

A Classification Framework for Educational Modeling Languages in Instructional Design

Luca Botturi¹, Michael Derntl², Eddy Boot³, Kathrin Figl²

¹ *NewMinE Lab, University of Lugano, Switzerland*
luca.botturi@lu.unisi.ch

² *Research Lab for Educational Technologies, University of Vienna, Austria*
michael.derntl@univie.ac.at; kathrin.figl@univie.ac.at

³ *TNO Human Factors, The Netherlands*
eddy.boot@tno.nl

Abstract

The integration of advanced learning technologies in education has made the design and development of instructional units and courses a complex task. Instructional design languages are proposed as a conceptual tool to achieve more creative design solutions and to enhance communication in design teams. This paper reviews the state of the art in the development, application, and research concerning the use of design languages in education and e-learning. The review reports on relevant literature and on both theoretical and empirical studies. As basis for further research, the authors propose a taxonomy of design languages and a framework for possible application of design languages in instructional design and e-learning practices.

1. Introduction: About Design Languages

When an architect is in charge of designing a new house, s/he usually starts – right after what an engineer would call a requirements analysis – with some sketch about the division and uses of the available space. The architect would then refine this and translate the design solution into a representation that the client could see, understand and discuss, and then into some executive plans that she would hand out to the construction staff. Architects exploit a number of such visual representations of the design for analyzing design problems, thinking about solutions, and communicating with stakeholders and other partners. The ability to use such visual representations is a core part of an architect's skill set – as it is for graphic designers, software designers, musicians, and for all those involved in a design activity with a long tradition.

The rise of e-learning, in combination with the introduction of “new learning” models, has significantly increased the complexity of learning technology [1]. As a response, educational modeling languages are emerging in Instructional Design (ID) as new conceptual tools in order to deal with this complexity. To the purposes of this paper, we will call them instructional design languages, shortened to *ID languages*.

Currently, little is known about the character of instructional design languages and their potential applications. In general, the use of ID languages allows the development of reflective practice and potentially enhances a more thorough understanding and reuse of eLearning solutions in lifelong learning.

The goal of this paper is twofold, namely classifying existing ID languages and presenting their potential applications to the ID practice. The next section introduces some basic concepts about design languages in general, while Section 3 briefly reviews existing ID languages. Section 4 describes a classification framework for ID languages based on their formal features, and then exploits the framework to classify existing design languages. Section 5 proposes a simple classification of the possible uses of design languages in education, and tries to apply it to the reviewed languages. Finally, Section 6 presents a summary of research studies about the use and effectiveness of design languages in this field.

2. Basic Concepts

Before approaching the core of the paper it is important to clarify some basic terms.

A *design language* is a set of concepts that support structuring a design task and conceiving solutions [2]. For example, “group work” or “online resource” can be two concepts belonging to an online instruction

design language. Design languages can include any number of concepts and relationships between them, thus extending their lexicon and syntax at different degrees. This can make the language more expressive and at the same time more complex and potentially difficult to learn. In order to make complexity manageable, some design languages are organized into *layers* that capture different perspectives. For example, the design diagrams of the Unified Modeling Language (UML) [3] capture different features or views of the same application; likewise, the web design language W2000 [4] is organized into three layers: hyperbase, access, and publishing.

A design language is a mental tool, but can be expressed, and thus turned into a communication means, through a *notation system*, i.e., a set of signs and icons that allow representing a design problem or solution so that it is perceivable by our senses [2]. Notation systems are usually visual, mostly exploiting text and diagrams, and support the production of design documents. If representing layered design languages, notation systems usually exploit multiple diagram types. For example, UML's standardized diagrams provide a standard notation system for different conceptual layers of UML. It is also possible to have "private" or idiosyncratic design languages with no notation system – indeed a common situation in less structured design fields such as game design [5] or instructional design [6]. Notation systems can vary considerably: Some of them are formal and standardized (such as the diagrams of UML), while some of them are rather sketchy (such as the diagrams of E²ML, see below). And of course, designers can use sketchy versions of formal notation systems.

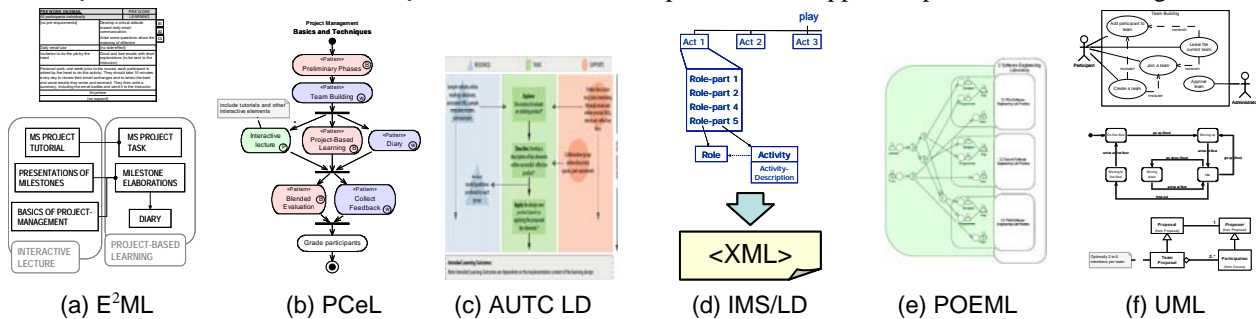


Figure 1: Examples of visual notations of selected ID languages.

The combination of design language and notation system is a central concept in the definition of a design team or community, as a shared language is the medium for the creation of shared culture. From a practical point of view, a language is fundamental for a community to share their practices [7] and to engage in reflective thinking (see for example Schön's

"reflection on action" [8]). The use of design languages allows designers to generate and share *design patterns*, i.e., the gist of a design solution to be adapted and reused over and over again [9] [10]. Finally, it is important to notice that in any complex design activity, design languages complement other design languages. For instance, in e-learning, instructional design languages are complementary with hypermedia design models, etc. [11].

3. ID Languages Review

This section briefly presents and provides references for some recently proposed design languages mainly in the field of education and e-learning. To give the reader a glimpse of how the visual notations of these languages look like, we included a small example of each in Figure 1.

E²ML [6] [12] is a simple design language coupled with a visual notation system consisting of multiple interrelated diagrams. It was developed as a thinking tool for instructional designers and for enhancing communication within large e-learning projects.

The *PCeL pattern* initiative [13] defines a UML profile for modeling learning scenarios (activities) and environments (structures). Its primary use lies in the context of blended learning, where a clear view on the online and face-to-face elements of course design is essential for deriving appropriate web support [14].

In the AUTC project [15] a number of ICT-based learning designs are proposed as generic guides (e.g. for "structured problem solving") based on concrete implementation examples, much in the spirit of design patterns. The approach provides scenario diagrams for

visually arranging involved resources, tasks, and so-called "supports."

IMS Learning Design (IMS/LD) [16] is based on the Educational Modeling Language [17] and defines a modeling technique and XML binding for describing roles and activity sequences within an environment of learning objects and services. Properties, conditions

and notifications are provided at further levels. The primary goal is to provide a means for exchange and (semi-) automatic execution of learning designs.

The *perspective-oriented educational modeling language* (POEML) [18] integrates workflow and groupware aspects into educational modeling and focuses on a separation of eleven different perspectives of educational practices (e.g., social, organizational, temporal, etc.). It can be used to model educational scenarios on different aggregation levels (lessons, curricula), and it offers a set of patterns for modeling in each of the perspectives.

UML [19] was originally proposed as a software design language, but some of its diagrams (e.g., use cases or interaction diagrams) can also be used for supporting communication in e-learning development by modeling particular instructional situations, or the application of instructional applications. It is also widely known and referenced, so we include it here.

4. ID Languages Classification Scheme

Different design language features address different ways of thinking: A highly formal language like UML fits the way of thinking of a more accurate and technically-oriented person better, while rather sketchy, informal languages are more suitable for creative and intuitive mindsets. The main goal of this section is proposing a classification scheme that grasps and organizes some basic differences in ID languages based on their formal features, i.e., the static properties of a design language, characterizing its “interface” to the users. The classification is partly a generalization of the framework developed by Boot, Nelson, van Merriënboer and Gibbons [22] presented below, and is comprised of an array of five features, labeled F1–F5:

F1: Stratification (nominal: *flat, layered*). A layered language offers a set of tools or representations for describing entities of different types, such as people and roles, activities, or learning materials. On the other hand, a flat language would collect entities of all types into a single representation. For example, UML takes a layered perspective; Gibbons [20] proposes a structure of 7 layers in order to organize the discussion about ID languages.

F2: Formalization (interval: *formal, informal*). A formal language defines a stringent, closed set of concepts and rules for composition of concepts in order to describe designs. For instance, XML or UML are formal languages, while sketches or dialogs are more open and informal. Other design languages may combine formal and informal descriptions.

F3: Elaboration (ordinal: *conceptual, specification, implementation*). Each particular design

language is able to provide more or less detail of a specific artifact. The three levels of elaboration are taken from Fowler [21]: The conceptual level allows for a general, aggregate view on the design, indicating its rationale and main elements; the specification level provides means for a more comprehensive description, including all elements; the implementation level represents the highest level of detail achieving maximum precision.

F4: Perspective (nominal: *single, multiple*). While layered languages foresee the use of multiple representations for different entities, multiple-perspective languages exploit different tools for representing more than one view *on the same entities*. For example, E²ML offers two overview diagrams, one for chronological relationships among learning activities, and one for structural relationships. Note that both perspectives can be at the same level of elaboration and located on the same layer, i.e., F4 is independent from F1 and F3.

F5: Notation System (nominal: *none, textual, visual*). If a language exposes a notation system, this can be primarily non-visual (=textual, e.g. IMS/LD) or visual (e.g. UML).

Table 1 proposes a tentative classification of the reviewed ID languages according to the scheme introduced in above. Note that we tried to depict the *primary* value for each feature (e.g., even though E²ML can be used to model at different levels of detail, its intended primary elaboration level is “conceptual”).

Table 1: Classification of selected ID languages.

	<i>Stratification</i>	<i>Formalization</i>	<i>Elaboration</i>	<i>Perspective</i>	<i>Notation</i>
<i>E²ML</i>	Flat	Semi-formal	Conceptual	Multiple	Visual
<i>PCeL</i>	Layered	Semi-formal	Conceptual	Single	Visual
<i>AUTC</i>	Flat	Informal	Specification	Single	Visual
<i>IMS/LD</i>	Layered	Formal	Specification	Single	Textual
<i>POEML</i>	Layered	Formal	Implementation	Multiple	Visual
<i>UML</i>	Layered	Formal	Conceptual / Specification	Multiple	Visual

4. Possible Applications for ID Languages

The proposed analysis of formal features clearly indicates that there are differences among ID design languages. But how are they used? In what situation can they make a difference? Currently, the literature does not provide sound evidence of the real applications of ID languages – actually, not even for UML in the domain of software development.

Nevertheless, each design language was developed with a specific use framework in mind. In this respect this section proposes a simple framework for classifying the possible applications of design

languages and then it describes the target applications for which each of the ID languages in our review was originally developed.

Communication. The first axis in the application framework concerns the main objective of the ID language, with two values: (a) *Reflective* (personal) means that the language is used primarily for personal creative thinking. This is useful for formally-bent or visually-oriented people and for designers in the first conceptual stages of design in which they do not yet collaborate with other designers and stakeholders; (b) *Communicative* (community) indicates that the language is used to communicate with other designers or stakeholders. This is useful for interdisciplinary design teams involving different views/roles.

Creativity. The second axis describes the relationship between the design language and the generation of design solutions: (a) *Generative* means that the language can be used as a means of exploring the design space and creating and refining design solutions and alternatives, e.g. during redesign. (b) *Finalist* means that it is used to formalize and “freeze” the final design solution, e.g., for creating a final IMS/LD specification of an e-learning module.

The axes are laid out in the *usage classification box* shown in Figure 2, along with the areas occupied by the languages considered in this paper.

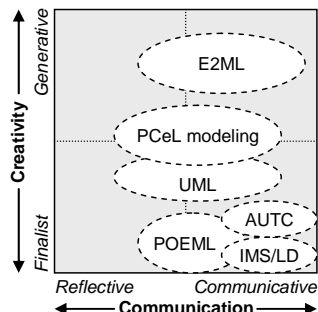


Figure 2: Uses of selected ID languages.

Note that a single design language need not to be located on a single spot, but may occupy a range on an axis or an area in the classification box, respectively. For example, most languages can be used for reflective *and* communicative purposes. Furthermore, note that the generative use of a design language is not to be confused with Alexander’s [9] notion of generativity of patterns: You can describe generative patterns with a finalist language. Finally, note that formal features of ID languages do not limit their possible use, or even abuse. For example, the formal features of a car (physical features such as size, or economical features such as costs and fuel consumption) do not limit the

use of a four-wheel drive for both heavy transportation as well as a glamorous car for MTV-clips.

5. Studies on ID Languages

Most literature on ID languages concerns the specification of such languages and their theoretical justification. However, the first empirical studies are starting to appear.

Boot, Nelson, van Merriënboer, & Gibbons [22] compared the efficiency of communication by means of ID languages between instructional designers and software programmers. A 3D-model was constructed to support instructional designers to select ID languages based upon their need for three of the formal features as described above, namely stratification (F1), formalization (F2), and elaboration (F3). Two conditions were created. Both conditions used a layered ID design language (F1), but the first condition used mainly informal descriptions (F2) with less detail (or elaboration, F3) whereas the second condition used mainly formal descriptions (F2) with more detail (F3). The results showed that the 3D-model is able to support instructional designers in selecting ID languages that suits their needs, and that more formal, detailed ID design languages communicate the instructional design more efficient to software programmers. Also a study by Botturi [23] indicates that instructional designers perceive design languages in general, and E²ML in particular, as a potentially useful tool for their practice. The participants in this study indicated that a design language can be useful for making consistent design decisions (e.g., matching goals and specific design decisions), and to revise the project development at different stages. Accordingly with the purposes for which it was developed, the participants found E²ML mainly useful as a generative tool, with a finalist use in the archival and reuse of designs. The main issue identified in the study is the steep learning curve of design languages, especially for non-visually-oriented people, which might hinder its use. This study also proposes a structured framework for evaluating the communicative impact of the introduction of a design language in a team’s or community’s practice. In short, there is small but growing evidence that ID languages are useful, although there is a huge space of improvement. The main point seems to be to collect good practices and case studies, in order to verify the effectiveness of design languages; second, to make ID languages easily learnable by potential users.

6. Conclusions

In this paper, we started with the definition of some basic concepts about design languages and notation systems for exploring their application in instructional and e-learning design. Subsequently, we proposed a set of five formal features to describe them, and we tried to classify a set of existing design languages according to these features. Possibly, other features are relevant, and should be further investigated. Our point is that a structured classification of design languages is important in order to identify particular affordances and (dis)advantages of each language (or Learning Technology standard) in order to evaluate its suitability to specific projects or situations.[24]. We then defined possible uses or applications of design languages in this field, in order to provide a glance of their potential uses. Again, our framework identifies a subset of applications, which might be extended with the growth of this research field. Finally, we proposed a short review of studies about ID design languages. Empirical studies show that (support in) selecting and using appropriate ID language is beneficial to instructional designers. One of the implications of those studies is that the use of ID languages should be an important issue in training and supporting instructional designers. Therefore, further research such as the study described in this paper is required, in order to validate and provide support tools for this purpose.

7. References

- [1] W.M.G. Jochems, J.J.G. van Merriënboer, and R. Koper, *Integrated E-Learning: Implications for Pedagogy, Technology and Organization*. London: Routledge Farmer, 2003.
- [2] A.S. Gibbons, and E.K. Brewer, "Elementary principles of design languages and design notation systems for instructional design". In J.M. Spector, C. Ohrazda, A. Van Schaack, and D. Wiley (Eds.), *Innovations to instructional technology: Essays in honor of M. David Merrill*, Lawrence Erlbaum Associates, Mahwah NJ, pp. 111-129.
- [3] G. Booch, *Object-oriented analysis and design with applications*, Benjamin/Cummings, Redwood City CA, 1994.
- [4] L. Baresi, F. Garzotto, and P. Paolini, Extending UML for modeling web applications, *34th Annual Hawaii International Conference on System Sciences*, Maui, 2001.
- [5] K. Salen, and E. Zimmermann, *Rules of Play*, MIT Press, Cambridge MA, 2004.
- [6] L. Botturi, "E²ML - A Visual Language for the Design of Instruction", *Educational Technologies Research & Development*, in press.
- [7] J. Lave, and E. Wenger, *Situated learning. Legitimate peripheral participation*. Cambridge University Press, Cambridge MA, 1991.
- [8] D. Schön, *The Reflective Practitioner. How professionals think in action*, Temple Smith, London, 1983.
- [9] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel. *A Pattern Language*. Oxford University Press, New York, 1977.
- [10] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA, 1995.
- [11] J. Armani, L. Botturi, L. Cantoni, M. Di Benedetto, and F. Garzotto, "Integrating Instructional Design and Hypermedia Design", *EDMEDIA 2004 (vol. 1)*, AACE, 2004, 1713-1719.
- [12] L. Botturi, "E²ML - Educational Environment Modeling Language", *EDMEDIA 2003 (vol.1)*, 2004, 304-311.
- [13] M. Derntl, "Patterns for Person-Centered e-Learning," PhD thesis, Faculty of Computer Science, University of Vienna, Austria, 2005.
- [14] R. Motschnig-Pitrik and M. Derntl, "Learning Process Models as Mediators between Didactical Practice and Web Support," *Proc. ER 2005 (LNCS 3716)*, Springer Verlag, pp. 112-127.
- [15] AUTC, "Learning Designs," available at <http://www.learningdesigns.uow.edu.au/>, 2003.
- [16] IMS Global, "IMS Learning Design Specification," <http://www.imsglobal.org/learningdesign/>, 2003.
- [17] R. Koper, "*Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML*," Technical Report, Open Univ. of the Netherlands, 2001
- [18] M. Caeiro-Rodriguez, L. Anido-Rifon, and M. Llamas-Nistal, "POEML: A Perspective-Oriented Educational Modeling Language Meta-model for Engineering E-learning Practices," *Proc. WWW 2006*, to appear.
- [19] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley, Reading, MA, 1999.
- [20] Gibbons A., "What and how do designers design? A theory of design structure", *Tech Trends*, 47, AECT, 2003, pp. 22-27.
- [21] M. Fowler, *UML distilled: A brief guide to the standard object modeling language*. Addison Wesley, Boston, MA, 2003.
- [22] E. Boot, J. Nelson, J.J.G. van Merriënboer, and A.S. Gibbons, "Stratification, Elaboration, and Formalization of Design Documents: Effects on the Production of Instructional Materials", submitted.
- [23] L. Botturi, "Visual Languages for Instructional Design: an Evaluation of the Perception of E²ML", *Journal of Interactive Learning Research*, 16(4), 2005, 329-351.
- [24] H.H. Clark, and S.E. Brennan, "Grounding in Communication". In L.B. Resnick, J.M. Levine, and S.D. Teasley (Eds.), *Perspectives on Socially Shared Cognition*, American Psychological Association, Hyattsville MD, 1991, pp. 127-149.