Koper, R., & Olivier, B. (2004). Representing the Learning Design of Units of Learning. *Educational Technology & Society*, 7 (3), 97-111.

Representing the Learning Design of Units of Learning

Rob Koper
Open University of the Netherlands
Educational Technology Expertise Centre (OTEC)
Valkenburgerweg 177, PO Box 2960
6401 DL Heerlen
The Netherlands
rob.koper@ou.nl
http://www.learningnetworks.org

Bill Olivier
Bolton Institute
Centre for Educational Technology Interoperability Standards (CETIS)
Deane St.
Bolton, BL3 5AB
United Kingdom
b.olivier@bolton.ac.uk
http://www.cetis.ac.uk

ABSTRACT

In order to capture current educational practices in eLearning courses, more advanced 'learning design' capabilities are needed than are provided by the open eLearning specifications hitherto available. Specifically, these fall short in terms of multi-role workflows, collaborative peer-interaction, personalization and support for learning services. We present a new specification that both extends and integrates current specifications to support the portable representation of units of learning (e.g. lessons, learning events) that have advanced learning designs. This is the Learning Design specification. It enables the creation of a complete, abstract and portable description of the pedagogical approach taken in a course, which can then be realized by a conforming system. It can model multi-role teaching-learning processes and supports personalization of learning routes. The underlying generic pedagogical modelling language has been translated into a specification (a standard developed and agreed upon by domain and industry experts) that was developed in the context of IMS, one of the major bodies involved in the development of interoperability specifications in the field of eLearning. The IMS Learning Design specification is discussed in this article in the context of its current status, its limitations and its future development.

Keywords

Learning Design, Learning Objects, Educational Modelling, Open Specifications & Standards

Introduction

It is said that while open specifications (or standards) make interoperability possible, they come at the price of limiting options. Current eLearning specifications appear largely limited to page turning and single learner tracking. This is in stark contrast to the wide range of experimental systems developed to date which provide courses with more advanced pedagogical approaches, but lack portability. The challenge is to develop a specification that lifts these limitations while maintaining interoperability.

Most current open eLearning specifications and platforms available for course development and presentation can only represent courses that are restricted to a certain type of pedagogy that can be summarized as: in order to learn, a single learner has to work through a sequence of learning objects. The underlying assumption is that learning is a process of consuming content. Teaching is envisioned as the art of selecting and offering content in a structured, sequenced way, and of tracking the learner's progress and assessing the acquired knowledge. Current educational practice is more complex and advanced than this. In most settings multiple interacting roles are involved: tutors interact with the learners and learners with each other, possibly in different roles in the learning environment. Furthermore, social-constructivist pedagogical approaches have introduced different (active, learner-centred and community-centred) models and have strong arguments to reject the structured knowledge consumption approach. The current interoperability specifications have to be extended to include the multi-role interactions and the various pedagogical models that are needed to provide real support for learners and teachers in more advanced and newly developing educational practices.

We extended the current eLearning specifications by developing an additional layer to express the 'learning design' of courses in an interoperable and abstract way. We assume that every educational practice can be represented in a design description, i.e. that even for the most open course or workshop, there are underlying design ideas and principles that could be captured in an explicit representation. Furthermore, we assume that the design of any course is driven by 'pedagogical models' that capture the teacher's beliefs about good teaching and learning. More precisely, a pedagogical model is a set of rules that prescribe how a class of learners can achieve a class of learning objectives in a certain context or knowledge domain in the most effective way. Examples of pedagogical models are mastery learning, problem-based learning, active learning, or any notions of teachers about good teaching and learning. An example of a rule within a pedagogical model could be: "when teaching a new concept in higher education, start with an advanced organizer", or: "…, provide authentic problem descriptions to stimulate learners to search for solutions and study underlying concepts and facts".

A ‘learning design’ is defined here as an application of a pedagogical model for a specific learning objective, target group and a specific context or knowledge domain. The learning design specifies the teaching-learning process. More specifically, it specifies under which conditions, what activities have to be performed by learners and teachers to enable learners to attain the desired learning objectives. A learning design can refer to physical resources (learning objects and learning services) that are needed during the teaching and learning process. The learning design and the included physical resources can be packaged into a ‘unit of learning’ (UOL). A unit of learning can be seen as a general name for a course, a workshop, a lesson, etc that can be instantiated and reused many times for different persons and settings in an online environment.

There are hundreds of different pedagogical models described in the literature and new models continue to be formulated. Modelling each separate model, and then developing tools to support it, would be an inefficient path to follow. For this reason we aimed at the development of a more abstract notation that is sufficiently general to represent the common structures found in these different pedagogical models. With such a notation, learning designs for concrete units of learning can be specified that are applications of a specific pedagogical approach. A layered approach is followed: the notation is used to describe the learning design, and the learning design along with referenced physical resources are packaged in a unit of learning. The unit of learning can then be distributed and instantiated many times in many different eLearning runtime systems. This provides a powerful means of creating more effective, cost-efficient, flexible and advanced eLearning courses.

In this article we present and discuss the representational language for learning designs that we developed. It is currently available as the IMS Learning Design specification (LD). The detailed specification itself can be obtained from the IMS website (IMSLD, 2003). This article summarizes the analysis behind the specification, provides evaluative comments and indicates future directions.

Needs in education

Current needs and trends in education point to directions that are different from the ones reflected in the current eLearning specifications. Some of the major trends that are related to the requirements of learning technology specifications are (from Howell, Williams and Lindsay, 2003):

- Instruction is becoming more personalized: learner-centred, non-linear and self-directed.
- The distinction between face-to-face and distance education is disappearing through the use of eLearning. Courses that can be followed at a distance and blended distance and face-to-face approaches will be dominant in future.
- Lifelong learning is becoming a competitive necessity, resulting in a need for interoperable, networked learning (e.g. interoperable learning networks and portable learner dossiers).
- Academic emphasis is shifting from course completion to competency attainment.
- Traditional faculty roles are changing toward more specialized roles (course designer, tutor, etc).
- Faculty members demand decreased workloads, especially while working with learning management systems or online collaborative and conference environments. More automated support in the work process of faculty members is needed.

Based on a study of current pedagogical models, Merrill (2003) summarized them as follows: "… the most effective learning products or environments are those that are problem-centred and involve the student in four distinct phases of learning: (1) activation of prior experience, (2) demonstration of skill, (3) application of skill and (4) integration of these skills into real-world activities". Instead of transferring facts to learners, the major focus should be on the attainment of complex skills and competencies in authentic task situations (e.g. Van

Merriënboer, 1997). Merrill summarizes the underlying so-called 'first principles of instruction' as follows: Learning is promoted when:

- learners are engaged in solving real world problems.
- existing knowledge is activated as the foundation for new knowledge.
- new knowledge is demonstrated to the learner.
- new knowledge is applied by the learner.
- new knowledge is integrated into the learner's world.

The focus of Merrill's analysis and the pedagogical models he studied, takes the perspective of a single learner in a problem-situation. Other pedagogical developments add the notion of learning communities, communities of practice and collaboration (see e.g. Hooff, Elving, Meeuwsen and Dumoulin, 2003; Wenger, 1998; Retallick, Cocklin and Coombe, 1999). One of the current issues is the shift towards more social-constructivist approaches to learning (see: Brown, Collins and Duguid, 1989; Duffy and Cunningham, 1996). Effective education should be learner-centred, assessment-centred, knowledge-centred *and* community-centred (Bransford, Brown and Cocking, 2000). One of the underlying notions is that knowledge is not absolute, but is relative to the interpretation and beliefs of communities of practice. This social notion of knowledge means that facts, events, data and information can only be interpreted and acted upon when the social context is represented in the learning situation (e.g. Lave & Wenger, 1991).

These further educational requirements need to be reflected in the methods and instruments that support the learning and teaching process, including the design methods & tools, the runtime systems and the interoperability specifications. In this article we will not focus on the tooling, but mainly on the consequences of the new requirements for the interoperability specifications that capture the *design* of the learning and teaching process: the Learning Design (LD) specification.

Requirements for a learning design specification

The major requirement for the development of a LD specification is to provide a containment framework that uses and integrates existing specifications, and can represent the teaching-learning process (the LD) in a UOL, based on different pedagogical models – including the more advanced ones - in a formal way. More specifically, following the needs analysis provided above, a LD specification must meet the following specific requirements:

1. Completeness: The specification must be able to fully describe the teaching-learning process in a UOL, including references to the digital and non-digital learning objects and services needed during the process. This includes:
   - - Integration of the activities of both learners and staff members.
   - - Integration of resources (objects and services) used during learning.
   - - Support for both single and multiple user models of learning.
   - - Support for mixed mode (blended learning) as well as pure online learning.
2. *Pedagogical expressiveness*: The specification must be able to express the pedagogical meaning and functionality of the different data elements within the context of a LD. While it must be sufficiently flexible to describe LDs based on all kinds of pedagogies, it must avoid biasing designs towards any specific pedagogical approach.
3. *Personalization*: The specification must be able to describe personalization aspects within an LD, so that the content and activities within a UOL can be adapted based on the preferences, portfolio, pre-knowledge, educational needs and situational circumstances of users. In addition, it must allow the designer, when desired, to pass the control over the adaptation process to the learner, a staff member and/or the computer.
4. *Compatibility*: The specification must enable learning designs to use and effectively integrate other available standards and specifications where possible, such as the IMS (imsglobal.org) and IEEE LTSC (ltsc.ieee.org) specifications (see Olivier & Liber, 2003).

Because a LD specification should extend and integrate existing specifications, it must also inherit most of the more general requirements for interoperability specifications and standards:

5. *Reusability*: The specification must make it possible to identify, isolate, de-contextualize and exchange useful learning objects, and to re-use these in other contexts.
6. *Formalization*: The specification must provide a formal language for learning designs that can be processed automatically.
7. *Reproducibility*: The specification must enable a learning design to be abstracted in such a way that repeated execution, in different settings and with different persons, is possible.

The Learning Design specification

The LD specification, following common IMS practice, consists of: (a) a conceptual model that defines the basic concepts and relations in a LD, (b) an information model that describes the elements and attributes through which a LD can be specified in a precise way, and (c) a series of XML Schemas (XSD) in which the information model is implemented (the so-called 'binding') (d) a Best Practices and Implementation Guide (BPIG), (e) a binding document and example XML document instances that express a set of learning requirement scenarios. In the following sections we will focus on the conceptual analysis work that informed the Learning Design specification.

The conceptual model

Educational Modelling Language (EML, 2000; Hermans, Manderveld, and Vogten, 2004; Koper and Manderveld, in press) was selected as the base from which to develop the LD specification. The main changes made to the EML specification were:

- The EML Metadata model, based on Dublin Core (DC, 2003), was replaced by the IMS-IEEE LOM Metadata (IMSLOM, 2003).
- All EML content models were deleted, to make a clear distinction between the logic of the learning design and the referenced learning objects. EML uses a DOCBOOK (DOCB, 2003) content model. This was replaced by XHTML and some extensions (so-called 'global properties') that can be used within the context of XHTML through namespaces.
- The EML packaging structures were replaced by the IMS Content Packaging structures. As a consequence, all EML versioning mechanisms and the role-typing mechanism for resources were deleted.
- The EML testing elements were deleted and the IMS QTI allowed to be inserted in several places to replace these.
- Placeholders for other IMS specifications, such as IMS Simple Sequencing, were added to enable the Learning Design specification to serve as an integrative framework for other related IMS specifications in order to allow for the full modelling of UOLs.

However, the central core of the EML language was preserved in LD. Figure (1) provides an overview of the conceptual structure of the LD specification.

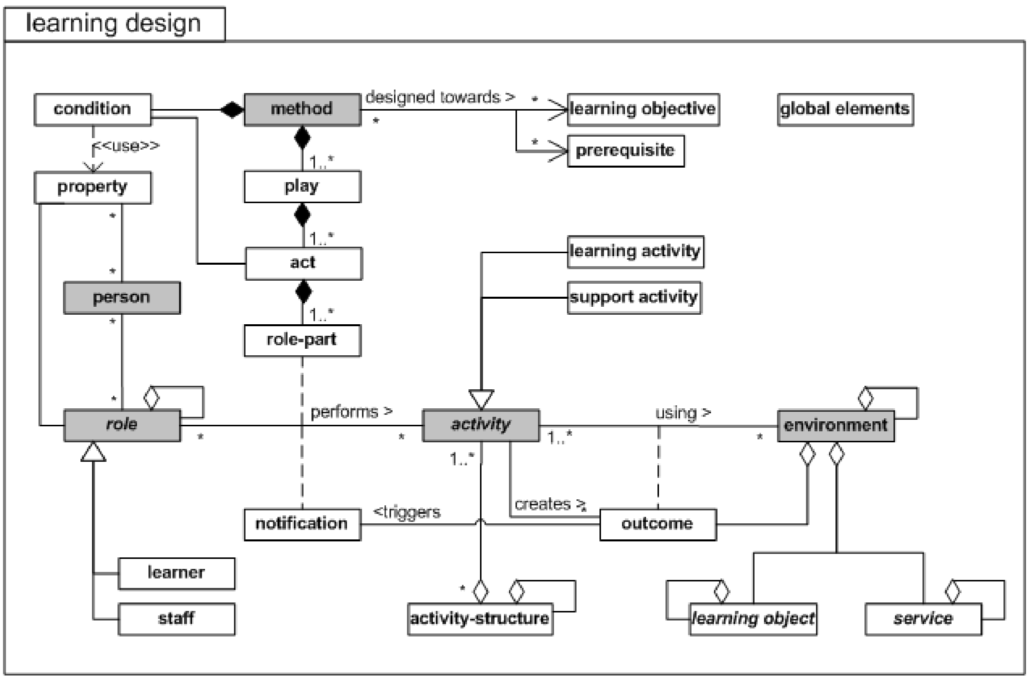

Figure 1: Conceptual structure of the LD specification

The core concept of LD, as expressed in figure 1, is that a learning design can be represented by using the following core concepts: A *person* gets a *role* in the teaching-learning process, typically a *learner* or a *staff* role.

In this role he or she works towards certain *outcomes* by performing *learning* and/or *support activities* within an *environment*. The environment consists of the appropriate *learning objects and services* to be used during the performance of the activities. Which role gets which activities at which moment in the process is determined by the LD *method,* or by a *notification* (both drawn as an association relationship with role – activity).

The LD *method* is designed to provide the coordination of roles, activities and associated environments that allows learners to meet *learning objectives* (specification of the outcomes for learners), given certain *prerequisites* (specification of the entry level for learners). This is the core part of the LD specification in which the teaching-learning process is specified. All the other concepts are referenced, directly or indirectly, from the method. The teaching-learning process is modelled using the metaphor of a theatrical play. A play has acts, and each act has one or more roles or parts. Acts follow each other in a sequence, although more complex sequencing behaviour can take place within an act. The roles within an act associate each role with an activity. The activity in turn describes what that role is to do and what environment is available to it within the act. In the analogy, the assigned activity is equivalent to the script for the part that the role plays in the act, although less prescriptive. Where there is more than one role within an act, these are 'on stage at the same time', i.e. they run in parallel. Thus a method consists of one or more concurrent *play*(*s*); a play consists of one or more sequential *act*(*s*); an act consists of one or more concurrent *role-part*(*s*), and each role-part associates exactly one role with one activity or activity-structure.

The *roles* specified are those of *learner* and *staff*. Each of these can be specialized into sub-roles. It is left open to the designer to name the roles or sub-roles and specify their activities. In simulations and games, for example, different learners can play different roles, each performing different activities in different environments.

Activities can be assembled into *activity structures*. An activity structure aggregates a set of related activities into a single structure, which can be associated with a role in a role-part. An activity-structure can model a sequence or a selection of activities. In a *sequence*, a role has to complete the different activities in the structure in the order provided. In a *selection*, a role may select a given number of activities from the set provided in the activity structure. This can, for instance, be used to model situations where learners have to complete two activities, which they may freely select from a collection of five activities contained in the activity structure. Activity structures can also reference other activity structures and external UOLs, enabling elaborate structures to be defined if required.

Environments contain the resources and references to resources needed to carry out an activity or a set of activities. An environment contains three basic entities: learning objects, learning services and sub-environments. Learning objects are any entities that are used in learning, e.g. web pages, articles, books, databases, software, and DVDs. The learning services specify the set-up of any service that is needed during learning, e.g. communication services, search services, monitoring services, and collaboration services. An example of set-up information is the specification of which LD roles have user rights in the learning service. This, for instance, enables automatic set-up of dedicated forums each time a LD is instantiated.

A method may contain *conditions*, i.e. If-Then-Else rules that further refine the assignment of activities and environment entities for persons and roles. Conditions may be used to personalize LDs for specific users. An example of such a personalization condition could be: "If the person has an exploratory learning style, Then provide an unordered set of all activities", or "If the person has prior knowledge on topic X, Then learning activity Y can be skipped".

The 'If' part of the condition uses Boolean expressions on the *properties* that are defined for persons and roles in the LD. Properties are containers that can store information about persons' roles and the UOL itself, e.g. user profiles, progression data (completion of activities), results of tests (e.g. prior knowledge, competencies, learning styles), or learning objects added during the teaching-learning process (e.g. reports, essays or new learning materials). Properties can be either global or local to the run of a unit of learning. Global properties are used to model portfolio information that can be accessed in any other unit of learning that is modelled with LD and has access to the same persistent storage for property data. Local properties are only accessible within the context of a specific run of a unit of learning and are used for temporary storage of data.

In order to enable users to set and view properties from content that is presented to them, so-called *global elements* are present in LD. These global elements are designed to be included in any content schema through namespaces. Content that includes these global elements is called '*imsldcontent*'. The preferred content schema is XHTML. Global elements can be included in the XHTML document instances to show (or set) the value of a property, for instance a table with progression data, a report added by a learner, a piece of text or URLs added by a teacher, etc.

LD also contains *notifications,* i.e. mechanisms to make new activities available for a role, based on certain *outcome* triggers. These outcomes are, for example, the change of a property value, the completion of an activity, or certain patterns in the user profiles. The person getting the notification is not necessarily the same person as the one who triggered the notification. For instance, when one learner completes an activity, then another learner or the teacher may be notified and set another activity as a consequence. This mechanism can be used to model adaptive task setting LDs, where the supply of a consequent activity may be dependent on the kind of outcome of previous activities. General pedagogical rules can also be implemented using the combination of conditions and notifications, e.g. "If a user has profile X, Then notify learning activity Y".

The information model and XML binding

The conceptual model is implemented as follows. A UOL is represented as a specific type of extended IMS Content Package (CP). It is extended by adding a LD element within the CP Organizations element (figure 2).

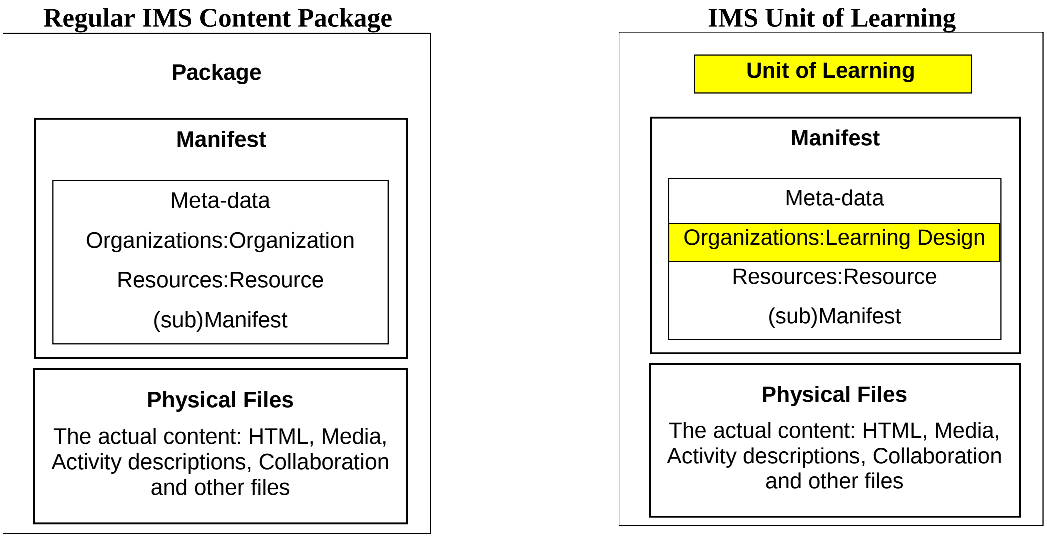

Figure 2: Comparison of a regular IMS CP and an IMS UOL

The LD element is itself a complex structure that includes elements that represent the conceptual model already outlined. The details of these elements are spelled out in the Information Model document, together with their behavioural specifications.

These elements are given an XML representation, or ‘binding’, provided as XML Schema that are designed to be included in an IMS CP XML structure.

The LD XML schema itself can be represented as a tree (figure 3).

The properties, activities and environments of the components element and the conditions of the method element all, in turn, have complex sub-structures but these are not shown here for the sake of simplicity.

A distinction is always made between the package (reflecting the UOL at the *class* level) and the run of that package (an *instance).* In creating instances from a package, some customization and localization may typically take place.

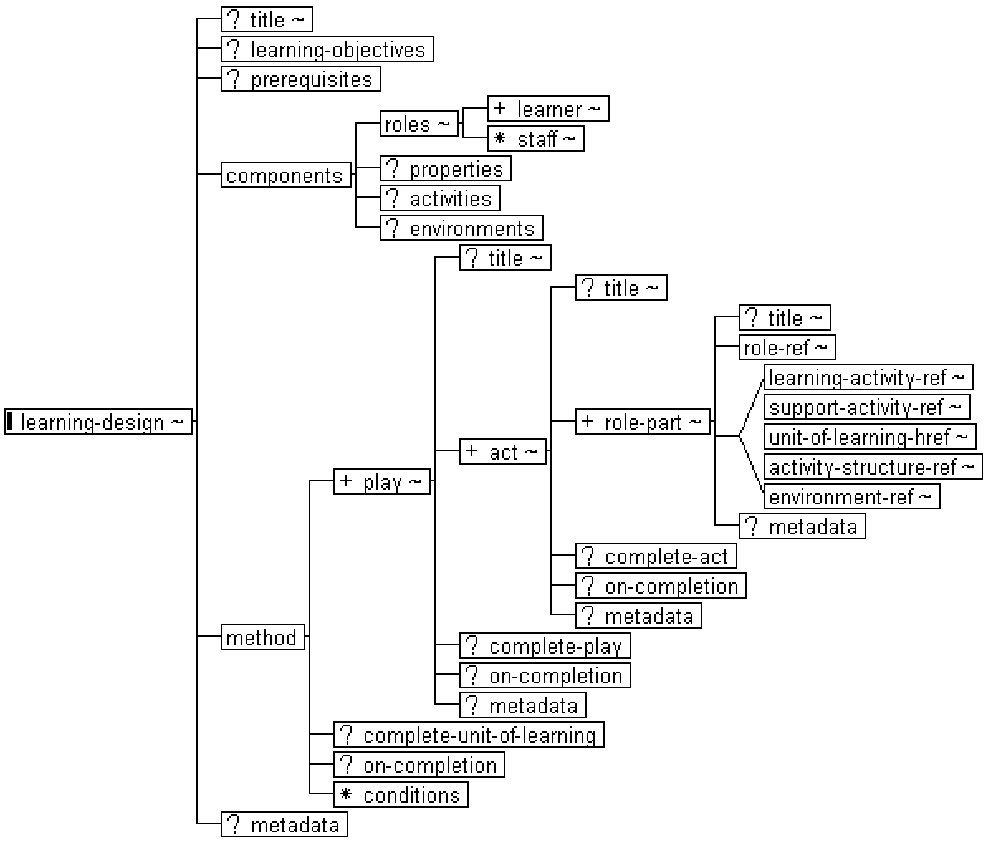

Figure 3: The LD schema represented as a tree.

A UOL package represents a fixed version of a UOL, with links to the underlying learning objects and service types. It may contain further XML document instances valid against the other appropriate schemas (IMS LD, IMS CP, IMS QTI, etc) along with the physical files that are referred to in a fixed version and URIs to other resources, including services. Such a package can be instantiated and run many times for different learners in different settings. If desired, it can also be adapted prior to instantiation in order to reflect local needs. This will create another version of the UOL and accordingly another

Evaluative comments

In the previous sections the requirements and the LD specification that is developed to meet these requirements were specified. In this section we will provide some evaluative comments about the fit of the LD specification to the requirements.

Because the IMS LD specification has only recently been released (Feb 2003) and because it is a relatively large specification, comprehensive, conforming implementations cannot be expected soon after its release. A proper evaluation of LD in its full scope will only become possible when supporting authoring, content management and runtime tools to create, store, share and interpret the LD document instances become available. However, there have been implementations of the Educational Modelling Language, EML, from which IMS LD was derived, and during the development of LD some validation activities were performed.

We will now evaluate the LD specification against each of the requirements outlined above.

Completeness

The specification describes the activities both of learners and staff in the teaching-learning process. It integrates resources and services within the environment in which activities are performed. It supports both a single and multiple user model of learning.
The EML specification included packaging, testing and content in the model, where LD references other specifications for these functions, like IMS Content Packaging, XHTML, global properties and IMS QTI. For the packaging and content part this delivers the same functionality, but the integration of QTI introduces additional problems. When interpreting a LD document instance, the LD interpreter must know that another specification is included to call another suitable interpreter. Another problem occurs when some events in the LD are dependent on test results. This requires integrating the property mechanisms of LD with QTI results. This work has now been started in IMS.

To test the requirement one can run a LD encoded course in several LD runtime systems. For any given UOL, each runtime system should deliver the same activities, with the same learning objects and services in the same order, with the same personalization facilities. This test can only be performed when different players become available. The internal tests at the Open University of the Netherlands (OUNL) with four different EML and LD runtime systems, provides some support for the assumption that this requirement has been met. Although each system implemented a different version of EML, which provides a bias, in all cases the conversion of the courses from the one EML version to the other was done automatically. A course created in EML and run in system X, was converted to a later EML version and run in system Y with the same functionality, but with a different user-interface (see for instance the interfaces discussed in: Hermans, Manderveld & Vogten, 2004; Paas & Firssova, 2004).

Pedagogical expressiveness

Literature study

One approach to testing the pedagogical expressiveness of the LD specification is by performing literature driven tests, i.e. studying the pedagogical models described in literature and testing whether learning designs based on these models can be represented in LD. Most of the groundwork of abstracting, testing and refining the core concepts, language and the XML Schema was carried out during the three year development of EML before LD was started. Models in literature have been studied (see Koper, 2001; Koper & Van Es, in press) in the three major streams of instructional theories and models (see Greeno, Collins & Resnick, 1996): empiricist (behaviorist), rationalist (cognitivist) and pragmatist-sociohistoric (social-constructivist & situationalist). Furthermore we studied different handbooks that summarize approaches (e.g. Gagné & Briggs, 1979; Reigeluth, 1983, 1999; Roblyer and Edwards, 2000), different websites with lesson plans (e.g. ERIC, 2003) and websites with pedagogical patterns (e.g. Ryder, 2003; Pedagogical Patterns Project, 2003). EML/LD was able to represent the UOLs described in literature and websites.

Check the core aspects of current pedagogical models

As discussed in the introduction, current views on pedagogical models can be summarized as learner, knowledge, assessment and community centered. We have looked into whether it is possible to represent these different aspects in LD.

LD can represent *learner centered* approaches, as it is possible to define learning designs where the learner is fully in control: e.g. learners can select activities and ask for support. Underlying such learner-centered approaches are designs that provide options, resources and access to support, and are created prior to the learners' involvement. In principle, however, it is also possible that a UOL, including the learning design, is designed specifically for and by learners themselves. This process can be supported by appropriate design tools and automated support services.

'Knowledge-centered' means two things according to Bransford *et al* (2000): a) the representation of information that is shared and acted upon by people in order to acquire new knowledge, and b) the attainment of the competencies that are required in certain communities of practice. LD can represent the former through learning objects within the environment and the latter through learning activities. In competency-based education, learners increase their level of competency of a given type by carrying out study tasks of increasing complexity,

starting from a baseline level. Learners work through a collection of study tasks until they have acquired the desired level of competence. The competence level itself can be specified in the objectives section. This is typically done by referencing a competency that is kept in an external competency map (as compared to normal learning objectives which are specified locally in the course itself).

Assessment is supported in several ways, depending on the type of assessment. Traditional multiple choice test items are supported through the inclusion of IMS QTI as discussed earlier. More advanced forms of assessment (e.g. portfolio-assessment and peer-assessment) are typically more integrated with the learning activities and can be modeled using LD in combination with its property mechanism. The properties can store, process and retrieve portfolio data. The notification mechanisms can support peer-review mechanisms by triggering certain persons to review work which has been produced.

Communities are supported in a variety of ways. First, it is possible to store the results of others' activities within the properties during the run of a UOL. These results can be viewed by new learners and provide a sense of community without direct peer-interaction. More direct interaction can be specified according to principles of collaborative learning where different learners jointly perform tasks or solve problems in a shared environment. Heterogeneous communities are supported by defining different roles (e.g. experts, novices), who interact with each other according to various learning design principles. LD is particularly strong in the modeling of multi-user, multi-role learning environments.

Evaluating learning scenarios

Another evaluation approach to test pedagogical expressiveness has been followed. In this approach a variety of courses which are desired or have been used in practice, are mapped into the XML language to see if it can adequately represent them. Dozens of UOLs of varying size and granularity, using different pedagogical approaches, have been encoded in EML and LD. During the development of LD, such a process was set up to validate its expressiveness. Members of IMS were asked to provide learning scenarios, or 'use cases' (i.e. narratives of a teaching-learning process) to test whether they could be expressed. Ten such use cases from universities and industry were submitted and presented in the Best Practices and Implementation Guide (BPIG, see IMSLD, 2003). These could all be represented without requiring changes in the LD model. Some concrete examples of the use cases that were sent in and represented in LD are discussed here. These varied from classical instructional design to more advanced approaches.

LD can support the modeling of *Role Play* learning. The roles of LD can be sub-typed, allowing a designer to specify any roles they need, with appropriate activities assigned. It also supports their coordination and sequencing at runtime. An example is provided in the BPIG (the Treaty of Versailles scenario) representing a complex, multi-player, multi-institutional role-play that lasts for several weeks and also includes project-based learning and blended (face-to-face and online) learning.

In *Learning by Doing* (activity-based learning), instructions are provided for carrying out a particular task. This task may be computer-based or it may be performed separately from a computer, with the computer being used to provide instructions, re-playable examples of how to perform the task, and separate skill sessions that can be undertaken before the main task is tackled.

In *problem-based learning*, a problem is set which will challenge learners to apply their existing understanding, knowledge and skills to finding solutions. In the case where more than one learner is involved, participants seek to clarify the problem, agree on an approach or method they will adopt, and share out the task in some way amongst themselves. The group may identify the learning goals of the problem and individuals then embark on their agreed tasks. The group then reconvenes to share their findings and/or their work. They evaluate these and on the basis of their evaluation they may draw up one or more solutions. These may be repeated and refined. Eventually they present the solutions. These may then be discussed with a teacher or facilitator and/or an assessment made of the group and/or individual contributions.

Programmed instruction was a very early form of computer-supported learning, developed to support Skinner's Analysis of Behaviour. In its basic form, the learner is provided with a sequence of simple tasks, whereby each task has a high chance of being completed successfully. When completed successfully, positive reinforcement is provided and the next task is presented. In variations, remedial tasks or adaptation is provided, using predefined conditions.

Literature Circles is a technique used to develop learners' discussion skills, based on discussing literary works they have read. Often arranged in small groups, the learners are given a book to read and assigned roles each with assigned tasks. For instance, a 'Discussion Director' will produce a list of questions, reactions, etc, while an 'Artful Artist' will produce a set of drawings and a 'Word Wizard' will supply definitions for any unfamiliar words. Other roles may be also be used. After reading the book and completing their preparatory tasks, they meet to discuss the book with the Discussion Director as Chair. On completion of this book, they are then assigned another book and the roles are rotated. This process is repeated with different books until each member has played each role.

Personalization

The design of personalization in LD is supported through the conditions and property mechanism. Personal characteristics can be measured and stored in properties. Conditions can be defined to adapt the learning design to the learner characteristics in runtime. This generic mechanism can support a variety of personalization examples:

- Adaptation of the learning design method, given a learner's needs or characteristics. In a Jazz course the OUNL developed in EML, the learning styles of learners were measured with a validated learning style test. On the basis of the test findings they were advised to follow a particular learning approach, either more exploratory or more structured. Users were only advised: they could change the learning approach themselves if they wished to do so.
- Sometimes students in different fields have to follow the same course (e.g. students in economics and social sciences may both have to take an introductory statistics course). In LD it is possible, within the same LD, to hide or show different examples for different student profiles.
- Some students want to study at their own pace and not in a group, while others want to have extensive support from peers or tutors. With LD it is possible to provide the same course with multiple support and group options, for example, doing the course in a self-study mode or in a group, and with or without tutoring.
- It is possible to provide extra, remedial activities or learning objects or examples for students with certain prior knowledge gaps.
- It is possible to have a different LD method for different phases in the learning process: when studying the UOL for the first time, the LD method is designed for learning; when coming back after first time completion, the LD method supports the repetition and later the LD method optimizes reviewing the learning content.
- Support staff can also benefit from personalization, e.g. some teachers only want to see the activities that they have to perform to support the students and others want to view this in the context of the learning activities of the learners.

In the EML practice of the OUNL, we see that conditions are used in almost every course design. In principle, a course can contain any number of personalization conditions. However, it is up to the designer to decide which personalization conditions it is prudent to include. In general, every personalization condition brings extra design and test work during course development. This is only worth the effort if the user gains or if the expected number of users is high enough. This should be further investigated in future.

Compatibility

LD acts as an integration framework for other specifications that are needed to specify an eLearning course. In addition to being designed to act as an add-in component to the IMS Content Packaging specification, it can use different metadata schemas like the IEEE Learning Object Metadata (LOM), or the Dublin Core. The generic mechanism is to include these schemas through the use of XML Namespaces. In principle this can be done anywhere in any XML document instance, but LD specifies specific placeholders for the prudent inclusion of these schemas. Every container element in LD has a placeholder for metadata. Testing schemas (e.g. IMSQTI, 2003) are included in the environment. IMS Simple Sequencing can be included in the environment to sequence the entities or within an aggregated learning object to sequence the underlying building blocks (e.g. chapters in a book). The IMS Reusable Definition of Competency or Educational Objectives can be used to specify competencies, learning objectives and prerequisites. SCORM content can be included in the environment (ADL, 2003). IMS Enterprise can be used for mapping learner and staff roles when instantiating a UOL. The IMS Learner Information Profile (IMSLIP, 2003) can be used to import and export persistent learner property structures to and from an LD runtime system.

The basic problem with the inclusion of external schemas is that the runtime engine must be able to call all the necessary sub-runtime engines that support the different schemas. In practice this is rather problematic, and most organizations are advised to restrict their use of embedded schemas as support for these will be implementation dependent, at least until common practice is established. More generally, application profiles will need to be included with UOL packages to specify what content types can be expected in the package.

Reusability

The position taken in LD is that reusability can be defined at different levels of reuse (see Koper, 2003), ranging from whole units of learning, through learning designs and learning methods, to learning activities, learning objects and learning services, but that not all objects are intended for reuse. Different scenarios apply. A learning object may often be used in a new context without any adaptation. In most cases, however, an object is found within its context, from which it can be isolated, but must be repurposed (adapted) for use in other contexts (e.g. Doorten *et al.,* in press). This depends on issues such as granularization (e.g. Wiley, 2002; Duncan, 2003) and implicit and explicit dependencies between objects. LD supports the different reusability scenarios in principle, because objects can be searched for using the higher-level context of use provided by a UOL, and then isolated and repurposed. In EML practice we also have seen different examples of reuse (e.g. one school for higher professional education reuses the same EML complex environment structure for every course, but with different activities specified). However, the issue of reuse in general is underdeveloped in theory and practice and needs further elaboration.

Formalization – automatic processing

EML and IMS LD provide a semantic specification, i.e. the names and structures are chosen in such a way that they can be understood by human beings (as opposed to computers). This has many advantages, but the question then becomes whether such a specification can be interpreted and managed by computers. LD is an 'under the hood' technique that is not meant to be visible to any end-user. It is typically used in authoring tools, content management systems and runtime systems to create, store, share and present eLearning courses. These tools should be developed in such a way that they are optimally suited to the job of authoring, storing, sharing and presenting eLearning courses with high usability for the specific target group. Specifically the authoring tools should hide the complexity of XML tagging and interpreting. Truly user-friendly tools have yet to be developed, but for the test of the formalization requirement we have already built several prototypical systems (see Koper & Manderveld, in press for an overview of the different runtime systems that were developed to test the formalization requirement). The major findings were that: authoring can be done in a generic XML editor, but because of usability issues it is better to develop a more generic, usable authoring tool that exports valid LD at any moment in the development process (see Brouns, 2003). EML and LD can be interpreted in runtime systems. Several EML and LD systems have already been implemented or are currently being developed (e.g. Edubox at OUNL and Reload at JISC/CETIS). In the context of the EU project Alfanet (IST-2001-33288), tools for LD authoring and runtime are being developed and further refined (see also the Open Source LD engine CopperCore (2004).

For content management, any system that is able to store and retrieve XML files and fragments is suitable. We explored the use of several systems with EML and LD and didn't find any specific LD related problems in using them.

Reproducibility

The LD specification doesn’t contain any information about specific groups, details, dates or service facilities that would bind a design to a specific context—this kind of information needs to be supplied when a design is instantiated. EML and LD courses that have been produced to date can be instantiated as often as needed for different people at different times and places. This supports the “write once, deploy many times, in many places” principle required in the context of eLearning, blended learning, distance education and other mass-education approaches.

Future Developments

The above discussion points to several areas for further exploration. The following points are all necessary:

- Evaluation of how well LD meets the pedagogical expressiveness requirement.
- Better integration of QTI into the LD.
- More work on prudent personalization rules.
- Further work in the field of reuse.

All these require better and more complete tools and systems (authoring, content management, delivery) to be available.

The integration of QTI and LD has to be studied further. This will focus on the mapping of QTI results to LD Properties. It may prove that some minor adjustments in the specification are needed to accommodate the integration more effectively, particularly to allow learning and formative assessment to be integrated smoothly. Similarly the effective integration of LD with any of the above specifications may require some adjustments, but changes are expected to be at the level of minor details.

Another area for development is the provision of additional schemas for new types of learning services to be integrated into the environment (as separate modules), together with a generic multi-user service schema. There is a potentially large number of learning services. They will extend what can be done in online learning, but the danger is that this will occur at the expense of interoperability. There will thus be much to be gained from an abstract, generic service definition. In the meantime, the ability to plug in new service schemas will be necessary and these will form the basis for exploring commonalities needed to inform a generic service schema.

A taxonomy of pedagogies is a common request as this would enable people to search for learning designs according to the embedded pedagogy. However, this is a difficult task as there is no commonly agreed taxonomy. If agreement can be achieved, it will need to be extensible as pedagogical approaches are likely to evolve with the technology.

Some areas for future development include the integration or coordinated use of ontologies (W3C Web Ontology Working Group: OWL - Web Ontology Language: www.w3.org/2001/sw/WebOnt), topic maps (XML Topic Map XTM specification: http://www.topicmaps.org; Park, 2003) and broader aspects of the semantic web as a layer that could both be separate from but link to learning designs by reference or become more integrated as part of the learning design itself. In general, the inclusion of semantic web principles would allow, for instance, software agents to interpret the underlying pedagogical structure of learning designs so that they are able to support designers in creating new designs, or supporting learners during learning.

The possibility of using LD for search purposes has to be explored. Authors are aware of the subject, the learning objectives, and the characteristics of the learners they are developing for. They can therefore be expected to find it easier and less onerous to create this higher-level metadata than providing metadata for every learning object and resource they include in a design. By the same token, it is these higher-level characteristics that they are likely to be most comfortable searching for. By searching at the level of learning design, they would be presented with complete UOLs, with the links to all the resources used by it, together with the activities describing how they are used. An appropriate authoring system, linked to a series of learning design repositories, would enable them to browse, review, extract and reuse or modify the content, learning objects, learning activities and entire learning designs. It thus provides a framework for metadata that more directly meets the needs of authors.

General agreement on learner characteristics is needed if persistent learner information is to be used across learning designs to personalize learning. At present, accessibility requirements are the first such learner characteristics to be defined as extensions to IMS Learner Information, and further work in IMS is being undertaken for Accessibility extensions to Metadata. However, learning styles, such as representational modalities, and many others offer possible candidates.

Last but not least, a number of tasks are necessary for the adoption of any specification. On the technical side, a key task is establishing consistent interoperability across implementations. Specific tasks leading to adoption include:

- The provision of one or more open source reference implementations.
- Support for implementers, typically through an industry association.

- Consistent implementation of LD in authoring tools and runtime system supported by Plugfests to establish interoperability.
- Awareness raising and training of authors and learning providers.
- Authoring tools geared to different pedagogical, learning development methods and cultural needs.
- Repositories that understand the structure of LD to analyse designs and make its components searchable and available for reuse

To accomplish these tasks it is necessary to establish appropriate communities of practice with good communication between them. The European Framework 6 Project UNFOLD (IST-1-507835) will support the development and operation of such communities of practice. These will build on the existing Valkenburg Group, providing communities of practice for software developers, learning designers and authors, and learning providers. Each has a different perspective and role to play in moving LD from a paper standard into effective use.

Conclusions

This article has outlined some of the requirements, capabilities and potential of LD to significantly enhance what can be done in online learning:

- Coordination of multiple users (single user becomes a special case).
- Integration of learning objects and services.
- Providing a learning activity layer over learning objects and services.
- Supporting generic properties and conditions enabling dynamic personalization/adaptation, including accessibility.
- The ability to support multiple pedagogical approaches through a single notation.

Because LD provides a formal language for expressing learning designs, it affords the intriguing possibility of comparing and contrasting different learning designs in a more systematic way than has been possible to date.

However, the greatest potential advantage of LD is that it offers support not just for rather simplistic learning designs (sequenced learning content), but also for newer, and more realistic pedagogical approaches that put the learning and teaching process in the centre, rather than the learning content. This approach to modelling more advanced teaching-learning processes, with the possibility of automating parts of the process merits serious implementation efforts and usage trials. Only then can these possibilities be fully tested, their current limitations determined, and the need for future enhancements made clear.

References

ADL (2003). *SCORM*, Retrieved July 19, 2004 from http://www.adlnet.org/.

Bransford, J., Brown, A. L., & Cocking, R. R. (2000). *How People Learn: Brain, Mind, Experience, and School* (expanded edition), Washington: National Academy Press.

Brown, J. S., Collins, A., & Duguid, P. (1989) Situated Cognition and the Culture of Learning. *Educational Researcher, 18* (1), 32-42.

Brouns, F. (2003). Editing IMS Learning Design. *OTEC TD Report*, Heerlen: Open Universiteit Nederland, Retrieved July 19, 2004 from http://learningnetworks.org/forums/attachment.php?s=&postid=220.

CopperCore (2004). *Open Source IMS Learning Design Engine (software)*, Retrieved July 19, 2004 from http://sourceforge.net/projects/coppercore.

DC (2003). *Dublin Core Metadata Website*, Retrieved July 19, 2004 from http://dublincore.org.

DOCB (2003). *DOCBOOK Website*, Retrieved July 19, 2004 from http://www.docbook.org.

Doorten, M., Giesbers, B., Janssen, J., Daniëls, J, & Koper, E. J. R. (in press). Transforming existing content into reusable 'LOs'. In R. McGreal (Ed.), *Accessible education using learning objects*, London: RoutledgeFalmer.

Duffy, Th. M., & Cunningham, D. J. (1996). Constructivism: implications for the design and delivery of instruction. In D.H. Jonassen (Ed.), *Handbook of Research for Educational Communications and Technology* (pp. 170-198), New York: Macmillan.

Duncan, C. (2003). Granularisation. In A. Littlejohn (Ed.), *Reusing Online Resources: A Sustainable Approach to eLearning* (pp. 12-19), London: Kogan Page.

EML (2000). *Educational Modelling Language Website,* Retrieved July 19, 2004 from http://eml.ou.nl.

ERIC (2003). *ASK ERIC Lesson Plans Website*, Retrieved July 19, 2004 from http://www.eduref.org/Virtual/Lessons/index.shtml.

Gagné, R. M., & Briggs, L. J. (1979). *Principles of instructional design* (2e Ed.), Fort Worth , TX : HBJ College Publishers.

Greeno, J. G., Collins, A. M., & Resnick, L. B. (1996). Cognition and Learning. In D. C. Berliner & R. C. Calfee (Eds.), *Handbook of Educational Psychology* 9pp. 15-46), New York: Simon & Schuster Macmillan.

Hermans, H., Manderveld, J. M., & Vogten, H. (2004). Educational modelling language. In W. Jochems, J. van Merriënboer, & R. Koper (Eds.), *Integrated E-learning* (pp. 80-99), London: RoutledgeFalmer.

van den Hooff, B., Elving, W., Meeuwsen, J. M., & Dumoulin, C. (2003). Knowledge Sharing in Knowledge Communities. In M. Huysman, E. Wenger & V. Wulf (Eds.), *Communities and Technologies* (pp. 119-142), Dordrecht: Kluwer Academic Publishers.

Howell, S. L., Williams, P. B., & Lindsay, N. K. (2003). *Thirty-two Trends Affecting Distance Education: An Informed Foundation for Strategic Planning,* Retrieved July 19, 2004 from http://www.westga.edu/~distance/ojdla/fall63/howell63.html.

IMSLD (2003). *IMS Learning Design. Information Model, Best Practice and Implementation Guide, Binding document, Schemas*, Retrieved July 19, 2004 from http://www.imsglobal.org/learningdesign/index.cfm.

IMSLIP (2003). *IMS Learner Information Package. Information Model, Best Practice and Implementation Guide, Binding document, Schemas,* Retrieved July 19, 2004 from http://www.imsglobal.org/profiles/index.cfm.

IMSLOM (2003). *IMS-LOM Metadata. Binding document, Schemas,* Retrieved July 19, 2004 from http://www.imsglobal.org/metadata/index.cfm.

IMSQTI (2003). *IMS Question and Test Interoperability. Information Model, Best Practice and Implementation Guide, Binding document, Schemas*, Retrieved July 19, 2004 from http://www.imsglobal.org/question/index.cfm.

Koper, E. J. R., & Manderveld, J. M. (in press). Modelling reusable, interoperable, rich and personalised units of learning. *British Journal of Educational Technology*.

Koper, E. J. R. (2001). *Modelling Units of Study from a Pedagogical Perspective: the pedagogical metamodel behind EML*, Heerlen: Open Universiteit Nederland, Retrieved July 19, 2004 from http://eml.ou.nl/introduction/docs/ped-metamodel.pdf.

Koper, E. J. R. (2003). Combining re-usable learning resources and services to pedagogical purposeful units of learning. In A. Littlejohn (Ed.), *Reusing Online Resources: A Sustainable Approach to eLearning* (pp. 46-59), London: Kogan Page.

Koper, R., & van Es, R. (in press). Modeling units of learning from a pedagogical perspective. In R. McGreal (Ed.), *Online education using learning objects*, London: RoutledgeFalmer.

Lave, J., & Wenger, E. (1991). *Situated learning: legitimate peripheral participation*, Cambridge: Cambridge University Press.

Merrill, M. D. (1994). *Instructional Design Theory*, Englewood Cliffs: Educational Technology Publications.

Merrill, M.D. (2003). *First Principles of Instruction*, Retrieved July 19, 2004 from http://www1.moe.edu.sg/itopia/download/abstracts/Applying%20First Principles of Instruction to Technology-Based Education.pdf.

Olivier, W., & Liber, O. (2003). Learning Technology Interoperability Standards. In A. Littlejohn (Ed.), *Reusing Online Resources: A Sustainable Approach to eLearning* (pp. 146-155), London: Kogan Page.

Paas, F, & Firssova, O. (2004). Usability evaluation of integrated e-learning. In W. Jochems, J. van Merriënboer, & R. Koper (Eds), *Integrated E-learning* (pp. 112-125), London: RoutledgeFalmer.

Park, J. (2003). Topic Maps, the Semantic Web and Education. In J. Park (Ed.), *XML Topic Maps: Creating and Using Topic Maps for the Web* (pp. 507-529), Boston: Addison-Wesley.

Pedagogical Patterns Project (2003). Retrieved July 19, 2004 from http://www.pedagogicalpatterns.org/.

Reigeluth, C. M. (1999). What is Instructional-Design Theory and How is it Changing? In C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models: a New Paradigm of Instructional Theory* (Vol. II) (pp. 5-29), Mahwah: Lawrence Erlbaum.

Reigeluth, C. M. (1983). *Instructional design theories and models: an overview of their current status*, Hillsdale, NJ: Lawrence Erlbaum.

Retallick, J., Cocklin, B., & Coombe, K. (1999). *Learning Communities in Education*, London: Routledge.

Roblyer, M. D., & Edwards, J. (2000). *Integrating Educational Technology into Teaching* (2^{nd} Ed.), Upper Saddle River, NJ: Merrill, Prentice Hall.

Ryder, M. (2003) *Instructional Design Models*, Retrieved July 19, 2004 from http://carbon.cudenver.edu/~mryder/itc_data/idmodels.html#prescriptive.

Van Merriënboer, J. J. G. (1997). *Training Complex Cognitive Skills: a four component instructional design model for technical training*, Englewood Cliffs: Educational Technology Publications.

Wenger, E. (1998). *Communities of Practice: Learning, Meaning and Identity*, New York: Cambridge University Press.

Wiley, D. (2002). *The Instructional Use of Learning Objects*, Retrieved July 19, 2004 from http://www.reusability.org/read/.