Model based Engineering of Learning Situations for Adaptive Web Based Educational Systems

Thierry Nodenot
LIUPPA - IUT Bayonne
3 avenue J. Darrigrand
64115 Bayonne, France
Tel : 33 5 59 57 43 53
Thierry.Nodenot@iutbayonne.univ-pau.fr

Christophe Marquesuzaa
LIUPPA - IUT Bayonne
Château Neuf – Place Paul Bert
64100 Bayonne, France
Tel : 33 5 59 57 43 45
Christophe.Marquesuzaa@iutbayonne.univ-pau.fr

Pierre Laforcade
LIUPPA – faculté des sciences
Avenue de l'Université
64014 Pau, France
Pierre.Laforcade@univ-pau.fr

Christian Sallaberry
LIUPPA – IAE
Avenue de l'Université
64014 Pau, France
Tel : 33 5 59 40 80 67
Christian.Sallaberry@univ-pau.fr

ABSTRACT

In this paper, we propose an approach for the engineering of web based educational applications. The applications that we focus require advanced functionality for regulating and tutoring learners' activities (dynamics of learning). Our approach aims at proposing models, not only to describe details of such learning situations, but also to characterize the constraints that the Learning Management System exploiting such situations must satisfy; in this sense, this approach also contributes to the specification of the Adaptive Web Based Educational System (AWBES) fitted to a particular learning situation. Moreover, this approach for the engineering of learning situations conforms to current software engineering research works.

Categories and Subject Descriptors

D.2.1 Requirements/Specifications, D.2.2 Design Tools and Techniques, H.1.2 User/Machine Systems.

General Terms

Design, Human Factors, Languages.

Keywords

Architectures and designs for web-based learning delivery environments, specification of educational applications, Models and Metamodels, UML Language.

1. INTRODUCTION

For years, numerous research prototypes (intelligent tutoring systems, adaptive hypermedia applications) have remained unused in learning laboratories. More recently, industry brought to Education some Learning Management Systems (LMS) like BlackBoard [1] or WebCT [2] to promote intensive web-based Education. Now, courses using such LMS are very widespread, even if their tutoring capabilities are quite limited.

The purpose of this article does not consist in evaluating or criticizing theses systems but rather to study the design problems that can occur when an educational designer wants to develop an Adaptive Web Based Educational System fitted to a particular learning situation.

In the first part, we present several systems that we consider to be Adaptive Web Based Educational Systems (AWBES). When specifying a learning situation, we suggest that it is necessary to also specify how the components of a given AWBES will meet the requirements of such a situation. Thus, in the second part we present an approach for describing such learning situations together with the constraints that must satisfy the chosen AWBES. Considering focus of this paper, we deliberately conceal the IMS-LD specification [3]; the reader will find in [4] such a comparison between our models and the IMS-LD specification.

Conclusions present our current works in order to validate and to improve the models proposed into this paper.

WWW 2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005

2. TOWARDS DISTRIBUTED AND ADAPTABLE LEARNING MANAGEMENT SYSTEMS

If you go by current Open Distance Learning standards, all the learning situations look similar. We do not agree with this. From an industrial viewpoint of education, we can accept that a learning situation consists of a predefined sequence of activities mainly based on information presentations (whatever rich the offered interactions are: simple HTML pages, Flash or Toolbook animations, etc). Like many people, we expect more from Web-based education. This expectation is strengthened when we consider new types of Learning Management Systems (LMS), such as the uPortal or OKI infrastructures in the United States, or such as some CampusSource projects like the OpenUss container in Europe. Unlike monolithic platforms like WebCT or Blackboard, these new systems are based on an open and interoperable architecture providing the developers with a set of components. Such an architecture facilitates the development of an environment enhancing the training process thanks to the integration of educational services put at the user disposal by application servers distributed all over the web [5].

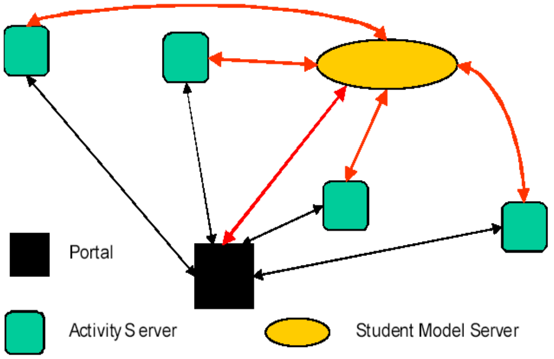


Figure 1: A distributed educational application [5].

Such activity servers can serve parameterized questions and assessments, annotated examples and solutions for given exercises, dynamic activity sequencing, etc. Technology is now available to implement such distributed applications with open standards like the java language, the Apache Software Foundation toolkit, etc.

- The educational portal "uPortal" (see http://mis105.mis.udel.edu/ja-sig/uportal/) implements the "Channel registry" in order to interoperate with other servers according to predefined protocols. For example, the "Remote Channel Proxy" is a channel allowing to very easily plug, in a uPortal channel, services using the SOAP protocol. This channel allows to communicate with and present to the learner the contents of a channel living in another instance of uPortal, somewhere on the Web.
- The OpenUss container (see http://openuss.sourceforge.net/openuss/) is built on top of the J2EE standard (Java 2 Enterprise Edition) that implements Application Service Providers which are necessary for the execution of distributed services, whether educational or not.

Researchers and others educational designers are now aware of the advantage they can take from working in a projects community: they can design and implement interoperable educational components or Web services that they put at the community's disposal, they can use educational components developed by other researchers, etc. These numerous exchanges facilitate the evaluation of the research works, and contribute to go beyond untested prototypes. It is yet possible to exploit numerous educational components or services: DELI (see http://delicon.sourceforge.net/) is an open source library that allows server applications to access CC/PP (Composite Capabilities / Preference Profiles) included in W-HTTP requests. ELM-ART (see www.psychologie.uni-trier.de:8000/projects/ELM/elmart.html) helps students to solve news problems by suggesting them relevant successful problem solving cases from an earlier experience. COW (Cooperative Open Workflow) is a component allowing to link teaching activities according to pedagogical events specified by the teacher when he/she formalizes these activities [6]. Other components are also available for sequencing educational contents and assessing learner skills (see http://test.com or see http://www.webassign.net/)

Whatever the quality of available components, developing an educational environment from such components is not an easy job, particularly when the goal consists in implementing an Adaptive Web Based Educational System (AWBES).

The "Adaptive" qualifier can take different meanings [7], [8]: *Adaptive Guidance* means providing the learner an optimal navigation path through the learning material according to his personal preferences, goals, navigation history and/or tested knowledge. *Adaptive presentation* means the adaptation of the content itself according to the same information as with adaptive guidance. *Adaptive collaboration* enables the generation of collaboration groups (with adequate functionality) of users with matching learning goals or finding the most competent user for answering a question about a specific learning topic. In the context of this research, we propose another meaning for the "Adaptive" qualifier: *specialization of an existing component* in order that the services provided by this component fit the requirements of the learning situation that it will serve. [7] studied this question in his PhD thesis dedicated to "Contextual Web Services for Teaching"(cf figure 2).

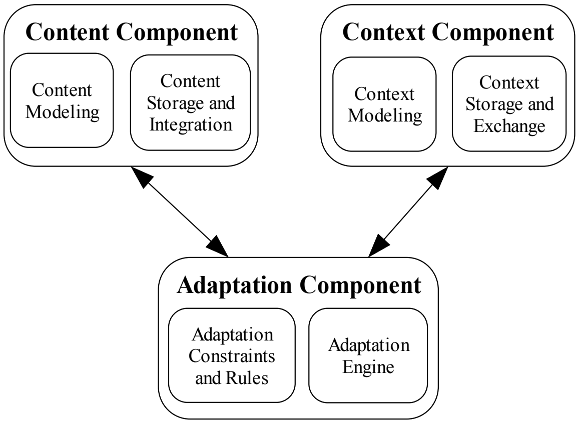


Figure 2 : Content / Context Adaptation.

He distinguished:

- The *"user" context:* learners, human tutors acting within a learning situation (possibly a cooperative learning situation). The learning situations should conform to the behavior of these actors.
- The "*environment*" *context* clarifies the real characteristics of the LMS (or any other software) from which the learning situation is exploited.
- The "*time*" context allows to define how absolute and relative time acts upon the dynamics of the learning situation proposed to the actors.
- The "*location*" context specifies, for a distributed platform, both the location of available components and the information expected by these components for a correct functioning. This dimension is not essential for our design model; we therefore do not consider this specific context in the continuation the paper.

We consider that the educational specification must lead the designers to define how the context (user, environment, time) acts upon the contents to be taught. We try to link:

- On the one hand, a learning scenario as defined by the designer without taking into account the constraints of a real environment of tools (the contexts),
- On the other hand, some abstractions of software components. Some of them can have an educational aim (e.g. a component for sequencing learning activities from a learner profile), others must be specialized to behave as educational components (e.g. a "chat" component, a "whiteboard" component, a "clock" component, …).

In the next part, we shall propose an approach for the engineering of learning situations. This approach aims at specifying the learning situation together with the Learning Management System that will later enable students learning from this situation.

3. OUR VIEWS ON ENGINEERING LEARNING SITUATIONS

The models presented in this part of the paper have been designed to help us developing a Learning Management System (LMS) fitted to Cooperative Problem Based Learning Situations (PBLS) [9]. Our aim was not to produce such a LMS from scratch but to constraint the behaviour of the components embedded in a given LMS (here OpenUss) to enable this system to fit PBLS-like educational requirements. To this end, our approach consists in mapping some abstract models of the components of a given LMS with some models of the learning scenarios defined for a particular cooperative PBLS.

At the moment, we are in the process of checking all these models against a cooperative PBLS called Smash[1] [10] and we are still upgrading these models when they fail to fit with the Smash PBLS educational requirements.

3.1 Three-Dimensional models

The proposed models are part of a UML Profile called CPM for « Cooperative Problem based learning Metamodel» [11], [12]. These models have:

- a vertical dimension that enables one to represent a Learning Situation (particularly a cooperative PBLS) at different levels of abstraction (from an external view to a very concrete description of the learning scenario and of the learners activities).
- a generalization dimension that promotes designs based on two levels of models : *the metamodel* of the CPM Profile from which educational designers can produce dedicated *models* suited to fit the requirements of the specific learning situation that they want to formalize. From these dedicated models, the designers can instantiate *objects* that will be put at the learners disposal according to the learning scenario.
- an horizontal dimension that enables one to split a learning specification into different focused models. Each model has its own objectives and terminology. All these models are instances either of the CPM Metamodel or of other well known software engineering metamodels.

In the next paragraphs, we detail the concepts of the CPM Profile, stressing on the added value of the three-dimensional models.

3.2 Added value of the vertical dimension

The next figure describes the different abstraction levels of the CPM Metamodel.

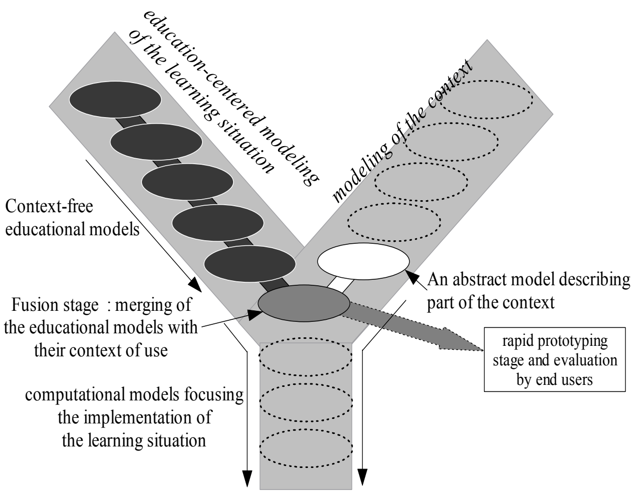

education-centered modeling of the learning situation
modeling of the context
Context-free educational models
An abstract model describing part of the context
Fusion stage : merging of the educational models with their context of use
rapid prototyping stage and evaluation by end users
computational models focusing the implementation of the learning situation

Figure 3: Vertical Dimension of the models.

This figure that stems from the "Y" model of the MDA Architecture [13], shows the different levels of models that we consider in the CPM profile : these models are represented as coloured ovals (black, white, grey). Dotted ovals represent models that we do not consider in the CPM profile. In figure 3,

- The left part of the "Y" is purely educational. Models therein enable one to describe the learning situation to be later implemented, the predicted learning scenario, the dynamics of the activities proposed to the learners, ...

  Since we consider that modelling the learner context is particularly important, we decided (as detailed in part 3.4.2) to consider the models focusing the user (its cognitive abilities and behaviours) as part of our context-free educational models (in the IMS-LD specification, the authors followed a similar approach).
- The right part of the "Y" is mainly out of the scope of our CPM profile. In the MDA architecture, the models appearing in that part of the « Y » describe the technological environment of a system (Platform Description Model or PDM): the lower the models, the more abstract (and stable) the specification of the platform. In the CPM profile, we only consider these abstract specifications of the components that constitute the context of execution of a learning situation (cf the white oval in figure 3).
- The fusion stage (see the gray oval in figure 3 consists in adapting the context-free educational models of a learning situation (black ovals in figure 3) with its context of execution (white oval in figure 3).
- The lowest part of the « Y » is not in the scope of our profile (see dotted ovals). On the contrary, from our models, we use prototyping techniques to evaluate how end-users learn from the described learning situation. Later on, we shall be thinking of transforming our models into platform specific models (PSM) what will enable us to check the compatibility of our metamodel with technical standards like ADL SCORM for instance.

We repeat here that the objective of our profile is not to propose a full set of metamodels covering the different contexts that we consider being important to characterize learning

[1] The Smash PBLS focuses good driving behaviours and Primary teachers helped us validating (with their pupils) the Learning Scenario of this particular cooperative PBLS.

situations. We rather propose to design some mappings between the concepts of our metamodel and concepts of other current and future metamodels, particularly those proposed by the OMG consortium.

We are studying mappings of CPM with several metamodels:

- The Software component metamodel from the UML 2.0 specification [14] which is particularly convenient for describing at a high abstraction level the services offered by the components of a LMS.
- Part of the profile called « UML Profile for Schedulability, Performance, and Time Specification » [15], in order to describe how absolute and relative Time sequencing has an effect on the dynamics of a learning scenario (Time is particularly important for the regulation of cooperative learning situations).
- The model of process execution proposed by the OMG consortium [16] to handle activity execution and termination, events raised during the lifecycle of an activity, etc. This model is useful to enable educational designers detailing how learning activities must be dynamically regulated at runtime. Yet; the description of the static part of this process model (activities to be executed, data and events to be tracked, etc.) remains in the focus of our CPM metamodel (left part of the "Y").

From this engineering approach, Pierre Laforcade proposed in [17] his works allowing to merge models of a learning activity with some abstractions of software components that are embedded in a given LMS. We shall give an example of such a fusion in part 3.4.5.

3.3 Added value of the generalization dimension

There is a wide range of formalisms for describing learning situations. Such formalisms can take the form either of a computational language, or of an ontology (with its terminology and assertions), or of an XML dialect.

We chose to describe learning situations from formalisms based on the UML language. This enables us to efficiently model the dynamics of learning activities: we consider that other formalisms like OKBC (cf http://www.ai.sri.com/~okbc/okbc-2-0-3.pdf) or OKBC by-products -such as « Protege-2000 » models [18]-, like the learning ontologies used by R. Mizogushi's research team [19], [20], fail to efficiently describe such dynamics of activities. On the contrary, in contrast of what is often said, the UML class diagram is not the central diagram of a UML specification; this diagram offers one view among others of a system and UML proposes State-Transition diagrams, activity and collaboration diagrams that provide the designers with a convenient toolkit for describing dynamics of (learning) activities.

Next figure presents different levels of models for modelling learning situations.

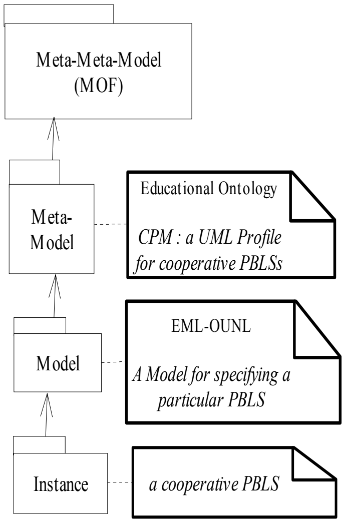


Figure 4: models and metamodels.

Models. It is the level at which the modeling of problems, solutions and systems occurs and it is used to formalise specific expressions regarding a given subject.

Ontologies and meta-models. An *ontology* provides a list of the basic elements giving rise to a common vocabulary for a domain application and the relationship and dependencies between them[19], [21]. It is possible to add formal definitions to prevent any unexpected interpretation of these concepts and necessary relations and constraints which can also be formally defined as a set of axioms. Furthermore, most advanced ontologies are executable. On this basis, [20] defined an ontology forming the conceptual structure for the definition and construction of CSCL environments and for the analysis and assessment of group collaboration.

A *meta-model* is for specifying the syntax and the semantics of a class of models. It can be used to define a set of concepts and properties of a particular domain; these concepts become the basic building blocks for specifying a particular problem. It is in this sense that any ontology may be considered as a kind of meta-model. Among available metamodels, a UML Profile provides useful add-ons.

UML Profile. A UML profile is a variant of UML which uses the extension mechanisms of UML [22] in a standardised way and for a particular purpose (see UML 1.4 in the 2.14.4 Semantics): " … *A profile stereotype of Package provides a mechanism for grouping one or more related refinements (extensions) of standard UML Semantics. These extensions are normally intended for a particular way of using and interpreting UML, such as a specific domain or purpose. Profiles can contain stereotypes, tag definitions and constraints. They can also contain data types that are used by tag definitions for (informally) declaring the types of values that can be associated with tag definitions. In addition, a profile package can specify a related model library and identify a subset of the UML Meta model that is applicable for the profile. In principle, profiles merely refine the standard semantics of UML by adding further constraints and interpretations that capture domain-specific semantics and* *modeling patterns. They do not add any new fundamental concepts.*" Figure 4 and these previous definitions show that an ontology, a meta-model and a UML profile have quite similar characteristics. Unlike a model such as IMS-LD, educational ontologies, educational meta-models and a UML profile for co-operative PBLSs are not supposed to deal with the total knowledge of the target world. They represent an agreement within a community from which practitioners can then draw the particular model of any educational situation they want to describe.

A Meta-meta model defines the language used for creating meta-models. The MOF (Meta-Object Facility) [23] is such a language from which the UML language is described.

So "Meta-modelisation" is an added-value modelling tool [24], that we use:

1. to design models fitted to a particular educational designer or to a particular type of learning situation. The CPM profile represents a backbone to the target world and it represents an agreement within a community. From such an agreement, practitioners can then draw the particular model of any educational situation they want to describe: one can design educational models whose level of complexity goes with the complexity of the learning situation to be designed [4].
2. to connect different models or metamodels representing on the one hand the learning situation, on the other hand the context of use of such a situation (contexts Time, Environment, …). Since all these models are based on the same meta-metamodel, the MOF (Meta Object Facility), connections of models become easier.

The profile mechanism, as defined in the UML reference Model, is natively interpreted and exploited by UML CASE tools for drawing UML models. Recently, we implemented this profile using the Profile Builder of the "Objecteering" UML case tool. Now, even if it is not our primary objective, our UML profile for co-operative PBLSs can provide practitioners with CASE tools but without any extra work.

Up to now, we did not study tools and techniques for transforming our CPM based models into other models or languages, for example into models in accordance with the IMS-LD specification.

3.4 Added value of the horizontal dimension

We decomposed the CPM profile into several packages that represent complementary views of a learning situation (at a given level of abstraction). In figure 4, we summarize the CPM structure.

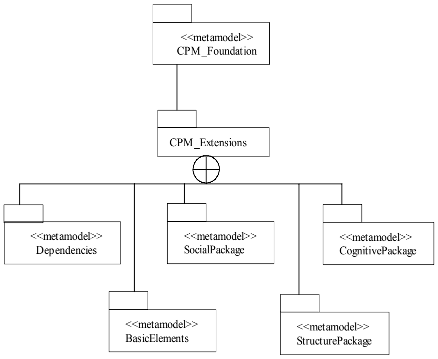


Figure 5: The structure of the CPM Metamodel.

3.4.1 The Basic Elements Package

Below is the top level package presenting the Basic Elements of our UML Profile. The semantics are the same for this basic elements ("Actor", "Role", "Activity structure", "Scenario", "Resource", "Constraint") as in IMS-LD. But in our Package, these elements can be linked by means of a dependency relation. Instances of dependency are "ConsistsOf"/"IsEquivalent" from Resource to Resource, "RefersTo"/"ConsistantWith" from Scenario to Scenario, "Precedes"/"Implies" from Activity to another, "Governs"/"Promotes" from Scenario to Activity Structure, "Handles"/"Requires" from Educational Component to Resource, "Aggregates"/"Specializes" from Educational Component to Educational Component.

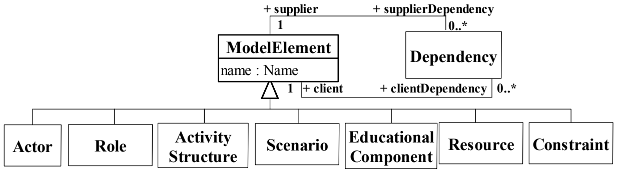


Figure 6: Basic elements Package.

The following paragraphs and figures will present extensions of this initial package. In the different figures, reused/specialized elements are in grey. These extensions of the basic elements focus on the following items: a) providing the Model with cognitive properties necessary to trace the learning/tutoring behaviours of actors (cf. cognitive package); b) doing a decomposition of an educational situation (scenario) into simpler learning/tutoring activities (cf. structural package); and c) managing co-operative work including sharing of resources and of learning/tutoring activities (cf. social package).

3.4.2 The cognitive Package

The cognitive package deals with information to model the components of a PBLS: these include learner's (mis)conceptions, predicted obstacles that a teacher wants learners to overcome, goals and success criteria within the PBLS, resources available to the learners, and so on.

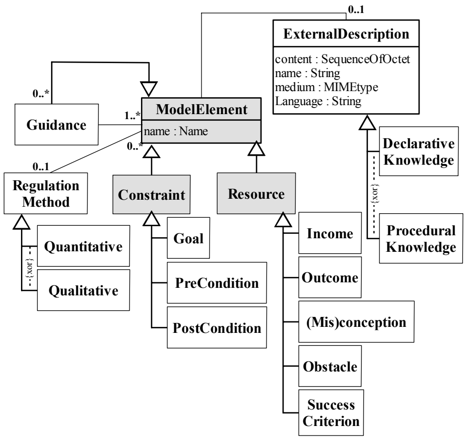


Figure 7: Cognitive Package.

In the previous figure, each of these elements is a subclass of the root Class called the "ModelElement" Class. At this level, we have added some extra cognitive information:

- Each ModelElement has an "ExternalDescription" which can be either Declarative or Procedural Knowledge. The content property of this ExternalDescription serves to describe information elicited under the form of a set of predicates, an algorithm, a state-transition diagram, etc
- As defined in [25], some guidance may be associated to each Model Element. This guidance provides more detailed information (FAQ, …) to the practitioners about the associated ModelElement.
- A Regulation Method can be linked to each ModelElement. Any regulation method which is quantitative is based on valuable quantities such as predicted answers, state-based behaviours of the learner, etc. If the regulation method is qualitative, then it is based on qualitative factors which can be evaluated by inference mechanisms on some Knowledge Base (see the Declarative and the Procedure knowledge classes). We presented in [9] UML stereotypes (such as action, transition, state and event) which enabled us to deal with guidance and the regulation methods connected to these learning activities.

3.4.3 The structural package

Calling on the Activity Theory [26], our structural package incorporates concepts for decomposing a global learning situation into simpler elements which can be assigned to learners or tutors. The basic unit is called an *activity:* structuring elements are called either *activity structure* (for decomposing an identified complex activity into simpler activities) or *scenario* (for identifying more independent activity structures). This package allows the designer to define his own links (concept *Sub_scenario_link* and concept *SubActivityLink*) in order to refine the decomposition process of a learning situation. Instances of links are: then", "or", "xor", "implies", "synchronized", etc. By contrast, IMS-LD only allows to decompose a scenario in terms of *method*, *play* and *act* (like our *Scenario* and *Activity Structure* levels). IMS-LD nor does offer more then a single link (with a "composed of" semantics) to decompose a learning session into simpler elements (such as "play" into "acts"). From a structural point of view then, IMS-LD is only a particular instance of this part of our meta-model.

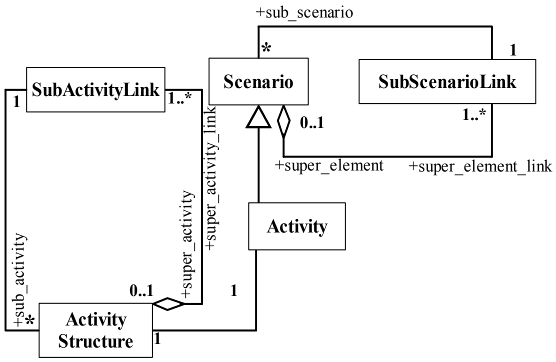


Figure 8: Structural Package.

3.4.4 The social Package

To start with the broadest terms, we consider actors, their roles and particular activity structures in which those actors are engaged. This is why we have included a *play/assigned to* dependency between *Actor* and *Role*, and an *involve* dependency between *Role* and *ActivityStructure* (cf. figure 3).

As can be seen in figure 6, any collaboration must involve *interactions* between *roles* and such interactions will lead to sharing *resources* and *activities*. It follows therefore that an *interaction protocol* involves different *roles*: one initiates the interaction and another or several others can be involved in this interaction.

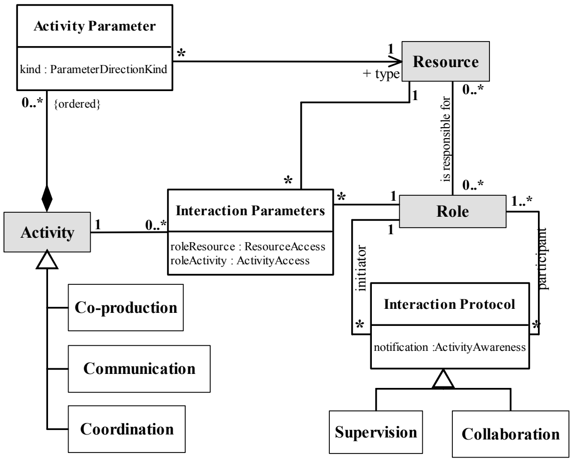


Figure 9: Social Package.

Within an interaction, roles can be considered to be at the same hierarchical level: this is called a *collaborative* interaction. However, a role can be senior in rank and it is then a *supervised* interaction. It follows that, within an interaction, the supervisor role may decide to modify activity parameters, resource and communication parameters assigned to the supervised roles.

Another feature incorporated into the social package is the clover of Ellis [27] which characterises any collaboration activity by three spaces allowing respectively to ***communicate***, ***co-produce*** and ***co-ordinate***. We use this when we are describing an interaction: thereby, roles execute communication, co-production or co-ordination activities. Each role/activity couple is described by *roleActivity* parameters. From these parameters, it is possible (at design time) to describe/to evaluate (at runtime) whether a role can be engaged in synchronous or asynchronous communication.

It is also possible to describe/evaluate whether a role can send/receive messages from other roles, and also whether it can attach any documents with these messages. The value of these parameters can be modified at runtime, depending on the defined activity structure or on the actions of a supervisor.

Furthermore, in this social package, two or more roles share resources within an interaction. Each role/resource couple is described by *roleResource* parameters. For example, if we consider a *co-production,* these parameters will describe (at design time and at runtime)/evaluate (at runtime) what kind of access a role may have on a particular resource i.e. creating, deleting, updating and reading. These parameters will also define the access parameters on each resource [28]: *Lockable* (floor control in order to authorise access or not), *Controllable* (runtime access control mechanisms i.e. read/write), *Observable* (runtime control in order to notify an access), etc.

Finally, within an interaction, it is important for each role to share views with others about its particular activities and about the activities in which the other roles are engaged. This awareness capability is managed by means of the *notification* attribute (Boolean value).

3.4.5 The Fusion_CL Package

In the previous figures, we have presented the concepts of the metamodel that enable educational designers to describe the role of actors in a learning scenario, their cognitive abilities and behaviours, the different activities (possibly, cooperative activities) in which they will be engaged, the resources that they can use and create, etc.

We present here the Fusion_CL package that focuses the “Environment” context. This package enables us to link abstract representations of software components (see the right part of the Y model in figure 3) with learning activities described with the previous packages of CPM (see the left part of the Y model in figure 3); UML specifications of these software components describe the functionality of an ideal Learning Management System : sequencing component, chat component, whiteboard component, etc. can then be represented as UML models and stored in a library for eventual reuse and specialisation. This package is an adaptation to our educational context of the UML 2.0 Component description [29].

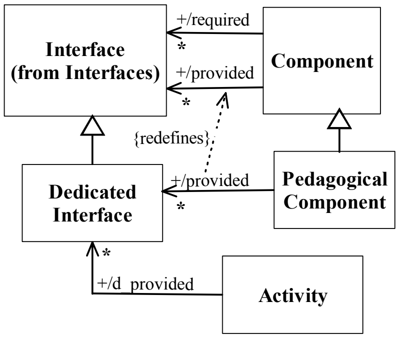
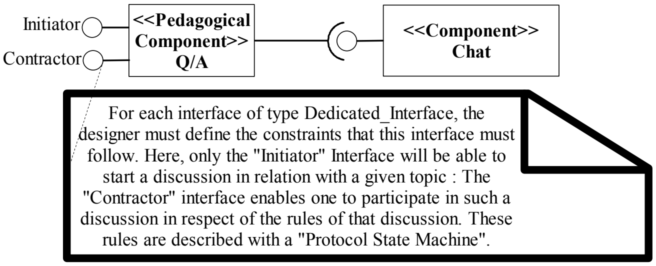


Figure 10: The Fusion_CL package.

Figure 10 details the metamodel for the Fusion_CL Package while figure 11 presents an instance for this metamodel. The metamodel maps an activity from the CPM Metamodel with concepts for describing Software Components (taken from the UML 2.0 Superstructure Specification).



For each interface of type Dedicated_Interface, the designer must define the constraints that this interface must follow. Here, only the "Initiator" Interface will be able to start a discussion in relation with a given topic : The "Contractor" interface enables one to participate in such a discussion in respect of the rules of that discussion. These rules are described with a "Protocol State Machine".

Figure 11: An example of model from the Fusion_CL package.

The instance (right part of the figure) is a model for a specialised « chat » component (whose name is Q/A for Question / Answer). This model states that only a tutor role can initiate an exchange with one or several learner roles (this « chat » was designed to enable a pedagogue organize debates with learners in order to study how one of them solved an exercise, also to carry out improvements if necessary). The specification of the « Q/A chat » component is documented in the note of figure 11: we used UML State transition Diagrams to formalize such behaviours for the “Initiator” and “Contractor” Interfaces.

When specified, such a component can be stored in a library of components and put at the disposal of other pedagogues if interested.

The Fusion_CL Package presented therein is a first step for mapping context free educational models (represented by the CPM metamodel) with abstract models describing their context of use (see the Y model in figure 3). At the moment, we formalize different types of educational components by specializing software components provided by the OpenUss platform.

4. Examples and application

In this part of the paper, we present some examples of models inferred from the CPM profile. We currently use the CPM Profile to exchange with pedagogues and practitioners in order to both:

- describe the way of Teaching/Learning that they favour.
- describe some particular learning situations that they would like to implement using a given Learning Management System.

If we consider the different levels of models presented in figure 4,

- describing some way of Teaching / Learning consists in producing some descriptions at the “Model” level (cf figure 12).
- while describing a particular learning situation consists in producing descriptions at both levels of “Model” (cf figure 13) and “Instance” (cf figure 14).

In the next page, we present some UML diagrams that we extracted from the whole set of diagrams dedicated to the design of the Smash Problem Based Learning Situation (PBLS). These diagrams must be considered as examples of what can be done and not of what must be done.

Figure 12 is a model that was produced to describe the conception of a given pedagogue about the concept of PBLS.

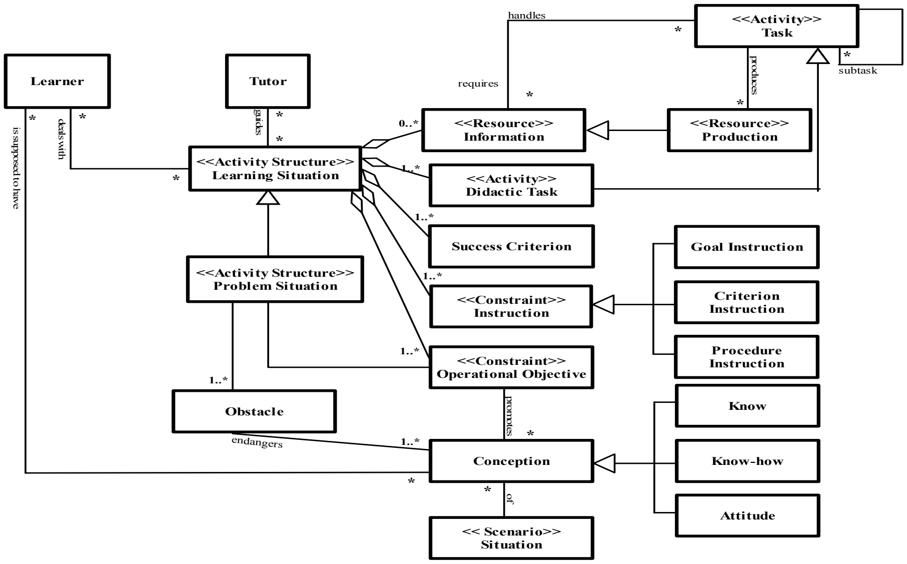


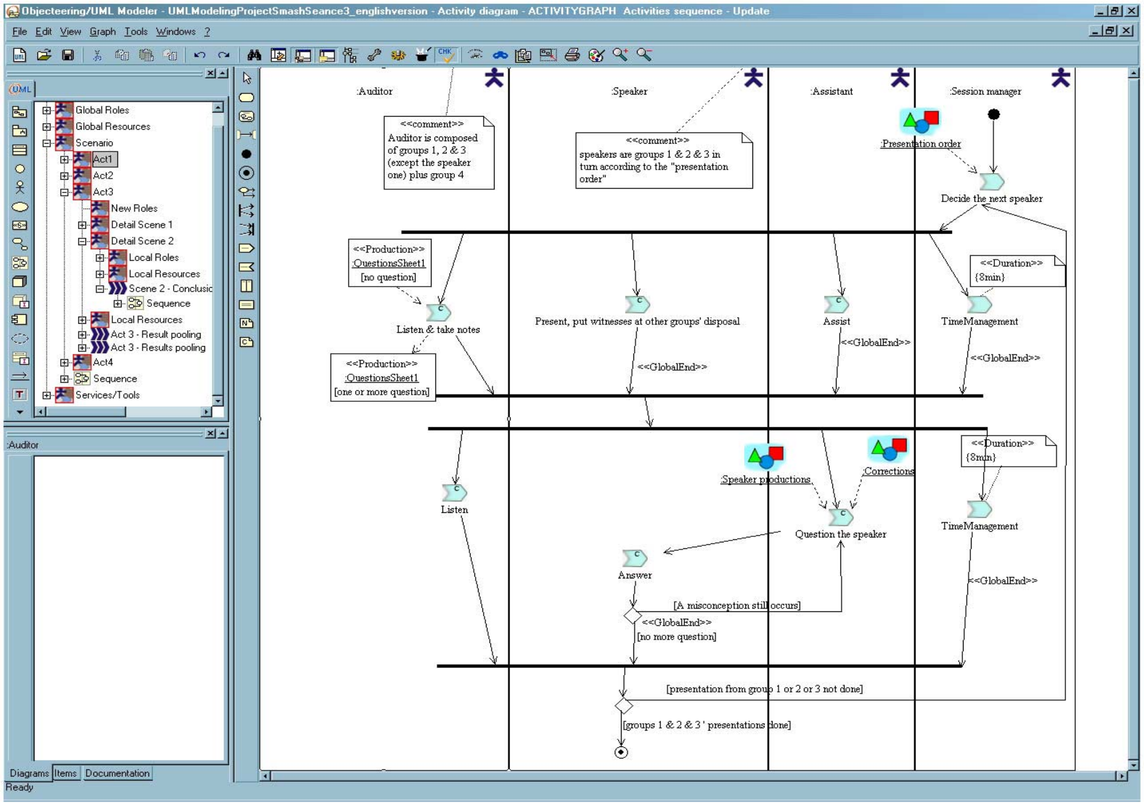
Figure 12: A Model defining the characteristics of a PBLS (inspired from [30]).

The concepts of figure 12 are either concepts of the CPM Metamodel or specializations of these concepts (see the name of their stereotype written between the symbols << and >>). Broadly speaking, the connections between these concepts follow the UML notation (inheritance relation, aggregation relation and association relation). But these connections can also be considered as specializations of the Dependency concept (see the Basic Elements Package in figure 6).

The model tells that a PBLS is a special Learning Situation whose main operational objectives promote some conceptions of a given situation (considering that the learners that will face this PBLS have probably some misconceptions about this situation).

An example of such an Operational Objective for our Smash PBLS could be: “The learner understands that driving safely consists not only in driving in accordance with the highway code but also in paying attention to all that goes on”. The situation could be the story of an accident. An example of (mis)conception could be: “Since I drive at an allowed speed with respect of the roadsigns, I do not endanger myself and the other drivers”.

From this model, the pedagogue can explain the details of the Smash PBLS. He can choose between the available types of UML diagrams to provide both static and dynamic models of the Smash PBLS. Next figures provide examples of such diagrams.


Objecteering/UML Modeler - UMLModelingProjectSmashSeance3_englishversion - Activity diagram - ACTIVITYGRAPH Activities sequence - Update

<<comment>>
Auditor is composed of groups 1, 2 & 3 (except the speaker one) plus group 4

<<comment>>
speakers are groups 1 & 2 & 3 in turn according to the "presentation order"

Figure 13: an example of dynamic diagram for the Smash PBLS.

Figure 13 is a snapshot of a dynamic diagram implemented with the Objecteering UML Case tool (the CPM Profile is also implemented with this Case tool). The diagram describes how different learners (playing different roles: Speaker and Listener) are supposed to cooperate within one Scenario. Within the Smash PBLS, this scenario enables different groups of learners to present and confront their works and conclusions from their previous analyses of different witnesses of a given accident.
Such a dynamic model is useful to infer the contextualized interfaces of tools that the actors will use at exploitation stage (see the Fusion_CL Package in paragraph 3.4.5).

Static diagrams are also important to describe the Smash PBLS. For example, we used several static diagrams to describe:

- the different security rules that the learners should know and should respect as safe drivers (knows, know-hows and attitudes),
- unsafe behaviours of drivers that must be incorporated within the witnesses of the studied accident (and corresponding safe behaviours for the same typical situation),
- …

In figure 14, we give some examples of instances for one of the static diagrams that we draw to describe security-rules. Among the whole set of security-rules that could be of interest, these instances detail which ones are in the focus of the PBLS (and of the targeted audience for this PBLS).

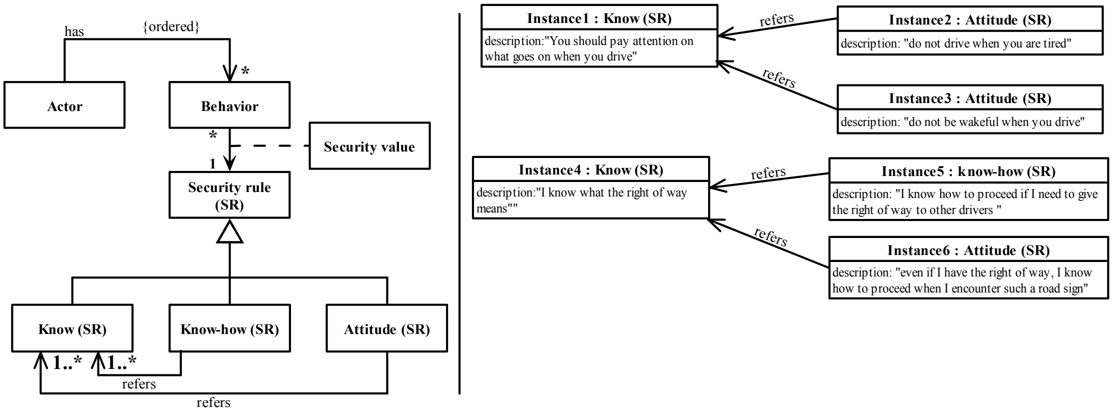

has
{ordered}
*
Actor
Behavior
*
1
Security value
Security rule (SR)
Know (SR)
Know-how (SR)
Attitude (SR)
1..*
1..*
refers
refers
Instance1 : Know (SR)
description:"You should pay attention on what goes on when you drive"
refers
Instance2 : Attitude (SR)
description: "do not drive when you are tired"
refers
Instance3 : Attitude (SR)
description: "do not be wakeful when you drive"
Instance4 : Know (SR)
description:"I know what the right of way means""
refers
Instance5 : know-how (SR)
description: "I know how to proceed if I need to give the right of way to other drivers "
refers
Instance6 : Attitude (SR)
description: "even if I have the right of way, I know how to proceed when I encounter such a road sign"

Figure 14: At left, a static diagram for describing security-rules. Some instances of such security-rules for the Smash PBLS (right)

5. Conclusions

In this paper, we proposed an approach for the engineering of web based educational applications. This approach aims at specifying the learning situation together with the Learning Management System that will later enable students learning from this situation. Needs to go towards such an approach are clear with Adaptive Web Based Educational Systems (AWBES) that can manage existing educational components distributed over the Web. We listed different types of educational components that can specialise a given AWBES, for instance uPortal or the OpenUss Container or the OKI infrastructure.

We emphasized problems that educational designers can encounter when specifying learning situations for such AWBES. Thus, we have proposed an approach for describing such learning situations together with the constraints that must satisfy that AWBES. The design models that we propose are based on the UML language. One facet of these models is their horizontal dimension (different views of a learning situation at a given abstraction level). Another facet is their vertical dimension (abstract/implementation views for the same element). The last facet is their generalization dimension that promotes mappings between metamodels to merge purely educational models with models describing the context of execution of a learning situation.

Perspectives of this work are numerous. First of all, we must propose some means for adapting the models of the CPM profile to designers with no computer-science background: we are thinking of putting at pedagogues' disposal some predefined design patterns and convenient methodological guidelines (UML is not a method but just a language). We are also studying a plan of action for the evaluation of the proposed packages of the CPM metamodel. We have already checked these models at a reasonable scale while specifying the Smash Problem Based Learning Situation (PBLS). Now, we are thinking of publishing on the Web the whole set of models that we drew for the Smash PBLS. This will probably facilitate exchanges with other research teams working in this field; by the way we shall get the feedback that we need to consolidate and to improve the CPM profile. Finally, our approach converges towards the modelling of Educational Components with the UML as well as the potential associated implementation of these components. Capturing learning and teaching activities that are basically cognitive processes comes up against inappropriate formalisms. We think that a suitable formalism is together helpful for representing all aspects of PBLS and easily leads to concrete software entities. Thus, we focus on the segmentation of activities in order to deliver components that can be quite easily assembled and deployed to provide opened and flexible distance learning platforms.

6. Acknowledgement

This research is supported by the Aquitaine Region (France), project number 200330204009A and Aquitaine Euskadi Navarre.

7. References

[1] BlackBoard, *Blackboard Course Management System*. http://www.blackboard.com

[2] WebCT, *WebCT Course Management System*. http://www.webct.com

[3] IMS, *IMS Learning Design Information Model*. 2003, IMS Global Learning Consortium. http://imsglobal.org/learningdesign/.

[4] Nodenot, T., et al. *Knowledge Modelling of Co-operative Learning Situations: Towards a UML profile*. in *11th International Conference on Artificial Intelligence in Education, International AI-ED Society*. 2003. Sydney (Australia).

[5] Brusilovsky, P. *A Distributed Architecture for Adaptive and Intelligent Learning Management Systems*. in *Workshop "Towards Intelligent Learning Management Systems" , 11th International Conference on Artificial Intelligence in Education*. 2003. Sydney (Australia).

[6] Peter, Y. and T. Vantroys. *A Flexible Workfow System for the Enactment of Pedagogical Scenarii*. in *4th International Conference on Information Technology Based Higher Education and Training*. 2003. Marrakech (Morocco).

[7] Schneemayer, G., *Contextual Web Services for Teaching*, in *Institut für Informatik*. 2002, Ludwig Maximilians Universität: München (Allemagne).

[8] Brusilovsky, P. and H. Nijhawan. *A framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities*. in *AACE World Conference on E-Learning, Elearn 2002*. 2002. Montreal (Canada).

[9] Sallaberry, C., et al. *Information modelling within a Net-Learning Environment*. in *12th Conference on Information Modelling and Knowledge Bases*. 2002. Krippen, Swiss Saxony (Deutschland): IOS Press.

[10] Oudot, C., *Modélisation UML d'une situation d'apprentissage coopérative*. 2003, Rapport de stage du DESS "Sciences Cognitives Appliquées", Université de Bordeaux 2.

[11] Laforcade, P., et al. *Profiling Co-operative Problem-Based Learning Situations*. in *IEEE International Conference on Cognitive Informatics (ICCI 2003)*. 2003. London (UK): IEEE.

[12] Nodenot, T., et al. *A UML Profile incorporating separate viewpoints when modeling Co-operative Learning Situations*. in *IEEE International Conference on Information Technology : Research and Education*. 2003. Newark (USA).

[13] OMG, *Model Driven Architecture (MDA)*. 2001. http://www.omg.org/docs/ormsc/01-07-01.pdf.

[14] OMG, *UML 2.0 Superstructure Specification*. 2002. http://www.omg.org/docs/ptc/03-08-02.pdf.

[15] OMG, *UML Profile for Schedulability, Performance, and Time Specification*. 2002. www.omg.org/technology/documents/formal/schedulability.htm.

[16] OMG, *Workflow Management Facility Specification v1.2*. 2002. http://www.omg.org/docs/formal/00-05-02.pdf.

[17] Laforcade, P. and F. Barbier. *UML Modeling for Cooperative Problem-Based Learning Situations:Towards Educational Components*. in *2003 IRMA International Conference*. 2003. Radisson Warwick Hotel, Philadelphia Pennsylvania, USA.

[18] Fridmann Noy, N., R.W. Fergerson, and M. Musen. *The knowledge model of Protégé-2000: combining interoperability and flexibility*. in *Proceedings of 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2000)*. 2000. Juan-Les-Pins.

[19] Mizogushi, R. and J. Bourdeau, *Using Ontological Engineering to Overcome Common AI-ED Problems.* International Journal of Artificial Intelligence in Education, 2000. volume 11, n°2.

[20] Barros, B., et al. *Applications of a Collaborative Learning Ontology*. in *MICAI'2002 Mexican International Conference on Artificial Intelligence*. 2002: Springer-Verlag.

[21] Mizogushi, R. *Ontology-based systematization of functional knowledge*. in *TMCE2002:Tools and methods of competitive engineering*. 2002. Wuhan (China).

[22] OMG, *White Paper on the Profile Mechanism*. 1999, Object Management Group Analysis and Design Task Force(Contact P. Desfray).

[23] OMG, *Meta-Object Facility (MOF) Specification*. 2000. http://www.omg.org/technology/documents/formal/mof.htm

[24] Breton, E., *Contribution à la représentation de processus par des techniques de méta-modélisation*. 2002, Thèse de doctorat de l'Université de Nantes.

[25] OMG, *The Software Process Engineering Management Metamodel (SPEM)*. 2001. http://cs.ua.edu/630/Notes,%20etc/OMG%20SPEM/submission%202%20-%20ad-2001-02-08.pdf.

[26] Engeström, Y., R. Miettinen, and R.L. Punamäki, *Perspectives on activity theory.* Cambridge University Press, 1998.

[27] Ellis, C. and J. Wainer. *A conceptual model of Groupware*. in *ACM Conference on Computer Supported Cooperative Work (CSCW'94)*. 1994. Chapel Hill, North Carolina (USA).

[28] Rubart, J. and P. Dawabi. *A UML Profile for Modeling Groupware*. in *8th International Workshop , CRIWG'2002*. 2002. La Serena (Chile): Published in J. M. Haake & J. A. Pino (Ed.): Groupware: Design, Implementation, and Use.

[29] Bruel, J.-M. and I. Ober. *The new UML 2.0 Component Model : Critical View*. in *Proceedings of Euromicro Conference*. 2003. Belek (Turquie).

[30] Meirieu, P. *Apprendre... Oui mais comment?* Collection Pédagogies. 1994, ESF Edition.