HANDBOOK OF
Visual Languages for
Instructional Design
Theories and Practices

Luca Botturi and S. Todd Stubbs
The more complex instructional design (ID) projects grow, the more a design language can support the success of the projects, and the continuing process of integration of technologies in education makes this issue even more relevant.

The Handbook of Visual Languages for Instructional Design: Theories and Practices serves as a practical guide for the integration of ID languages and notation systems into the practice of ID by presenting recent languages and notation systems for ID; exploring the connection between the use of ID languages and the integration of technologies in education, and assessing the benefits and drawbacks of the use of ID languages in specific project settings.

**Topics Covered**

- Aesthetics in instructional design
- Comparison of visual instruction design languages
- UML visual language
- CPM visual language
- Culture-based model
- Design drawing
- Educational environment modeling language
- End-user understanding of design languages
- Globalization of instructional design
- IMS learning design notation system
- Instructional design
- Instructional software design and production
- LDL visual language
- Learning flow patterns
- MoCoLADe visual language
- MOT+ visual language
- Paper drawing as means to innovation
- Performance case modeling
- pcEML visual language
- Technology-based learning systems
- Visual design tools
- Visual instructional design languages
Chapter XVI

Comparing Visual Instructional Design Languages: A Case Study

Luca Botturi
University of Lugano, Switzerland

Daniel Burgos
The Open University of the Netherlands, The Netherlands

Manuel Caeiro-Rodriguez
University of Vigo, Spain

Michael Derntl
University of Vienna, Austria

Rob Koper
The Open University of the Netherlands, The Netherlands

Patrick Parrish
The COMET® Program/University Corporation for Atmospheric Research, USA

Tim Sodhi
The Open University of the Netherlands, The Netherlands

Colin Tattersall
The Open University of the Netherlands, The Netherlands

ABSTRACT

This handbook testifies that research on VIDL is lively, and has produced a number of interesting design languages and tools. This chapter wants to support readers in understanding the similarities and differences of some of the VIDL presented in the previous chapters, not in theory, but applying them to a specific instructional design case.

Copyright © 2008, IGI Global. Distributing in print or electronic forms without written permission of IGI Global is prohibited.
CONCLUSION

We have showed how this specific case study can be fully modeled with IMS-LD. IMS-LD is intended to model pedagogies and not specific services. Services are taken into consideration along the full learning flow although the actual delivery is made by any external provider. In order to model such a case an IMS-LD editing tool (Bolton, 2004; Miao, 2005; Van der Vegt, 2005) or a general purpose XML based tool (Altova, 2006) are needed (see Chapter XIII for some further analysis). In addition, an engine and a player capable to run the Unit of Learning are also needed, like CopperCore (Vogten & Martens, 2005), LRN or Sled (OUUK, 2005). But tools need still a low level knowledge of the specification to come across with this type of results, involving Level B elements. Therefore, the modeling is technically possible but still too difficult for a non-technical user. Editing facilities need to be more accessible to non-technical users in order to develop, implement and reach an easier and further use of this type of case studies in reality. A higher level visual metaphor would help. In this direction, some LD visual editors are currently under development (TENCompetence, 2005; UCM, 2006) but with no actual outcome to be used yet.

REFERENCES


Figure 28. Definition of the method: Who does what and when
Miao, Y (2005) *Cosmos LD Editor*.

ENDNOTE

1 Regarding tool support, to create a course's coUML models, the only "tools" really required are pencil and paper. Electronic versions can be created with simple drawing tools or with any office application, as it makes use of basic drawing shapes only. There is currently no freely available editor for coUML models.