A Visual Educational Modeling Language to Help Teachers in Learning Scenario Design

A. Retbi, M. Khalidi Idrissi, and S. Bennani

Abstract—The success of an e-learning system is highly dependent on the quality of its educational content and how effective, complete, and simple the design tool can be for teachers. Educational modeling languages (EMLs) are proposed as design languages intended to teachers for modeling diverse teaching-learning experiences, independently of the pedagogical approach and in different contexts. However, most existing EMLs are criticized for being too abstract and too complex to be understood and manipulated by teachers. In this paper, we present a visual EML that simplifies the process of designing learning scenarios for teachers with no programming background. Based on the conceptual framework of the activity theory, our resulting visual EML focuses on using Domain-specific modeling techniques to provide a pedagogical level of abstraction in the design process.

Keywords—Educational modeling language, Domain Specific Modeling, authoring systems, learning scenario.

I. INTRODUCTION

THE E-learning field is constantly improving to best meet learners' needs by providing high quality educational content. In this sense, E-learning is a subject of much research, especially in terms of design and production of a quality educational content [1-2]. Indeed, the content is the raw material of E-learning and the challenge is not to provide learners with an encyclopedic content, but with the resources and forms of interaction that can implement training activities (scenarios) provided in the course (problems, projects, readings, exercises, etc.).

The design of educational content, initially centered on the content, has become pedagogy-centered, moving the focus to the activities of the learner and the teacher and their interactions with the content. The emergence of Educational Modeling Languages "EML" brought a first response to these new requirements by proposing a formalization of relations between actors, activities, resources, tools and services [3].

EMLs [4-5] aim to assist teachers and provide them with models suited to the instructional design of various learning situations through the explicit separation of activities and resources without relying on a particular pedagogical approach (behaviorism, cognitive, constructivism, social-constructivism).

However, until now, there was a gap between the interest in educational modeling languages and their practical use by teachers they are intended to. In practice, teachers find it difficult to handle these EMLs due to either the use of a metaphorical or ambiguous terminology quite different from that used by the teacher or the complexity or lack of educational dimension.

To overcome this issue, it is necessary to ensure a good compromise between, on the one hand, the simplicity of manipulation via a graphical notation and a domain vocabulary and on the other hand, the power of expression allowing a precise description of the learning scenario. Thus, we thought of designing a visual educational modeling language inspired from the conceptual framework of the activity theory [6] coupled with the potential of a Domain-Specific Modeling "DSM" approach. Therefore, our language will be more accessible to teachers who can use it directly to design learning scenarios and will also facilitate the operationalization in learning management systems of scenarios generated.

The rest of this paper is organized as follows. In section II, the current studies on visual EML-oriented domain are briefly reviewed. The key concepts proposed of a visual EML-oriented domain based on adaptive sequencing of learning actions are detailed in section III. In section IV, the experimental results using Domain Specific Modeling approach are shown. Finally, Section V provides some conclusions and suggests future research directions.

II. RELATED WORKS ON VISUAL EML-ORIENTED DOMAIN

To meet the new demands for quality teaching combined with an ease of handling, much interest is focused on visual EML among which we quote:

− MOT+LD (Modeling using Object Types) [7]: a graphical editor which combines the concepts of IMS-LD [8-9] and the graphical notation language of MOT. The pedagogical model is used to graphically create learning scenarios according to IMS-LD's Level A and export them automatically in that format. The approach is interesting but:

o It is primarily intended for MOT modeling experts who need to produce Learning Design specifications which in fact are complex to implement and require a high design time.

o It does not take into account the levels B and C of IMS-LD and therefore does not support the dynamic aspect of personalization in the learning units.

− CPM (Cooperative Problem-based learning Metamodel) [10-11]: a modeling language that targets the design of cooperative problem-based situations. It is proposed as a UML

A. Retbi is with RIME Team, Networking, Modeling and E-learning, Mohammadia Engineering School, Mohammed V University Agdal, Avenue Ibnsina B.P. 765 Agdal, Rabat, Morocco (e-mail: asmaa.retbi@gmail.com).

M. Khalidi Idrissi is with RIME Team, Networking, Modeling and E-learning, Mohammadia Engineering School, Mohammed V University Agdal, Avenue Ibnsina B.P. 765 Agdal, Rabat, Morocco (e-mail: khalidi@emi.ac.ma).

S. Bennani is with RIME Team, Networking, Modeling and E-learning, Mohammadia Engineering School, Mohammed V University Agdal, Avenue Ibnsina B.P. 765 Agdal, Rabat, Morocco (e-mail: sbennani@emi.ac.ma).

profile to express educational scenarios. However, since it was designed for educational engineers who have mastered the UML formalism, the CPM language remains quite difficult for teachers to handle. In particular, it is considered [12-13] as a specific PBL (Problem-based Learning) approach and therefore does not allow educational neutrality which is a major goal of EML.

− Pleiades [14-15]: based on an astronomical metaphor, it provides a formal framework and an operating method for reusing scenarios. It is primarily intended for teachers and instructional designers to create learning scenarios. This method follows a metaphorical approach similar to the one adopted by the IMS-LD. This kind of formalism uses a vocabulary which is different from the application domain's. Consequently, it is neither simple to understand nor easy to generalize to all possible learning situations and educational approach.

− ISIS (Intentions-Strategies-Interactional-Situation) [16-17]: a conceptual model which aims to promote the design, reuse, adaptation and sharing of learning scenarios between designers. This is a model that uses as its starting point the pedagogical intentions and which is interested in explaining different levels of learning design: Intention, Strategy, and Interactional-Situation. Although this formalism uses a domain specific modeling approach, it does not support the dynamic aspect of the activities' sequencing during the specification of learning scenario's progression.

Overall, the above-mentioned visual EMLs are either dedicated to a particular type of learning situation (CPM), or are technically complicated to use by teachers (MOT +), or use ambiguous and unfamiliar terminology (Pleiades), or do not support the dynamic logical sequencing in learning activities (ISIS). Thus, we thought of the need to define a visual EML which:

− Will be domain-oriented and based on familiar terminology to teachers and thus simple and easy to manipulate.

− Will support the educational dimension of a learning scenario by introducing the concept of adaptive sequencing within a learning activity.

III. The Key Concepts for the Proposed Visual EML

Through a participatory design approach, our idea is to provide teachers with an expression language and environment close to their domain while allowing them to design the progression of the learning activity regardless of the pedagogical approach.

To meet the required educational effectiveness of his course, a teacher must develop the content and the related activities to be carried out following an execution order. In other words, the teacher will develop his pedagogical method of teaching the course in question, taking into account the prerequisites of the learners, the learning objectives and the environment.

From this first description of a learning scenario, we can state that any educational approach can be summarized as a logical sequence of steps or phases. The realization of this description was inspired from the activity theory [18]. In particular, we relied on the hierarchical structure of the activity proposed by Leontiev [19], namely: activity, action and operation. Activity (higher level) is carried through chains of actions (intermediate level) which are themselves conducted through operations (lower level). The action is always directed toward a conscious goal (intermediate to what motivates the activity) and is performed in the context of an activity.

This hierarchical structure allows us to represent the interactions between learners and the interactions between learners and the learning environment. Indeed, the key concept in Elearning is the activity of the learner, as outlined by [20] "knowledge is constructed by the individual through actions he performs on objects".

Thus, we propose that a learning scenario be described as a composition of activities. Each activity will consist of an adaptive sequencing of actions. The latter is expressed as a flow using simple and/or conditional transitions and synchronizations. More explicitly, for each learning scenario, we define:

• Activities and objectives associated with them.

• Actions that make up the activity and which execution contributes to the performance of activities. It is also necessary to provide a description of the action by identifying:

− Services and educational tools, i.e. communication tools (email, chat, forum, etc.) or information tools (push, documentary database, FAQ, etc.).

− Learning resources (text, audio, video, etc.) necessary to the performance of the action.

− Intentions that motivate the action. The achievement of these intentions will contribute to fulfillment of the objectives assigned to the activity.

• Adaptive sequencing of actions, indicating the execution order of actions and the nature of the transition from one action to another (progression). This sequencing is expressed by a simple transition, a conditional transition or a synchronization.

To implement this adaptive sequencing within learning activities through a visual educational modeling language, we decided to exploit the potential of Domain Specific Modeling "DSM" approach to define Domain Specific language. The adoption of the DSM approach will facilitate the specification and the operationalization of an adaptable learning scenario (i.e. containing adaptive sequencing).

IV. Outline of the Implementation

The development of our visual EML will focus on two key phases: the development of the DSL and the specification of templates for code generation from models. In this section, the two phases will be briefly explained.

A. Development of the DSL

Domain Specific Modeling is a design and development

approach centered on the concept of problem-related domain. The core idea of this approach is to build a "Domain Specific Language" DSL; "a custom language that targets a small problem domain, which it describes and validates in terms native to the domain" [18-19]. The key feature of DSL is that it employs familiar concepts of the problem domain to allow users to construct models of their applications [20-21].

Using DSM modeling techniques in order to develop a visual EML facilitates three main aspects [22-23-24]:

− The design of a modeling language,

− The generation of domain specific code in low level languages,

− The provision of a framework for modeling and code generation to the developers.

The design process which is done via Microsoft DSL Tools [25], involves the steps below [26]:

− Building the meta-model file .dsl with the concepts described previously and defining its structure, i.e. Domain Classes and their relationships. Each class refers to a domain concept. An excerpt from the meta-model is illustrated in Fig. 1.
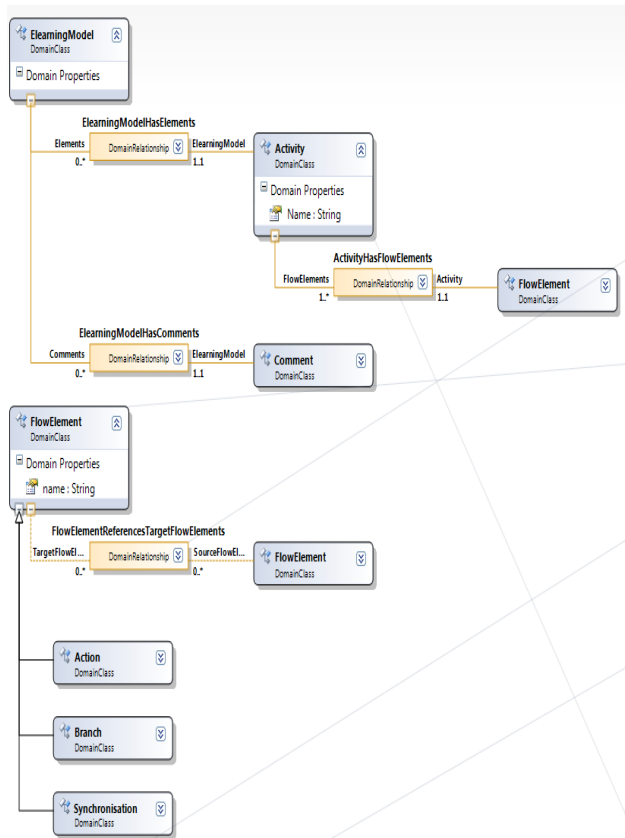

Fig. 1 The core of the conceptual metamodel

"ElearningModel" represents the meta-model's root class to which all Domain classes such as "Activity" are connected. All Domain classes are connected to "ElearningModel" with "one-to-many" embedding relationship.   An "Activity" is associated with a set of "FlowElement". The concept of "FlowElement" represents the expression of adaptive sequencing of actions. There are three types of elements constituting "FlowElement":

− "Action" is the core concept of sequencing,

− "Branch" which expresses the condition that controls the sequencing of actions

− "Synchronization" that allows the execution of parallel actions.

− Describing, in a separate area of DSL Designer, the meta-model's graphical notation which defines geometric shapes representing classes and connector lines representing relationships.

− Defining the mapping between the graphical notation and the meta-model by linking the geometric shapes to their corresponding classes and connector lines to their relationships.

− Building the visual language Toolbox which is intended to represent activities, actions and adaptations of a learning scenario.

B. The Code Generation

The templates specification for code generation from models, instance of the meta-model, is facilitated by the use of T4toolBox "Text Template transformation Toolkit" [27-28] tools.

```
<#@ template inherits="Microsoft.VisualStudio.TextTemplating.VSHost.ModelingTextTransformation" #>
<#@ import namespace= "System.Collections.Generic" #>
<#@ output extension=".xml" #>
<#@ EducationalFlow processor="EducationalFlowDirectiveProcessor" requires="fileName='Sample.eml'" #>
<?xml version="1.0" encoding="utf-8"?>
<ElearningModel>

<#

    foreach (Activity eachActivity in this.ElearningModel.Elements)
    {
#>
    <Activity name="<#= eachActivity.Name #>" type="<#=eachActivity.GetType().Name#>">
    <#
    PushIndent("  ");
        foreach (FlowElement eacheElement in eachActivity.FlowElements)
        {
        string htmlName = eacheElement.GetType().Name;
        int index= eachActivity.FlowElements.IndexOf(eacheElement);

        if (eacheElement is Branch)
        {
#>
        <Branch name="<#= eacheElement.name #>" type="<#=eacheElement.GetType().Name#>">

        </Branch>
    <#
```

Fig. 2 An excerpt of the XML template

T4 is a template-based code generation engine available in Visual Studio and standardized by Microsoft. By using the T4 Text Templating Language, developers can implement an interpreter to translate the domain models into executable

specifications used to synthesize automatically various software artifacts. A template file is created that navigates the model and emits boilerplate text interspersed with text calculated from the model [27-28]. Fig. 2 shows an excerpt of the body of the xml template developed. The template code generation can automatically produce a two-tier system involving a database and an HTML form.

Generating code from DSM meta-model allows us to:

− Save the produced scenario in a machine-interpretable formalism (XML) thus promoting the operationalization in Learning Management Systems.

− Facilitate the sharing, exchange and reuse of learning scenarios within the teaching community via a pedagogical model repository and a learning scenario repository.

V. CONCLUSION

In this paper, we present a prototype of a visual EML that aims to simplify learning scenario design. The key idea is to use a flow-oriented visual notation to create adaptive sequencing within learning activities. This research is based on domain specific modeling techniques providing teachers with a user-friendly graphical editor and familiar vocabulary to manipulate models and easily create a complex learning scenario. As a future work, we are planning to:

- Carry out an exhaustive evaluation of the results
- Identify and formalize, in collaboration with teachers, learning scenario patterns that contribute to create a repository of teaching practices.

REFERENCES

[1] M. Prat, "e-learning Réussir un projet : pédagogie, méthodes et outils de conception, déploiement evaluation", France: Editions ENI, 2008.
[2] C. Ernest, "E-learning : guide pratique pour une e-pédagogie", France : Editions Cépaduès, 2008.
[3] R.Koper, "Modelling units of study from pedagogical perspective : the pedagogical meta-model behind EML", Open University of Netherlands, 2001, Available: http:// dspace.ou.nl/handle/1820/36.
[4] CEN, "Educational Modeling Language : overview", Available: http://www.cen-ltso.net/main.aspx?put = 196, 2012
[5] R.Koper, C.Tatersall, "Learning design: a handbook on modelling and delivering networked education and training", Netherlands:Springer Verlag, 2005.
[6] B. Nardi, "Context and Consciousness: Activity theory and human-computer interaction", Cambridge, MA: MIT Press, 1996.
[7] G. Paquette, I. De la Teja, M. Leonard, K. Lundgren-Cayrol, O. Marino, "An Instructional Engineering method and tool for the design of units of learning", Ed. in R. Koper & C. Tattersall . Learning Design - A Handbook on Modelling and Delivering Networked Education and Training, Netherlands: Springer Verlag, 2005.
[8] IMS Global Learning Consortium, "IMS Learning Design Specification", Available: http://www.imsglobal.org/learningdesign/, 2001
[9] A. Lejeune, "IMS Learning Design : Etude d'un langage de Modélisation pédagogique, Revue Distance et Savoirs", vol. 2, n°4, Lavoisier, 2004, p. 409-450
[10] P. Laforcade, T. Nodenot, C. Sallaberry, "Un langage de modélisation pédagogique basé sur UML", Revue STICEF Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation, N°12 (Numéro spécial: Conceptions et usages des plates-formes de formation), 2005.
[11] P. Laforcade, "Méta-modélisation UML pour la conception et la mise en oeuvre de situations-problèmes coopératives", Ph.D. Thesis, Université de Pau et des Pays de l'Adour, France, 2004.
[12] H. EL Kechai, "Conception collective de scénarios pédagogiques dans un contexte de réingénierie : Une approche par la métamodélisation située", Ph.D. thesis, Maine University, Maine, France, 2008.
[13] E. Villiot-Leclercq E., "Modèle de soutien pour l'élaboration et la réutilisation de scénarios pédagogiques", Ph.D thesis, Université Joseph Fourier/Université de Montréal, France, 2007.
[14] E. Villiot-Leclercq, "La méthode des Pléiades: un formalisme pour favoriser la transférabilité et l'instrumentation des scénarios pédagogiques", Revue STICEF, Volume 14, 2007.
[15] E.Villiot-leclercq E., JP.David JP., "Le formalisme des pléiades pour la conception et l'adaptation de patrons de scénarios pédagogiques", in EIAH 07, 2007, pp. 269-274.
[16] V. Emin, "ScenEdit: an authoring environment for designing learning scenarios", ICALT'08, IEEE International Conference on Advanced Learning Technologies, Santander, 2008.
[17] V. Emin, "A goal-oriented authoring approach to design, share and reuse learning scenarios", European Conference on Technology Enhanced Learning - ECTEL, 2008.
[18] P. Dessus, "Des théories psychologiques pour guider le design pédagogique", France : Eds. B. Charlier & F. Henri, 2010.
[19] A.N. Leontiev, "The problem of activity in psychology, the concept of activity in soviet psychology", NY: M.E.Sharpe, 1979.
[20] DSM (Domain-Specific Modeling) Forum Main Page, Available: http://www.dsmforum.org/, 2004
[21] S. Kelly, J. Tolvanen J., "Domain-Specific Modeling: Enabling full code generation", Canada: Ed. John Wiley & Sons, 2008.
[22] S.Cook, G. Jones, S. Kent, and A. C. Wills, Domain-Specific development with Visual Studio DSL Tools, Ed. Addison-Wesley, 2007.
[23] J. Greenfield and K. Short, Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools, Ed. John Wiley and Sons, 2004.
[24] M. Mernik, J. Heering, and A. M. Sloane, "When and How to Develop Domain-Specific Languages", ACM Computing Surveys, 2005, pp. 316–344
[25] Microsoft DSL tools, Available: http://msdn.microsoft.com/library/bb126259.aspx, 2012
[26] Development of a Feature Modeling Tool using Microsoft DSL Tools, GIRO Technical Report 2009-1.ver 1.0, Available: http://www.giro.infor.uva.es/fmt.pdf, 2009
[27] O. Sych, T4Toolbox, Available: http://www.olegsych.com/2007/12/text-template-transformation-toolkit/, 2008.
[28] CodePlex open source community, T4Toolbox, Available: http://t4toolbox.codeplex.com/, 2010.

Asmaâ Retbi received the Engineer degree in Computer Science in 1997 and the DESA degree in Computer Science in 2000. She is currently PhD Student in Computer Science with RIME team at the Mohamed V University Agdal, Mohammadia School of Engineers (EMI). His research interests include: E-learning, Authoring tool, MDE and Domain Specific Modeling.

Mohammed Khalidi Idrissi received the Doctorate degree in Computer Science in 1986 and a PhD in Computer Science in 2009. He is currently a Former Assistant chief of the Computer Science Department at the Mohammadia School of Engineers (EMI) and a Professor at the Computer Science Department-EMI. His research interests include: SI, ontology, Web services, MDA, elearning and evalution.

Samir Bennani received the Engineer degree in Computer Science in 1982, the Doctorate degree in Computer Science and the PhD in Computer Science in 2005. He is currently a Former chief of the Computer Science Department at the Mohammadia School of Engineers (EMI) and a Professor at the Computer Science Department-EMI. His research interests include: SI, Modeling in Software Engineering, Information System, eLearning content engineering, tutoring, assessment and tracking.