

From Project Based Collaborative Learning (PBCL) Scenario Elaboration toward PBCL Scenario Able to be Implemented in a Target Platform: PBCL Scenario Transformation Tool

Firas Abdallah

University Institute of Technology, Lebanese University, Lebanon

Abstract: In the context of Technology Enhanced Learning System (TELS) engineering, a Project Based Collaborative Learning Scenario Transformation Tool (PBCL-STT) is proposed. It allows instructors implementing Project Based Collaborative Learning (PBCL) scenarios in a chosen target Learning Management System (LMS) or in an Internet-enabled system for use in projects. This proposal is situated in the framework of an overall strategy consisting firstly, in providing instructors, with a PBCL meta-model to be used in their PBCL scenarios built up. Secondly, it consists in providing them with a model of functionalities dedicated to PBCL. This functionalities model will help them choosing a LMS or an Internet-enabled system for use in projects the most suitable to their PBCL scenarios. Finally, it will provide instructors a Project Based Collaborative Learning Scenario Transformation Tool (PBCL-STT) that transforms their PBCL scenarios into PBCL scenarios able to be implemented in their chosen platform. This paper explains the computer architecture of this transformation tool, which adopts a Model Driven Architecture (MDA) approach. Especially, it explains the elaboration of transformation rules between the PBCL meta-model and that of the chosen platform. A comparison between a PBCL scenario transformed into a PBCL scenario able to be implemented in Moodle platform from one side, and into a PBCL scenario able to be implemented in phpCollab platform from the other side, is established. The objective of this comparison consists in measuring correspondence and semantic losses in each transformation.

Keywords: Technology Enhanced Learning System (TELS), Project Based Collaborative Learning (PBCL), learning scenario, Learning Management Systems (LMS), Internet-enabled system for use in projects, meta-model, MDA, models transformation.

Received February 27, 2017; Accepted August 8, 2017

1. Introduction

The context of research of this paper is the Technology Enhanced Learning System (TELS) engineering. Especially, it focuses on the implementation of Project-Based Collaborative Learning (PBCL) situation supported by ICT (Information and Communication Technology). This paper is a part of a methodology and work process proposed in the framework of [1, 2, 3]. Previous works focused on the definition, specification and design of PBCL situation [1, 2]. In [1], a PBCL meta-model was proposed and explained to allow instructor elaborating a PBCL scenario. This PBCL meta-model was proposed since the major proposal made by the IMS Global Learning Consortium, the Instructional Management Systems-Learning Design (IMS-LD) [28, 22], is not satisfying to express a PBCL scenario by instructors. The existing experiments [13, 25] have demonstrated that this approach has some limits and cannot answer the instructors' pedagogical needs. "It is not always easy

to model a collaborative learning situation using IMS-LD level B and C" [25]. Garcia-Robles R. et al. [19], adopting the Problem Based Learning (PBL) method which is part of the PBCL, advice instructors to use this method in blended learning environment. They consider that the IMS-LD should be used just as an eLearning methodology because it "was developed to support the design, implementation running of UoLs in eLearning environments." A design language is a "mental tool" [11], but can be communicated through a notation system. This latter, usually visual, is mainly using text and diagrams. "A design language is what designers use to communicate design plans and intentions to each other" [15]. The DSL (Domain Specific Language) is used to express models common concepts belonging to the same domain. It is constructed starting from informal models, recommendation in natural or semi-natural language written generally in UML (Unified Modeling Language). DSLs are specifically designed to answer particular problem or application domain. They allow domain experts understanding, validating, modifying,

testing, and sometimes even developing DSL programs. DSLs can be defined for non-technical stakeholders, such as instructors, biologists, etc. since they make knowledge explicit [36].

Laforcade [23] had proposed the modeling of PBL (Problem Based Learning) scenario by using a PBL dedicated language CPM (Cooperative PBL Meta-model). Martel [25] had proposed LDL (Learning Design Language) meta-model allowing sequencing organizational and social aspects. The difficulty of IMS-LD, the lack of extensions added to this standard [21], and the validity of domains dedicated language (business meta-models) drive to propose a PBCL dedicated meta-model. This latter is considered as a Domain Specific Modeling Language regrouping common concepts for models in the same area.

The existing commercial and open sources learning systems (LMS) [37] seem to not be adapted and compatible to PBCL situation. So and as mentioned in previous studies, the adopted strategy in the methodology proposed has consisted in assisting instructors choosing the platform that answers the best their pedagogical needs. It is more practical and economical to choose an existing platform than developing a new one since this approach seeks to capitalize and to reuse what it is offered in existing platforms.

To allow instructors choosing the platform that best suits their needs, a model of functionalities has been proposed [3] helping instructors / designers formulating their needs through their learning scenario. Instructors may be asked to choose a platform among several available to them or assess the extent to which a certain platform is well suited to their needs (scenario).

Berggren D. *et al.* [8] had proposed to modify LMS by adding it an execution engine dedicated to the considered EML. Dodero, J. M. *et al.* [16] tried reducing the gap between EMLs and LMSs by the means of web services. The strategy of this study consists in allowing instructor implementing his PBCL scenario by the chosen platform, the process of work (Figure 1) consists in proposing to the instructor/designer a Project Based Collaborative Learning Scenario Transformation Tool (PBCL-STT) allowing to transform his PBCL scenario into a PBCL scenario able to be implemented by the chosen platform. The transformed scenario is provided to the pedagogical engineer, charged to deploy it in the target platform.

In the software engineering domain, the Model Driven Architecture (MDA) oriented approach has been recommended in the Technology Enhanced Learning System (TELS) engineering domain [29]. In [2], a model transformation approach was proposed allowing the integration of the PBCL scenarios in a platform. Certainly, the Model-Driven Engineering (MDE) design approach which is based on models, has

been supported. Its software process methodology develops systems by refining models. The principle key in this approach is the model transformation technique. Instructors can design learning scenarios based on the PBCL meta-model. Then, these PBCL scenarios are transformed to learning scenarios conformed to the chosen platform meta-model [2]. Practically, it is not evident to implement transformations between two different domains because the two meta-models (of the PBCL and that of the chosen platform) specifying the concepts and relations of each domain are able to be close but also very distant in terms of semantic "distance". Such transformations can thus produce semantic losses (certain information specified in a scenario disappears) [24].

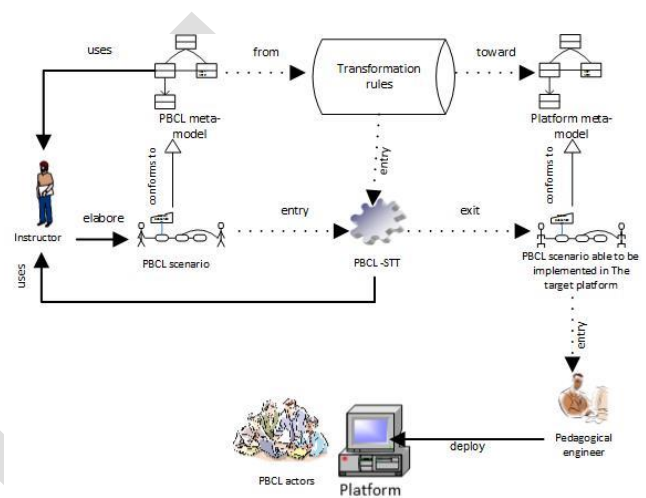


Figure 1. Process of Work.

The model of functionalities [3] has an objective to minimize the semantic losses after the transformations. It allows choosing the platform having its meta-model concepts and relations closest to the meta-model concepts and relations that define instructors' PBCL scenarios.

In [2], the Moodle platform, being used in the University Institute of Technology (UIT) of Laval, has been taken as a case study to illustrate the use of Model Driven Architecture of the software engineering domain in the TELS engineering domain. Transformation rules from the PBCL meta-model toward the Moodle meta-model were elaborated. From the other side, some researchers use an approach close to industrial projects' one [34]; this latter can be qualified as techno-economic approach. George S.[20] highlights the use of project management techniques and tools in PBCL context. In a publication studying the teaching of usage project management tools [30]; the authors consider the usage of PM tools as part of project manager competencies [32]. The ACM/IEEE reference curriculum for Computer Science [4], in the context of Instructional Units (IUs) to teach this skill, specifies that students have to learn the usage of a PM tool to develop a project schedule, allocate resources,

monitor the project activities, etc. Among the commercial and open sources PM tools available [17, 30], PhpCollab has been chosen to implement the same PBCL scenario transformed and implemented in Moodle [2].

In this paper, transformation rules from the PBCL meta-model to the meta-model of PhpCollab collaborative platform have been developed. These rules allow operationalizing the PBCL scenario in PhpCollab platforms. This paper proposes also a Project Based Collaborative Learning Scenario Transformation Tool (PBCL-STT) with a graphical user interface which will be used by the PBCL instructor/designer in the objective to upload his PBCL scenario and transform it to a PBCL scenario able to be implemented in a target platform.

Throughout the remainder of this paper (Section 2), concepts of the MDA approach and how they are introduced in instructional learning design are presented. In section 3, transformation rules allowing transforming PBCL scenario to PBCL scenario able to be implemented in a target platform are explained; PhpCollab is chosen as a case study. In section 4, a comparison between the transformed learning scenario from PBCL toward Moodle (explained in [2]) and toward PhpCollab, is deduced allowing qualifying the semantic losses of each transformed learning scenario. In section 5, a Project Based Collaborative Learning Scenario Transformation Tool (PBCL-STT) is proposed. This tool transforms a PBCL scenario into a scenario able to be implemented in a target platform. An architecture capitalizing in a base of rules the transformation rules from the PBCL meta-model to several existing platform meta-models is explained. In a final section, a conclusion and perspectives elements are given.

2. Adopt MDA Approach in TELS Engineering

To reconcile the two approaches: educational and technological, a model driven approach MDA (Model Driven Architecture) was adopted. MDA is a method to organize and manage system architectures; it is supported by automated tools and services for both defining the models and facilitating model types [10].

This approach, proposed by the OMG (Object Management Group), divides the vision of a system in three point of views [29]: 1 / CIM (Computer Independent Model) corresponding to the business model independent on any computer implementation, 2 / PIM (platform Independent Model) corresponding to a vision of the system independent of any technology, 3 / PSM (platform Specific Model) corresponding to a dependent on a specific platform vision.

The use of MDA approach is powerful in the building up of system architectures. It consists of multi-level abstraction models which allows refining

models at any level. The use of this approach gives an obvious idea of systems since models simplifies complex ideas to be communicated and explained [10]. Since the PBCL meta-model captures the common vocabulary of Project Based Collaborative Learning, a proposed PBCL scenario is then located at the CIM level. Moreover, the PBCL scenario able to be implemented in the target platform is located at PSM level, having a vocabulary conform to the target platform.

3. Transformation of PBCL Scenario Into a PBCL Scenario Able to be Implemented in a Target Learning Platform or in an Internet-Enabled System for Use in Projects

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 [35]. The transformation of a 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.

3.1. ATL (Atlas 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 proposed by the OMG [33, 9]. In the context of this 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 [9]. 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 [6]. 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) [12].

3.2. ATL Transformation Rules Elaboration

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 [29], defined by the OMG and Ecore [12, 5], introduced with the Eclipse Modelling Framework (EMF) [12] [5].

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.

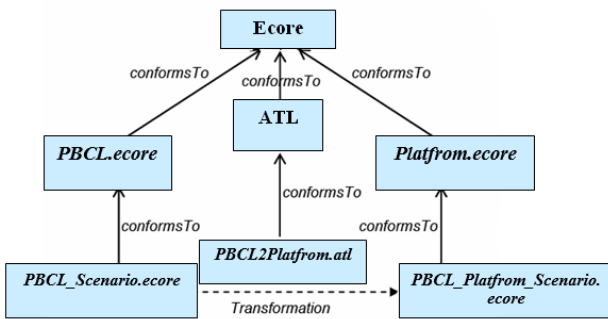


Figure 2. ATL Transformation from PBCL toward the target platform.

Figure 2 provides an overview of the PBCL to the target platform ATL transformation process. This

figure introduces the name of the files (in italics) that are going to encode the models (*PBCLscenario.ecore*, *PBCL_Platform_Scenario.ecore*), the meta-models (*PBCL.ecore*, *Platfrom.ecore*) and the ATL transformation (*PBCL2Platfrom.atl*) that will be handled during the design of the PBCL to the target platform ATL transformation. Note that the transformation to be designed (*PBCL2Platfrom.atl*) will have to conform to the ATL transformation meta-model. The ATL syntax is detailed in [6].

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 construction manner of the source PBCL meta-model is explained in details in [1], the target Moodle platform meta-model was explained in [2], and the target PhpCollab platform meta-model is shown (Figure 3).

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 “neighbor” 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) [24].

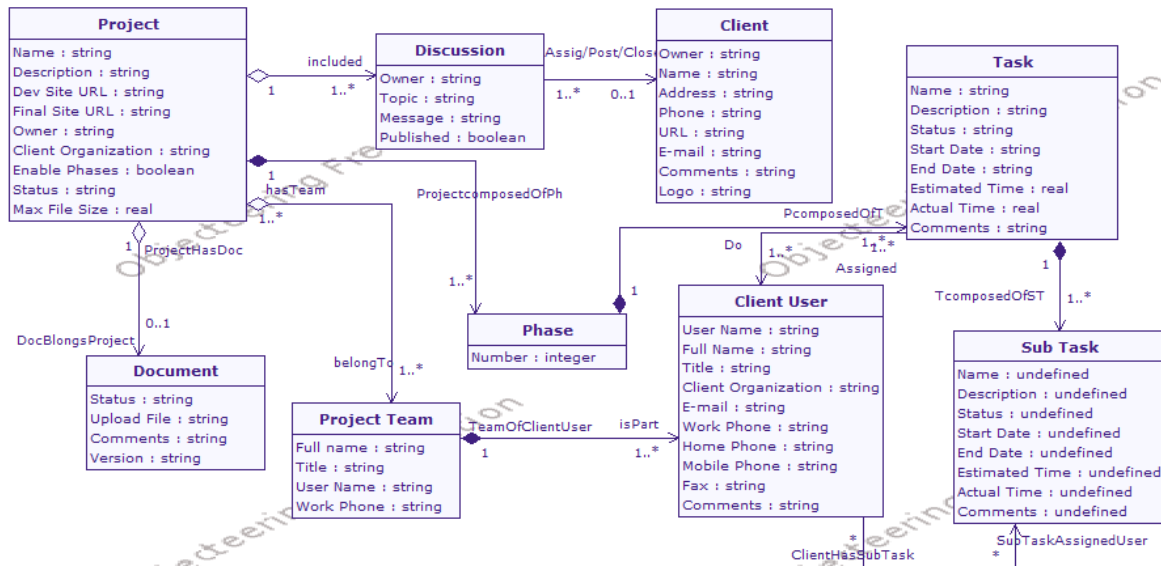


Figure 3. PhpCollab meta-model.

3.3. Transformation Rules from PBCL Meta-Model Toward Phpcollab Meta-Model

Two open source systems have been chosen as target platform to transform a PBCL scenario. The chosen PBCL scenario is developed that conforms to PBCL

meta-model. This scenario was elaborated for the first year in computer science at the UIT of Laval starting from PBCL meta-model. The transformation of this PBCL scenario toward Moodle LMS has been explained in [2]. In the following, seven transformation rules are elaborated from the PBCL meta-model towards the PhpCollab one. These transformation

rules, allowing the PBCL scenario implementation in the PhpCollab platform techno-centered universe, clarify as well the semantic loss qualification, compared to the PBCL, of each rule application. An extraction of transformation rules elaborated since PBCL meta-model towards the PhpCollab one is shown in the table below (Table 1).

Since PhpCollab has predefined phases (Planning, Design, Production, Testing, and Sign-off) [18], the *Step* concept of PBCL meta-model is transformed into the *Task* one of PhpCollab.

For example, the *Step2Task* rule consists of simple and ‘complex’ codes:

```

1. rule Step2Task {
2.   from
3.     s :PBCL!Step
4.   to
5.     t :PhpCollab!Task (
6.       Name <- 'Step number ' + s.Step_number + ': ' +
s.Step_title,
7.       StartDate<- s.Step_start_date,
8.       EndDate<- s.Step_end_date,
9.       Description <- s.getStepLearningObjective(),
10.      AssignedUser<- s.StepHasActor,
11.      Comments <-s.getDomain(),
12.      EstimatedTime<- s.getStepPrerequisite()
13.    )
14.  }

```

A matched rule (*Step2Task*) [6] 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 “Task” model element from a “Step” model element. It initializes the *Name*, *StartDate*, *EndDate*, *Description*, *Comments*, and *EstimatedTime* features of the generated “Task”. The bindings also enable to initialize reference features (line 10). The *Name* of the “Task” corresponds to the *Step_number* of the “Step” concatenated with the *Step_title* of the “Step” and preceded by the expression ‘Step number’. The *StartDate*, *EndDate* and *Description*, of the “Task” correspond to the *Step_start_date*, and *Step_end_date* of the “Step”. PhpCollab does not have an equivalent elements of the *LearningObjective*, *Domain*, and *Prerequisite* in PBCL. Then, *helpers* are used in lines 9, 11, and 12 (correspond to functions in Java) to regroup respectively all *LearningObjective* values in the *Description* attribute of the “Task”, all *Domain* values in the *Comments* attribute of the “Task”, and all *Prerequisite* values in the *EstimatedTime* attribute of

the “Task”, without any duplicate. Below an example of a “helper”¹ is shown.

```

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

```

The PBCL scenario (represented in XMI format in the left part of the Figure 4) is converted successfully into PBCL scenario able to be implemented in PhpCollab (represented in XMI format in the right part of the Figure 4).

4. Comparison between transformation rules from PBCL meta-model toward the meta-models of Moodle and that of PhpCollab

To compare the results between the two PBCL scenarios able to be implemented in Moodle and PhpCollab platforms, a brief description of each system is explained.

Moodle [27], founded in 1999, is an open source software package. It is developed and implemented in PHP, and it uses a traditional Apache server and a relational database management system. Moodle is developed in the context of blended learning, distance education, flipped classroom and other e-learning projects in schools, universities, workplaces and other sectors [27].

PhpCollab [18] is an open source internet-enabled system for use in projects that require collaboration over the internet. Organizations, such as consulting firms, that rely on a division between firm-side and client-side information will benefit most from use of phpCollab [31].

Moodle, which is a LMS designed for socio constructivist approach, is used in PBCL context which requires the use of project management techniques and tool. PhpCollab, which is an internet-enabled system for use in industrial projects, is used in learning context. To compare the PBCL transformed into PBCL scenario able to be implemented in PhpCollab and in Moodle, a recapitulative table (Table 2) is shown. In this table, the letter **C** indicates that the concept of the target platform corresponds to the concept of the PBCL meta-model, the letter **R** indicates that the transformation from PBCL into the target platform has resolved by using an alternative concept or by regrouping respectively all the concept values in an

¹A “Helper” is the equivalent of an UML additional operation

Table 1. Transformation rules toward PhpCollab.

Transformation Rules	PBCL		PhpCollab		Semantic loss
	Class	Attribute	Class	Attribute	
Project2Project	Project	Project_title	Project	Client Organization	
		Project_identifier		Name	
		Project_start_date		Dev Site URL	
		Project_end_date		Final Site URL	
Project2Project	LearningObjective in relation with Project	Description	Project	Description	Concatenation
Step2Task	Step	Step_number + Step_title	Task	Name	
		Step_start_date		Start Date	
		Step_end_date		End Date	
	Domain in relation with Step	Description		Comments	Concatenation
	Prerequisite in relation with Step	Description		Estimated Time	Concatenation
	LearningObjective in relation with Step	Description		Description	Concatenation
Task2SubTask	Task	Task_number + Task_titleTask_nature	SubTask	Name	
		Task_start_date		Start Date	
		Task_end_date		End Date	
	Prerequisite in relation with Task	Description		Estimated Time	Concatenation
	Resource in relation with Task	Description		Comments	Concatenation
	Objective in relation with Task	Description	Description	Concatenation	
Actor2ClientUser	Actor	First_name	Client	User Name	
		Name		Full Name	
		Type		Title	
		Comments			
		Client Organization			
		E-mail			
		Work Phone			
		Home Phone			
		Mobile Phone			
		Fax			
Actor2ClientUser	Role in relation with Actor	Role_name	Client	Comment	Concatenation
Staff2Groups	Staff	Staff_name	Groups	FullName	
		Staff_identifier		Title	
Production2Document	Production	Production_name	Document	Comments	
		File_name		Upload_File	
Tool2Discussion	Tool	Description	Discussion	Message	
		Forum_name		Topic	
		Tool_identifier		Owner	
Tool2Activity	Chat		?		PhpCollab does not comprise an equivalent concept
Tool2Activity	Gantt chart		?		Gantt chart does not exist in PhpCollab
Metric2Quiz	Factor	Description	?		PhpCollab does not comprise an equivalent concept
Metric2Quiz	Metric	Metric_name	?		PhpCollab does not comprise an equivalent concept
		Metric_type			
		Description			

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns="PBCL">
  <Project Project_title="Web Site" Project_identifier="1"
Project_start_date="15/01/08" Project_end_date="15/02/08">
    <ProjectHasActor ID_Actor="FA" Name="Abdallah"
First_name="Firas" Type="Designer">
      <ActorPlaysRole Role_name="Facilitator"
Role_identifier="F" />
    </ProjectHasActor>
    <ConsistsStep Step_title="Project Definition"
Step_number="1" Step_start_date="15/01/08"
Step_end_date="21/01/08">
      <StepBelongsDomain Description="Computer" />
      <StepBelongsDomain Description="Administration" />
      <StepHasProduction Production_name="Production Step"
File_name="Project_definition.doc" />
      <StepHasLO Description="Learn to defin a project" />
      <ConsistsTask Task_title="Feasability study"
Task_number="1" Task_start_date="15/01/08"
Task_end_date="17/01/08" Task_nature="Individual">
        <TaskHasObjective Description="Justify the project
starting" />
        <TaskHasProduction Production_name="Production task"
File_name="feasability_study.doc" />
      </ConsistsTask>
      <ConsistsTask Task_title="Staffs definition"
Task_number="2" Task_start_date="18/01/08"
Task_end_date="19/01/08" Task_nature="Individual">
        <TaskHasObjective Description="Staffs construction"
File_name="Staffs.doc" />
        <TaskUsesTool Description="Mail" Mail_name="Staffs
definition" Forum_name="StaffDefinition"
Chat_name="Negociation" Wiki_name="FinalStaffDefinition" />
      </ConsistsTask>
    </ConsistsStep>
  </Project>
</xmi:XMI>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0"
xmlns:xmi="http://www.omg.org/XMI"
xmlns:PhpCollab="PhpCollab">
  <PhpCollab:Project ClientOrganization="Web Site"
Description="" DevSiteURL="15/01/08"
FinalSiteURL="15/02/08" Name="1"/>
  <PhpCollab:Task Name="Step number 1: Project Definition"
Comments="Administration and Computer"
Description="Learn to defin a project"
StartDate="15/01/08" EndDate="21/01/08"
EstimatedTime="" />
  <PhpCollab:SubTask Name="Task number 1: Feasability study
(Individual)" Comments=""
Description="Justify the project starting"
StartDate="15/01/08" EndDate="17/01/08"
EstimatedTime="" />
  <PhpCollab:SubTask Name="Task number 2: Staffs definition
(Individual)" Comments=""
Description="Staffs construction" StartDate="18/01/08"
EndDate="19/01/08" EstimatedTime="" />
  <PhpCollab:SubTask Name="Task number 3: Writing a work
planing (Individual)" Comments=""
Description="Fix the coming work framework"
StartDate="20/01/08" EndDate="21/01/08"
EstimatedTime="" />
  <PhpCollab:ClientUser Comments="Facilitator"
FullName="AbdalLah" Title="Designer"
UserName="Firas" />
  <PhpCollab:Document Comments="Production task"
UploadFile="feasability_study.doc" />
  <PhpCollab:Document Comments="Production task"
UploadFile="Staffs.doc" />
  <PhpCollab:Document Comments="Production task"
UploadFile="Project_plan.doc" />
  <PhpCollab:Document Comments="Production Step"
UploadFile="Project_definition.doc" />
</xmi:XMI>
```

Figure 4. PBCL scenario and PBCL scenario able to be implemented in PhpCollab.

Table 2. Recapitulative Table.

PBCL	Project & Step & Task	Learning Objective & Objective	Resource	Actor	Role	Staff & Productio n	Forum & Domain & Prerequis ite	Chat	Gant Chart	Factor	Metric
Moodle	3R	2R	C	C	C	2C	3R	C	NR	NR	C
PhpCollab	3C	2R	R	R	R	2C	3R	NR	NR	NR	NR

attribute. NR indicates that the transformation is not resolved and there is a semantic loss.

Moodle has 7 C, 8 R, and 2 NR. PhpCollab has 5 C, 8 R, and 4 NR. Theoretically, the transformation toward Moodle is less “expensive” and has the minimum semantic losses. Practically, it depends on the context use of PBCL scenario. For example, if the PBCL scenario is used in a professional context without the needs to use factors and metrics for students’ evaluation activity, the transformation of the PBCL scenario into PBCL scenario able to be implemented in PhpCollab may be will more practical since the main concepts of project management (Project, Step, and Task) have taken C. In the next section, there is a proposal of a Project Based Collaborative Learning Scenario Transformation Tool (PBCLSTT) helping instructors transforming PBCL scenario toward a chosen target platform without the need to have technical skills in Eclipse platform and its plugins.

5. Project Based Collaborative Learning Scenario Transformation Tool (PBCL-STT)

The developed transformation rules require the installation of Eclipse platform and some plug-ins. The transformation of the PBCL scenario requires also its loading in Eclipse because it is already created under another graphical modeling tool (ex. ModX) that can generate an XMI file. ModX [26] is a graphic tool used to create both model and MOF-based meta-model, it allows non-technical users to create models graphically and then generate an XMI format file in order to be compatible with other MDA tools.

Since this process is not simple and easy for the instructor, a graphical user interface tool is developed. It helps instructors loading their PBCL scenarios, generated in XMI file by ModX (or other modeling tool that can generate an XMI file), and executing the developed transformation rules.

The proposed computer architecture (Figure 5) allows the definition of transformation rules base from the PBCL meta-model toward several meta-models

that correspond to several platforms. It capitalizes in a base of rules all the developed transformation rules. The PBCL-STT transformation tool graphical user interface (GUI) is related to a base of transformation rules. When the instructor chooses the “wished” platform, he specifies it in the GUI. This tool searches, in the base of transformation rules, the rules that correspond to the chosen platform. If the transformation tool finds the needed rules, it runs them and transforms the PBCL scenario into a PBCL scenario able to be implemented in the chosen platform. In case of this tool does not find the needed transformation rules in the base, it becomes necessary that the technician interacts to define and develop these rules to index and store them in the base of rules to be capitalized.

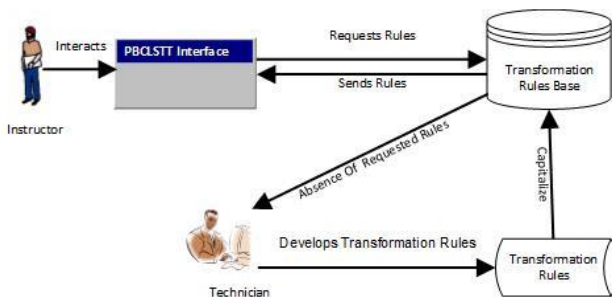


Figure 5. Computer Architecture.

To implement this architecture, the ATL transformation rules should be defined independently and then called from the programming language code defining the graphical user interfaces of the tool. The ATL community of programmers had published on their web site [7] a Java code allowing the execution of ATL transformation rules.

5.1. Run ATL transformations rules starting from Java

To run ATL transformation rules starting from Java code, a simple *RunTransfoJava* [7] program defined in the page of ATL/Developer Guide is used. This program shows how to run ATL transformations programmatically, including the loading (injection) and saving (extraction) of corresponding required meta-models (PBCL and target platform meta-models) and models (PBCL scenario and PBCL scenario able to be implemented in the target platform).

The main idea is to encode the “.ecore” meta-models (eg PBCL.ecore, Moodle.ecore and PhpCollab.ecore) and develop the ATL transformation rules aside, in files with an .atl extension (eg PBCL2Moodle.atl, PBCL2PhpCollab.atl).

At transformations runtime, the EMF environment with ATL plug-ins generates automatically an .asm file (eg PBCL2Moodle.asm, PBCL2PhpCollab.asm, etc.). This file with the two meta-models files (the PBCL meta-model and that of the platform chosen by the user) must be run from *RunTransfoJava* class.

For each target platform, there is a definition of a *RunTransfoJava* class allowing the running of the PBCL meta-model, its meta-model and the models corresponding to the PBCL scenarios. Each Java file (*RunTransfoJava* class) is called from a graphical user interface (GUI).

Consequently, the PBCL-STT tool is composed of: Java classes containing all the functions needed to run the ATL transformation rules (a Java file per platform), and also files for the interfaces developed with the Java language.

In the subsection below, the PBCL-STT is presented.

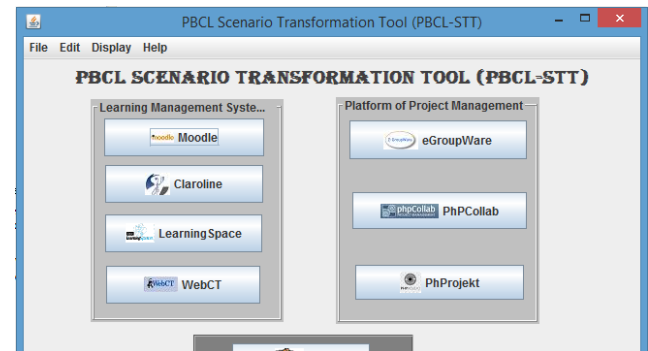


Figure 6. Main interface.

5.2. Presentation of the PBCL-STT

When launching this tool, the main interface also called home interface is displayed (Figure 6). It consists of two blocks of main buttons plus an “Exit” button. The first block (on the left) contains buttons allowing choosing a LMS platform, and the second block (on the right) contains buttons allowing choosing a project management platform. By clicking one of these buttons, instructor specifies to which platform he wishes to transform his PBCL scenario.

All functions presented above are also accessible from the main menu. Instructor can open the main menu “File” to choose “Transform” and after he chooses between “Transform toward a LMS” or “Transform to a project management platform” to find LMS and project management platforms. When he chooses a platform (LMS or project management platform), another menu opens, distinguished by the name of the platform selected above both buttons “Browse” and “Transform”. On the “Transform” button, there is also a logo of the chosen platform.

This interface allows the instructor to load his PBCL scenario. To load it, he must click on the button “Browse” allowing him to specify the location of the PBCL scenario on the computer or on an electronic device memory. After loading the chosen PBCL scenario in a “container” with a “scrollbar”, it is now ready to be transformed toward a PBCL scenario able to be implemented in the chosen platform. The instructor, simply, clicks on the “Transform” button designated by the logo of the target platform (Figure

7). After this click, a new window appears (Figure 8) containing the transformed scenario.

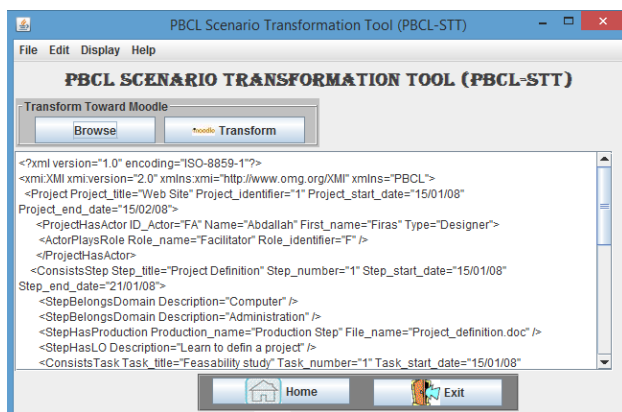


Figure 7. Loading PBCL scenario.

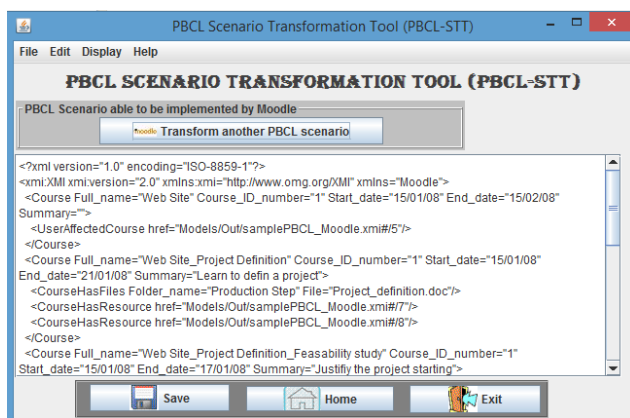


Figure 8. Transformed PBCL scenario.

In this window, the instructor can: click on the “Transform another PBCL scenario” button to return to the loading window of a PBCL scenario, save the transformed PBCL scenario by clicking on “Save” button at the bottom right, return to the main window by clicking the “Home” button to choose another platform, or exit the tool by clicking “Exit”.

6. Conclusion and perspectives

The proposed PBCL-STT graphical interface transformation tool capitalizes a transformation rules base from the PBCL meta-model toward several meta-models that correspond to several LMS and internet-enabled system for use in projects. It allows reducing the gap between instructor, expert in PBCL domain, and pedagogical engineer expert in technical platform. This is ensured by a drawing up a communication bond consisting in transforming a PBCL scenario into a PBCL scenario able to be implemented in a target platform. Therefore, the instructor elaborating his PBCL scenario by using the proposed PBCL meta-model, can transform it into a PBCL scenario which can be implemented in the target platform, using the developed transformation rules. The transformed PBCL scenario is given to the instructional engineer who is charged to deploy it in the platform. The

automation process starting from elaborating a PBCL and arriving to implementing it in the target platform is still not complete since it requires instructional engineer support. Therefore, further research should be concentrated on enabling instructor deploying his PBCL scenario on his chosen platform with the minimum instructional engineer support.

References

- [1] Abdallah, F., Toffolon, C. &Warin, B. (2007). “Assistance to project-based learning support: from learning models to platforms”, IADIS, Lisbon Portugal.
- [2] Abdallah, F., Toffolon, C. &Warin, B. (2008). “Models transformation to implement a Project-Based Collaborative Learning (PBCL) scenario: Moodle case study”, IEEE-ICALT, Santander, Cantabria, Spain.
- [3] Abdallah, F. (2014), “Proposing a Framework for Choice Assistance of a Technology Enhanced Learning System Dedicated to Project-Based Collaborative Learning”, *International Arab Journal of e-Technology*, Vol. 3, No. 3, pp. 163-170.
- [4] ACM, and IEEE Computer Society, Computer Science Curricula 2013, 2013.
- [5] Allilaire, F., Idrissi, T. ADT: Eclipse Development Tools for ATL. EWMDA-2, Kent, September 2004.
- [6] ATLAS group LINA, and INRIA Nantes. 2006. Atlas Transformation Language – ATL User Manual, INRIA.
- [7] ATL/Developer Guide, https://wiki.eclipse.org/ATL/Developer_Guide#Core_API
- [8] Berggren, D., Burgos, J.M., Fontana, D.,Hinkelman, V., Hung, A.,Hursh, and Tielemans, G., (2005), “Practical and Pedagogical Issues for Teacher Adoption of IMS Learning Design Standards in Moodle LMS”, *Journal of Interactive Media in Education*, p.Art. 3. DOI: <http://doi.org/10.5334/2005-2>
- [9] Bézivin, J.; Dupé, G; Jouault, F.; Pilette, G.; and EddineRougui, J. 2003. “First experiments with ATL model transformation language: Transformation XSLT into XQuery”, In OOPSLA 2003 Workshop, 1-18. Anaheim, California.
- [10] Bizonova Z., Ranc D., Drozdova M. 2007. “Model Driven E-learning Platform Integration” 2nd PROLEARN Doctoral Consortium in Technology Enhanced Learning, Crete, Greece.
- [11] Botturi, L., Derntl, M., Boot, E., &Figl, K. (2006) “A Classification Framework for Educational Modeling Languages in Instructional Design”, the IEEE International Conference on Advanced Learning Technologies (ICALT’06.

- [12] Budinsky, F., Steinberg, D., Ellersick, R., & Grose, T. (2004). Eclipse Modeling Framework. Addison-Wesley Professional, ISBN: 0131425420.
- [13] Burgos, D., Arnaud, M., Neuhauser, P., & Koper, R. (2008). "IMS Learning Design: la flexibilité pédagogique au service des besoins de l'information", Available: <http://www.epi.asso.fr/revue/articles/a0512c.htm>.
- [14] Canals, A., Le-Camus, C., Feau, M., Jolly, G., Bonnafous, V., and Bazavan, P., (2005), "Une Utilisation opérationnelle d'ATL: l'intégration de la transformation de modèles dans le projet TOPCASED", *Génie Logiciel*, (73), pp. 21–26.
- [15] Conole, G. (2013). Explorations in the Learning Sciences, Instructional Systems and Performance Technologies. *Designing for Learning in an Open World*, vol. 4, ISBN 978-1-4419-8516-3 ISBN 978-1-4419-8517-0 (eBook) DOI 10.1007/978-1-4419-8517-0. Springer New York Heidelberg Dordrecht London
- [16] Dodero, J. M., Martinezdel Val, A., & Torres, J. (2010), "An extensible approach to visually editing adaptive learning activities and designs based on services", *Journal of Visual Languages and Computing*, 21(6), pp.332-346.
- [17] Fabac, R., Radošević, D., & Pihir, I. (2010), "Frequency of use and importance of software tools in project management practice in Croatia", In: Proc. Of 32nd Int. Conf. on Information Technology Interfaces, Cavtat.
- [18] Feldman Michelle L. (2003). PhpCollab Team Manual, *Manual*.
- [19] Garcia-Robles, R., Diaz-del-Rio, F., Vicente-Diaz, S., & Linares-Barranco, A., (2009), "An eLearning Standard Approach for Supporting PBL in Computer Engineering", *IEEE TRANSACTIONS ON EDUCATION*, VOL. 52, NO. 3.
- [20] George S. (2002). "SPLACH: a Computer Environment Supporting Distance Project-Based Learning", World Conference on Educational Multimedia, Hypermedia & Telecommunications, ED-MEDIA, Denver, Colorado, USA.
- [21] Hernandez Leo, D.H., Asensio Pérez, J.I., & Dimitriadis, Y.A. (2004). "Imslearning design support for the formalization of collaborative learning patterns", the IEEE International Conference on Advanced Learning Technologies (ICALT'04), pp.350-354, Joensuu, Finland.
- [22] Koper R. (2006). "Current Research in Learning Design", *Educational Technology & Society*, pp. 13-22.
- [23] Laforcade, P. (2005). "CPM: Concepts, Techniques and Tools for an Educational Modeling Language", In the 4th IEEE International Conference on Cognitive Informatics, University of California, Irvine, USA.
- [24] Laforcade, P., Barré, V., & Zendagui, B. (2007). "Scénarisation Pédagogique et Ingénierie Dirigée par les Modèles. Cadre d'étude pour la définition de langages et environnements-outils de scénarisation pédagogique spécifiques à des domaines", *EIAH'07*, Lausanne, pp. 257-268.
- [25] Martel, C., Vignollet, L., & Ferraris, C. (2006). "Modelling the case study with LDL and implementing it with LDI", the IEEE International Conference on Advanced Learning Technologies (ICALT'06), The Netherlands, pp. 1158-1159.
- [26] ModX: Modelling Everything! <http://www.lifl.fr/modx>, Visited on 16/09/2016.
- [27] Moodle (2017) <https://Moodle.org> Visited on 10/09/2017
- [28] Norm IMS-LD, part 1, 2, and 3 (2003). "IMS Learning Design (Best Practice Guide, Information Binding, Information Model), *IMS*, Available: <http://www.imsglobal.org/learningdesign/index.html>.
- [29] Object Management Group, (2003). *Technical Guide to ModelDriven Architecture: The MDA Guide v1.0.1*.
- [30] Pereira, A., Gonçalves, R., & Wangenheim, C. (2013). "Comparison of open source tools for project management," *International Journal of Software Engineering and Knowledge Engineering*, vol. 23, n. 2, pp. 189-209.
- [31] phpCollab Project Management (2017) <http://www.phpcollab.com/> Visited on 10/09/2017
- [32] PMI – Project Management Institute (2013). *A Guide to the Project Management Body of Knowledge*, 5. ed., Newtown Square.
- [33] QVT (2002). OMG/RFP/QVT MOF 2.0 Query/Views/Transformations RFP.
- [34] Rak, I., Teixido, C., Favier, J., & Cazenau, M. (1990). "La démarche de projet industriel", Paris.
- [35] Sendall, S., & Kozaczynski, W. (2003). "Model Transformation: The Heart and Soul of Model-Driven Software Development", *IEEE Software*, Vol 20, Issue 5, pp. 42–45.
- [36] Strembeck, M., & Zdun, U. (2009) "An approach for the systematic development of domain-specific languages", *Softw. Pract. Exper.* pp. 1253–1292, published online 28 August 2009 in WileyInterScience (www.interscience.wiley.com). DOI: 10.1002/spe.936
- [37] Tho- Cursus inc., Formation et Culture Numérique (2017) : <http://thot.cursus.edu>



Firas Mohamad Abdallah is Associate professor at the Lebanese University, Saida, Lebanon. He holds PhD degree in computer science from the University of Maine, 2009. Dr. Abdallah has more than 10 years of teaching experience, including software engineering project management experience. His research interest has centered on Technology Enhanced Learning Systems (TELS) and Model Driven Architecture (MDA). He supervises some e-learning projects in, Management Information Systems and software engineering process.

LAMJET