A Domain–Specific Modeling approach for supporting the development of Visual Instructional Design Languages and tools

Pierre Laforcade, Boubekur Zendagui
LIUM/IUT de Laval
52 rue des Docteurs Calmette et Guérin
53020 LAVAL - FRANCE

FirstName.LastName@lium.univ-lemans.fr

Abstract

In this paper we present, discuss and illustrate our Domain-Specific Modeling orientation for helping communities of instructional designers to specify Visual Instructional Design Languages and to develop dedicated user-friendly graphical editors.

1. Context

Since the CPM language proposition [1], a visual language for defining and specifying Problem-Based Learning situations that has been tooled thanks to a dedicated Unified Modeling Language profile and a customization of the Objecteering CASE-tool, we have followed as a red-line a Model-Driven Engineering (MDE) approach for dealing with Visual Instructional Design Languages (VIDL) [2] and user-friendly graphical editors [3]. Finally our past experiences about graphical representations of learning scenarios and scenario transformations between different Educational Modeling Languages (EML) [4], lead us to the more focused Domain-Specific Modeling (DSM) orientation [5].

The DSM [6] is a software engineering methodology for designing and developing systems that involves the systematic use of graphic DSM-Languages to represent the various facets of a system. Several technical approaches and tools coexist, all proposing metamodeling techniques for expressing domain-specific vocabularies (abstract syntaxes), and proposing facilities to construct various notations (concrete syntaxes). These tools generate powerful and user-friendly dedicated editors for DSM languages.

2. DSM approach of VIDL

We proposed in [5] three categories for learning scenarios and languages from a separation of concerns reflecting different communities of practices.

Practitioners-centered VIDLs/Scenarios: the vocabulary is the one shared by a pluridisciplinary design team. They aim at easing the definition of learning scenarios, acting as a design guide and a support of thinking.

Abstract VIDLs/Scenarios: the vocabulary aims to be independent from any Learning Management System (LMS) and to be at a sufficient level of abstraction in order to support the pedagogical diversity while promoting exchange and interoperability of scenarios.

LMS-centered VIDLs/Scenarios: the vocabulary is specific to a dedicated LMS or other Technology Enhanced Learning (TEL) systems in order to act as a guide for the manual configuration of the technical systems by humans as well as for automatic configuration by machines when possible.

These categories share fuzzy frontiers between each other. We do not think instructional design processes handling learning scenarios must systematically follow all these categories. We do not propose a systematic way to transform scenarios from one to another. On the contrary we think that designers must be free to decide which EML/VIDL is useful according to their objectives and target public (human or machine interpretation). In our mind we would like to help the emergence of communities of practices about VIDLs.

We think that it is more important to provide instructional designers with techniques and tools for helping them in elaborating VIDLs and dedicated tools than proposing yet another specific VIDL or editor.

Within a DSM approach, such VIDLs have to provide productive models (not contemplative) that will rely on a double-notation (graphical/textual) in order to be both human-readable and machine-interpretable (no ambiguous semantics) to guide specific executions (simulations, predictions, etc.).
We encourage the explicitation of vocabularies via meta-modeling techniques because metamodels are productive artefacts that can be both used by machine to generate editors, realize scenarios transformations between VIDLs, etc., and by humans to compare the pedagogical expressiveness helping so the reuse, integration and transformation of scenarios/VIDLs.

The key-point of this DSM approach concerns the transformation from one type of scenario to another, to gain the objectives of the targeted category, when changed, or to exchange and reuse scenarios with other communities of practices that do not share the same learning domain, or to adapt to a different target public by changing the format of the learning scenarios.

3. Experiencing Eclipse DSM tools

Our proposition needs concrete DSM tools and techniques: tools for defining domain-oriented VIDL, tools/techniques for transforming learning scenarios. We chose to use a unified set of modeling frameworks, tooling, and standard implementations from the Eclipse Modeling Projects: EMF (main metamodel-oriented framework), GMF (graphical framework) and ATL (model-to-model transformation framework).

These tools have been experimented within several projects of different scopes for two of the three categories: a VIDL and an editor specific to a TEL system (the LEA system) have been proposed as well as practitioners-centered VIDLs and editors.

We illustrate in figure 1 the use of these tools for a very basic VIDLoader about a UML UseCase-like learning modeling for the specification of performing relations between roles and learning activities at a high-level of abstraction, and precedence/following relations between learning activities in a learning phase. According to the EMF/GMF engineering process, we have successively designed the domain model (metamodel on left of figure 1), the graphical definition model, the tooling definition model, and the mapping definition model. After a code generation step, a specific editor is generated. Scenarios realized with this editor are both human-readable (right of figure 1) and concretely serialized in a machine-interpretable format (XMI).

4. Issues

Although the DSM Eclipse tools have been successfully used to guide and generate 100% of final graphical editors, such tools still require a certain amount of technical skill. Also, the metamodeling technique require some experience. Indeed, specifying a metamodel is not a trivial activity. We still have to study the limits of expressiveness of this technique.

More complex user-friendly learning editors (with for example multiple views – structural at different levels, pedagogical, social, etc. – for a same learning situation) can be built on top of editors generated by DSM tools but software engineers will have to develop expertise of such frameworks. Finally, learning scenarios transformations have to be experimented in order to bridge the gap between VIDLs communities of practices built thanks to this DSM approach.

5. References