Towards a visual Educational Modeling Language for effective learning

Asmaâ Retbi, Farid Merrouch, Mohammed Khalidi Idrissi and Samir Bennani

Computer science Department, Mohammadia School for Engineers, Mohammed Vth University Agdal
BP. 765 Ibn Sina Av., Agdal, Rabat, Morocco

Abstract

Educational Modeling Languages (EMLs) are design languages intended to teachers for modeling any educational unit, independently of the pedagogical approach, context, users or resources employed. However, most existing educational modeling languages 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 authoring learning scenarios for teachers with no programming background. Based on the conceptual framework of the activity theory, our resulting visual EML uses an explicit flow representation to design adaptive sequencing within learning activities, especially for effective learning purposes.

Keywords: *Educational modeling language, Domain Specific modeling, Authoring systems, learning scenario.*

1. Introduction

The key purpose of the teacher is to create effective and stimulating opportunities for learning through high quality teaching that enables the development and progression of all learners [1]. This rule also remains valid for Elearning which success is highly dependent on the quality and effectiveness of educational courses for learners [2-3]. Indeed, an eLearning course can’t be summarized by a series of hypermedia documents exposed to the learner. It is necessary to consider different kinds of learning activities (knowledge assessment, practical work, experimental tests, exercises, projects, case study, etc.) and the relationships between these activities to provide effective learning to the learner. Effective teaching is essential for effective learning.

Accordingly, [4] recognizes that the key to a successful learning environment is based on the activities associated with them and not the objects of knowledge. He proposes to describe learning situations by using Educational Modeling Language “EML” which is defined as "a design language to create models of educational units" [5]. EMLs aim to assist teachers and provide them with suited models to the instructional design of various learning situations through the explicit separation of activities and resources without relying on a particular pedagogical approach (behaviorism, cognitivism, constructivism, social constructivism). Thus, the genesis of educational modeling languages contributed to the emergence, in the field of instructional design, of the learning scenario [6] concept as an essential component and a connecting thread of a learning situation. A learning scenario is defined as [7] "a realization of teachers’ didactic knowledge when carrying out an act of teaching, a course, a sequence, an entire curriculum." In the same optic but by focusing on a finer granularity of the scenario, namely activity, [8] states that "the learning scenario’s role is threefold: it defines in details the proposed activity to learners [...] it specifies the control of the learner's progress during this activity and it determines which educational assistance will be provided automatically depending on his progression".

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 EML 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 [9] 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 2, 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 3. In section 4, the experimental results using Domain Specific Modeling approach are shown. Finally, a conclusion is given in section 5.

2. Related works on visual EML-oriented domain

Many researches have been carried out in the field of educational modeling languages. In this section, we will focus on the main domain-oriented modeling languages which are positioned upstream of IMS-LD [10-11], namely:

- MOT+ (Modeling using Object Types) [12]: a graphical editor which combines the concepts of IMS-LD and the graphical notation language of MOT. The pedagogical model is used to create graphically learning scenarios according to IMS-LD's Level A and export them automatically in that format. The approach is interesting but:
    - It is primarily intended for MOT modeling experts who need to produce Learning Design specification which in fact is complex to implement and requires a high design time.
    - 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) [13-14]: a modeling language that targets the design of cooperative problem-based situations. It is proposed as a UML profile to express educational scenarios. In addition, 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 [15-16] as a specific PBL (Problem-based Learning) approach and therefore does not allow educational neutrality which is a major goal of EML.

- Pleiades [17-18]: 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) [19-20]: 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 proposed 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.

3. Key concepts of adaptive sequencing of learning actions

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. Thus, the learning scenario, as it will be designed, is a chain of logical steps or phases which:

- Contribute to the acquisition, the strengthening or the assessment of knowledge as outlined by [21]. [21] indicate that a learning scenario consists of four key elements: Do, Review, Learn and Apply.
- Respond to a given pedagogical approach.

Taking the example of two different instructional approaches, i.e. traditional transmissive learning and problem-solving learning, we may express each approach with the following sequencing:

- Traditional transmissive approach= courses, self-assessment exercises, if necessary reread the appropriate documentation, assessment.
- Problem-solving approach = identification and acknowledgment of the problem, development of possible solving methods, application of these methods, if they are correct as per the expected results, making the choice, etc.

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[22]. In particular, we relied on the hierarchical structure of the activity proposed by Leontiev [23], 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 [24] "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.
  - – Intention that motivates the action which led to achieving 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.
  We rely on a flow-oriented visual syntax that will be as follows:

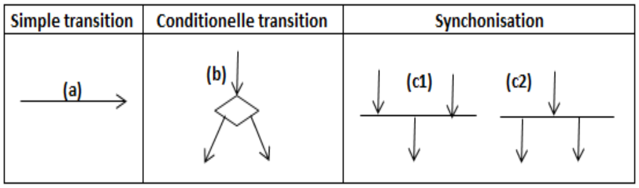


Fig. 1 Visual notation of adaptive sequencing of actions

– Simple transition: is a linear passage between two actions and without any condition. It is represented by an arrow (Fig.1a).
– Conditional transition: is a transition that checks if an *expression* of the action execution (number of correct answers, time limits, etc.) is valid, then action1 *else* action2 (Fig. 1b).
– Synchronization: it can manage the parallel execution of actions. It is represented by the notation shown in the Fig. 1c1 and Fig. 1c2

In order to illustrate this proposed notation, we took the example of "Geo Quiz 3" learning scenario made available by the research team of the University of Netherlands [25]. This is a quiz of five questions with multiple answers. The passage to one of four possible next actions depends on the number and the average of correct answers. Thus, for this example of learning activity, adaptive sequencing of actions is controlled by the learner’s performance. The visual scheme of our proposed method for "Geo Quiz 3" will be as follows:

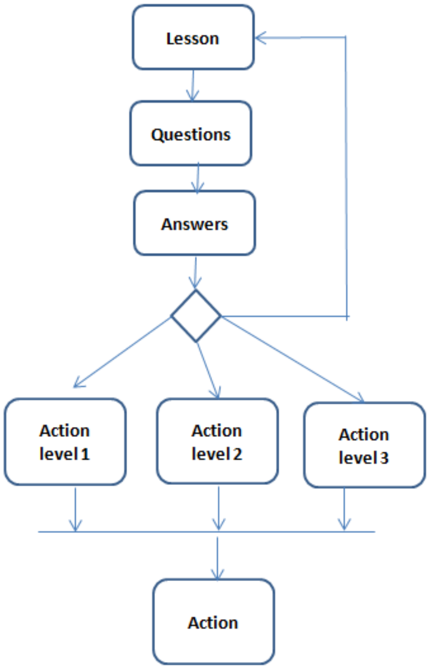


Fig. 2 Adaptive sequencing of actions for « Geo Quiz 3 »

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).

4. Experimental results

4.1 DSM approach and tools

4.1.1 DSM Definition

Domain Specific Modeling is a design and development approach centered on the concept of problem-related domain. The central 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" [26]. Unlike general-purpose languages that can provide solutions to a wide variety of problems, a DSL focuses on special and specific needs [26].

A DSL is a language with a high level of abstraction that uses familiar concepts and rules of the problem domain, making the model easier and less time consuming to understand, handle, and validate. In addition, the value of a visual modeling language lies in providing graphical notation that facilitates communication with domain experts.

Thus, using a DSM modeling approach in order to develop a visual EML will enable us to:

- Make the model more accessible to teachers not possessing modeling skills.
- Promote the sharing and exchange by the definition and appropriation of a common modeling Language.
- Provide teachers with a user-friendly graphical editor to manipulate the models.
- Improve the reliability of our solution by reducing development errors earlier in the process through expressing the concepts at a high level of abstraction.
- Facilitate migration of our solution from one technology to another, once the concepts have been captured in a model.

4.1.2 Disadvantages of standard modeling approaches: MDA and UML

UML is a general-purpose modeling language that offers diagrams and object-oriented programming primitives. Its semantics are unfamiliar and are often considered complicated and not easily accessible to domain experts. Furthermore, it is quite a monolithic standard created by general consensus. So, in order to be able to model some aspects, its range has to be reduced or extended by using mechanisms such as “Profiles”. Not all these mechanisms have the desired accuracy and can lead to serious deformation [27].

UML does not increase productivity since the core models are on the same level of abstraction as the programming languages. UML-based MDA tools do not increase productivity significantly either. An independent case study [28] of UML-based MDA showed productivity increase of 35%, which is far less than the 5-10 times more productivity gained from DSM [29].

In addition, the MDA approach is based mainly on the transformation of models from a high level of abstraction to lower ones [26]. These transformations are not automatic as they require the intervention of the developer who enlarges the model by specifying more details. By contrast, DSM aims to generate code directly and automatically without having to modify generated models. Furthermore, during the development process, the domain expertise needed for DSM is already available in the related domain while the MDA approach is based primarily on UML and requires prior knowledge of this standard.

4.1.3 DSM tools and technology

Several DSM environments now exist, including commercial products such as MetaEdit + [30], or Microsoft DSL tools [31-32] in addition to those from university research or open-source such as EMF [33] and GMF [34]. All these DSM tools propose metamodeling techniques that facilitate domain vocabulary definition and provide a simple approach to build appropriate notations. For our Visual EML prototype, we chose to use Microsoft DSL Tools to develop our visual modeling language as it offers the following advantages:

- Flexible representation of graphics.
- Simple for teachers to handle.
- Easy implementation of DSL graphics.
- Easy generation of machine-interpretable formalism.

4.2 Illustrations

On the basis of the DSL development methodology recommended by [35], the approach we adopted consists of five steps:

- In the context of our research and to address our problem, we decided to adopt a domain-specific modeling approach.
- Analysis of the educational content domain enabled us to clearly define and identify:
    - The range and scope of our domain
    - The terminology (vocabulary and ontology)
    - The concepts’ properties as well as relationships between them.
- Regarding the modeling language design, we chose to study and propose a new language that is in line with our goals.
- During the implementation stage, the Framework generated by the DSL language is tested.
- The deployment stage will enable us to assess the teachers’ satisfaction in addition to the maturity and completeness of the language.

When using DSL Tools, the design process follows the steps below [36-37]:

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

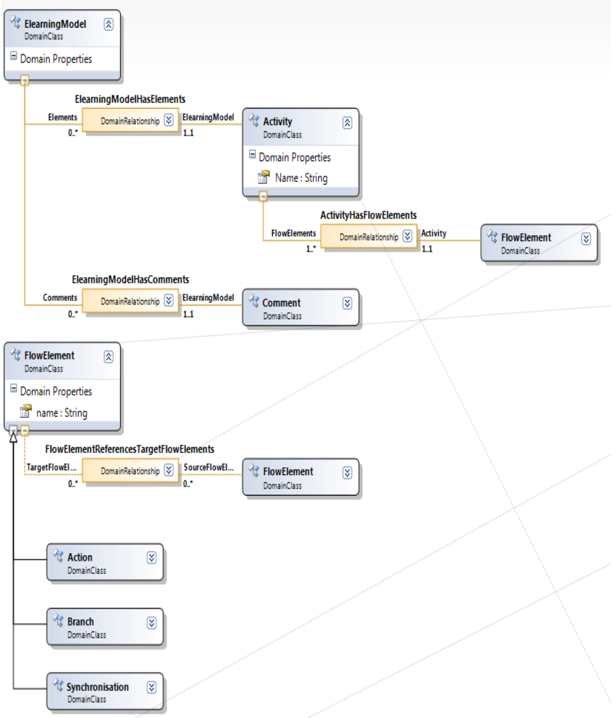


Fig. 3 The core of the conceptual metamodel

“ElearningModel” represents the metamodel’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 metamodel’s graphical notation which defines geometric shapes representing classes and connector lines representing relationships. Some shapes defined for our prototype are shown in Fig.2. The XML serialization of notation elements is automatically generated and stored in a file with .dsl.diagram extension.
- Defining the mapping between the graphical notation and the metamodel 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. The core language elements are depicted in Fig.4.

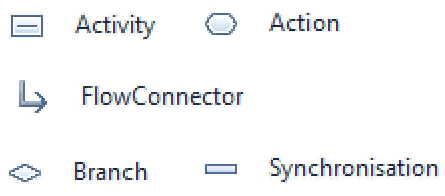


Fig. 4 The core visual language elements

The DSL Designer editor will thus enable the teacher to create, graphically and simply, models of learning scenarios. A simple metamodel of the “Geo Quiz 3” learning scenario is illustrated in Fig.5.

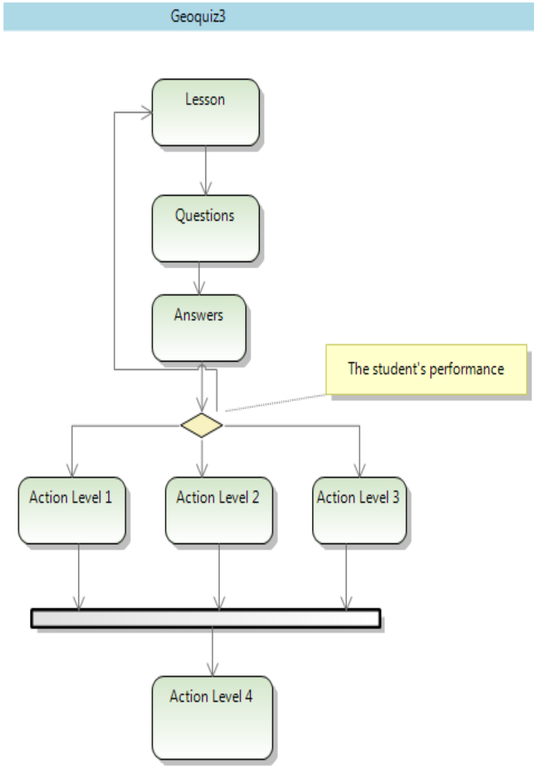
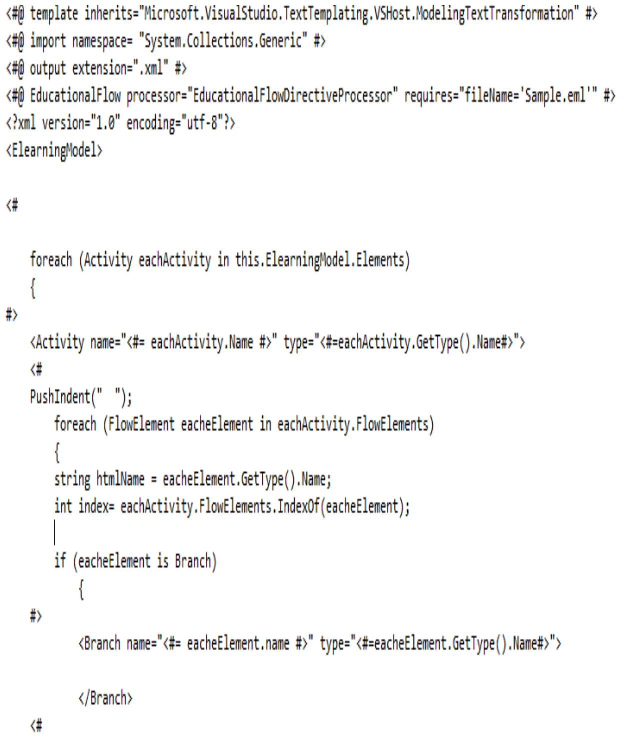


Fig. 5 The metamodel of the "Geo Quiz 3" learning scenario

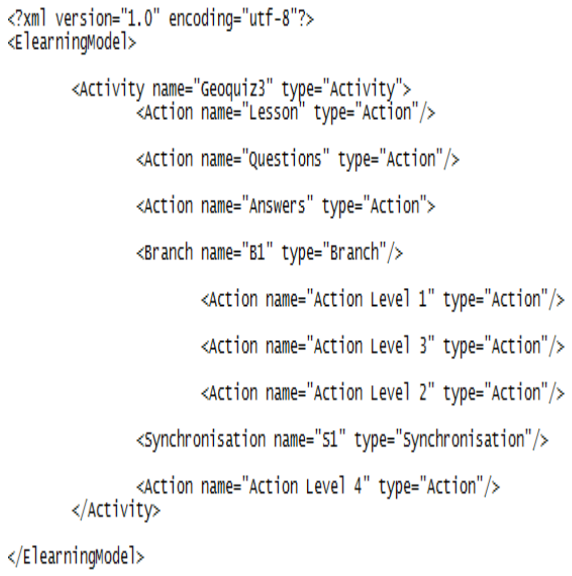
- Generating the xml file using the Text Templating engine available at T4toolbox “Text Template transformation Toolkit” [38-39]. T4 is a template-based code generation engine which can generate C#, T-SQL, XML or any other text files [38-39]. A template file is created that navigates the model and emits boilerplate text interspersed with text calculated from the model. Fig. 6 shows an excerpt of the body of the xml template developed.



```
<#@ 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. 6 An excerpt of the XML template

The execution of the XML template described above when applied to the “Geo Quiz 3” metamodel gives the following result:



```
<?xml version="1.0" encoding="utf-8"?>
<ElearningModel>

        <Activity name="Geoquiz3" type="Activity">
                <Action name="Lesson" type="Action"/>

                <Action name="Questions" type="Action"/>

                <Action name="Answers" type="Action">

                <Branch name="B1" type="Branch"/>

                        <Action name="Action Level 1" type="Action"/>

                        <Action name="Action Level 3" type="Action"/>

                        <Action name="Action Level 2" type="Action"/>

                <Synchronisation name="S1" type="Synchronisation"/>

                <Action name="Action Level 4" type="Action"/>
        </Activity>

</ElearningModel>
```

Fig. 7 the XML file generated from the "Geo Quiz 3" metamodel

Generating XML 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.

4. Conclusions

This paper presents a prototype of a visual EML that aims to simplify learning scenario design. It is intended to teachers not possessing technical skills, providing them with a user-friendly graphical editor and familiar vocabulary to manipulate models. The key idea is to use a flow-oriented visual notation to create adaptive sequencing within learning activities. In order to achieve reuse, exchange and sharing among teachers, we plan in our future work to:

- Identify and formalize, in collaboration with teachers, learning scenario patterns that contribute to create a referential of teaching practices.
- Make this referential available to teachers through assistance mechanisms, thus providing them with an authoring tool based on educational patterns.

References

[1] The lifelong learning UK, "New overarching professional, standards for teachers, tutors and trainers in the lifelong learning sector", 2007, Available: http://www.lifelonglearning.org.uk.
[2] M. Prat, "e-learning Réussir un projet : pédagogie, méthodes et outils de conception, déploiement evaluation", France: Editions ENI, 2008.
[3] C. Ernest, "E-learning : guide pratique pour une e-pédagogie", France : Editions Cépaduès, 2008.
[4] 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.
[5] CEN, "Educational Modeling Language : overview", Available: http://www.cen-ltso.net/main.aspx?put = 196
[6] Lejeune A., Pernin J-P., "A taxonomy for scenario-based engineering, Cognition and Exploratory Learning in Digital Age", CELDA, 2004, p.249-256.
[7] R.Koper, C.Tatersall, "Learning design: a handbook on modelling and delivering networked education and training", Netherlands:Springer Verlag, 2005.
[8] V.Guéraud, "Approche auteur pour les situations actives d’apprentissage: scenarios, suivi et ingénierie", Ph.D. thesis, Grenoble 1 University, Grenoble, France, 2005.

[9] B. Nardi, "Context and Consciousness: Activity theory and human-computer interaction", Cambridge, MA: MIT Press, 1996.

[10] IMS Global Learning Consortium, "IMS Learning Design Specification", Available: http://www.imsglobal.org/learningdesign/.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] E. Villiot-Leclercq, "La méthode des Pléiades: un formalisme pour favoriser la transférabilité et l'instrumentation des scénarios pédagogiques", Revue STICEF, Volume 14, 2007.

[18] 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.

[19] V. Emin, "ScenEdit: an authoring environment for designing learning scenarios", ICALT'08, IEEE International Conference on Advanced Learning Technologies, Santander, 2008.

[20] V. Emin, "A goal-oriented authoring approach to design, share and reuse learning scenarios", European Conference on Technology Enhanced Learning - ECTEL, 2008.

[21] W.F. Dennison, R. Kirk, "Do, review, learn, apply: a simple guide to experiential learning", UK: Oxford Blackwell Education, 1990.

[22] P. Dessus, "Des théories psychologiques pour guider le design pédagogique", France : Eds. B. Charlier & F. Henri, 2010.

[23] A.N. Leontiev, "The problem of activity in psychology, the concept of activity in soviet psychology", NY: M.E.Sharpe, 1979.

[24] M. Hubert, "Apprendre en projets : la pédagogie du projet-élèves", France : Chronique sociale, 1999.

[25] Open University of Netherlands, "Learning Network and Learning Design", 2005, Available: http://dspace.ou.nl/.

[26] S. Kelly, J. Tolvanen J., "Domain-Specific Modeling: Enabling full code generation", Canada: Ed. John Wiley & Sons, 2008.

[27] B. Combemale, "Ingénierie Dirigée par les Modèles (IDM) - État de l'art", 2008, Available : http://hal.archivesouvertes.fr/docs/00/37/15/65/PDF/mde-stateoftheart.pdf.

[28] Compuware, "Landmark study reveals 35 per cent Productivity Gains for businesses Using Model-Driven Architecture", 2003, Available: http://www.prnewswire.com/news-releases

[29] DSM (Domain-Specific Modeling) Forum Main Page, Available: http://www.dsmforum.org/.

[30] MetaCASE, MetaEdit+ Main Page, Available: http://www.metacase.com/.

[31] Microsoft DSL tools, Available: http://msdn.microsoft.com/library/bb126259.aspx.

[32] 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.

[33] Eclipse Modeling Framework (EMF) Main Page, Available: http://www.eclipse.org/emf/.

[34] Eclipse Graphical Modeling Framework (GMF) Main Page, Available: http://www.eclipse.org/gmf/.

[35] M. Mernik, J. Heering, and A. M. Sloane, *"*When and How to Develop Domain-Specific Languages*"*, ACM Computing Surveys, 2005, pp. 316–344

[36] J. Greenfield and K. Short, Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools, Ed. John Wiley and Sons, 2004.

[37] S.Cook, G. Jones, S. Kent, and A. C. Wills, Domain-Specific development with Visual Studio DSL Tools, Ed. Addison-Wesley, 2007.

[38] O. Sych, T4Toolbox, 2008, Available : http://www.olegsych.com/2007/12/text-template-transformation-toolkit/

[39] CodePlex open source community, T4Toolbox, 2010, Available: http://t4toolbox.codeplex.com/.

Asmaâ RETBI: Engineer degree in Computer Science in 1997, DESA degree in Computer Science in 2000, PhD Student in Computer Science With RIME team at the Mohamed V University Agdal, Mohammadia School of Engineers (EMI); Teacher at the Computer Science Department at the Mohammadia School of Engineers; Ongoing research interests: E-learning, Authoring tool, MDE and Domain Specific Modeling.

Farid Merrouch : Engineer degree in Computer Science in 1999, DESA degree in Software Engineering in 2006, PhD Student in Computer Science With RIME team at the Mohamed V University Agdal, Mohammadia School of Engineers (EMI); Ongoing research interests: E-learning, Authoring tool, MDE and Domain Specific Modeling.

Mohammed Khalidi Idrissi Doctorate degree in Computer Science in 1986, PhD in Computer Science in 2009; Former Assistant chief of the Computer Science Department at the Mohammadia School of Engineers (EMI); Professor at the

Computer Science Department-EMI; Ongoing research interests: SI, ontology, Web services, MDA, elearning and evalution

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