Preface

The Valkenburg Group

In March 2002, thirty-three experts in e-learning from four continents met each other for the first time in Valkenburg aan de Geul, a small village in the south of The Netherlands. Since then, the group, referred to as the Valkenburg Group, has met several times at different locations to explore how to improve the pedagogical quality of e-learning courses, in an interoperable way, with user-friendly tools. The general feeling of the experts was that most of the current e-learning offerings lack one or more of these aspects: they are of poor pedagogical quality, they lack portability, or they lack adequate tooling. Pedagogical quality is considered to be the key issue. To be successful, e-learning must offer effective and attractive courses and programmes to learners, while at the same time providing a pleasant and effective work environment for staff members who have the task of developing course materials, planning the learning processes, providing tutoring, and assessing performance.

Learning Design

The Valkenburg Group reached consensus on the idea that the Educational Modelling Language (EML) and the IMS Learning Design (LD) specification provide a good starting point towards this objective. EML was developed at the Open University of the Netherlands and was released in December 2000. EML was the input for the development of the LD specification by IMS, a consortium of global e-learning software companies and users (see imsglobal.org) and the specification was released in February 2003. Although EML and LD differ in structure, functionally they are more-or-less equivalent. With EML and LD, it is possible to develop and present advanced, interoperable e-learning courses that go beyond current implementations. The specifications were developed to describe an unlimited number of pedagogical approaches, both old and new, by abstracting from those described in the literature (eg, the collection of models described by Reigeluth in 1983 and in 1999). This abstraction level is referred to as a pedagogical meta-model (Koper 2000, 2002), and has been tested in practice in several implementations and with various courses in different settings (Koper & Manderveld 2004). For example, with EML/LD courses were described that are based on the active participation of learners in an interoperable way, such as:

- Educational role and game playing courses where multiple users perform a variety of interdependent tasks.
- Problem-based learning courses where teams of learners collaborate in problem solving and teachers have expert, assessment, coaching or monitoring roles.
- Learning community approaches based on social-constructivist principles, where the design of the learning environment stimulates collaboration and sharing of knowledge and resources.
- Performance support approaches, where learning tasks are assigned depending on assessed knowledge gaps.
- Adaptive courses where the pedagogical model, the learning processes and content are adapted to, for example, the learning needs, preferences and learning styles of learners.
- Peer coaching and assessment approaches, where learners support each other.

Koper and Van Es (in press) tested the pedagogical flexibility of LD more systematically. Their approach used an inventory of databases of pedagogical models available on the Internet (also called "lesson plans", see Van Es 2004 for an overview). Sixteen lesson plans were randomly selected from these databases, covering a variety of designs based on different pedagogical traditions (behaviorist, cognitive, social-constructivist). The lesson plans were all able to be coded in LD without any restrictions.

Learning Design provides a conceptual model for the description of teaching and learning processes. In a certain sense it works like a musical notation: it can capture the teaching and learning processes on paper. This makes the design explicit, it can be reflected upon by the designers themselves or by others, and it can be further refined and shared within a community of course developers. This feature is expected to increase the quality of courses in the long run.

IMS delivers XML Schemas (W3C 2004b) as an integral part of all its specifications. As a result, the learning designs of courses are expressed in XML to make the course machine-readable. This means that courses encoded using LD can be processed by runtime agents, making the delivery management of courses more efficient. In current e-learning systems, the teacher still has many mundane management tasks to perform to set up and maintain the environment. This can be automated to a large extent using LD.

The realisation of all these very desirable advantages of LD is, however, still a future perspective. The principles and standards are defined, but most of the tooling still has to be developed. It is exactly this aspect, namely the joint development of tools around LD, that has been the driving factor behind the Valkenburg Group. Currently the European Commission has strengthened the activity of the Valkenburg Group by funding by the UNFOLD project[1]. This book is one of the valuable resources used within this project, and some parts of the work of writing and editing this book were also sponsored by the UNFOLD project.

Development of the Ideas Behind Learning Design

It is helpful to understand the 'where, why, when and how' that went into the development of the concepts that inform LD. In 1997, the Open University of the Netherlands made a strategic decision that e-learning would be central to its future in terms of helping to innovate institutes for higher education and to renew its own educational system by implementing new competency-based models of education, integrated into an electronic learning environment. The university had to confront the fact that many different pedagogical approaches are in use in higher education and its own institution. A key issue was how these many different approaches should be expressed and supported on-line. Up to then, many interesting e-learning projects had provided innovative ways of support for particular pedagogical approaches, but were based on different systems, with different support needs, scalability, and other characteristics, each requiring its own integration effort with existing systems. The alternative of attempting to limit existing practice to the use of one or two pedagogical approaches was, if anything, even more problematic. An internally funded five-year R&D programme was therefore initiated to address this difficult dilemma.

In addition to surveying the pedagogical approaches actually in use within the university and its partners, the project team carried out extensive research into the variety of available pedagogical approaches, identifying over a hundred. The team then analysed these for common characteristics and, through a process of abstraction and experimentation, arrived at a 'pedagogical meta-language' that formed the base of EML. EML evolved in several iterations over a further two-year period of development. The development of EML went through three complete cycles of specification development, implementation in prototype software, trialling with users, evaluation of results, and redesign of the specification and prototype software. A key aim throughout these three iterations was to achieve the right balance between being sufficiently general to support the desired range of pedagogies, while at the same time being sufficiently specific to be useful and capable of supporting what was needed. EML v1.0 was released in December 2000 after three years of development and ex-

[1] UNFOLD (IST-2002-1_507835, January 2004 to December 2005) is funded under the European Union's Sixth Framework Programme. It is a Coordination Action within the Technology-enhanced learning and access to cultural heritage Action Line of the Information Society Technologies area.

perimentation. In 2001, the specification was accepted as the basis for the development of the new IMS Learning Design specification, and after almost two years of work and debate, the final 1.0 version of the IMS specification was made available to the public in February 2003.

The basic idea of EML and LD (we hereafter refer only to LD) is in essence simple. It represents a vocabulary which users of any pedagogical approach understand, and into which existing designs can be translated. The core of LD can be summarised as the view that, when learning, *people in specific groups and roles engage in activities using an environment with appropriate resources and services.*

Many approaches to learning expect learners to work in groups, as well as on their own. However, e-learning standards to date have only supported the model of single learners working in isolation, such as the model behind SCORM (ADL 2004b). An important capability of LD is its integration of discussions and more complex, collaborative approaches to learning into the model of content provision to the isolated individual learner. It is also desirable to integrate these two approaches so that both could be in a single unit of learning. Other requirements of EML and LD included:

- allowing learners to work in several groups so that each group could do different things at the same time to support more complex types of collaborations, as in project-based learning;
- allowing different learners to do the same things at different times, such as taking turns in different roles, or a large group accessing a limited resource (e.g. a remote telescope or other experimental equipment) in a sequence of smaller groups.

Some kinds of learning, such as those derived from programmed learning, require tight control by the system of the learning sequence, depending on the learners' response to tests; while others, such as role-plays, need to allow participants greater control over the course of events. Newer types, such as personalised learning and competency-based learning, have to respond conditionally to the characteristics of the learner, or their current state. To support such a wide variety of approaches to learning is hard, but these ideas, particularly when implemented in an open specification such as LD, make a bold attempt to lay down a foundation for the next generation of learning systems. It is also of great benefit to e-learning system developers to be able to support a wide range of pedagogical approaches using one language, rather than having to support one for each.

However, it should be borne in mind that, as with all first-generation e-learning specifications, LD can be expected to evolve and develop in response to the experiences gained from implementing and using it.

Goal of the Book

The goal of this book is to present the current state of the art in the development of e-learning courses using LD. It provides information about LD, how to implement it in practice, what tools to use, what pitfalls to avoid. It is based on the experience of members of the Valkenburg Group in building tools and using these tools in practice. The book also goes beyond the current state of the art by looking at future advancements.

It should be noted however, that LD is a fairly young specification. Large scale implementations and a full toolset for handling LD are still missing. As a consequence, we are, for example, not yet able to present rigorous summative evaluative findings, and most of the current applications aim at proving the concepts behind LD. The authors and editors are however convinced that the book will help the community of learning designers and LD tool developers to further advance the field.

Intended Audience

The handbook is designed to serve both those with an understanding of the LD specification, and those who are new to it. The target audience is e-learning *course and tool developers* interested in the innovation of e-learning. This includes people who want to improve the effectiveness and attractiveness of e-learning by applying interoperable designs in their courses, including active learning, collaborative learning, problem-based learning, gaming approaches and other multi-role learning activities. It also includes people who want to make teaching and learning using ICT more efficient, e.g. by decreasing the workload of teachers using the automated workflow possibilities of LD. And last but not least, it is intended for those who want to create truly interoperable courses, including all content, services and processes (and not only the interoperable sequenced content).

Conventions Used in the Book

Learning Design or learning design?

In the text, we use the term ‘Learning Design’ (with capitals) and its abbreviation, LD, when referring to the formal specification. At the time of writing, this is the IMS Learning Design Specification, version 1.0. This specification consists of three different items: an information model, a best practice and implementation guide, and an XML binding with a binding document.

We use ‘learning design’ (without capitals) when the human activity of designing units of learning, learning activities or learning environments is meant. This term is never abbreviated to ld. As a synonym the phrase ‘instructional design’ or ‘instructional systems design’ is used in this book, however some may argue that this has a slightly different accent in meaning. Consequently we use ‘the learning design’ when the result of the learning design activity is meant, i.e. a document describing the learning design in any formal or informal notation that is not LD. Furthermore, ‘the Learning Design’ is the part of a unit of learning that describes the XML learning design elements.

When the XML element <learning-design> is meant, we will use the notation ‘learning-design’ (with a hyphen).

Learning Design, Unit of Learning or unit of learning

The term ‘Unit of Learning’ (UOL) is used to describe an IMS Content Package that contains a learning-design element as its organisation. This use of the term is defined in the LD specification. We use the term ‘unit of learning’ to indicate all different kinds of formal and informal learning opportunities and events. Examples are courses, workshops, self-directed informal learning events, lessons, a curriculum, etc.

Suggested Reading Path

		Course developer	Tool developer
Part I Specification, Architectures and Tools			
1	An Introduction to Learning Design	•	•
2	The Learning Design Specification		•
3	Architectures to Support Authoring and Content Management with Learning Design		•
4	An Architecture for the Delivery of E-learning Courses		•
5	An Architecture for Learning Design Engines		•
6	A Reference Implementation of a Learning Design Engine		•
7	Learning Design Tools		•
Part II Designing E-learning Courses			
8	Basic Design Procedures for E-learning Courses	•	
9	An Instructional Engineering Method and Tool for the Design of Units of Learning	•	
10	Integrating Assessment into E-learning Courses	•	
11	Collaboration in Learning Design Using Peer-to-Peer Technologies	•	•
12	Designing Adaptive Learning Environments	•	•
13	Designing Educational Games	•	
14	Designing Learning Networks for Lifelong Learners	•	
15	How to Integrate Learning Design into Existing Practice	•	
Part III Experience			
16	Applying Learning Design to Self-Directed Learning	•	•
17	Applying Learning Design to Support Open Learning	•	
18	Using Learning Design to Support Design and Runtime Adaptation		•
19	The Edubox Learning Design Player		•
20	Delivery of Learning Design: the Explor@ System's Case		•
21	Challenges in the Wider Adoption of LD: Two Exploratory Case Studies		•
22	A Learning Design Worked Example	•	•

Acknowledgements

The editors and authors wish to thank the management and staff of the Schloss Dagstuhl International Conference and Research Center for Computer Science for providing a pleasant, stimulating and well organised environment for the writing of this book. Furthermore, we would like to express our gratitude to the members of the Valkenburg Group and the members of the Technology Development Programme of the Educational Technology Expertise Centre at the Open University of the Netherlands who acted as reviewers for the book chapters. Last but not least, we want to thank Mieke Haemers for the enormous effort she put into supporting the editors.