



HAL
open science

Re-engineering of the Apprenticeship Electronic Booklet: Adaptation to New Users Requirements

Lahcen Oubahssi, Pierre Laforcade, Philippe Cottier

► **To cite this version:**

Lahcen Oubahssi, Pierre Laforcade, Philippe Cottier. Re-engineering of the Apprenticeship Electronic Booklet: Adaptation to New Users Requirements. 2010 10th IEEE International Conference on Advanced Learning Technologies, Jul 2010, Sousse, Tunisia. pp.511-515, 10.1109/ICALT.2010.146 . hal-01226416

HAL Id: hal-01226416

<https://hal.science/hal-01226416>

Submitted on 21 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Re-engineering of the Apprenticeship Electronic Booklet : Adaptation to new users requirements

Lahcen Oubahssi, Pierre Laforcade 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
{lahcen.oubahssi, pierre.laforcade, philippe.cottier} @univ-lemans.fr

Abstract—In this article we illustrate and discuss a technocentric aspect of re-engineering realized on an existent TEL system: the Apprenticeship Electronic Booklet. Although this system has been designed with end-users following a participatory process, the first version had also been found too rigid 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 an internal modification of the system functionalities and parametrization facilities. 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 facility to configure booklets. This second work is particularly illustrated and discussed in regard to the promising underlying re-engineering process.

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

I. INTRODUCTION

Many research projects focus on the design and development of *Technology-Enhanced-Learning* (further TEL) environments or systems. Because of multiple causes, such as the difficulty for users to appropriate these systems in real situation or the functional incompleteness of TEL systems built as prototypes focusing on specific services, these research products have to be modified to continue to be used. Maintain and operate such TEL systems require developers to carry out re-engineering activities to find adaptability solutions for end-users.

The research work described in this article is part of a TEL re-engineering activity conducted within the computer science laboratory of the *Maine University* (LIUM, France). We are interested in the adaptability of architectures and functional models for TEL systems. Our works aim to propose reusable solutions in order to adapt TEL systems to end-users, both from a functional and techno-centered point-of-view. In this article, we present and discuss two re-engineering approaches realized on a specific case-study, the *Apprenticeship Electronic Booklet system* (AEB) [7]. The first solution focused 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 service to configure booklets.

The paper is organized as follows. After a brief introduction we present our research context and the re-engineering activity when considering TEL systems (section 2). We then present in section 3 the Apprenticeship Electronic Booklet system and we detail the problems and needs identified by the end-users of the first version of the AEB system. In section 4, we detail the works of re-engineering realized on the system. Finally, we conclude by discussing the possibility of reusing the re-engineering process we put into practice for other TEL contexts.

II. TECHNOLOGY ENHANCED LEARNING ENVIRONMENTS (TEL) RE-ENGINEERING AND RESEARCH CONTEXT

Re-engineering is generally associates to software. It always starts from an existing software and makes a shift of that software to anticipate new requirements [8]. The following re-engineering goals are most typical [10]:

- ◆ Moving to a new platform (technological shift that could include moving the logic to another platform in parallel to the existing version);
- ◆ Rewriting to implement another business logic, add more features etc. (process similar to implementing a new version of a product);
- ◆ Combined: shifting at the same time more than one element: the business logic, the technical base and the algorithmic structure.

From a more general viewpoint, re-engineering is defined according to [4] as “the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form”. The objective of system re-engineering is to improve the system structure and make it easier to understand.

[5] defines the educational re-engineering of TEL systems as the examination of a mediated learning and the modification of the learning in order to reconstitute it in a new form and to put in place a new teaching situation, taking better into account the uses. By analogy of the re-engineering classification of [2], [15] proposes 4 levels and types of change for re-engineering within the e-Learning domain:

- ◆ Conceptual (Re-think): deals with the background for re-engineering, e.g., new pedagogical strategy accepted.
- ◆ Requirements (Re-specify) of a module or an organization; deals with goals, provided knowledge, abilities and competencies.

- ◆ Design (Re-design): deals with learning activities, assessment types, informational, navigational, presentational structure, pedagogical scenario, etc.
- ◆ Implementation (Re-code): deals with learning resources, activities, scenarios, etc.

Our works are focusing on the lower level, although this level is generally influenced by the higher levels. For us, the implementation/code re-engineering of a TEL system is not reduced to the modification of the embedded *pedagogical intention* (ie. not only learning scenarios / activities and ressources) but also covers the modification of the core system (eg. the distant platform itself). Such re-engineering activities can be realized to take into account new functionalities, to favour adaptability services, as well as to improve appropriation to end-users.

While dealing re-engineering at implementation level, several techno-centric approaches can be realized according to various software engineering fields. For examples, mixing functional approaches has been used for the re-engineering of the modular SERPOLET platform [12][13], while others have preferred more technological-centered approaches like the use of educational software component [14], or SOA/Web services [3] [1], etc.

The re-engineering activity may also require some models of the end-users use in order to allow both abstraction and study of the details of the existing system, and then to guide, assist or even drive the changes, fixes or required transformations. These models can be used to detect anomalies, to make corrections, to develop automatically regression tests or to generate more complete corrections.

In this article we present and discuss two different software engineering approaches for the re-engineering of a TEL system at the implementation level. From the previous classification of software re-engineering goals [9], we can consider our work as part of the 'combined' category. Indeed, the first approach aims to realize the intern re-engineering of an existing TEL environment, i.e. to review the design and development of one part of the software in order to implement a new domain logic. In opposition, the second approach deals with technological shift as it aims to cover the new needs by realizing an external re-engineering: new functionalities are embedded in additional software artefacts out of the pre-existing single system, and a communication interface have to be added internally in order to ensure the interoperability of the different tools.

III. THE APPRENTICESHIP ELECTRONIC BOOKLET

The AEB is a *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 the trainers and the employers the possibility to evaluate their apprentice's knowledge acquisition, to perceive their progression in the training and to regulate it. The AEB project is the result of a partnership between the computer science laboratory of the *Maine University* (LIUM, France) and the apprentice's training center of the *Mayenne* cities (CFA, France).

It was designed in a participatory process involving researchers in the human science, computer science,

practitioners of the dual training (trainers, administrative and company) and apprentices. The AEB was developed as a parameterizable artifact that offers different functionalities to various end-users [7]. The actors of the apprentice following-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. The AEB functionalities are the result of the materialization of a set of scenarios described by the participants during the design process. In practice, these functionalities have been validated, but the whole system has become too rigid. Some functionalities are only accessible to certain actors, according to the designers recommendations.

Unfortunately, it has been found from the several apprentices' training center using the AEB system, that the AEB is too difficult to be used in new trainings whose general organization scheme does not conform to the initial one. For example, 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 him-self several times to the various profiles: the administrator making the management of the site, the training responsible developing the training booklet for several apprentices and tutors, the teacher managing classes of apprentices/tutors, etc.

We have therefore to change the booklet design as components to assign to actors, according to new situations: multiple roles for multiple functions. Two technological options have been taken: providing opportunities for booklet's designers to adapt the AEB system "on-line" or "off-line", according to different design contexts that can be met. The first one allows to re-set in-use a booklet, the second allows, in an educational team, to work locally on a machine with some configuration informations before deploying them on a AEB already available on a server. The main objective of this work is to improve and facilitate the use of the system. To do this, the system configuration exceeds the simple change of terminology, as allowed by the current official version, in order to adapt AEB systems to each institution and to offer modular functionalities and re-assignment of roles appropriate to each context.

IV. RE-ENGINEERING OF THE APPRENTICESHIP ELECTRONIC BOOKLET

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* [9] approach, and the import / export communication API added to the existing AEB system to ensure communications between the two computer artifacts (represented in Figure 1).

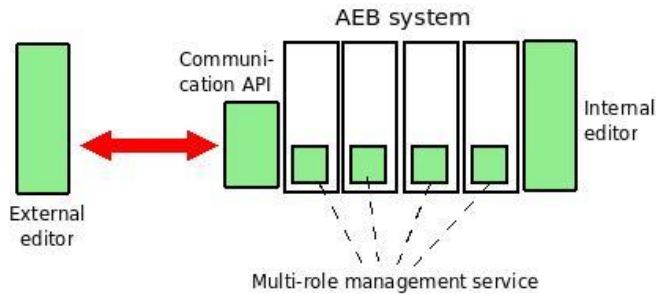


Figure 1. The communication API between external editor and internal editor

A. Development of a multi-role management service and internal configuration

The main objective of this work is 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.

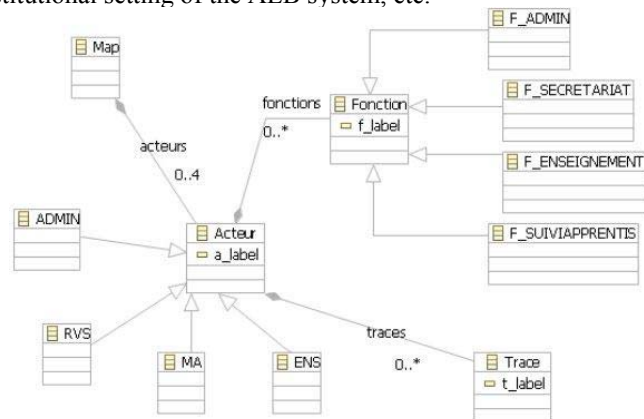


Figure 2. The functional meta-model of the multi-role management service

This internal re-engineering have implied the development of many lines of code in accordance to the current technological languages and choices of the current AEB version. We obtain 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 (represented in Figure 3),

- ◆ an 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 obtain 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 (Figure 3) by using drag-and-drop code techniques on a PHP page (the existing AEB system was developed using this technological language).



Figure 3. The internal graphical editor

B. Application of DSM techniques for the development of the external editor

To anticipate new uses in the design of the AEB system and to test the system, an external editor was also developed.

Similarly to the internal editor, the external editor 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 *Model Driven Engineering*, and more particularly in accordance to 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 engineering and re-engineering [11].

1) DSM domain and tools

The *Domain-Specific Modeling* [9] 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.

Several technical approaches coexist to support the specification of DSM. All these 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, tooling, and standard implementations from the Eclipse Modeling Projects [6]: EMF (main metamodel-oriented framework), GMF (graphical framework) and ATL (model-to-model transformation framework).

2) Use of the Eclipse EMF/GMF tooling

In the AEB case, the development of the external editor for the institutional configuration of booklets has set up a new field of experimentation and validation of the DSM approach. Indeed, 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 would then be in conformance with a meta-model specifying the domain terminology in terms of concepts, relations, properties and constraints.

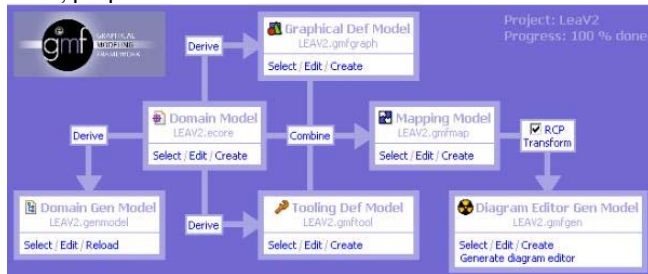


Figure 4. The GMF process and models to produce

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 (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. (see Figure 4 for a view of these various models and steps).

At this point of the iterative process a full-generated prototype of the booklet configuration editor was generated by the Eclipse tools. Finally, when this editor has been considered as in accordance with the domain and graphical objectives, we realized some additional activities, some of them with the help of the Eclipse framework, other ones requiring the writing of code:

- ◆ development of the editor as a plug-in for Eclipse at first,
- ◆ addition of constraints (use of OCL - *Object Constraint Language*) 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 the *Rich-Client Platform* (RCP) version in a second time (standalone),
- ◆ addition of support services / guidance (eg. cheat sheets for end-users, splash-screen, etc.),
- ◆ development of specific versions for different operating systems.

At the end, the external editor we produced takes the form of:

- ◆ an application providing a drawing space in which a graphical configuration can be modeled (Figure 5),
- ◆ a palette offering different domain elements for the configuration of the booklets institutional learning (actors, functionalities, parameters, etc.),

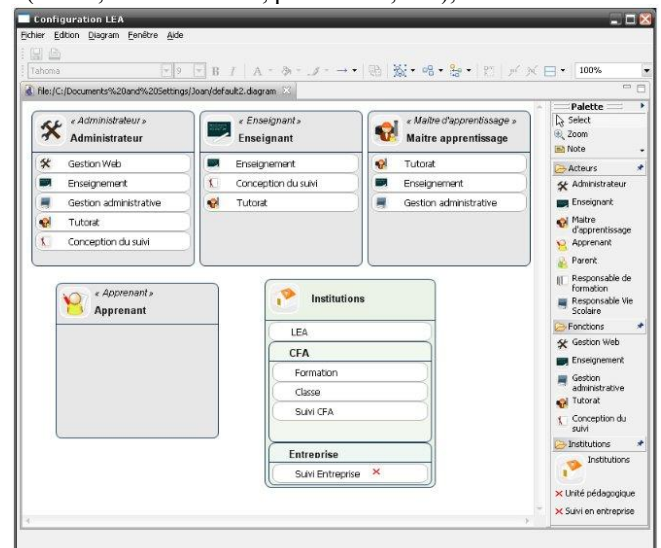


Figure 5. The external graphical editor

- ◆ a final graphical notation (set of icons and nested elements to conceptualize the functions performed by the actors), textual helping elements, etc.

This editor allows the graphical specification of configuration models but it can also open existing

configurations described thanks to this editor or via the AEB internal editor.

Finally, the AEB system can import or export a XML file configuration. It was necessary to develop within the existing AEB this new functionality according to the technological choices of the existing tool (PHP in this case). This new module plays the role of a bridge of communication between the external editor and the apprenticeship electronic booklet, and guarantees the interoperability of models configuration specified either using the graphical editor external or the internal one. Figure 6 is an example of the format used to describe this communication. This allows, in addition to a simplified design of a booklet, to consider future exchange of the booklets formats.

```
<Acteur xsi:type="LEAV2:ADMIN" terminologie="Administrateur">
  <Fonction xsi:type="LEAV2:F_ADMIN" droit="admin" nom="Gestion Web"/>
  <Fonction xsi:type="LEAV2:F_SECRETARIAT" droit="rva" nom="Gestion
administrative"/>
  <Fonction xsi:type="LEAV2:F_SR" droit="sr" nom="Conception du suivi"/>
  <Fonction xsi:type="LEAV2:F_SUIVIAPPRENTIS" droit="ma" nom="Tutorat"/>
  <Fonction xsi:type="LEAV2:F_ENSEIGNEMENT" droit="ens" nom="Enseignement"/>
  >
</Acteur>
```

Figure 6. Example of XML configuration file for a booklet

V. 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 demonstrate that the liberation of the technological choices related to the initial design of the TEL can offer more user-friendly and soundly computer artefacts.

Although the re-engineering work that we conducted was to extend the functionalities of an existing TEL while improving its ownership and its uses, we believe it can also be conducted within a context where the aim is to improve the specification of learning scenarios and their "deployment", or operationalization, on existing *Learning Management System* (LMS). We are currently preparing a similar re-engineering activity, according to the DSM approach. This work aims to help designers in specifying the

structural definition, as a first step, for a learning unit in accordance to the meaning of the MOODLE platform. We would like to provide teacher-designers with some external and graphical editors best-suited to structure successive prototypes of learning structures facilitating communication and understanding while keeping the insurance that the resulting scenarios will be compatible with the MOODLE LMS.

REFERENCES

- [1] I-Ajlan A. and Zedan H., 2007. E-learning (MOODLE) Based on Service Oriented Architecture. In Proc. of the EADTU's 20th Anniversary Conference, Lisbon, Portugal, 8-9 November, Lisbon-Portugal, vol. 1, p. 62-70.
- [2] E.J. Byrne, 1992. A conceptual foundation for software re-engineering. In Proceedings of the Conference on Software Maintenance, pp. 226-235. IEEE Computer Society Press.
- [3] Caron, P.-A., Derycke, A., & Le Pallec, X., 2005. The Bricoles project: support socially informed design of learning environment. In AIED'05, the 12th International Conference on Artificial Intelligence in Education, Amsterdam.
- [4] Chikofsky J.-E. et Cross J.-H., 1990. Reverse engineering and design recovery: A taxonomy. *Software IEEE*, 7 : 13-17.
- [5] Choquet C., Corbière A., 2006. Reengineering Framework for Systems in Education. *Educational Technology & Society*, 9(4), pp. 228-241.
- [6] Eclipse EMP, 2009. Eclipse Modeling Projects, <http://www.eclipse.org/modeling/>, retrieved from 2009.
- [7] El-Kechaï 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).
- [8] Henrad J., Hick J.M., Thiran P., Hainaut J.L., 2002. Strategies for Data Reengineering. In: Proc. Working Conference on Reverse Engineering, pp.211-220.
- [9] Kelly, S, Tolvanen, J.-P., 2008. Domain-Specific Modeling, ISBN: 978-0-470-03666-2, March 2008. Wiley-IEEE Computer Society Press.
- [10] Kumlander D., 2006. Personnel Motivating Software Reengineering. In: Proceedings of the 10th WSEAS International Conference on COMPUTERS, Vouliagmeni, Athens, Greece, July 13-15, pp. 826-830.
- [11] 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.
- [12] Ngomo N., Abdulrab H., Oubahssi L., 2005. Application Service Provider System: a new concept to provide interoperability between learning management systems. In Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, pp. 2763-2769, Norfolk, VA: AACE. Vancouver, Canada.
- [13] Oubahssi L., Grandbastien M., 2007. E-learning systems reengineering: Functional specifications and component based architecture. *Architecture Solutions for E-Learning Systems*, Claus Pahl (ed.), Edited by IGI Publishers.
- [14] Rebaï, I, De La Passardière, B, Labat, J-M, 2008. To store and retrieve software components for Interactive Learning *Environnements*: the ECR repository. *Int. J. Advanced Media and Communication*, N° 1, Volume 2, pp 73-95.
- [15] Tankeleviciene, 2008. Using reengineering techniques for distance study course improvement. In: MEL'08, the third International Conference Modern (E-) Learning, Varna, Bulgaria.