

RE-ENGINEERING OF TECHNOLOGY ENHANCED LEARNING SYSTEMS

The Case of the Apprenticeship Electronic Booklet

Pierre Laforcade, Lahcen Oubahssi and Philippe Cottier
*Université du Maine, LIUM (Laboratoire d'Informatique de l'Université du Maine)
IUT de Laval, 52 Rue des drs Calmette et Guérin, F-53020 Laval Cedex 9 France*

Keywords: Re-engineering, Technology Enhanced Learning Environments, Software Engineering, Model Driven Engineering, Adaptability.

Abstract: Maintain and operate TELs systems in real situations generally requires for developers to carry out re-engineering activities to find some adaptability solutions for TELs' users. In this article we illustrate and discuss a techno-centric aspect of re-engineering realized on an existent TEL system: the Apprenticeship Electronic Booklet. The first version had been found too rigid by its end-users in regard to the roles management and to the underlying academic structures. In order to improve this TEL system, two approaches of re-engineering have been conducted. The first solution focuses on a more classical internal modification of the system functionalities. The second re-engineering work follows a Domain-Specific Modeling approach that led us to propose a graphical editor communicating with the TEL system. This external component aims to provide end-users with a more user-friendly 'editor' to configure booklets.

1 INTRODUCTION

Maintain and operate *Technology-Enhanced-Learning* (further TEL) systems require developers to carry out re-engineering activities to find adaptability solutions to end-users. The research work described in this article is part of a TEL re-engineering activity conducted within the LIUM laboratory. We focus on the adaptability of architectures and functional models (Oubahssi et al., 07). Our works aim to propose reusable solutions in order to adapt TEL systems to end-users, both from functional and techno-centered point-of-views.

In this article, we present and discuss two re-engineering approaches - re-engineering in the meaning of (Chikofsky et al., 90) - realized on a specific case-study, the *Apprenticeship Electronic Booklet* system (AEB)(El-Kechaï et al., 06). The first solution focuses on an internal modification of the system functionalities and parametrization facilities. The second re-engineering work follows a *Domain-Specific Modeling* (DSM) approach that led us to propose a graphical editor communicating with the TEL system. This external component aims to provide end-users with a more user-friendly facility to configure booklets.

2 THE APPRENTICESHIP ELECTRONIC BOOKLET

The AEB is a Web-based *Technology Enhanced Learning* environment where informations concerning the apprentice's training progression is consigned. Its goal is to help them in the appropriation of their training and to give trainers and employers the possibility to evaluate their apprentice's knowledge acquisition, to perceive their progression in the training and to regulate it.

It was designed in a participatory process involving many researchers (human science, computer science), practitioners, and future end-users (trainers, administrative staff, apprentices). The AEB was developed as a parametrizable artefact that offers different functionalities to the various end-users (El-Kechaï et al., 06). The actors of the apprentices follow-up, the tutors, can evaluate and record the apprentice informations, each user creating a common area of work and communication with others. But to do this, teachers are required to design their own upstream booklet. In practice, these functionalities have been validated, but the whole system has become too rigid. Some functionalities

are accessible only to certain actors, in accordance with the designers recommendations. This AEB version proposed an internal web page expliciting the various roles involved in the organization scheme and their relationships, where administrators were only allowed to change the terminology for web pages customization purposes.

Unfortunately, it has been found from the several apprentices' training center that the AEB was difficult to be used in new trainings whose general organization scheme does not conform to the initial one. It can happen that a same person plays different roles from those specified in the general scheme or that some parts/relations of the structure are not necessary. For examples, a company manager involved in the students formation and in charge of a student's training have to handle two roles/logins; also, it generally happens that the administration of the booklets is devoted to a teacher of the same formation. In order to concretely realize tasks attached to these roles, one have to connect himself several times to the various profiles.

We had therefore to change the booklet design as blocks of functionalities to assign to various roles according to concrete contexts.

3 AEB RE-ENGINEERING

We describe in this section the realized re-engineering into two parts. In the first one we present the internal re-engineering approach: development of the multi-role management service and realization of an internal configuration editor. In the second part, we present the external re-engineering approach: the graphical editor realized thanks to a *Domain-Specific Modeling* (Kelly and Tolvanen, 08) approach, and the import / export communication API added to the existing AEB system to ensure communications between the two computer artefacts (represented in Figure 1).

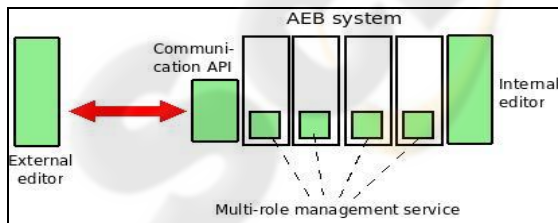


Figure 1: The communication API between external editor and internal editor.

3.1 Development of a Multi-role Management Service and Internal Configuration

The main objective of this work was to improve and facilitate the AEB's use, and meet a set of needs expressed by tool's users. Three steps were necessary to develop the multi-role management service and the internal configuration of the AEB system: extraction of the multi-role management conceptual model, modification of existing code to adapt the AEB system to the multi-role management, and addition of the internal editor and import/export facility.

To this aim, some specific re-engineering activities have been realized: the study and the analysis of both functional and conceptual models of the existing database, the reorganization of the system's functions in the form of modules (internal services), identification of the actors, definition of the rules and constraints for the multi-role management (for example the association of one or more modules of functions to an actor), definition of a new institutional setting of the AEB system, etc.

This internal re-engineering had implied the development of many lines of code in accordance to the current technological languages and choices of the current AEB version. We obtained as a result of this step:

- the specification of a new functional meta-model for the AEB system (represented in Figure 2),
- the development of an internal multi-role management service,
- a import/export facility to handle XML files describing AEB configurations.

This re-engineering work had to meet technical requirements of the database conceptual model of the original system. The development of the internal configuration of the AEB system also had the constraint to revise the design of the functional model of the system, ie. to define a new system architecture that takes into account the existing and the emerging user's needs. Finally, we obtained as a result of this work an improved version of the AEB system model and the definition of an "internal parameters module". This module allows users to plays different roles, and for each role, to perform one or more blocks of functionalities by using drag-and-drop code techniques on a PHP page (the existing AEB system was developed using this technological language).

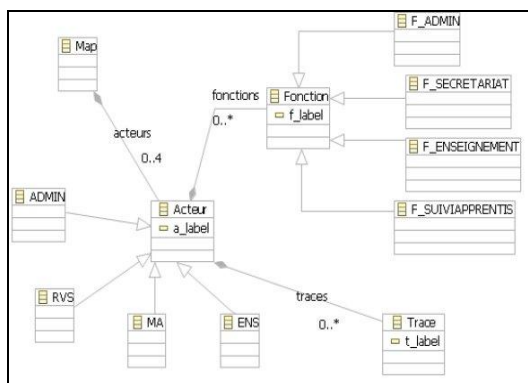


Figure 2: The functional meta-model of the multi-role management service.

3.2 Application of DSM Techniques

To anticipate new uses in the design of the AEB system and to test the system, an external editor was also developed. In opposition of the internal editor, this solution has the advantage to avoid following the technological constraints and choices from the existing system.

Similarly to the internal editor, the external one aims to graphically configure a booklet at the institutional level and to graphically ease the specification of roles and functionalities to perform for these roles. We decided to follow a development guided by models in the meaning of the DSM approach (*Domain-Specific Modeling*), because of our research results and our experience on the study and application of their theories and practices for TEL learning scenarios (Laforcade et al., 08). The development of the external editor for the institutional configuration of booklets was also for us a new opportunity to experiment DSM tools and techniques.

3.2.1 DSM Domain and Tools

The Domain-Specific Modeling (Kelly and Tolvanen, 08) is a software engineering methodology for designing and developing systems, most often IT systems such as computer software. It involves the systematic use of a graphic DSM Language to represent the various facets of a system.

All DSM tools propose meta-modeling techniques capable of expressing domain-specific vocabularies (abstract syntaxes), and propose facilities to construct various notations (concrete syntaxes). These editing frameworks are supporting the techniques and many more customizations with minimal programming effort. As a result, these tools

can generate powerful and user-friendly dedicated editors for DSM languages. They are kind of meta-CASE editors capable of generating CASE tools. The final editors give domain-designers the ability to graphically specify models from their domain, and propose some persistence facilities to load and store these models in a machine-interpreted format.

In our research works we chose to use a unified set of modeling frameworks and tools from the Eclipse Modeling Projects (Eclipse EMP, 09): EMF (main metamodel-oriented framework), GMF (graphical framework).

3.2.2 Use of the Eclipse EMF/GMF Tooling

From a DSM point-of-view, the graphical configuration of a booklet can be considered as a model of the configuration desired by the booklet's designer; this model being in conformance with a meta-model specifying the domain terminology in terms of concepts, relations, properties and constraints (Figure 2).

To achieve the design and development of this editor, several studies has been realized within several iterations and many specifications of models have been produced in accordance to the DSM approach when using the EMF/GMF tooling: the "booklet configuration" domain meta-model (reuse from the other re-engineering approach: Figure 2 is a graphical representation of the concrete 'ecore' meta-model specified with the EMF tooling), the XML schema describing how AEB configurations will be serialized, the meta-model for the graphical formalism, the meta-model for the 'palette' of the editor (set of basic concepts and relations available for drawing), the meta-model describing the mapping between the previous meta-models, etc.

At this point of the iterative process a full-generated prototype of the booklet configuration editor was generated by the EMF/GMF frameworks. When abstract (domain metamodel) and concrete (notation) syntaxes objectives have been reached, we realized some extra activities: addition of constraints to add some semantics on the domain meta-model (eg. an actor can only be instantiate once, it is not possible to add the same functionality twice for an actor, etc.), development of a *Rich-Client Platform* version (standalone), addition of support services/guidances, etc.

Finally, the resulting external editor takes the form of an application providing a drawing space in which a graphical configuration can be specified (see Figure 3). This editor can also be used to modify existing configurations specified thanks to

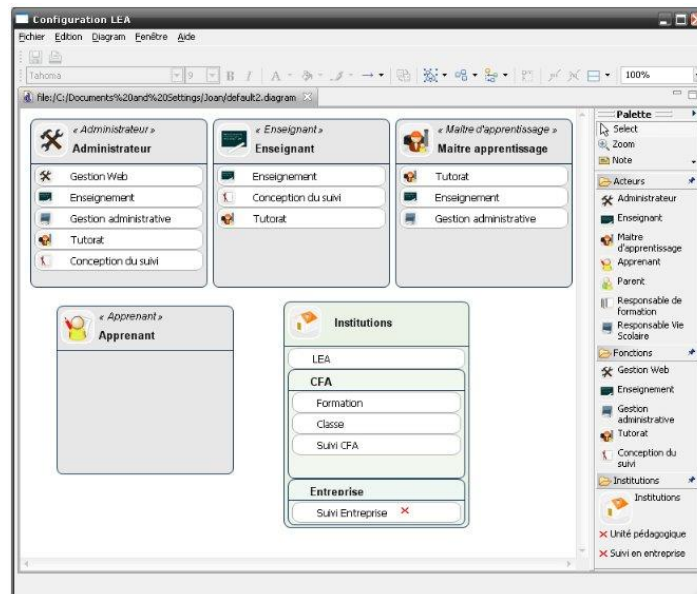


Figure 3: The external graphical editor.

the AEB internal editor (they use the same XML persistence format). Also, the AEB system can import or export a XML file configuration created with the external editor (interoperability between both internal and external editors).

4 CONCLUSIONS

In this article we discussed the problem of defining and implementing a TEL re-engineering according to a techno-centric view. Based on the re-engineering work done on the AEB system, we presented and discussed two approaches, internal and external, promoting the adaptability of architectures and functional models to facilitate their use. These two approaches, different from a development viewpoint, have required a preliminary common analysis and design to extract the functional model of the existing system. We also realized a re-engineering guided by models, in the way that the functional model identified from the multi-role system has been crystallized under the form of a domain meta-model. This functional meta-model was then used as a basis for the development of the external editor. The DSM tools that we used made it possible to exploit this meta-model to guide and generate most of the final code for the editor.

Concerning the need for the AEB system booklets configuration, we have proposed two editors, one internal and another one external to the system. Future experiments about these two different

editors will compare the ownership and usage of these tools and confirm the added-value of the external editor. Indeed, it remains to validate that the liberation of the technological choices related to the initial design of the TEL can offer more user-friendly and soundly computer artefacts.

REFERENCES

- Chikofsky J.-E. et Cross J.-H., 90. Reverse engineering and design recovery: A taxonomy. *Software IEEE*, 7 : 13–17.
- Eclipse EMP, 2009. Eclipse Modeling Projects, <http://www.eclipse.org/modeling/>, retrieved from 2009.
- El-Kechai H., Choquet C. 2006. Understanding the Collective Design Process by Analyzing Intermediary Objects, In: *ICALT'06, The 6th IEEE International Conference on Advanced Learning Technologies*, July 5-7, Kerkrade (The Netherlands).
- Kelly, S., Tolvanen, J.-P., 2008. *Domain-Specific Modeling*, ISBN: 978-0-470-03666-2, March 2008. Wiley-IEEE Computer Society Press.
- Laforcade P., Zendagui B., Barré V., 2008. A Domain-Specific-Modeling Approach to Support Scenarios-Based Instructional Design. In: *ECTEL'08, 3rd European Conference on Technology Enhanced Learning*, Maastricht, The Netherlands.
- Oubahssi L., Grandbastien M, 2007. *E-learning systems reengineering: Functional specifications and component based architecture*. Architecture Solutions for E-Learning Systems, Claus Pahl (ed.).