1-1-2008

Learning design: concepts

Susan J. Bennett
University of Wollongong, sbennett@uow.edu.au

Rob Koper
University of Netherlands

Recommended Citation

Bennett, Susan J. and Koper, Rob: Learning design: concepts 2008, 135-154.
https://ro.uow.edu.au/edupapers/691

Learning Design: Concepts

Rob Koper

Open University of the Netherlands

Sue Bennett

University of Wollongong

1 Introduction

Crucial in any learning process are the activities that learners undertake: reading, thinking, discussing, exploring, problem solving, etc. When learners are passive you cannot expect them to learn much. The primary role of any instructional agent, whether it is a teacher, the learners themselves or a computer, is to stimulate the performance of learning activities that will gradually result in the attainment of the learning objectives. The instructional agent defines the tasks, provides the contexts and resources to perform the tasks, supports the learner during task performance and provides feedback about the results. The learning activities that are needed to obtain some learning objectives are in most cases carefully sequenced according to some pedagogical principles. This sequence of learning activities that learners undertake to attain some learning objectives, including the resources and support mechanisms required to help learners to complete these activities, is called a *learning design*.

A *learning design language* is a notation that describes learning designs in a machine interpretable way. The most obvious use of such a learning design language is that it can be used to codify the learning design of a course (as a flow of activities) and then this code is interpreted with a runtime engine that can repeat the course over and over again for different users in different situations, adapted to the characteristics of the individual users in the course. When the course is designed well, the different actors do not have to be concerned much about the management of activities and information flow within the course: this is done automatically. Also the adaptation rules that are specified are applied automatically and consistently within the course runs. Furthermore, the necessary content and services are setup automatically and made available to the users at the right moment.

In this chapter we will concentrate on two questions. First, how to identify high quality learning designs and second, how to codify these learning designs in a machine interpretable way using IMS Learning Design (IMSLD, 2003).

2 High quality learning designs

Before using any learning design language, it is important to know which learning designs are highly effective for a certain target group, a certain domain and certain learning objectives. The Australian project of Agostinho, Oliver, Harper, Hedberg & Wills (2002) identified high quality designs that made exemplary use of information and communication technologies in higher education. High quality designs were defined as those that engage learners' prior knowledge and experiences, set learning effectively within the broader context, challenged learners through active participation, and encouraged learners to articulate their understanding to themselves and peers (Boud & Prosser, 2002). The high quality designs were selected because they emphasize active, constructive learning and address the need to cater for a diverse range of adult learners. (A full description of the project and the designs collected can be found at http://www.learningdesigns.uow.edu.au.)

In this approach a standard format is used to provide textual information about how the design was derived from theory and/or practice, the research or evaluative evidence to support the approach, guidance about how it should be implemented, and suggestions for how the design might be adapted to other learning contexts. This description is accompanied by a graphical representation developed to illustrate the learning design as it is experienced by a learner (Agostinho *et al*, 2002). An example of the format used for the graphical representation is shown in Figure 1.

This example, adapted from Bennett (2002), depicts a series of tasks that learners typically complete when undertaking an analysis of case materials, beginning with an individual analysis in which learners develop their own ideas, followed by small group and then whole class discussions in which learners refine their ideas through discussion and negotiation with other learners under facilitation by the teacher. Resources may be provided by the teacher to support the task, as the case materials are provided by the teacher in this example, or may be produced as part of the learning experience to be shared with others, for example the lists of key points derived from the cases which are refined through discussion activities. The supports provided may be in the form of personal interactions with the teacher or other students, which may occur in person or may be mediated through information and communication technologies (ICTs). Supports may also be provided in the form of documents, in this example as written instructions, templates and guiding questions. A timeline on the right hand side provides an indicative timeframe for the sequence. An example of the accompanying text description is available in Bennett (2002).

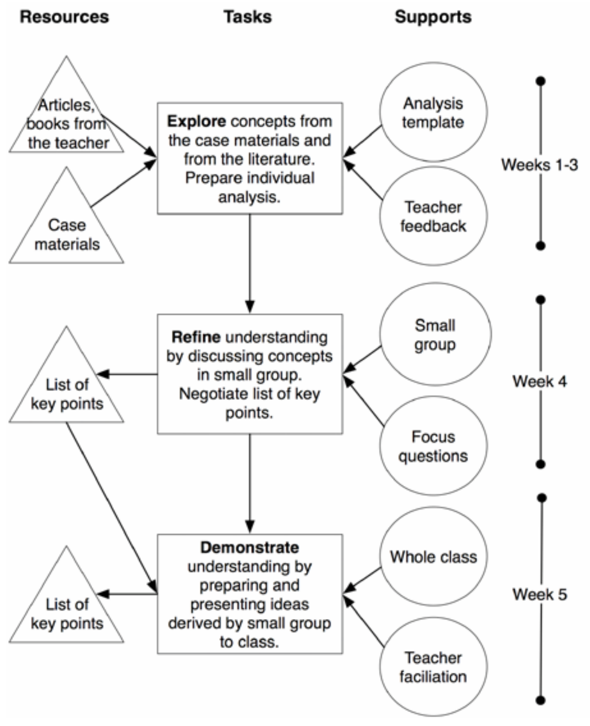

Figure 1: Learning design graphical representation for case analysis tasks

The learning design format makes it possible to represent any learning experience of any granularity in the form of a document. Important to the learning design concept is that the description communicates the general structure and logic of the learning sequence, but does not specify either the content or the particulars of the task or support. These decisions are left to the instructional agent (e.g. a teacher) acting on the guidance included in the learning design and on their understanding of their discipline and their knowledge of their students and institutional requirements.

The strategy of using learning designs to support the design process has its theoretical basis in case-based reasoning. Cases that describe how similar design problems have been solved have been shown to help teachers in designing new learning experiences (Bennett, 2005). Teachers do this by relating proven learning designs created by others to their own contexts, and then adapting the relevant features of a design to suit their learners. When used in this way learning designs promote a form of professional peer learning as described by Kreber (2003) in which university teachers are presented with new ideas that are grounded in the realities of teaching in higher education.

In recent research into how university teachers use learning designs, participants in a small study applied a problem-based learning design to different learning context and their design decisions and outcomes were recorded and analyzed (Bennett, Agostinho & Lockyer, 2005). The results indicated that the learning design description supported the teachers' design processes, and that the features and underpinning rationale of the learning design was evident in the different versions developed by the participants. The findings indicate this form of learning design was readily understood by the participants and sufficiently flexible to be adapted to different contexts. Further research is underway to test the learning design concept across a broader range of disciplines.

3 Applying the learning design in online courses

The next step is to develop the learning design approach as the basis for practical, relevant and flexible supports and tools that university teachers need as they design for online learning. Despite an array of expert advice and descriptive literature about online learning, many university educators find designing effective online learning experiences a significant challenge.

Designing consists of activities, such as planning schedules, writing course outlines, preparing materials, determining assessment tasks, and anticipating students' needs (Bennett & Lockyer, 2004). Designing may involve modifying a previous course, updating material or trying new strategies. Much of this design work occurs within the online environment of a learning management system (LMS) for administering, designing and facilitating online learning.

The key to using learning designs to support the design process is to provide software tools that link directly with the LMS. This strategy will provide support within the online environment, in the context and at the time it is needed. Rather than provide models to be applied or templates to be completed over which little discretion can be exercised, the strategy will give teachers the flexibility to customize the learning design to suit their context. This places the teacher in the mediating role of design decision maker rather than prescribing a particular approach, and seeks to further develop their professional knowledge and judgement. The process is depicted in Figure 2.

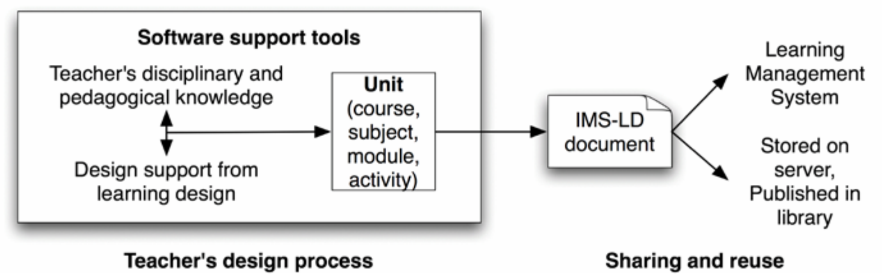

Figure 2: Supporting the teacher's design process

The process begins with a teacher interacting with software tools that allow him or her to select an appropriate learning design consistent with the learning objectives required. While working with the learning design to customize and adapt it for the local situation the teacher is creating a 'unit of learning', which may be a course, a subject, a module or an activity. The 'unit of learning', therefore, encapsulates all of the specifics of the tasks, resources and supports, which can in turn be expressed in the machine interpretable learning design language IMS Learning Design (IMS

LD). When the design is complete, the IMS LD document can be saved and imported into any LMS compliant with the standard.

After creating a lesson or course in an LMS and saving it as an IMS LD document, a teacher could share it within a teaching team, institution or digital library, allow it to be edited in any other LMS that complies with the standard, and the new version could be saved as a new IMS LD document. This strategy has great potential to not only make particular lessons or courses sharable so that they can be reused and adapted by others, but also for the learning designs on which they are based to be shared and reused.

4 The IMS Learning Design specification

4.1 Introduction

In the previous sections we discussed how high quality learning designs could be derived from practice and how teachers can use learning design tools and high-level representations. We also discussed how these designs could be coded in IMS LD in order to be used in any compliant LMS. In this section we will discuss the basic principles of IMS LD.

The IMS Learning Design specification (Koper & Olivier, 2004) is a standardized learning design language that was based on the work on Educational Modelling Language (EML, 2000; Koper, 2001; Hermans, Manderveld, and Vogten, 2004; Koper & Manderveld, 2004) at the Open University of the Netherlands.

When we started to develop learning design language EML we realized that we had to develop a meta-model of pedagogical approaches. There are hundreds of different pedagogical models described in the literature (e.g. see Koper, 2001; Reigeluth, 1983, 1999). There are many so-called lesson plans shared on the Internet (Van Es, 2004) and new models, lesson plans and best practices continue to be formulated. Modelling each separate example and then developing tools to support it,would be a very 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 courses (and other 'units of learning' as they are called in IMS LD) can be specified that are applications of a specific pedagogical approach.

4.2 The requirements

The major requirement for the development of any learning design language is to provide a containment framework that uses and integrates existing specifications as much as possible, and which can represent the teaching-learning process (the learning design or LD) in a unit of learning (UoL), based on different pedagogical models – including the more complex and advanced ones – in a formal way. More specifically a LD specification must meet the following requirements:

1. The notation must be comprehensive. It must describe the teaching-learning activities of a unit of learning in detail and include references to the learning objects and services needed to perform the activities. This means describing:
    - How the activities of both the learners and the staff roles are integrated.
    - How the resources (objects and services) used during learning are integrated.
    - How both single and multiple user models of learning are supported.
2. The notation must support mixed mode (also called blended learning) as well as pure online learning.
3. The notation must be sufficiently flexible to describe learning designs based on all kinds of theories and so must avoid biasing designs towards any specific pedagogical approach.
4. The notation must be able to describe conditions within a learning design that can be used to tailor the learning design to suit specific persons or specific circumstances.
5. The notation must make it possible to identify, isolate, de-contextualize and exchange useful parts of a learning design (e.g. a pattern) so as to stimulate their re-use in other contexts.
6. The notation must be standardized and in line with other standard notations.
7. The notation must provide a formal language for learning designs that can be processed automatically.
8. 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 IMS 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.

4.2 The conceptual model

The pedagogical meta-model that has been developed to represent different kinds of learning designs is at the heart of the IMS LD specification. It provides the conceptual structure of the specification as well as its underlying theoretical model (see figure 3).

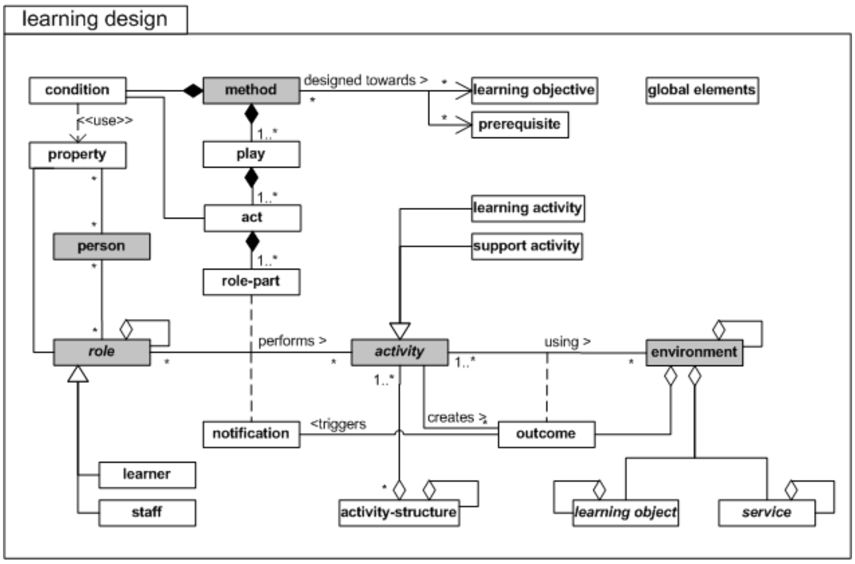

Figure 3: Conceptual structure of the LD specification

The core concept of LD, as expressed in Figure 3, is that a learning design can be represented by using the following core concepts: A *person* takes on a *role* in the teaching-learning process, typically a *learner* or a *staff* role. In this role he or she works towards certain *learning objectives* 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. Figure 4 contains an example of the use of these labels in a photograph of a classical learning design: a classroom setting.

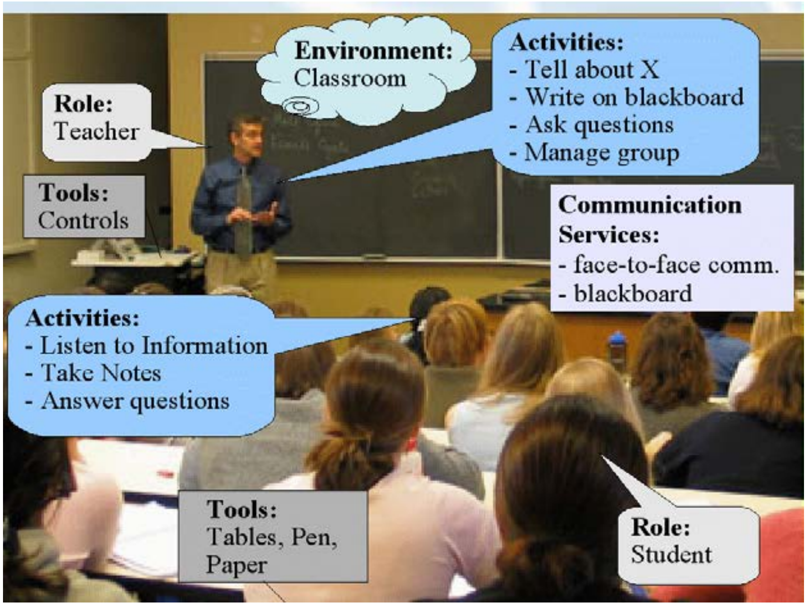

Figure 4. Labelling a classroom setting with IMS LD concepts

You can imagine that this type of labelling is possible on any photograph of any teaching-learning event, whether this is classroom teaching, self-study, group collaborations, field experiments, etc. However, photographs are static and the teaching-learning process is dynamic, so labelling of the visible entities is not sufficient. What is needed is an additional process description. This process description is provided in the *method* section of IMS LD. The method is designed to provide the co-ordination 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).

The method section 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 in which 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 people’s 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 '*imsld-content*'. 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 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".

4.3 The information model and XML binding

The conceptual model is implemented as follows. A UoL is represented as an IMS Content Package (CP). A CP has an organization part that represents how items are organized in the package. Normally the organization part represents nothing more than a hierarchy of items, but the CP specification allows replacement of the organization structure by any other structure. In IMS LD the organization part of a CP is replaced with a <learning-design> element (figure 5).

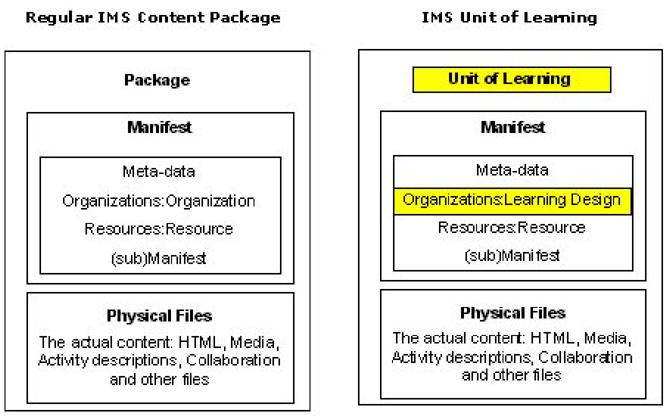

Figure 5. In IMS LD the organization element of a regular IMS Content Package is replaced with the Learning Design elements

The <learning-design> element is a complex structure that includes elements that represent the conceptual model already outlined. The details of these elements are detailed in the Information Model document, together with their behavioural specifications.

The learning design elements have an XML schema binding that can be represented as the tree in figure 6.

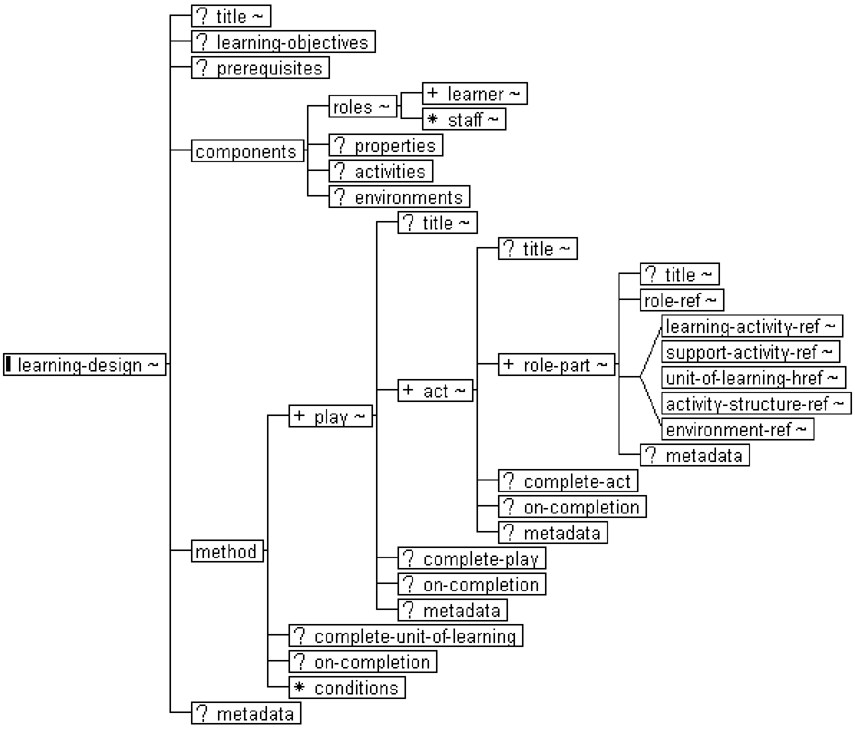

Figure 6. The LD schema represented as a tree.

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.

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 UoL package.

5 Interpreting IMS LD

When a UoL is specified in IMS LD the result is a zip file. Running this zip file requires a runtime engine that handles at least the following five tasks:

1. A validation of the zip file to ensure that only valid IMS LD is processed. Validation includes both technical and semantic checks and the validation results are reported.
2. Creation of one or more instances of the zip-file (this is called a 'run').
3. Assignment of persons to the specific roles in the run and setup of the required communication and collaboration services like forums, chats, wikis.
4. Interpretation of the IMS LD and delivery of personalized and sequenced learning activities, content and services according to the rules defined in LD. This is achieved by keeping track of the user's progress and settings.
5. The concept of a run is described in (Vogten *et al,* 2005, 2006; Tattersall *et al*, 2005) and is comparable with parallel classes in a school. A school may have different parallel classes: each with the same objectives and content, but with different learners and teachers. The same classes (runs) are also repeated year after year with different students (and sometimes different teachers), although the versions of the learning design may be adapted in between different runs. So, a run is an instance of a course with specific learners and teachers and is executed in a specific timeframe.A runtime engine must be able to setup and manage runs of UoLs packages.

An IMS LD runtime engine must be able to interpret every IMS LD zip file package. The challenge is that LD is a *declarative* language, meaning that it describes *what* an implementation must do, it does not specify *how* this should be done. Furthermore, LD is an *expressive*, i.e. semantically , language that enables expression of learning designs in a clear, natural, intuitive and concise way, closest to the original problem formulation. This expressive and declarative nature complicates the implementation of an engine that can interpret the specification. For this reason we implemented an open source runtime engine, called CopperCore (Martens & Vogten, 2005; see also www.coppercore.org) to serve as a reference implementation of IMS LD handling. CopperCore can be used by any LMS to handle LD packages or be used as an example for the recoding of an LMS native runtime engine.

The CopperCore runtime engine does not provide user interfaces: it only provides APIs to build a dedicated user interface. For demonstration purposes CopperCore is provided with a simple user interface (CopperCore Player, see figure 7), but a better implementation of a player is the SLED player (see McAndrew, Nadolski & Little, 2005; see also sourceforge.net/projects/ldplayer).

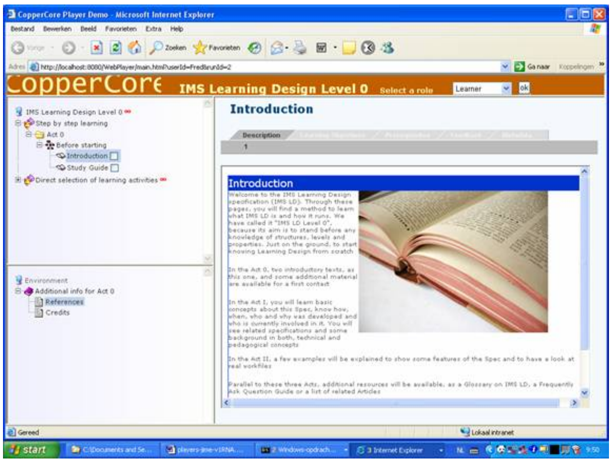

Figure 7. The CopperCore Player

6 Conclusion

In this chapter we have introduced the basic concepts in the field of learning design research. We defined what learning designs are, how high quality learning designs can be identified and described, how learning designs can be coded in IMS LD and how IMS LD code can be interpreted by a runtime engine and presented by a LD Player.

In the conclusion of this chapter we will now concentrate on the issues that have been identified and studied in the past three years. These issues are summarized from the research of many different researchers, for example those who have reported their work in two special issues of journals (JIME, 2005 and ET&S, 2006), an edited book (Koper & Tattersall, 2005) about IMS LD and international conferences such as ASCILITE and ICALT. The issues are summarized below.

The first issue is related to the identification of *high quality learning designs*. One of the ideas in this area is to identify and use learning design patterns. These patterns can be used to support learning designers to develop high quality learning designs in specific areas by combining and adapting several patterns to a course. One direction of research is to search for solutions to derive these patterns from effective IMS LD coded courses. A pattern detection mechanism will then analyze the code to look for patterns (Brouns *et al*, 2005). Another approach is to capture best practices and learning design knowledge of teachers using textual descriptions and graphical representation that can be readily understood by other teachers (e.g. the approach described by Agostinho *et al* (2002) and further developed by Bennett *et al* (2005)).

The second issue is the development of learning design *authoring tools*. This includes the following issues:
a) The development of a graphical representation for learning designs, like the ones found in MOT+ (Paquette *et al*, 2006), LAMS (Dalziel, 2003) and ASK-LDT (Karampiperis & Sampson, 2005). Also the representation of learning designs that is described in figure 1 is an alternative.
b) The support for reuse of the learning design knowledge of teachers and experienced instructional designers (Hernández-Leo, Harrer, Dodero *et al*, 2006).
c) The question how learning designers should be supported with tools and how teachers should be supported with tools in specific contexts (i.e. that support the teacher as a designer, Bailey *et al*, 2006; Bennett *at al*, 2006);
d) The integration of learning design and assessment editors (e.g. IMS QTI) in a single authoring environment (Vogten *et* al, 2006; Giacomini Pacurar *et al*, 2006; Joosten-Tenbrinke *et al*, in press).

The third issue is the further development of *learning design players*. This includes the following issues:

a) How to integrate the variety of specifications (e.g., IMS LD, IMS QTI, SCORM, IMS LIP) and the connections to other systems in an e-learning infrastructure (student administration, portfolio systems, financial systems) into a single, easy to use learning environment? (Van Rosmalen *et al*, 2006)
b) How to instantiate and integrate communication and collaboration services that are called by IMS LD, e.g. forums, wikis, chats (Weller *et al*, 2006; Vogten *et al*, 2006))
c) How to design a usable, powerful and flexible graphical user-interface for a player environment?
d) How to integrate IMS LD into existing Learning Management Systems like Moodle, Blackboard, dotLearn and LAMS (see for instance Berggren *et al*, 2005)?
e) How to integrate learning design authoring systems and learning design players, including the question how to deal with runtime adaptations (Zarraonandia, Fernández, & Dodero, 2006)?
f) How to use an integrated set of learning design tools in an integrated way in a variety of settings e.g. in universities, training, blended learning (Sloep *et al*, 2006).

Furthermore, there are several additional issues to mention, such as proposals to change the current XML schema binding to an ontology language like OWL. Amorim *et al* (2006) and Knight *et al* (2006) propose such a binding to integrate learning objects and learning designs to represent specific pedagogical approaches and to build software agents that operate on the learning design knowledge to support in the development of units of learning. A last point to mention is the work in the area of the evaluation of the expressiveness of IMS LD (Caeiro-Rodriguez, 2005; Van Es & Koper, 2006).

Acknowledgement

The work on this chapter has been sponsored by the TENCompetence Integrated Project that is funded by the European Commission's 6th Framework Programme, priority IST/Technology Enhanced Learning. Contract 027087 (www.tencompetence.org)

References

Agostinho, S., Oliver, R., Harper, B., Hedberg, H., & Wills, S. (2002). A tool to evaluate the potential for an ICT-based learning design to foster high-quality learning. In A. Williamson, C. Gunn, A. Young & T. Clear (Eds.), *Proceedings of Australasian Society for Computers in Learning in Tertiary Education*. (pp. 29-38). Auckland, NZ: UNITEC.

Amorim, R. R., Lama, M., SÃ¡nchez, E., Riera, A., & Vila, X. A. (2006). A Learning Design Ontology based on the IMS Specification. Educational Technology & Society, 9 (1), 38-57.

Bailey, C., Zalfan, M. T, Davis, H. C., Fill, K., & Conole, G. (2006). Panning for Gold: Designing Pedagogically-inspired Learning Nuggets. Educational Technology & Society, 9 (1), 113-122.

Bennett, S. (2002). *Description of A technology-supported constructivist learning environment that uses real-life cases to support collaborative project work.* Retrieved November 7, 2006, from Learning Designs Web site: www.learningdesigns.uow.edu.au/LearningDesigns/exemplars/info/LD1/index.html

Bennett, S. (2005). Using related cases to support authentic project activities. In A. Herrington & J. Herrington (Eds.), *Authentic learning environments in higher education* (pp. 120-134). Hershey, PA: Idea Group Inc.

Bennett, S. & Lockyer, L. (2004). Becoming an online teacher: Adapting to a changed environment for teaching and learning in higher education. *Educational Media International*, 41(3), 231-244.

Bennett, S. Agostinho, S., & Lockyer, L. (2005). Reusable learning designs in university education. In T. C. Montgomerie & J. R. Parker (Eds.), *Proceedings of the IASTED International Conference on Education and Technology* (pp.102-106). Anaheim, CA: ACTA Press.

Bennett, S., Agostinho, S., Lockyer, L., Harper, B., & Lukasiak, J. (2006). Supporting university teachers create pedagogically sound learning environments using learning designs and learning objects. *International Journal on WWW/Internet, 4*(1), 16-26.

Berggren, A., Burgos, D., Fontana, J.M., Hinkelman, D., Hung, V., Hursh, A., & Tielemans, G. (2005). Practical and Pedagogical Issues for Teacher Adoption of IMS Learning Design Standards in Moodle LMS. *Journal of Interactive Media in Education, 2005/02*. ISSN:1365-893X [jime.open.ac.uk/2005/02].

Boud. D., & Prosser, M. (2002). Key Principles for High Quality Student Learning in Higher Education: A framework for evaluation. *Educational Media International, 39*(3), 237-245.
Brouns, F., Koper, R., Manderveld, J., Van Bruggen, J., Sloep, P., Van Rosmalen, P., Tattersall, C., & Vogten, H. (2005). A first exploration of an inductive analysis approach for detecting learning design patterns. *Journal of Interactive Media in Education*, *2005/03*. Retrieved November 21, 2006, from http://www-jime.open.ac.uk/2005/03/.
Caeiro-Rodriguez, M., Anido-Rifon, L., & Llamas-Nistal, M. (2005). A perspective and pattern-based evaluation framework of EMLs' expressiveness for collaborative learning: application to IMS LD. Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies, ICALT 2005, Taiwan, 5-8 July 2005. Retrieved November 21, 2006, from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1508604
Dalziel, J. (2003). Implementing learning design: the Learning Activity Management System (LAMS). Available at www.melcoe.mq.edu.au/documents/ASCILITE2003 Dalziel Final.pdf
EML (2000). Educational Modelling Language (version 1.0 and version 1.1). Website. Retrieved November 21, 2006, from http://dspace.ou.nl/handle/1820/81 and http://dspace.ou.nl/handle/1820/80
ET&S (2006). Current Research in Learning Design (special issue). *Educational Technology & Society*, 9(1), 13-175. Retrieved November 21, 2006, from http://www.ifets.info/others/
Pacurar, E. G., Trigano, P., & Alupoaie, S. (2006). Knowledge base for automatic generation of online IMS LD compliant course structures. Educational Technology & Society, 9 (1), 158-175.
Hermans, H., Manderveld, J., & Vogten, H. (2004). Educational Modelling Language. In W. Jochems, J. van Merrienboer, & R. Koper, Integrated e-Learning (pp. 80-99). London: RoutledgeFalmer
Hernández-Leo, D., Harrer, A., Dodero, J. M., Asension-Pérez, J. I., & Burgos, D. (2006). Creating by reusing Learning Design solutions. *Proceedings of 8th Simposo Internacional de InformÃ¡tica Educativa*, León, Spain: IEEE Technical Committee on Learning Technology. Retrieved November 21, 2006, from http://dspace.ou.nl/handle/1820/788
IMSLD (2003). *IMS Learning Design.* Information Model, Best Practice and Implementation Guide, Binding document, Schemas. Re-

trieved November 21, 2006, from http://www.imsglobal.org/learningdesign/index.cfm

IMSLIP (2003). *IMS Learner Information Package.* Information Model, Best Practice and Implementation Guide, Binding document, Schemas. November 21, 2006, from http://www.imsglobal.org/profiles/index.cfm

IMSLOM (2003). *IMS-LOM Metadata.* Binding document, Schemas. Retrieved Retrieved November 21, 2006, 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 November 21, 2006, from http://www.imsglobal.org/question/index.cfm

JIME (2005). Advances in Learning Design (Special Issue). *Journal of Interactive Media in Education, 2005/01*. Retrieved November 21, 2006, from http://www-jime.open.ac.uk/2005/01/.

Joosten - ten Brinke, D., Van Bruggen, J., Hermans, H., Latour, I., Burgers, J., Giesbers, B., & Koper, R. (in press). Modeling Assessment for Re-use of Traditional and New Types of Assessment. *Computers in Human Behaviour*.

Karampiperis, P., & Sampson, D. (2005). Designing learning services for open learning systems utilizing Learning Design. In Uskov, V. (Ed.), *Proceedings of the 4th IASTED International Conference on Web-based Education*, Anaheim, CA, USA: ACTA Press, 279-284.

Knight, C., GaÅ¡eviÄ‡, D., & Richards, G. (2006). An Ontology-Based Framework for Bridging Learning Design and Learning Content. Educational Technology & Society, 9 (1), 23-37.

Koper, E.J.R. (2001). *Modelling Units of Study from a Pedagogical Perspective: the pedagogical metamodel behind EML*. Heerlen: Open Universiteit Nederland. Retrieved November 21, 2006, from http://hdl.handle.net/1820/36

Koper, E.J.R., & Sloep, P. (2003), *Learning Networks: connecting people, organizations, autonomous agents and learning resources to establish the emergence of effective lifelong learning*, Learning Technology Development Programme 2003 – 2008. Heerlen: Open University of the Netherlands. Retrieved October 3, 2003, from http://www.learningnetworks.org/

Koper, E.J.R., & Manderveld, J. M. (2004). Modelling reusable, interoperable, rich and personalised units of learning. *British Journal of Educational Technology*, 35 (5), 537-552.

Koper, R., Tattersall, C. (eds.) (2005). *Learning Design: a handbook on modelling and implementing network-based education & training*. Heidelberg: Springer Verlag.

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

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. (2004). Modeling units of learning from a pedagogical perspective. In R. McGreal (Ed.), *Online education using learning objects*. London: RoutledgeFalmer.

Kreber, C. (2003). The scholarship of teaching: A comparison of conceptions held by experts and regular academic staff. *Higher Education, 46*(1), 93-121.

Martens, H., & Vogten, H. (2005). A reference implementation of a Learnin Design engine. In: Koper, R. & Tattersall, C., *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 91-108). Berlin-Heidelberg: Springer Verlag

McAndrew, P., Nadolski, R., & Little, A. (2005). Developing an approach for Learning Design Players. *Journal of Interactive Media in Education, 2005/14.* Retrieved November 21, 2006, from http://www-jime.open.ac.uk/2005/14/

Paquette, G., Léonard, M., Lundgren-Cayrol, K., Mihaila, S., & Gareau, D. (2006). Learning Design based on Graphical Knowledge-Modelling. *Educational Technology & Society, 9* (1), 97-112.

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

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*, Volume II (pp. 5-29). Mahwah: Lawrence Erlbaum Associates, Publishers.

Sloep, P.B., Van Bruggen, J., Tattersall, C., Vogten, H., Koper, R., Brouns, F., Van Rosmalen, P. (2006), Innovating education with an educational modelling language: two case studies. *Innovations in Education and Teaching International*, 43(3), 291-301.

Tattersall, C., Vogten, H., Brouns, F., Koper, R., van Rosmalen, P., Sloep, P., & van Bruggen, J. (2005). How to create flexible runtime de-

livery of distance learning courses. *Educational Technology & Society*, 8 (3), 226-236.
Van Es, R. (2004). Overview of online databases with lesson plans and other learning design methods. Retrieved November 21, 2006, from http://hdl.handle.net/1820/102.
Van Es, R., & Koper, R. (2005) Testing the pedagogical expressiveness of IMS LD. *Educational Technology & Society*, 9 (1), 229-249.
Van Rosmalen, P., Vogten, H., Van Es, R., Passier, H., Poelmans, P., & Koper, R. (2006). Authoring a full life cycle model in standards-based, adaptive e-learning. *Educational Technology & Society*, 9 (1), 72-83.
Vogten, H., Koper, R., Martens, H. & Tattersall, C. (2005). An Architecture for Learning Design Engines. In: Koper, R. & Tattersall, C., *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training* (pp. 75-90). Berlin-Heidelberg: Springer Verlag
Vogten, H., Tattersall, C., Koper, R., van Rosmalen, P., Brouns, F., Sloep, P., van Bruggen, J. & Martens, H. (2006). Designing a Learning Design Engine as a Collection of Finite State Machines. *International Journal on E-Learning*. 5 (4), pp. 641-661
Vogten, H., Martens, H., Nadolski, R., Tattersall, C., van Rosmalen, P., & Koper, R. (2006). Integrating IMS Learning Design and IMS Question and Test Interoperability using CopperCore Service Integration. *Proceedings of International Workshop in Learning Networks for Lifelong Competence Development*, *TENCompetence Conference*. March 30th-31st, Sofia, Bulgaria: TENCompetence. Retrieved June 30th, 2006, from http://dspace.learningnetworks.org
Weller, M., Little, A., McAndrew, P., & Woods, W. (2006). Learning Design, generic service descriptions and universal acid. *Educational Technology & Society*, *9* (1), 138-145.
Zarraonandia, T., Fernández, C., & Dodero, J. M. (2006). A Late Modelling Approach for the Definition of Computer-Supported Learning Process. *Proceedings of Adaptive Hypermedia*. June, Dublin, Ireland. Retrieved November 21, 2006, from http://dspace.ou.nl/handle/1820/753