Domain Specific Supports for Design Rationale of Open Pedagogical Scenarios

El Amine Ouraiba, Christophe Choquet, Philippe Cottier

***Abstract*—Our works take place in the research field of Technology Enhanced Learning systems engineering. In this paper we deal with the design rationale of open pedagogical scenarios (OPS). We have proposed a generic model of OPS based on the QOC formalism, and an incremental/iterative engineering process of OPS. In order to provide the dedicated supports for practitioner teachers/designers we have followed a constructive approach of instructional design based on Model-Driven Engineering and Domain-Specific Modeling. To verify our proposal we took Hop3x domain as experimentation area. The Hop3x's DSEML is described by a metamodel of open learning sessions which is defined basing both on the generic model of OPS and Hop3x-specific educational domain semantic, and accordingly a graphical editor of sessions has been developed thanks to EMF/GMF tooling.**

***Index Terms*—Design rationale, QOC, open pedagogical scenario, model-driven engineering, domain-specific modeling**

I. Introduction

TEL (Technology Enhanced Learning) systems are complex environments that mobilize human agents (learner, teacher) and artificial ones in interactions conceived in order to improve the quality of the human learning [14]. The design of these systems is a significant effort for learning institutions [35]. However, these educational technologies have not always the necessary flexibility for use in real educational contexts that often requiring the rapid adaptations to new and often unexpected events [11]. Indeed, TEL environments should be designed as"open" in which the teacher himself is able to lead the adaptation and reengineering of learning system at an abstract level. We consider that a pedagogical scenario could be considered as a model, "a simplification of a system built with an intended goal in mind" [4], where the system is the TEL situation itself and the modeling goal is the organization of the learning activity.

Manuscript received October, 2011; revised November, 2011. This work was supported by the Computer Science Laboratory of Le Mans University, France. Domain Specific Supports for Design Rationale of Open Pedagogical Scenarios.

El-A. O., C. C., P. C. Authors are with the SRC Department, University's Institute of Technology of Laval, Le Mans University, France ({el_amine.ouraiba, christophe.choquet, philippe.cottier}@univ-lemans.fr).

According to [3], "such models must necessarily be open, deliberately and strategically imprecise, objects which raise the reflection [39], allowing a collaboration between researchers and practitioners".

We consider that is relevant to investigate the instructional design rationale of Open Pedagogical Scenarios (OPS) [34] that can be adapted according to execution context [36]. There are different trends in literature about learning scenarios design and adaptation [36]. Some works ([1] [2] [43] [46]) are based specially on IMS Learning Design where the learning scenarios are produced on conformity with the metamodel of this specification. However, the main problem here is that "teacher must understand the IMS-LD's concepts and metaphor" and this is not always easy in practice [36]. Moreover, considering these limits in terms of distance between the practitioners and specifications, as well as the ones of the tools proposed, others approaches (like [17]) found in the Model-driven Engineering (MDE) techniques and the graphical approaches the potential that allows the designers to work at abstract level, far from the technical complexity [34][35]. In addition, again in a MDE paradigm, Domain-Specific Modeling (DSM) approach proposes to define models using domain specific languages (DSL) based on a metamodel simpler but more focused on the business domain of designers [26]. That allows indeed a high productivity and quality by facilitating the generation of code from models designed at a high-level of abstraction through avoiding semantic losses caused by transformations.

By our work we want to overcome the difficulties a practitioner teacher can encounter when using generic EMLs and existing editors for designing Open Pedagogical Scenarios (OPS) [34] that can be adapted according to execution context. In this paper, we use the QOC (Questions, Options, Criteria) model to investigate the instructional design rationale of OPS [37]. The section 2 presents the two approaches of instructional design. The section 3 gives an overview about the design rationale and QOC and explains thereafter our investigation of the OPS design rationale using QOC. In section 4 we present the MDE/DSM paradigm and how we instantiate it for supporting teacher both at design and run time of OPS. Section 5 presents an implementation by EMF/GMF of our proposal about learning sessions of Hop3x TEL system. We conclude our paper by current and future works.

(Advance online publication: 12 November 2011)

II. TWO APPROACHES OF INSTRUCTIONAL DESIGN

The preoccupation of a teacher who wants to use computer technologies for his teaching activity is to design a teaching situation as a scenario that responds to pedagogical problem that he has [11]. The scenario could be described with the help of an Educational Modeling Language (EML) [27], defined by a specific metamodel which is itself linked by conformity relations with the scenario. Within this framework, we distinguish two approaches of instructional design [12] [18]:

- A "classical" design process here considered as "*interpretative approach*", where an existing EML (such as IMS LD) is chosen for specifying a scenario. Here, the designers (mainly the teachers) have to appropriate the semantics of the EML in order to transform their specific domain model into the metamodel of the chosen EML. However, practitioners and the literature [17] [21] notice the lack of appropriation of the EMLs semantic and the difficulty of use of existing editors in practice [34][35]. The risk when choosing a pre-defined metamodel is that certain particularities may emerge in situ (in the real learning situation). These particularities cannot be always anticipated at design-time. Their description cannot be supported by the predictive pedagogical scenario because the metamodel chosen to express it does not allow this. The metamodel should empower designers to adapt and evolve their scenarios [18]. In addition, using a generic EML imposes a modeling structure based on some metaphor. To use it, it is necessary to model the scenario in accordance with this structure [18].

- To address the inadequacies of the interpretative approach, a more iterative design process is considered as "*constructive approach*", where the designers, generally helped by modeling specialists, build the metamodel of their specific domain (and thus, their "domain specific" EML) and use it for specifying their scenarios. This approach is much closer to the DSM one and engages the designers in an iterative design process, eventually supported by reengineering phases. Moreover, according to [22]: the greatest weaknesses in instructional design are to stop at the "*'delivery"'* stage to learner the design's product. Actually, the designing activity of a pedagogical scenario must continue at runtime. According to [10]: it must not be simplified just as a preliminary modeling act of an artifact that is exogenous to the real context of its usage process; it must be continued in the activity of users themselves. This requires the development and the use of models that are endogenous to usage contexts and that may be evolving in parallel (simultaneously) with theirs metamodel.

III. DESIGN RATIONALE OF OPEN PEDAGOGICAL SCENARIOS

A. *Design rationale with QOC formalism*

According to [31], Design Rationale (DR) emphasises working with explicit representations not only of possible design solutions, but also of the reasons and processes behind them. In other words, DR is the explicit listing of decisions made during a design process, and the reasons why those decisions were made [24]. DR can be used, according to Burge and Brown [8], for many aims: design verification, evaluation, maintenance, reuse, teaching, communication, assistance, and design documentation.

DR is used by research communities in several science areas such as software engineering, mechanical design, artificial intelligence, civil engineering, knowledge management, cognitive science, and human-computer interaction research. But there is still very little experience of applying the DR in the TEL engineering area, particularly in the instructional design. Indeed, DR can be adopted as a framework for justifying the reasons behind pedagogical decisions taken at design process of units of learning, in order to allow the understanding, recreation, and/or adaptation of design production [37].

To supporting design rationale, several frameworks are proposed, such as: IBIS [9], DRL [28], DIPA [29], and QOC [31]. According to [25], the studies have concluded that the designers wanted a method requiring less effort to keep this logic, while maximizing the possibility of its reuse. For its simplicity and relevance of its elements we choose to rely on the potential of the QOC model (Questions, Options, Criteria), proposed by MacLean [31], for investigating the instructional design rationale. The QOC is a semi-formal notation which allow producing a graphical representation of DR. [31] noted that QOC can be used for representing the design space around the artifact being produced, thus situating this artifact in a broader context than would otherwise be the case [32]. According to [32] the diagrams can help designer to explain, elaborate, compare, and review design ideas and issues. As it is showed in the left side of figure 1, QOC represents design reasoning [32] as a network of "*Questions*" which highlight key design issues, "*Options*" which represent alternative solutions to these issues and "*Criteria"* to explicitly describe the methods to evaluate the options, such as the requirements to be satisfied or the properties desired. A solid line between a criterion and an option means that the criterion is favorable for option, otherwise it is unfavorable. The preferred option is framed. This allows the designer to read enough to understand the reasons for or against the various options [32].

B. Design rationale of open pedagogical scenarios

[45] notices that the pedagogical choices of a teacher are rarely made explicit. Then, it is difficult to really understand the criteria that led him/her to take such decision at the expense of another deal with a particular teaching-learning context. We choose QOC for capturing instructional design rationale in order to improve learning design quality by arguing decisions thanks to explication of design criteria and to capture evaluated variants to avoid duplication of effort in the future lifecycles.

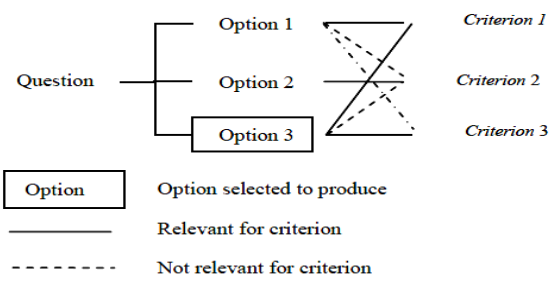


Fig 1. Model of QOC (Questions, Options, Criteria).

We have defined a model of OPS inspired from the QOC one (see fig2). In order to achieve his/her *pedagogical objective*, the teacher/designer has to define the elements of *static layer* at the design-time. However, the multitude of the possible execution contexts requires him/her to describe different ways for conducting learning session. We call these different ways "*variants*", where each *variant* is intended to be executed in a particular teaching-learning *context* characterized by a set of *indicators* describing teaching-learning circumstances, where each indicator is described with the meta-language UTL (Using Tracking Language) [13]. Thanks to the semantic open points, designer specifies the diverse variants for combining them into a single integrated model, called "*open pedagogical scenario-OPS*".

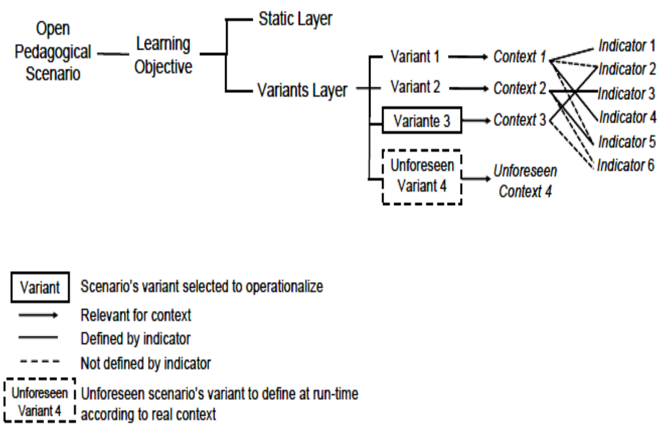


Fig 2. Model of Open Pedagogical Scenario.

C. Incremental and iterative engineering process of open pedagogical scenarios

We consider that an OPS is a continually evolving entity in different contexts in which it is executed. Its state must depend strongly on the circumstances of its use (execution). But, this poses several challenges to the teacher. He must be able to specify all parts of open pedagogical scenario, to operationalize it, to adjust it dynamically, to evaluate the relevance of performed adaptations according to the overall educational objective, to decide the capitalization of these adaptations because they are very important strategic knowledge that would be reused, in future designs of new scenarios, or in improvements of existing scenarios in a context of reengineering.

Our objective is to articulate the engineering process of OPS around the teacher. We consider this process as incremental and iterative where a teacher could late some specifications in order to fit with a given context and to avoid unnecessary specifications. We distinguish four main phases of engineering process of an open pedagogical scenario: design-time, deployment, run-time and reengineering (see Fig 3).

Design-time

Teacher has to define static elements of OPS which guarantee the achievement of the overall educational objective, and according to the foreseen teaching/learning contexts he has to specify a set of predicted variants or reuse the capitalized ones at the last lifecycles.

Deployment

Before its execution, an open pedagogical scenario must be personalized in order to meet specific circumstances of a given teaching/learning context by taking into account only the most relevant variant among all variants previously defined and capitalized. Thus, the individualized scenario can be used for a thinner design in order to produce an executable specification for generating instances that are suitable to such set of requirements. However, in order to avoid the selection of inconsistent variants that lead to undesirable situations, the validation of OPS personalization must be made at a high level of abstraction in order to ensure the persistence and consistency.

Run-time

It's the learning session's execution phase where static elements and selected variant are used in the learning situation's real context. In this phase, some indicators, which characterize the actual context of the learner's activity, are calculated (we use here the Usage Tracking Language for modeling and calculating these indicators, see [38] for details). Based on these indicators, the teacher, in his/her tutor role, could adapt the session by modifying the pedagogical scenario structure. By responding to new emerged needs, he can adjust some elements in order to maintain and improve

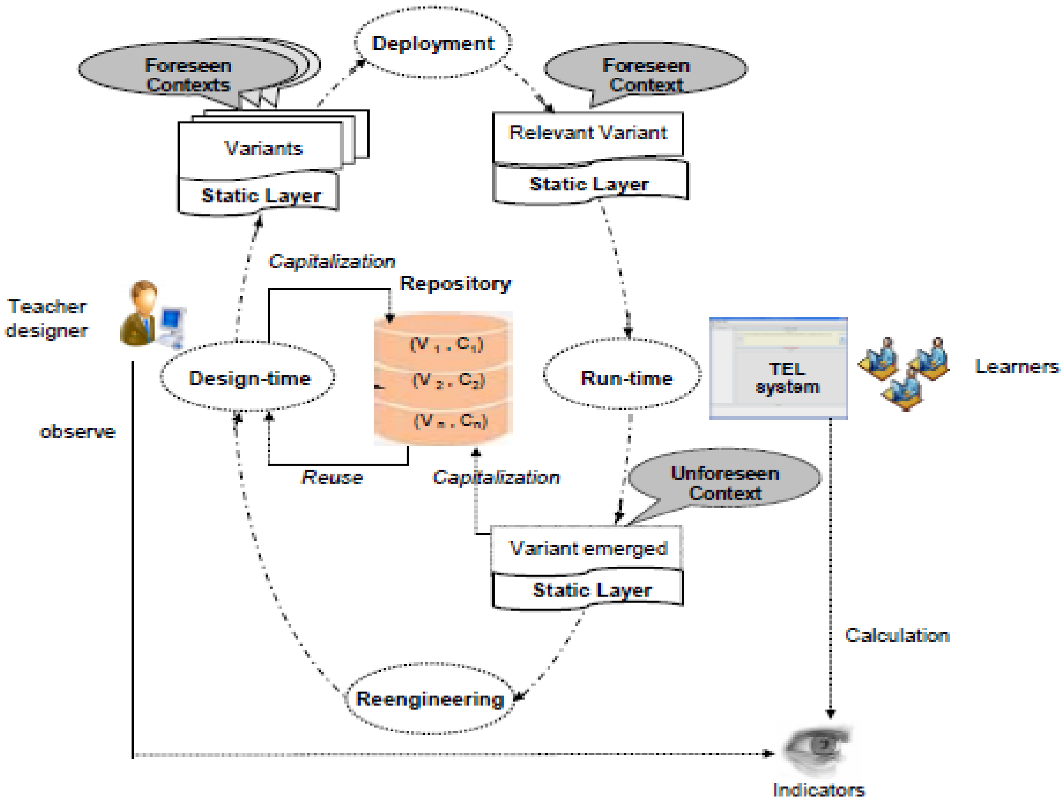


Fig 3. Incremental and iterative engineering process of an open pedagogical scenario.

the learning quality. This adjustment is not only a choice between some pre-specified elements but also the remove or addition of new elements that are necessary for the proper conduct of learning session. The set of performed adaptations causes the emergence of new variants. Then, teacher decides if he/she capitalizes or not these emerged variants with their own relevant execution contexts in OPS structure as predictive variants for reusing, sharing and re-engineering.

Reengineering

The reengineering of an open pedagogical scenario is not so easy for teachers who have to take pedagogical decisions. By using indicators that are calculated from tracks of learning session execution, a teacher can evaluate the relevance of the performed adaptations, e.g. of the emerged variants. The new version of the scenario must be generated by integrating positive adaptations. This integration remains under the control of the teacher while he decides if a new emerged variant improves the scenario or not. In the case where this variant corresponds to a new context for which it is relevant, or is better than existing variant, teacher can integrate it with its relative context into the structure of the scenario. If the teacher notices that a variant is relevant in most contexts, its elements can be considered as mandatory for achieving the scenario's objective. The variants that are never or rarely used can be removed to avoid the scalability problem caused by overloading of variations repository.

Indeed, this approach based on variants avoids to design every time the same pedagogical scenario. It can promote the systematic reuse of common or proven practices in a specific educational domain, and reduces the modeling effort for teacher. However, generally teachers haven't good technical competences, so it is relevant if we allow them using their own business language (DSEML) and provide them user-friendly modeling tools for designing, adapting and managing open pedagogical scenarios in a high level of abstraction.

IV. The MDE/DSM approch

The MDE/DSM represents a pragmatic and robust approach which has best practices and dedicated tools. It allows a high productivity and quality by facilitating the generation of code from models designed at abstract level.

A. *MDE/DSM principles*

Model-Driven Engineering (MDE) is basically a software development approach. It is an enhancement of the Model-Driven Architecture (MDA) approach, initially proposed by Object Management Group (OMG) in 2001 [33] to provide a solution to the problem of software technologies continual emergence that forces companies to adapt their software systems every time a new "hot" technology appears.

MDE focuses on creating productive models that describe the elements of a system [41] and guide the implementation. The MDE's goal is to define models that can be

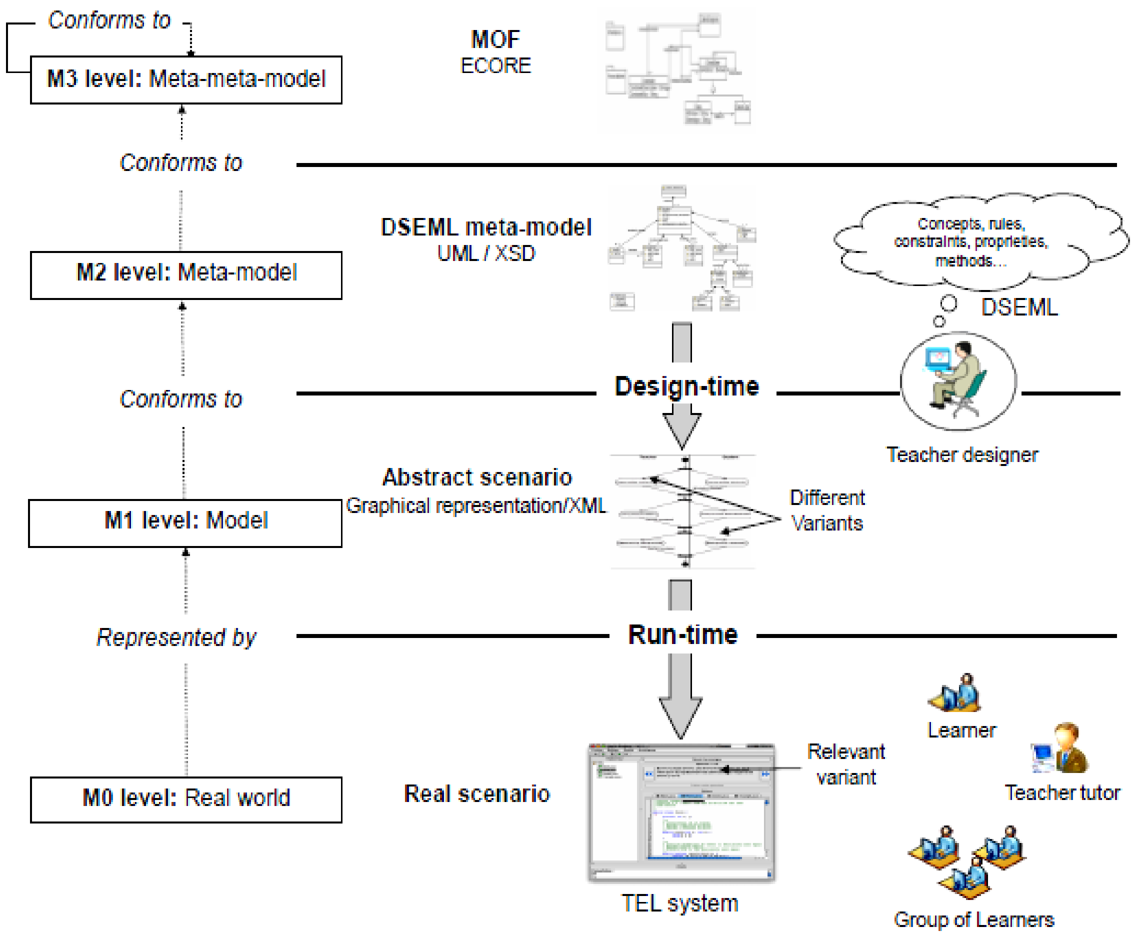


Fig 4. OMG layers view of the open pedagogical scenarios engineering.

operationalized and manipulated by computer. This implies that the produced models (1) conform to the explicit and formal metamodel; and (2) they represent without ambiguity an aspect (a point of view) of artifact to produce. These relations of conformity (conforms-to) and representation (represents) are the basis of MDE [5].

Indeed, this approach is attractive for our problematic because (1) it advocates the development of productive models, which helps the designer to control the choice of development and implementation; and (2) it allows to working directly into the business world of the target application by defining its domain-specific languages.

Moreover, among the advantages of MDE we retain the possibility of reuse and capitalization of both models and practices (transformation and transcription rules between models), the ability to "project" the business knowledge expressed within abstract models (Computation Independent Model – CIM) towards concrete and platform dependent ones (Platform Specific Model – PSM). However, MDE is also regarded as too simplistic and normative [20]. For this, [7] [42], again in an MDE approach, prefer to define their models using domain specific languages (DSL) based on a metamodel simpler but more focused on the business domain of designers. A domain is defined by [15] as "*an area of knowledge: scoped to maximize the satisfaction of the requirements of its stakeholders, including a set of concepts and terminology understood by practitioners in that area*".

The Domain Specific Modeling approach (DSM) was defined (1) to reduce the complexity of the transformations and the *semantic losses* they generate, and (2) raise the *level of abstraction* beyond programming by specifying the solution in a language that directly uses concepts and rules from a specific problem domain [26]. The principle here is to develop a DSL, tailored for specifying artifact which instruments a specific activity in a specific context. This DSL has to be formal but its metamodel reflects the domain of the users: the modeling vocabulary used is the domain one. Then, code generators could be developed for directly transform models expressed with a DSL into a specific technological platform framework.

B. MDE/DSM for instructional design rationale

We adopt MDE/DSM approach for providing the necessary supports to practitioner teachers in order to allow them design and adapt the OPS at a high-level of abstraction. We consider in this framework that a scenario, for being really designed and manipulated by a teacher, has to be considered as a domain specific model, expressed with a DSEML (Domain-Specific Educational Modeling Language) situated in his/her teaching context and rooted in his/her practices. In such a paradigm, MDE techniques have to support the transformation of the scenario from domain specific representation to operationalized one, both at the design phase to support the operationalization and at runtime to support the dynamic adaptation [34].

The figure 4 illustrates how we instantiates MDE/DSM paradigm to support teacher both at design and run time. According to the OMG layers view, the DSEML's metamodel is specified at the M2 level conforming to a meta-metamodel (MOF-MetaObject Facility) defined at the M3 level. This metamodel, inspired from the QOC, is the generic model of OPS (see Fig 2). It should formalize the semantic of the teacher own educational domain by describing him/her business language (vocabulary, rules, constraints, etc). The abstract OPS must be specified at the M1 level as models in conformity to DSEML's metamodel. At the design-time of OPS, designer instantiates the generic elements of metamodel for specifying the mandatory elements, the open points, the foreseen variants with their relevant contexts, etc. At run-time, mandatory elements and selected variant are operationalized on TEL system where they are executed in the learning real context. In unforeseen situations, new variants can be emerged thanks to dynamic adaptations which can be performed, at real time, by defining the elements (*open points*) that are not yet done for responding to new emerged needs.

V. MDE/DSM FOR SUPPORTING DESIGN RATIONALE OF HOP3X'S LEARNING SESSIONS

To verify our proposal we took Hop3x domain as experimentation area. Hop3x is a practical works TEL environment [16] developed for learning and teaching object-oriented programming languages like Java, C, Ruby, etc. It is mainly structured around a specific Java editor/compiler where the student has to solve programming exercises. Actually, the tutor could intervene during the session by providing hints and assessments; he/she is instrumented by a feedback system which provides indicators on the learner's activity.

Our objective is to provide the dedicated means to teachers for helping them to design, at an abstract level, the open practical works sessions [35]. For this, the MDE/DSM approach is adopted for concretely formalizing the instructional design rationale of open learning sessions.

Firstly, we investigated the semantic of Hop3x: in order to extract the domain specific concepts and rules we collected and analyzed the use cases of Hop3x thanks to direct observation of the activity of learners and teachers involved in a Hop3X session. Then based on this, we specified the metamodel that describes the Hop3x's DSEML. This metamodel of open practical works sessions formalizes in fact the semantic of Hop3x field by specifying the meaning of each concept and how it can be used according to domain's rules and respecting constraints. In the last step, a Hop3x-specific editor that is user-friendly. It was generated from the DSEML's metamodel. This editor makes available as specification tools the concepts and rules that are handled usually in the Hop3x practices. A teacher could use this editor for designing the practical works sessions graphically at an abstract level.

A. *Use of the EMF/GMF tooling*

The Eclipse Modeling Projects [19] provides a unified set of modeling frameworks, tooling, and standard implementation. In the following, we use the EMF (Eclipse Modeling Framework) and GMF (Graphical Modeling Framework) because they facilitate code generation for building tools and other applications based on a structured metamodel [44]. Our objective was to specify a metamodel which describes the Hop3x's DSEML, and then generating from this metamodel the code of the editor thanks to tools provided in EMF/GMF.

Thanks to a preliminary study on the Hop3x's usual practices and based on the OPS model proposed above (see Fig 2) we have defined a metamodel which describes the Hop3x's DSEML. Technically, this metamodel is an ECORE model where ECORE is the MOF-like meta-metamodel in EMF. Figure 6 illustrates this metamodel in the class-diagram-oriented view proposed by the ECORE graphical internal editor of EMF.

The code of editor has been generated automatically from Hop3x's DSEML metamodel thanks to the EMF/GMF tooling. This editor provides a graphical-view of the models which are namely the Hop3x learning sessions (see Fig 7). By using this editor the teachers who want to use Hop3x can design graphically the practical works sessions at an abstract level compared to the manual creation of XML files as it is the case currently.

Finally, thanks to this editor, designer can generate learning sessions in the XML format required (readable) by Hop3x system. Figure 8 shows an example of a Hop3x learning session generated as an XML file after its design by the specific editor.

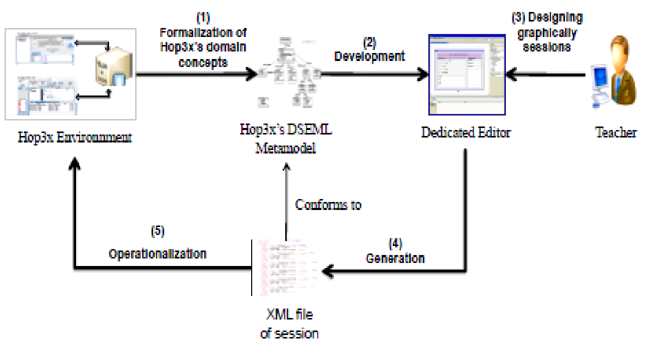


Fig 5. Development process of Hop3x's DSEML and dedicated editor.

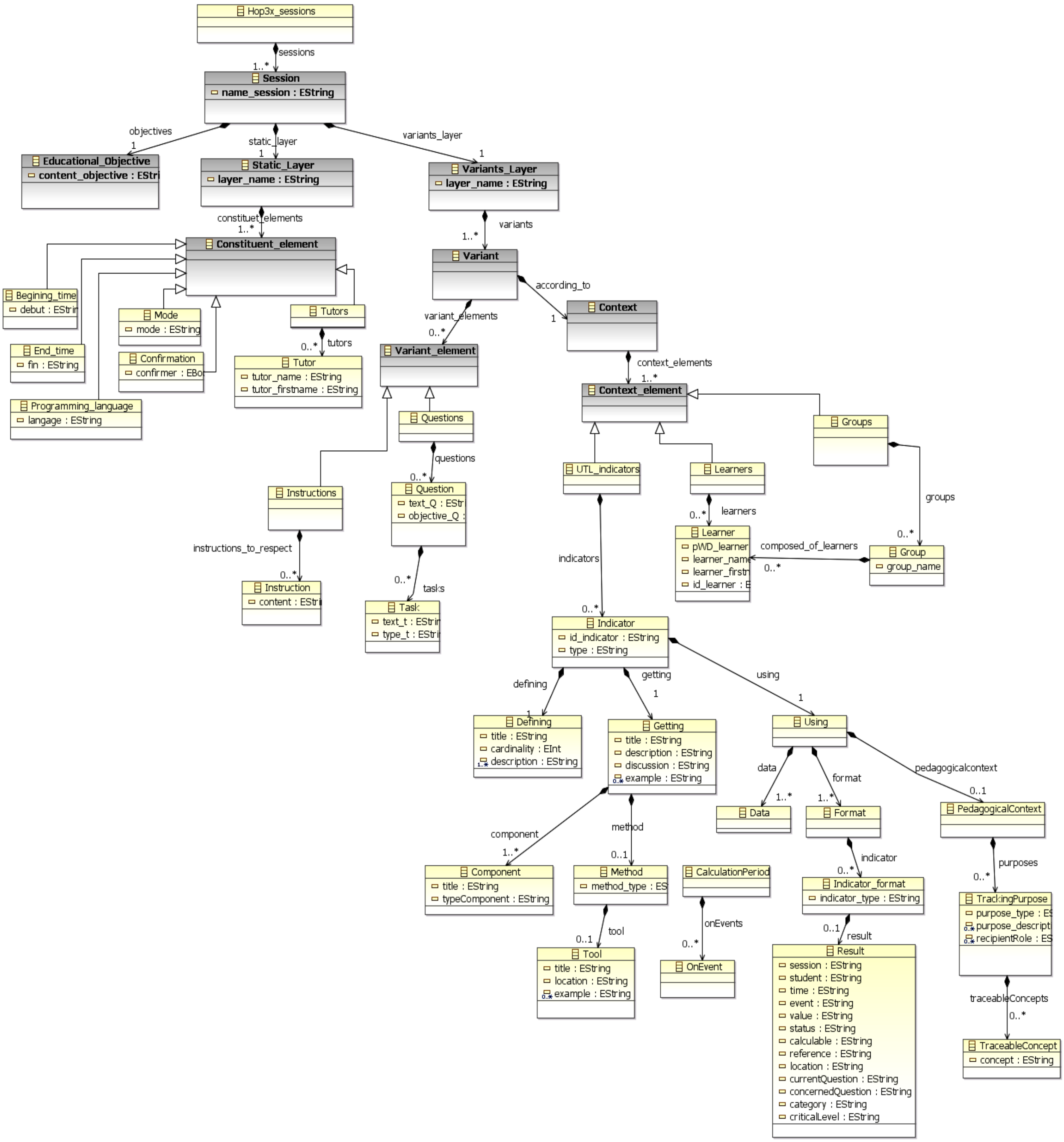


Fig 6. Hop3x's Domain Specific Educational Modeling Language metamodel.

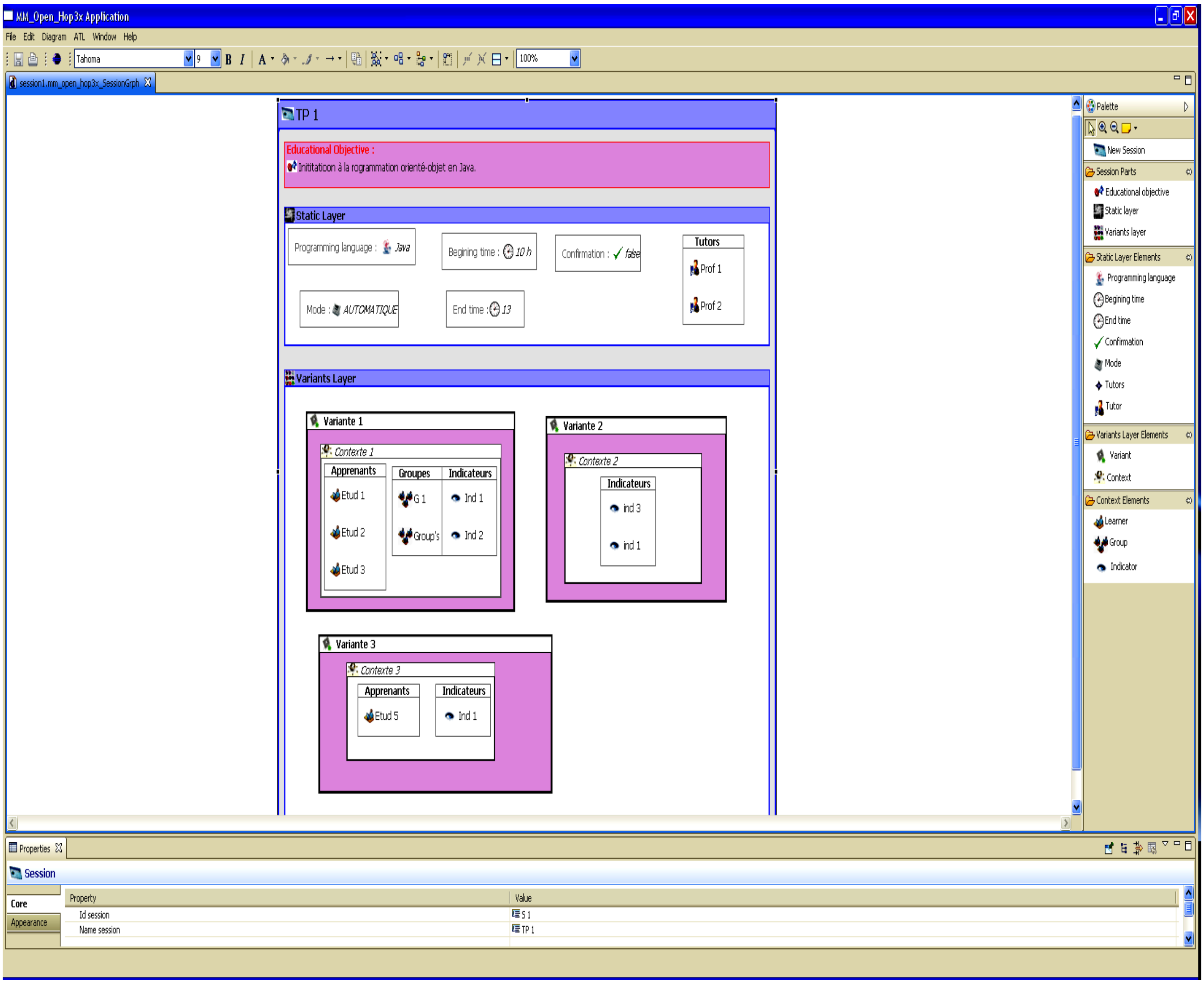


Fig 7. Example of a Hop3x learning session designed graphically by the specific editor.

B. *Testing experimentation*

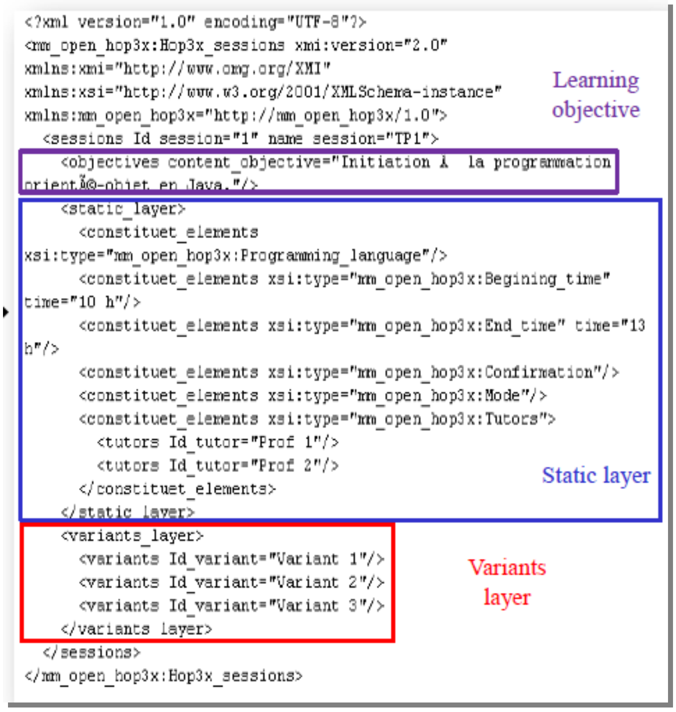
We have conducted a testing experimentation of the specific editor with the students who preparing the “Professional License in the Design and Realization of Multimedia Services and Products”. These 22 students had particularly lessons related to learning design. As work they had to describe a practical learning session of Hop3x using two editors separately, the first one is generic (Reload Editor [40] which implements IMS LD [23] and the second one is the specific editor which we have developed based on Hop3x’s DSEML (see Fig 7).

Our goal was simply to verify which editor was intuitive enough to enable the autonomy of its user. Beyond this testing, we have noted the interests of "putting in the hands of users" a specific editor, freed from the conceptual and technical barriers of learning session’s representation.

VI. Conclusion and Future Works

In this paper we have investigated the design rationale of open pedagogical scenarios. In order to concretely formalize this, we have adopted a constructive approach of instructional design based on the use of the MDE/DSM paradigm. This pragmatic paradigm facilitates indeed to provide the necessary supports for designers for allowing them to perform the instructional design rationale at a high level of abstraction. To verify our proposal we took Hop3x as experimentation area. We aim to help practitioner teachers to have a reflection about their design rationale. Our objective is to provide them the dedicated supports for designing open learning sessions at an abstract level. For doing this, we have defined a metamodel of OPS based both on the generic model of OPS and Hop3x-specific educational domain semantic, and then a graphical editor has been developed thanks to EMF/GMF tooling. This dedicated editor allows teachers to design open learning sessions at an abstract level freed from the conceptual and technical barriers.

Using this editor we are conducting iteratively interviews with Hop3x's users in order to promote the expression of new dynamic adaptation requirements. The information gathered from these interviews will also help us for adapting Hop3x's functionalities for transforming it into a more open TEL system [35].




```xml
<?xml version="1.0" encoding="UTF-8"?>
<mm_open_hop3x:Hop3x_sessions xmi:version="2.0"
xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mm_open_hop3x="http://mm_open_hop3x/1.0">
  <sessions Id_session="1" name_session="TP1">
    <objectives content_objective="Initiation Ã  la programmation
orientÃ©-objet en Java."/>
    <static_layer>
      <constituet_elements
xsi:type="mm_open_hop3x:Programming_language"/>
      <constituet_elements xsi:type="mm_open_hop3x:Begining_time"
time="10 h"/>
      <constituet_elements xsi:type="mm_open_hop3x:End_time" time="13
h"/>
      <constituet_elements xsi:type="mm_open_hop3x:Confirmation"/>
      <constituet_elements xsi:type="mm_open_hop3x:Mode"/>
      <constituet_elements xsi:type="mm_open_hop3x:Tutors">
        <tutors Id_tutor="Prof 1"/>
        <tutors Id_tutor="Prof 2"/>
      </constituet_elements>
    </static_layer>
    <variants_layer>
      <variants Id_variant="Variant 1"/>
      <variants Id_variant="Variant 2"/>
      <variants Id_variant="Variant 3"/>
    </variants_layer>
  </sessions>
</mm_open_hop3x:Hop3x_sessions>
```

Fig 8. XML file of a Hop3x learning session designed by the specific editor.

REFERENCES

[1] A. Berlanga, "Adaptive Learning Designs: Personalized and Reusable Instruction in Learning Environments". Ph.D. thesis in Computer Science. University of Salamanca, Spain, 2006.

[2] D. Burgos, C. Tattersall, and R. Koper, "Representing adaptive eLearning strategies in IMS Learning Design". Proc. TEN Competence Conference. Sofia, Bulgaria, March 31st, 2006.

[3] M. Baker, "Rôle et usage des modèles en Environnements Informatiques pour l'Apprentissage Humain". 5ème école thématique du CNRS sur les EIAH Personnalisation des EIAH 7-12 juillet 2007 Saint Quentin sur Isère, France

[4] J. Bézivin and O. Gerbe, "Towards a precise definition of the OMG/MDA framework".Automated Software Engineering,USA,2001.

[5] J. Bézivin, M., Blay-Fornarino, M. Bouzeghoub, J. Estublier, and J. M. Favre. "Rapport de Synthèse IDM". Synthèse de l'AS CNRS sur l'Ingénierie Dirigée par les Modèles, 2005.

[6] L. Becky and L. Hooey. "Requirements for a Design Rationale Capture Tool to Support". NASA paper, 2006.

[7] V. Bhanot, D. Paniscotti, A. Roman, and B. Trask. "Using Domain-Specific Modeling to Develop Software Defined Radio Components and Applications". In: the 5th OOPSLA Workshop on Domain-Specific Modeling (DSM'05), San Diego, California, USA, 2005.

[8] J. Burge and D.C. Brown, "Design Rationale: Types and Tools,Technical Report", Worchester Polytechnic Institute, Computer Science Dept,1998.

[9] J.E., Conklin and M.L. Begeman. "gIBIS: A Hypertext Tool for exploratory Policy Discussion", ACM Transactions on Office Informations Systems, 1998, pp 303-331.

[10] P. Cottier and H. El-Kechaï, "L'usager concepteur en situation: Conception collective d'un livret électronique d'apprentissage (LÉA)".In: Ingénierie des systèmes d'information, ISSN1633-1311. vol.14, n°3, 2009, p162.

[11] P. Cottier, C. Choquet, P. Tchounikine, "*Repenser l'ingénierie des EIAH pour des enseignants concepteurs*". In*: Usages, usagers et compétences informationnelles au XXIème siècle*, Jérôme Dinet (ed.), p.159-193. Édité par Hermes Lavoisier, ISBN 978-2-7462-2193-2.

[12] C. Choquet, "Ingénierie et réingénierie des EIAH – L'approche REDiM". Habilitation à diriger les recherches en informatique, Université du Maine, 2007, France.

[13] C. Choquet and S. Iksal (20, "Usage Tracking Language: a Meta - Language for Modelling Tracks in TEL Systems", In: International Conference on Software and Data Technologies (ICSOFT), 11-14 Septembre 2006, Setubal (Portugal), pp. 133-138, Edited by INSTICC, *ICSOFT 06* n°2, Filipe J., Shishkov B., Helfert M. (ed.), ISBN 978-972-8865-69-6.

[14] C.I. Pena de Carrillo, C. Choquet, C. Després, S. Iksal, P. Jacoboni, A. Lekira, E.A. Ouraiba, D. Pham Thi Ngoc D. "*Engineering and Reengineering of Technology Enhanced Learning Scenarios Using Context Awareness Processes.*", coord. : Graf, S. ; Lin, F. ; Kinshuk ; McGreal, R. In Intelligent and Adaptive Learning Systems: Technology Enhanced Support for Learners and Teachers, edited by IGI Global, isbn : 9781609608422, Scientific Book Chapter.

[15] K. Czarnecki and U.W. Eisenecker. "Generative Programming:Methods, Tools, and Applications". Addison Wesley, 2000.

[16] C. Després, P. Jacoboni, P. "Hop3x: Un Environnement de Suivi de TP de Programmation". Software, Université du Maine, 2010.

[17] J.M. Dodero, C. Tattersall, D. Burgos, and R. Koper, "Transformational Techniques for Model-Driven Authoring of Learning Designs". H. Leung et al. (Eds.): Advances in Web Based Learning – ICWL 2007, LNCS 4823, Springer-Verlag Berlin Heidelberg, Vol.4823, 2008, pp. 230 – 241.

[18] H. El-Kechaï. "Conception collective de scénarios pédagogiques dans un contexte de réingénierie Une approche par la métamodélisation située". Phd thesis of the Université du Maine, Université du Maine, 2008, France.

[19] Eclipse EMP, "Eclipse Modeling Projects", http://www.eclipse.org/modeling/. Last view 12/2010

[20] J.M. Favre, J. Estublier, and M. Blay-Fornarino. "L'ingénierie dirigée par les modèles – Au delà du MDA", Traité IC2, série Informatique et Systèmes d'Information, Hermès, Lavoisier, Paris, 2006.

[21] D. Griffiths, J. Blat, R. Garcia, H. Vogten, and K. Kwon, "Learning Design Tools". In Learning Design – A Handbook on Modeling and Delivering Networked Education and Training, R. Koper and C. Tattersall (Eds). ISBN 3-540-22814-4, Springer-Verlag, 2005, pp. 114-116.

[22] F. Henri. "Modèles du sujet pour la conception". Revue de l'éducation à distance, vol.22, n°.1, 2007, pp101-106.

[23] IMS Global Learning Consortium, "IMS Learning Design Specification, Version 1 - Final Specification", 2003, available at: http://www.imsglobal.org/learningdesign/

[24] Alex P. J. Jarczyk, Peter Löffler , and Frank M. Shipman Iii, "Design Rationale for Software Engineering: A Survey", 25th Hawaii International Conference on System Sciences, 2, 1992, pp. 577-586

[25] L. Karsenty, "Capitaliser le contexte des décisions en conception", livre Management des connaissances : modèles d'entreprise et applications, , 2001, p.49-69

[26] S. Kelly, and J. Tolvanen, "Domain-specific modeling: enabling full code generation". Wiley-IEEE Computer Society Press. ISBN: 978-0-470-03666-2, 2008, 448 pages.

[27] R. Koper, "Modeling units of study from a pedagogical perspective The pedagogical metamodel behind EML"; OUNL Technical report,Heerlen, The Netherlands, 2001.

[28] M. Lewkowicz, and M. Zacklad, "La capitalisation des connaissances tacites de conception à partir des traces des processus de prise de décision collective", 1998

[29] M. Lewkowicz, and M. Zacklad, "A guide through the construction of a groupware of efficient knowledge management", 2000

[30] J. Lee, "Design Rationale Systems: Understanding the Issues". *IEEE Expert* 12 (3), 1997, pp 78–85

[31] A. Maclean, R.M. Young, V.M.E. Bellotti, and T.P. Moran, "Questions, Options, and Criteria: Elements of Design Space Analysis", Human-Computer Interaction, Vol.6, 1991.

[32] D. McKerlie, and A. MacLean, "Experience with QOC Design Rationale". INTERCHI'93, Amsterdam : ACM, 1993

[33] OMG, “MDA specification guide. Version 1.0.1”, 2006, accessible at: http://www.omg.org/docs/omg/ 03-06-01.pdf, consulted 01/05/10.
[34] E.A. Ouraiba, C. Choquet, P. Cottier, C. Després, and P. Jacoboni, “Engineering of open learning scenarios: the case of Hop3x learning scenarios”.Proceedings of IEEE ICALT’10,Tunisia, 2010, pp. 264-268
[35] E.A. Ouraiba, C. Choquet, P. Cottier, “Opening TEL systems for teachers: A Domain-Specific Modeling & Model-Driven Engineering Approach”. The 3rd International Conference on Computer Supported Education (CSEDU’11), Noordwijkerhout, Netherlands, May 2011.
[36] E.A. Ouraiba, C. Choquet, P. Cottier, “Domain-Specific Modeling Approach To Support Instructional Design Rationale”. Proceedings of IEEE ICALT’11, Athens, Georgia, USA, 6-8 July 2011, pp. 205-206.
[37] E.A. Ouraiba, C. Choquet, P. Cottier, “Instructional Design Rationale with QOC: a Model Driven Engineering and Domain Specific Modeling Approach”. The 2011 International Conference of Information Engineering, The World Congress on Engineering organized by the International Association of Engineers (IAENG), London, U.K, July 2011.
[38] D. Pham Thi Ngoc, S. Iksal, C. Choquet, and E. Klinger, “UTL-CL: A Declarative Calculation Language Proposal for a Learning Tracks Analysis Process”. Proceedings of IEEE ICALT’’09, Riga, Latvia, July 2009, pp.681-685.
[39] M. Puustinen, M. Baker, and K. Lund GESTALT, “A framework for redesign of educational software”. Journal of Computer Assisted Learning 22, , 2006, pages 113.
[40] The Reload Project, The University of Bolton, The University of Strathclyde and JISC, 2004. Available at: http://www.reload.ac.uk/editor.html
[41] D.C. Schmidt, 2006. “Model-Driven Engineering”. *IEEE Computer*. 39 (2), 2006.
[42] J.P. Tolvanen. “Domain-Specific Modeling: How to Start Defining Your Own Language”, DevX.com, 2006, accessible at: http://www.devx.com/enterprise/Article/30550
[43] O.C. Santos, J.G. Boticario, and C. Barrera, “aLFanet: An adaptive and standard-based learning environment built upon dotLRN and other open source developments”. Foro hispano de .LRN y software libre educativo, Madrid, Spain, May 2005.
[44] D. Steinberg, F. Budinsky, M. Paternostro. “*EMF: Eclipse Modeling Framework, Second Edition”*. Publisher: Addison Wesley Professional, 2008
[45] E. Villiot-leclercq, “La méthode des Pléiades : un formalisme pour favoriser la transférabilité et l’instrumentation des scénarios pédagogiques”, research paper, Vol.14, 2007
[46] T. Zarraonandia, J.M. Dodero, C. Fernández, I. Aedo, and P. Díaz, “Iterative Design of Learning Processes”. Chapter 15 in Computers and Education: E-learning - from theory to practice, Springer Netherlands, ISBN978-1-4020-4913-2, 2007, pp.163-177.