



HAL
open science

Models transformation to implement a Project-Based Collaborative Learning (PBCL) scenario: Moodle case study

Firas Abdallah, Claudine Piau-Toffolon, Bruno Warin

► To cite this version:

Firas Abdallah, Claudine Piau-Toffolon, Bruno Warin. Models transformation to implement a Project-Based Collaborative Learning (PBCL) scenario: Moodle case study. The 8th IEEE International Conference on Advanced Learning Technologies - ICALT 08, 1st to 5th July 2008 Santander, Cantabria, Spain, Jul 2008, Santander, Spain. hal-01861671

HAL Id: hal-01861671

<https://hal.science/hal-01861671>

Submitted on 24 Aug 2018

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

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

Models transformation to implement a Project-Based Collaborative Learning (PBCL) scenario: Moodle case study

Firas Abdallah, Claudine Toffolon, Bruno Warin (*)
LIUM - IUT of Laval, France, () LIL-Littoral University, France*
{firas.abdallah, claudine.piau-toffolon}@lium.univ-lemans.fr
() warin@iutcalais.univ-littoral.fr*

Abstract

In this paper, a Project-Based Collaborative Learning (PBCL) meta-model is proposed. It allows a teacher to build up a PBCL scenario and to implement it in a learning system which is not designed to this learning method. A MDA (Model Driven Architecture) approach is described to elaborate transformation rules between the PBCL meta-model and that of the learning system. Moodle platform, used in the UIT of Laval, is the chosen case study.

1. Introduction

The work described in this paper focuses on the definition, specification, design, and implementation of learning situation supported by ICT (Information and Communication Technologies), especially, Project-Based Collaborative Learning (PBCL). PBCL is often recommended in the academic world. It has been a part of the instructional culture since Dewey, Decroly, Freinet, etc. and it is opposed to the behaviorist pedagogy which is based on a unilateral and passive transmission of the knowledge from the teacher to the student.

The specification of PBCL situation is described by learning scenario. The traditional approach is mainly focused on mechanisms that facilitate the (re)engineering, repurposing and the effective use of existing digital resources to support learning and teaching (depending on the platform selected). A new approach recommends starting from a learning scenario to lead to a learning system such as a web site for example. According to this approach, the learning scenario is the central part of a learning environment [1].

To allow teacher to elaborate a PBCL scenario, a meta-model dedicated to the PBCL is proposed. This meta-model is briefly described in section 2. It is explained in details in [2]. The PBCL meta-model is a

challenging proposal to the limits of existing approach to express PBCL scenario. In particular, the major proposal made by the IMS Global Learning Consortium, the Instructional Management Systems-Learning Design (IMS-LD) [3, 4], is not satisfying to express a PBCL scenario by teachers. Many experiments have been carried out that shown the limits of this approach [5, 6]. A meta-model is a domain specific modeling language which is used to express the common concepts for models in the same area. It is build from informal models, recommendations in a natural language and semi-formal models usually written in the Unified Modeling Language (UML).

Most of commercial or open sources learning systems are available [7], but none of these systems seems to be adapted to PBCL context. In the present study, the strategy is to assist the teacher to implement his PBCL scenario in the chosen target platform, rather than developing a new one. A models transformation approach is proposed allowing the integration of the PBCL scenarios in a platform. The Model Driven Architecture (MDA) oriented approach in the software engineering domain [8] can be recommended in the instructional engineering field as well. Indeed we support a design approach based on models, so-called Model-Driven Engineering (MDE). In the MDE software process methodology, refining models develops a system. The model transformation technique is a principle key within this approach. It allows traceability between the different models produced at the different abstraction levels. Within this approach, teachers can design a learning scenario based on the PBCL meta-model. Then, this scenario is adapted to a chosen platform. To illustrate this proposal, Moodle platform is taken as a case study being used in the University Institute of Technology (UIT) of Laval.

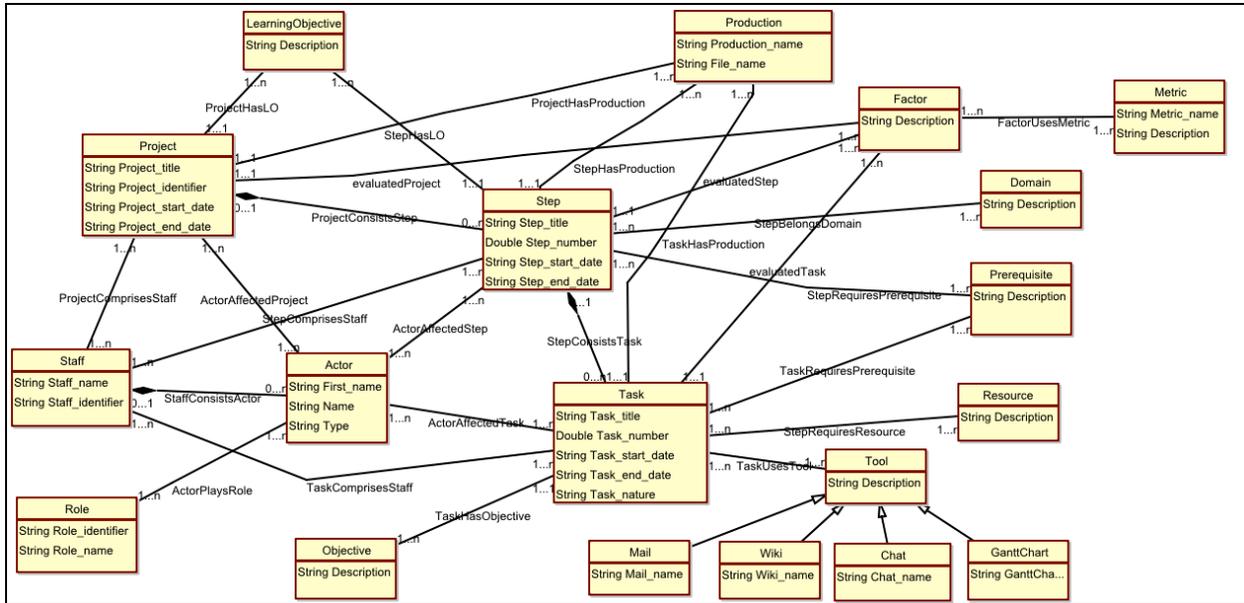


Figure 1. PBCL meta-model

2. Project-Based Collaborative Learning (PBCL) meta-model

The PBCL meta-model is built according PBCL concepts, theories (theoretical study), and practices (empirical study). From the existing literature on this topic, George PBCL characteristics were selected [9] to describe the Project-Based Collaborative Learning. This meta-model is presented as a diagram class in UML language (Figure 1). The construction manner of this PBCL meta-model is explained in details in [2]. An example of a PBCL scenario is developed that conforms to PBCL meta-model. This scenario was elaborated for the first year INFO at the UIT of Laval starting from PBCL meta-model. To elaborate this scenario, ModX tool is used [10]. This is a graphic tool allowing creating both model (PBCL scenario) and MOF-based meta-model (PBCL meta-model). It supports XMI format in order to be compatible with other MDA tools (The XMI file of PBCL scenario generated will be used in ATL transformation tool). To elaborate the PBCL scenario model, a PBCL meta-model has to be created (Figure 1). The PBCL scenario (model level) is created in a graphical way. It consists of proposing to students a computer development project. This project is made of steps (from the "project definition" step to the "validation" one). These steps are divided into tasks. For all these steps and tasks, prerequisites, objectives, learning objectives, tools, productions, and evaluation factors have been defined. To be completed, the scenario is accomplished by a clear definition of all actors (Student, designer, expert,

and tutor) and their roles (Facilitator, writer, technical, etc. role).

3. Introduce MDA approach in learning instructional design

To reconcile the two approaches: learning and technology, a MDA approach is adopted. This development approach proposed by OMG (*Object Management Group*) divides the vision of a system in three focal points of view [8]: The Computation Independent Model (CIM) corresponding to a domain model is independent of any computer implementation. The Platform Independent Model (PIM) is a view of a system from the platform independent viewpoint. The Platform Specific Model (PSM) is a view of a system from the platform specific viewpoint.

Through the three MDA models CIM / PIM / PSM, MDA seems appropriate for learning scenarios. The viewpoint of Lafordade et al. [11] is adopted. It proposes a new framework inspired by that of MDA. This new framework is represented by the proposal for different types of possible learning scenarios. These learning scenarios are categorized into three types and inspired by three MDA viewpoints: 1) Scenario at Knowledge level (SK): captures the teacher's common vocabulary associated with specific learning approaches, known platforms. The purpose of this scenarios type is to guide and facilitate (to the teacher of a community) design activity used in defining the learning scenario. 2) Abstract Scenario (AS): its vocabulary is independent from that of the e-learning platform (IMS-LD). 3) Specific Scenario to an e-

learning platform (SS): its vocabulary is consistent with that of a specific platform.

The software engineering process requires consideration of the three MDA approach viewpoints. However, the learning scenario process does not necessarily take into account these three aspects [11]. That means it is entirely appropriate to make the transformation of SK to SS. Since the PBCL meta-model captures the common vocabulary related to the Project Based Collaborative Learning, the PBCL scenario proposed above is situated at SK level. On the other hand, the PBCL scenario able to be implemented in Moodle is situated at the SS one, having a vocabulary that corresponds to the Moodle platform.

4. Transforming PBCL scenario to PBCL scenario able to be implemented in Moodle

Transformation activities are performed as automated processes that take one or more source models as inputs and produce one or more target models as outputs, while following a set of transformation rules. This process is referred as model transformation [12]. The transformation of MDA model is established by a relation (mapping) between a source model and a target one. Each model is described by a meta-model, which identifies the model characteristics. The mapping is then defined as a translation between the initial meta-model and the target one.

4.1. ATL transformation language

To make transformation, it is necessary to have transformation tools. These tools are based on transformation language having to respect the QVT (Query / Views / Transformations) standard [13]. In the measurement of our work, the commercial tools (e.g. MIA) are excluded. Among a number of "free" tools such as MTF, MTL, QVTP... ATL (Atlas Transformation Language) is selected because it answers best the problems of models transformation [14].

The ATL abstract syntax is specified as a MOF meta-model and a corresponding textual concrete syntax has been defined [15]. In the field of Model-Driven Engineering (MDE), ATL provides developers with a mean to specify the way to produce a number of target models from a set of source models [16]. ATL is made available as a part of the ATL Development Tools (ADT). The current ADT distribution runs over the Eclipse platform, and is based on the Eclipse Modeling Framework (EMF) [17].

4.2. Transformation rules

In the field of model engineering, models are considered as first class entities. A model has to be defined according to the semantics provided by its meta-model: a model is said to conform to its meta-model. In the same way, a meta-model has to conform to a meta-meta-model. In this three layers architecture (models, meta-models, meta-meta-model), the meta-meta-model usually self-conforms to its own semantics (e.g. it can be defined using its own concepts). Existing meta-meta-models include, by example, MOF [13], defined by the OMG and Ecore [17], introduced with the Eclipse Modelling Framework (EMF) [17].

A major feature in model engineering is to consider, as far as possible, all handled items as models. The model transformation itself has to be defined as a model. This transformation model has to conform to a transformation meta-model that defines the model transformation semantics. As the other meta-models, the transformation meta-model has, in turn, to conform to the considered meta-meta-model.

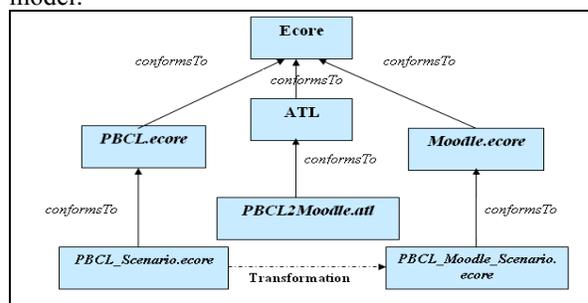


Figure 2. PBCL to Moodle ATL transformation

Figure 2 provides an overview of the PBCL to Moodle ATL transformation process. This figure introduces the name of the files (in italics) that are going to encode the models (*PBCLscenario.ecore* generated by ModX in a XMI file, *PBCL_Moodle_Scenario.ecore*), the meta-models (*PBCL.ecore*, *Moodle.ecore*) and the ATL transformation (*PBCL2Moodle.atl*) that will be handled during the design of the PBCL to Moodle ATL transformation. Note that the transformation to be designed (*PBCL2Moodle.atl*) will have to conform to the ATL transformation meta-model. The ATL syntax is detailed in [16].

A rule defines the "mapping" between the classes of the meta-models as well as the handling rules of the attributes and the classes relations. Performing a model transformation requires a clear understanding of the abstract syntax and semantics of both the source and target. The source PBCL meta-model was presented in

Figure 1, and now a vision of the target Moodle platform meta-model is shown (Figure 3).

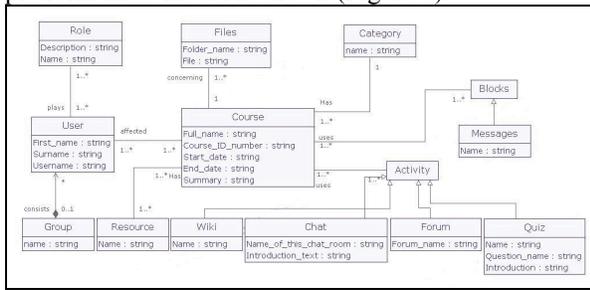


Figure 3. Moodle meta-model

In general, the transformations between two different domains are difficult to be implemented in practice since the two meta-models specifying the concepts and relations of each domain are able to be close but also very distant in terms of semantic "distance". Moreover, certain meta-model concepts/relations do not have necessarily "neighbour" all corresponding (concepts/relations) in the other one (and vice versa): such transformations can thus produce semantic losses (certain information specified in a scenario disappears) [11]. Twelve transformation rules are elaborated from the PBCL meta-model towards the Moodle one. These transformation rules have two goals: the first one is the PBCL scenario implementation in the Moodle platform techno-centred universe. The second one is the semantic loss qualification, compared to the PBCL, of each rule application. In the table below (Table 1), an example of this transformation rules elaborated since PBCL meta-model towards the Moodle one is shown:

Table 1. Transformation rules

Transformation Rules	PBCL		Moodle		Semantic loss
	Class	Attribute	Class	Attribute	
Project2Course	Project	Project_title Project_identifier Project_start_date Project_end_date	Course	Full_name Course_ID_number Start_date End_date	Moodle does not comprise three concepts equivalent to: Project, Step, and Task: impossibility of navigation between these three levels of conceptualization.
Step2Course	Step	Project_title+Step_title (concatenation) Step_number Step_start_date Step_end_date	Course	Full_name Course_ID_number Start_date End_date	
Task2Course	Task	Project_title+Step_title+Task_title (concatenation) Task_number Task_start_date Task_end_date	Course	Full_name Course_ID_number Start_date End_date	
Task2Course	Task	Task_nature	Category	name	
Actor2User	Actor	First_name Name Type	User	First_name Surname Username	
Role2Role	Role	Role_identifier Role_name	Role	Description Name	
Project2Course	LearningObjective in relation with Project	Description	Course	Summary	

For example, the *Step2Course* rule consists of simple and 'complex' codes:

```

1. rule Step2Course {
2.   from
3.     s : PBCL!Step
4.   to
5.     c : Moodle!Course (
6.       Full_name <- s.self.ParentStep.Project_title + '_' +
s.Step_title,

```

```

7.     Course_ID_number <- s.Step_number,
8.     Start_date <- s.Step_start_date,
9.     End_date <- s.Step_end_date,
10.    Summary <- s.getStepLearningObjective(),
11.    UserAffectedCourse <- s.StepHasActor
12.  )
13. }

```

A matched rule (*Step2Course*) is composed of a source pattern and a target pattern. The source pattern of a matched rule is defined after the keyword "from". It enables to specify a variable model element that corresponds to the type of source elements the rule has to match. The target pattern of a matched rule is introduced by the keyword "to". It specifies the elements to be generated when the source pattern of the rule is matched, and how these generated elements are initialized. This rule aims to produce a "Course" model element from a "Step" model element. It initializes the *Full_name*, *Course_ID_number*, *Start_date*, *End_date* and *Summary* features of the generated "Course". The bindings also enable to initialize reference features (line 11). The *Full_name* of the "Course" corresponds to the *Project title* of the "Project" concatenated with the *Step title* of the "Step". The *Course_ID_number*, *Start_date*, and *End_date*, of the "Course" correspond to the *Step_number*, *Step_start_date*, and *Step_end_date* of the "Step". Moodle does not have an equivalent element of the *LearningObjective* in PBCL. Then, a *helper* is used in line 10 (correspond to a function in Java) to regroup all *LearningObjective* values in the *Summary* attribute of the "Course", without any duplicate. The written helper is shown below.

```

1. helper context PBCL!Step def: getStepLearningObjective(): String
=
2.   self.StepHasLO->collect(e | e.Description) ->
3.     asSet()->
4.       iterate (LearObj; acc : String = " |
5.         acc +
6.         if acc = "
7.           then LearObj
8.           else 'and ' + LearObj
9.         endif);

```

4.3. Results

As result, the PBCL scenario elaborated by ModX (represented in ecore format on the left of Figure 4) is successfully converted to a PBCL scenario able to be implemented in Moodle (represented in ecore format on the right of the Figure 4). The instructional engineer, expert in Moodle platform, can now deploy the transformed scenario in the platform.

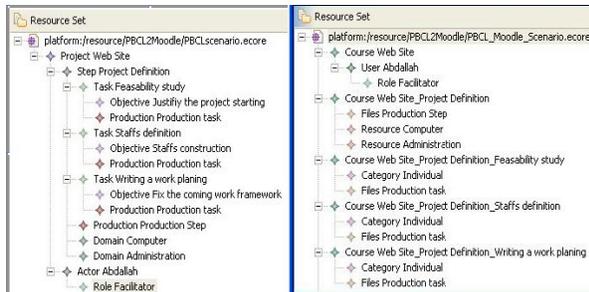


Figure 4. PBCL scenario and transformed PBCL scenario

5. Conclusion

Transforming a PBCL scenario into a PBCL scenario able to be implemented in Moodle renders possible to draw up a communication bond between the teacher and the instructional engineer. The teacher, expert in the PBCL domain, can now elaborate his scenario by using the proposed PBCL meta-model. Then, he can transform it into a PBCL scenario which can be implemented in Moodle, using the developed transformation rules. The transformed PBCL scenario is given to the instructional engineer who is charged to deploy it in Moodle.

The current state of transformation rules requires installation of the Eclipse platform and some plug-ins, plus the loading of the PBCL scenario to be transformed from ModX. This process is not obvious to the teacher. Thus, a graphical interface tool will be developed. It will help the teacher uploading his PBCL scenario generated by ModX and using the developed transformation rules. As future work, the process of automation will be continued to enable the teacher deploying its PBCL scenario into the target platform with less instructional engineer support.

6. References

- [1] J.P. Pernin, and A. Lejeune, “Dispositifs d’apprentissage instrumentés par les technologies : vers une ingénierie centrée sur les scénarios”, *TICE*, France, 2004, pp. 407-414.
- [2] F. Abdallah, C. Toffolon, and B. Warin, “Assistance to project-based learning support: from learning models to platforms”, *IADIS*, Lisbon Portugal, 6-8 July 2007.
- [3] Norm IMS-LD: IMS Learning Design (Best Practice Guide, Information Binding, Information Model), IMS, February 2003
- [4] R. Koper, “Current Research in Learning Design”, *Educational Technology & Society*, 2006, pp. 13-22.
- [5] D. Burgos, M. Arnaud, P. Neuhauser, and R. Koper, “IMS Learning Design: la flexibilité pédagogique au service des besoins de l’e-formation”, Visited 07/01/2008 - <http://www.epi.asso.fr/revue/articles/a0512c.htm>
- [6] C. Martel, L. Vignollet, and C. Ferraris, “Modelling the case study with LDL and implementing it with LDI”, *In ICALT’06*, The Netherlands, 2006, pp. 1158-1159.
- [7] <http://thot.cursus.edu>
- [8] Object Management Group, 2003. *Technical Guide to Model Driven Architecture: The MDA Guide v1.0.1*.
- [9] S. George, “SPLACH: a Computer Environment Supporting Distance Project-Based Learning”, *World Conference on Educational Multimedia, Hypermedia & Telecommunications ED-MEDIA*, Denver, Colorado, USA, June 2002, pp. 558-593.
- [10] ModX: Modelling Everything! Visited 07/01/2008 - <http://noce.univ-lille1.fr/projets/ModX/>
- [11] P. Laforcade, V. Barré, and B. Zendagui, “Scénarisation Pédagogique et Ingénierie Dirigé par les Modèles. Cadre d’étude pour la définition de langages et environnements-outils de scénarisation pédagogique spécifiques à des domaines”, *EIAH’07*, Lausanne, 2007, pp. 257-268.
- [12] S. Sendall, and W. Kozaczynski, “Model Transformation: The Heart and Soul of Model-Driven Software Development”, *IEEE Software*, Published by the IEEE Society, 2003, pp. 42-45.
- [13] OMG/RFP/QVT MOF 2.0 Query/Views/Transformations RFP. October 2002.
- [14] A. Canals, C. Le-Camus, M. Feau, G. Jolly, V. Bonnafous, and P. Bazavan, “Une Utilisation opérationnelle d’ATL : l’intégration de la transformation de modèles dans le projet TOPCASED”, *Génie Logiciel (73)* :21—26, 2005.
- [15] J. Bézinvin, G. Dupé, F. Jouault, G. Pitette and J. Eddine Rougui, “First experiments with the ATL model transformation language: Transformation XSLT into XQuery”, *OOPSLA 2003 Workshop*, Anaheim, California, 2003, pp. 1-18.
- [16] ATLAS group LINA, and INRIA Nantes, *Atlas Transformation Language – ATL User Manual*, ATLS group LINA & INRIA Nantes, Nantes, February 2006.
- [17] Budinsky, F., D. Steinberg, R. Ellersick, and T. Grose, *Eclipse Modeling Framework*, Addison Wesley Professional, Location, ISBN: 0131425420, 2004.