Chapter XIII
Visual Design of Coherent Technology–Enhanced Learning Systems:
A Few Lessons Learned from CPM Language

Thierry Nodenot
Université de Pau et des pays de l'Adour, France

Pierre Laforcade
Université du Maine, France

Xavier Le Pallec
Université de Lille, France

ABSTRACT

Visual instructional design languages currently provide notations for representing the intermediate and final results of a knowledge engineering process. As some languages particularly focus on the formal representation of a learning design that can be transformed into machine interpretable code (i.e., IML-LD players), others have been developed to support the creativity of designers while exploring their problem-spaces and solutions. This chapter introduces CPM (computer problem-based meta-model), a visual language for the instructional design of problem-based learning (PBL) situations. On the one hand, CPM sketches of a PBL situation can improve communication within multidisciplinary ID teams; on the other hand, CPM blueprints can describe the functional components that a technology-enhanced learning (TEL) system should offer to support such a PBL situation. We first present the aims and the fundamentals of CPM language. Then, we analyze CPM usability using a set of CPM diagrams produced in a case study in a 'real-world' setting.

INTRODUCTION

For several years, the IMS-LD specification (IMS, 2003b) has been the subject of converging theoretical and practical works from researchers and practitioners concerned with Learning Technologies.

The IMS-LD specification is now well documented (Hummel, Manderveld, Tattersall, & Koper, 2004; Koper et al., 2003; Koper & Olivier, 2004) and widely used for the semantic representation of learning designs. A *learning design* is defined as the description of the teaching-learning process that takes place in a unit of learning (Koper, 2006). The key principle in learning design is that it represents learning activities and support activities being performed by different persons (learners, teachers) in the context of a unit of learning. These activities can refer to different learning objects that are used/required by these activities at runtime (e.g., books, software programs, pictures); they can also refer to services (e.g., forums, chats, wikis) used to communicate and collaborate in the teaching-learning process.

Thus, IMS-LD is an educational modeling language that provides a representation of the components of a learning environment in a standardized XML schema that can be executed by compliant e-learning platforms. According to the classification framework defined in Botturi, Derntl, Boot, and Gigl, (2006), IMS-LD is an example of a finalist-communicative language: it is not intended to enable designers to produce intermediate models of the learning design being studied, nor to provide significant methodological support for designers to build a final representation complying with the IMS-LD specification.

Initially, designers had to use XML editors (like *XMLSpy*) to benefit from all IMS-LD expressive capabilities (levels A, B, C). *Reload*, a tree and form based authoring tool, was the first editor to significantly improve this situation. Chapter XV of this handbook provides an extensive presentation of currently available IMS-LD compliant tools (Tattersall, 2007):

- LD-editors like *Reload (Reload, 2005), CopperAuthor (CopperAuthor, 2005), etc.*
- *Visual tools to support practitioners in the creation of IMS-LD compliant designs by means of using collaborative pattern-based templates (Hernández-Leo et al., 2006).*
- Authoring environments for IMS-LD designs like the *ASK Learning Designer Toolkit – ASK-LDT* (Sampson, Karampiperis, & Zervas, 2005).
- Runtime engines able to interpret a LD-scenario like *CopperCore* (Vogten & Martens, 2003).
- learning management systems able to interpret LD scenarios: *dotLRN (Santos, Boticario, & Barrera, 2005), LAMS (Dalziel, 2006), Moodle (Berggren et al., 2005), etc.*

However, standards like IMS-LD (2003) and IEEE LOM (2002) start from the principle that even though learning theories are not pedagogically neutral, neutral reference models and standards can still be designed: '*The aim is not to set up a prescriptive model but an integrative pedagogical meta-model which is neutral since it models what is common with any pedagogical model*' (Koper, 2001); this assumption promotes the concept of de-contextualized learning objects that can be specified once, and then reused to design learning scenarios relying on instructivist (acquisition metaphor) or constructivist (knowledge creation metaphor) principles.

This chapter proposes another way to address the design of learning scenarios. On the one hand, we consider that socio-constructivist learning scenarios must be designed in context. On the other hand, we think that even the final results of an instructional design (ID) process should clearly state the mapping between the contextualized activities specified by designers and the functionalities provided by a given learning management system (LMS).

In the first section, we present various on-going research work focusing on languages defined

25 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/visual-design-coherent-technology-enhanced/22097

Related Content