercore currently implements Level A. Levels B and C are planned for the near future

Design Characteristics	
1: What are the main concepts or entities built into the software?	
	As in IMS-LD
2: What is the model of 'Activity' built into the software	
	As in Ims-LD
3: What is the Workflow model built into the software	
	As in IMS-LD
4: What are the UI characteristics of the software	
	There is no UI (except the test UI's provided with the software) just API's (Application Programmer Interface)
5: For run-time environments - how interactive is the design once it is underway?	

	It should be able to handle the full range of interactions that IMS-LD is intended to support. What that amounts to in practice remains to be seen.

Technical Characteristics	
1: What form is the software in (web-based, stand-alone app etc?)	
	J2EE application based on Enterprise JavaBeans
2: What are the technical requirements to run the software?	
	A relational database for storage and a J2EE application server

Edubox

Edubox was originally built by the OUNL as a learning management system that implemented EML and was used on a variety of OUNL courses. After a quiet period, Edubox has now re-emerged with a recent announcement of a new strategic alliance between the OUNL and Blackboard to use Edubox to introduce learning design and more sophisticated activity management into their course management system. Edubox is currently being re-engineered to support IMS Learning Design and will be available later in 2004.

Intended Purpose	
1: Description of intended purpose	
	In its EML form, Edubox was a student run-time environment, the IMS-LD version will be both an authoring tool and a run-time environment
2: Who is the system for? Who else is involved?	
	Edubox is intended for use by teachers and learners
3: Perspective or world-view of the designers	
	The OUNL team that created Edubox has been instrumental in the development of IMS-LD. It is also the same team that have been responsible for the development of Copercore. Whereas that product is open-source, Edubox will be a commercial offering, although details of prices and licensing have not yet been finalised.
4: Scope	
	Both authoring environment and run-time environment
6: Integration and outputs	
	The integration between Edubox and Blackboard will be a full single sign-on integration.
7: Does the software implement IMS-LD? (if so what level: A, B or C?)	
	Level A initially, Levels B and C to follow.

Design Characteristics	
1: What are the main concepts or entities built into the software?	
	As for EML / IMS - LD
2: What is the model of 'Activity' built into the software	
	As for EML / IMS - LD
3: What is the Workflow model built into the software	
	As for EML / IMS - LD
4: What are the UI characteristics of the software	
	A web-browser based interface with a tree viewer pane for navigation and a content / activity viewer pane.
5: For run-time environments - how interactive is the design once it is underway?	
	- There is a limited content update possibility, but if an author wants changes that affect the basic structure of the course, the course should be published again. - Changes in style in the runtime environment are possible if the author has assigned this possibility when publishing the course. It is possible to decide when publishing which stylesheets users (students) may choose. The students can change the style in the runtime environment. For example change from an English environment to a Dutch environment.

Technical Characteristics	
1: What form is the software in (web-based, stand-alone app etc?)	
	Browser for the client, the server needs to run the web player app.
2: What are the technical requirements to run the software?	
	Information from the OUNL as follows: “we use an AIX-machine (RS/6000) that runs AIX 5.2, with WebSphere 4.0.1, an Oracle 7 Database en an IBM HTTP Webserver. Clients may choose to place a second AIX-machine (RS/6000) with IBM HTTP Webserver outside of the firewall for security reasons. We use as machines an IBM P-Series 650 en an IBM C20, but depending on the intensity of use you may install heavier or lighter machines”

Learning Activity Management System (LAMS)

www.lamsinternational.com

LAMS represents the most comprehensive implementation of the concept of Learning Design available to date. Although it is still quite limited in its functionality and flexibility (for example sequences cannot be created or adapted easily on the fly), it includes innovative design features that put it at the forefront of current tools for activity management. LAMS was reviewed in-depth as part of this author's report on elearning environments earlier this year (Britain and Liber, 2004).

LAMS has already made a significant contribution to this area by demonstrating how software tools can support different models of e-learning practice than the ones that we have become accustomed to expect from commercial VLEs. It will be interesting to see how LAMS develops over the next 12 months and how the commercial VLE players and open-source tools respond to the challenge laid down by LAMS. What LAMS does not do is create exportable Learning Designs that can be run in other environments. But, as yet neither has anything else, and it is only when IMS-LD begins to be used in earnest that the extent of technical difficulties associated with accommodating the nuances of external environments and tools will begin to be realised.

Intended Purpose	
1: Description of intended purpose	
	LAMS is an online web-based system for creating, managing and delivering sequences of collaborative learning activities. The visual authoring environment is designed to be easy to use by non-technical teaching staff and the run-time features allow real-time monitoring of the performance of learners.
2: Who is the system for? Who else is involved?	
	It is designed for use by teachers and students of any level of technical expertise
3: Perspective or world-view of the designers	
	The designer of LAMS – James Dalziel is highly interested in making e-learning work with a range of pedagogical approaches. His aim is to further the creation of innovative software tools that will support a variety of pedagogies. LAMS already has some original activity construction tools built into it and the aim is to create more. James has also been part of the Learning Design Working Group and so is well acquainted with the aims of Learning Design and also the technical difficulties of achieving those aims in a shareable, re-usable form, particularly when the characteristics of the software running a design are unknown. For this reason LAMS has implemented the concept of Learning Design but not the specification, consequently the designers refer to the software as inspired by Learning Design, rather than an implementation of the current specification
4: Scope	
	LAMS constitutes both an authoring environment and a run-time management and delivery environment
6: Integration and outputs	
	LAMS runs online via a web-server. At the present time LAMS is proprietary software owned by WebMCQ ltd. As a matter of design as explained above LAMS sequences can only be run within the LAMS environment.
7: Does the software implement IMS-LD? (if so what level: A, B or C?)	
	No

Design Characteristics
1: What are the main concepts or entities built into the software?

	Activities – both individual and collaborative or group-based *Sequences* – Sequential Workflow design *Groups* – Students can be aggregated into groups and sub groups for the performance of actvities
2: What is the model of 'Activity' built into the software	
	The activities are one of the main innovative features of LAMS as the designers have made an effort to create activities that are more oriented to teaching and learning than generic collaborative tools such as chat and conferencing. E.g there is a 'chat and scribe' tool; a polling tool; a Q&A tool. There are plans to develop a tools API that will allow a greater variety of activity tools to be developed.
3: What is the Workflow model built into the software	
	The workflow model is based on sequencing of activities with stop points to allow control of run-time behaviour.
4: What are the UI characteristics of the software	
	The visual drag n' drop interface for sequencing of activities is a great improvement over tabular 'field-editing' UI's in terms of ease of use for non-technical users.
5: For run-time environments - how interactive is the design once it is underway?	
	A high-level of interactivity is possible in LAMS including rich real-time monitioring. Because the sequences produced in LAMS are designed to run in LAMS, many of the technical problems facing IMS-LD designs about peculiarities of external run-time environments and external tools about which the authoring environment is ignorant are avoided.

Technical Characteristics	
1: What form is the software in (web-based, stand-alone app etc?)	
	LAMS is a web application that runs through a standard browser
2: What are the technical requirements to run the software?	
	The browser must be capable of supporting Flash.

EduPlone LearningSequence

http://eduplone.net

EduPlone is a learning content management system built on the open-source content management system - Plone, combined with the Web application server and development environment - Zope. Because Plone has a built-in workflow engine that can handle roles, activities and sequences, it has allowed the development of the LearningSequence product which provides initial support for Learning Design Level A (Kraan, 2004b). According to the CETIS article by Wilbert Kraan, the capability for rich collaborative designs is limited by the nature of the workflow in Plone which is much more designed around that of a single person's activity.

Intended Purpose	
1: Description of intended purpose	
	The aim of the LearningSequence product is to build on the Plone workflow engine to provide basic Learning Design functionality within the EduPlone environment.
2: Who is the system for? Who else is involved?	
	Like all Zope products the LearningSequence product is available either for use in EduPlone by end-users (teachers and learners) or to be adapted by software developers under the open-source license governing its use.
3: Perspective or world-view of the designers	
	EduPlone is built to serve an educational model they refer to as Webdidactics. The basic idea of webdidactics is individualised 'Knowledge Objects' to support the learner in directing their personal learning journey. Learning sequences can be created by a teacher to assist learners navigating their way through a topic
4: Scope	
	It is both an authoring and run-time environment.. Although the limitations of the Plone workflow engine mean that the scope for collaborative activity designs is limited at present. (see Kraan, 2004c)
6: Integration and outputs	
	Learning Sequences are exportable in IMS-LD format. The LearningSequence product as with all products built on Zope/Plone technology can be integrated into other systems built on Zope or Plone.
7: Does the software implement IMS-LD? (if so what level: A, B or C?)	
	Yes it implements basic support for Level A Learning Design

Design Characteristics	
1: What are the main concepts or entities built into the software?	
	Activities which are essentially learning objects consisting of content + metadata arranged into *Sequences* as a *workflow*
2: What is the model of 'Activity' built into the software	
	An activity is a sequence of knowledge objects created by a teacher for a particular learner.
3: What is the Workflow model built into the software	
	The workflow capability lies in being able to produce a sequence outline.
4: What are the UI characteristics of the software	
	Not assessed.
5: For run-time environments - how interactive is the design once it is underway?	

Technical Characteristics	
1: What form is the software in (web-based, stand-alone app etc?)	
	Web-based as for Zope/Plone
2: What are the technical requirements to run the software?	
	Not assessed

Lobster

www.lobster-online.co.uk

Lobster is an easy to use content authoring and structuring tool that has the added benefit of being an online environment that allows for collaborative authoring using a consistent database of assets.

As a content structuring tool, it is a point of argument as to whether Lobster is a learning design tool. Lobster allows creation of content that could specify an activity although the only activity-creation tools included with Lobster are question and test tools. Lobster allows the construction of a sequence of content. If each page of content, contains a task for learners to perform then, is this equivalent to a Learning Workflow?

Conceptually, Lobster raises the issue of how far the rich pedagogical metadata of LOM actually comprehends the notion of 'activity' within the notion of 'object'. E.g. it has been argued that a learning object by definition has planed outcomes, which suggests that it too must be activity-based, even if that activity is expressed at the level of structuring of content and requests-for-content. What does the notion of activity in LD add, and how is it intrinsically different?

The main difference between Lobster (and other content authoring tools) and what may be expected of a learning design tool is that Lobster has no notion of 'People' (roles or groups) built into it, so there is no sense in which activities expressed as interactions between people can form part of the design.

Intended Purpose	
1: Description of intended purpose	
	Lobster is an online content / course authoring tool. It allows the non-technical educational practitioner to build and sequence learning objects using pre-defined templates. The structured content that has been authored in Lobster can then be accessed as either a stand-alone web-based course with its own navigation or incorporated into a course developed within a VLE.
2: Who is the system for? Who else is involved?	
	The system is designed for use by non-technical educational practitioners who don't have sufficient technical confidence to use a web development package such as FrontPage or Dreamweaver and who don't have a knowledge of instructional design principles.
3: Perspective or world-view of the designers	
	The developers of Lobster come from a commercial training / elearning background. Their view is that Lobster makes a valuable contribution to content development for elearning contexts within the educational sector because the tool is both online and easy to use by non-technical practitioners. Their view of design for learning is very much based around a structured content and mcq assessment model.
4: Scope	
	The software tool is an authoring environment not a player. Sequences of learning objects authored in Lobster can either be run independently through a web browser or could be used in conjunction with a VLE. However it should be noted that there is no integration between the Lobster content and activities within the VLE. From the learning design perpective, Lobster is simply a content-authoring environment and so is ignorant of the groups, activities, roles and environment that the content is used to support.
6: Integration and outputs	
	Lobster projects are published as either a zip file or an IMS content-package and then can be imported into a VLE if desired.
7: Does the software implement IMS-LD? (if so what level: A, B or C?)	
	No

Design Characteristics	
1: What are the main concepts or entities built into the software?	
	The main entity within Lobster is that of the Learning Object. The learning object in this case essentially consists of a wrapper around some free-format online content which may be an html page, a word or powerpoint document or a flash animation. Anything that can be rendered in a web browser can be included as the content. The Lobster authoring environment provides preformed templates for the creation of learning objects which allow the content to be 'wrapped' with images and text to contextualise the content.
2: What is the model of 'Activity' built into the software	
	There is no explicit model of activity within Lobster. Activities in various forms can be presented to learners as part of content that is created or imported into Lobster as in any browser-based system. Lobster does have seven different templates for question and test designs which can form part of a learning object (structure). Question pages can be typed as 'scored' for formal assessment or 'unscored' for learner self-testing.
3: What is the Workflow model built into the software	
	Sequencing of information screens to form learning objects
4: What are the UI characteristics of the software	
	The user works through a web-browser to load assets into a repository within Lobster. The assets can then be assigned to information screens using the lobster templates. All the templates have the same general UI layout which has content appearing in a main window surrounded by optional images and text in the left-hand border and a tools menu in the
5: For run-time environments - how interactive is the design once it is underway?	
	A Lobster project can be adapted at any point, but if used in conjunction with a VLE it would then have to be imported into the VLE again.

Technical Characteristics	
1: What form is the software in (web-based, stand-alone app etc?)	
	Lobster runs through a web-browser. It can be hosted either locally or using the service of the training foundation
2: What are the technical requirements to run the software?	
	Internet Explorer 5.5 on PC

3. Discussion and Conclusions

The outcomes of the review of software tools conducted in this study are summarised in the table in appendix 1.

The main conclusion to be drawn from this review is that software development in this field is still at an immature stage although there are several exciting strands of development in progress. This means that whilst some software has been completed and other products are soon to be completed, few of the systems reviewed here have been widely used in practice as yet.

A significant step forward in the widespread visibility of learning design is the Blackboard / Edubox announcement of a strategic alliance. It will be interesting to see how the other major commercial VLE vendors respond to Learning Design in the light of this announcement.

Most of the tools we have looked at here aim to implement the IMS-LD specification at some level. By contrast LAMS has been developed to implement the concept of learning design but is not intended to implement the specification in its current form. There are other software tools in existence, which we have not looked at in detail here, that support aspects of the process of learning design but have not been specifically developed with 'learning design' in mind.

It will be interesting to see how IMS-LD works in practice and just how re-usable designs turn out to be, given dependence on unknown environments and services. At the moment the only collaborative activity tools (services) defined by the IMS-LD are send-mail and conferencing. A wider variety of tools need to be developed and it is up to the practitioner community to both demand pedagogically useful tools and to be involved in their development.

These points highlight the influence of both top-down and bottom-up approaches to software development in learning design. The IMS-LD specification (and its predecessor EML) are examples of top-down attempts to specify in advance a framework for capturing all the salient information about a learning and teaching situation. However, as some software designers have noted, to write software to conform to the specification means conforming to one particular viewpoint on what is required from tools for learning design, which may hamper the creativity of the software designer. Meanwhile the bottom-up influence of creative software development should continue to help shape and refine the specifications.

Additionally, experience with other IMS interoperability specifications such as the Enterprise specification has shown that coordination and agreement between implementers has been a key factor in achieving interoperability by resolving inevitable differences in the interpretation of the semantics of fields. This is likely to be even more the case with a specification as complex as the Learning Design specification.

What is clearly missing at the present time is feedback from communities of practitioners on both the specification and the wider concept of learning design. A very worthwhile recent development in this area is the European Framework 6, UNFOLD, project. Clearly the Elearning and Pedagogy programme should establish links with this project.

In this report we have followed both Dalziel (in press) and Beetham (2004) in drawing a distinction between software tools that support the implementation of the IMS Learning Design Specification, which constitute the majority of tools we have examined here, and tools that support the more general process of learning design or 'designing for learning'. Of the tools we have looked at here LAMS exemplifies this category. Further work is required to look more closely at the contribution of these tools and at the capabilities of LAMS in particular.

In this report we have elaborated on a distinction that is currently becoming popular between 'Learning Activities' and 'Learning Objects'. The notion of learning objects has increasingly become linked with single-learner, content-delivery instructional approaches within e-learning, whilst the notion of ‘learning activities’ appeals to those designing for a more interactive, multi-learner context. The general concepts of learning activities and learning design are indeed attractive. However, for them to be computationally useful they need to be grounded in a specification. IMS-LD provides one way to do this, by stating in precise terms what it is that an activity is assumed to involve and how it relates to content and other components. Any alternative formulation of these ideas whether in the form of software, specification or description should also aim to be precise and grounded as this enhances understanding whereas vague descriptions and allusions to ideas that are not actually implemented only serve to cause confusion. This report represents an attempt to provide a clear summary of the current state of the field of learning design.

4. Recommendations

The following recommendations arise from the work conducted in this report. This report was funded as part of the e-learning and pedagogy programme and the principle aim of these recommendations is to assist JISC in directing the further activities of this programme related to learning design:

4.1 In this report we have pursued the idea that the concept of learning design can be usefully distinguished from the implementational level. That is, the embodiment of the concept in the IMS-LD specification, EML or any other specification. Further work needs to be conducted to examine the range of approaches to ‘designing for learning’ in use by teachers and lecturers and the software tools that are or could be used to support these activities.

4.2 LAMS is one of the most interesting tools at the present time as it is the first software tool to persuasively demonstrate the concept of learning design in practice. Consequently further work should be conducted examining the benefits and limitations of the LAMS software in UK HE and FE contexts.

4.3 Conduct ongoing work with the IMS-LD authoring tools and runtime environments reviewed in this report as they become available, to examine how well the IMS-LD specification caters for the pedagogical needs of teachers and learners. This work may be expected to feedback into both the ongoing development of the specification and the development of tools to support learning design. Key bodies in conducting this work are CETIS and the UNFOLD project mentioned above. It is recommended that this programme works in collaboration with both of these groups where possible.

References

Beetham, H. (2004) Review: Developing E-learning models for the JISC Practitioner Communities.

Britain, S. and Liber, O. (2004) A Framework for the Pedagogical Evaluation of E-learning Environments.

Dalziel (in press) The Development of the Learning Activity Management System (LAMS).

Koper E.R.J. (2001). Modelling Units of Study from a Pedagogical Perspective: the pedagogical meta model behind EML (http://eml.ou.nl/introduction/docs/ped-metamodel.pdf)

Kraan, W. (2004a) Coppercore to power Learning Design implementations. http://www.cetis.ac.uk/content2/20040126154220

Kraan, W. (2004b) Eduplone reveals initial IMS Learning Design support http://www.cetis.ac.uk/content2/20040415020953

Appendix 1: Summary Table of Software Systems reviewed

	Coppercore	**EduBox**	**Eduplone**	**LAMS**	**Lobster**	**Reload Software**
Description / Purpose / Scope	A runtime engine to allow developers to incorporate IMS-LD into their applications.	EML authoring and run-time environment IMS-LD version currently in development	A Zope/Plone based System that implements basic support for learning activity sequences using IMS-LD	Both an authoring and a runtime environment for learning activity sequences consisting of LAMS learning activity tools.	An Authoring environment for sequences of learning objects	A learning design editor, implementing IMS-LD and a player are currently in development.
Who is it for	Application Developers	Teachers and Learners	Teachers	Teachers and Learners	Teachers	The editor is for teachers with knowledge of IMS-LD
Activity management and Workflow	It provides 3 API's that cover publication, administration and delivery of IMS-LD as well as a test environment	The player runs EML sequences.	Sequences of knowledge objects	Sequences of LAMS activity tools with real-time monitoring and interactive behaviour	Sequences of content	The editor will allow the creation of learning designs in IMS-LD format. The player will run IMS-LD learning designs
Sharing and Reuse	Currently IMS-LD level A. Future versions will implement B and C	IMS-LD version in development.	Basic IMS-LD (level A) support.	Export of sequences outside LAMS is not supported	Content Packaging. SCORM.	IMS-LD level A planned for July 2004.
User Interface	None	Tree navigation plus content view pane	unknown	Various. Sequence construction using box and arrows	Tree navigation plus content view pane	Tabs and editing fields
Integration and Licensing	Open Source can be integrated at a software level using APIs	Commercial proprietary software. Integration with Blackboard using single-sign on	Open Source Plone / Zope product	Commercial proprietary software	Commercial proprietary software	Open Source software