

Un langage de modélisation pédagogique basé sur UML

Pierre Laforcade [LIUM, Le Mans]

Thierry Nodenot, Christian Sallaberry [LIUPPA, Pau]

■ **RÉSUMÉ** : Cet article a pour but de présenter un langage de modélisation pédagogique issu d'un travail exploratoire sur l'apport potentiel du formalisme UML pour la conception et la mise en œuvre de formations à distance. Ce travail a été mené pour un type particulier de situation d'apprentissage, les situations-problèmes coopératives. Toutefois, cet article n'insiste pas sur les caractéristiques intrinsèques de ces situations d'apprentissage car l'intérêt principal est de présenter une approche originale de modélisation graphique capitalisable à d'autres situations d'apprentissage. Nous proposons ainsi le langage CPM, basé sur une spécialisation d'UML via les profils, dans le but de permettre l'élaboration de modèles de design pédagogique en amont des langages plus formels actuels tels que la spécification standard IMS-LD. Le langage CPM a été implanté dans un Atelier de Génie Logiciel UML existant et un prototypage d'environnement-auteur a pu être expérimenté afin d'envisager l'aide à la création, au suivi et à la maintenance des modèles pour un ingénieur pédagogique. Une mise à l'essai du langage sur un cas d'étude concret a permis de dresser les usages potentiels des modèles produits. Une expérimentation supplémentaire a porté sur la génération de modèles IMS-LD à partir de sous-modèles exprimés à l'aide du langage CPM. Toutefois un travail d'expérimentation reste encore nécessaire afin d'établir un cadre d'usage plus précis quant à l'approche UML pour la modélisation pédagogique. Ensuite, il s'agira d'aborder l'aspect méthode, complémentaire au langage et garantissant la bonne conception des modèles. Finalement, nous évoquons le rôle important et grandissant que doit jouer la méta-modélisation dans une perspective d'ingénierie de conception dirigée par les modèles pour les EIAH.

■ **MOTS CLÉS** : Modèles d'apprentissage ; méta-modélisation ; UML ; scénario pédagogique ; plateformes de formation à distance.

■ **ABSTRACT** : This article presents our results about an exploratory research work on the potential additions of the UML formalism for the design and implementation of distant learning situations. Our work focus on the specific Problem-Based Learning situations but could be suitable for other learning theories and approaches. Then we propose the CPM language based on a UML profile specialization in order to describe instructional design models upstream formal ones specified with standards like IMS-LD. Our language is implemented within an existent UML-CASE tool which has been customized in order to experiment end-users (instructional designers) facilities for the creation/support/maintenance of CPM models. The language has been tested on a complete case-study. We also briefly present and discuss additional works extending our contribution beyond the design stage. From these results, we stress the potentials of a Model-Driven instructional approach.

■ **KEYWORDS** : Instructional models; meta-modelling; UML; learning scenario; Open Distant Learning Platforms

- [1. Introduction](#)
- [2. Modèles et langages pour le design pédagogique](#)
- [3. Le langage CPM](#)
- [4. Expérimentations](#)
- [5. Conclusion](#)
- [Références bibliographiques](#)
- [Références à des sites Internet](#)

1. Introduction

Cet article présente et discute les résultats d'une étude centrée Génie Logiciel pour le domaine pluridisciplinaire de l'Ingénierie des EIAH. Notre intérêt s'est porté sur le champ de la formation à distance et plus précisément au niveau de la phase de conception des formations. Nous nous intéressons aux modèles de conception de type design pédagogique (Paquette, 2004) dont les objectifs sont i) de faciliter la spécification *prescriptive* des formations (quelles sont les ressources ? les critères de succès à spécifier ? les activités à proposer ? les rôles à jouer ? etc.) et ii) d'agir comme supports de réflexion et de communication pour l'équipe pluridisciplinaire chargée de la conception.

Nous avons restreint notre cadre de travail à une forme particulière d'apprentissage. Il s'agit de l'apprentissage par situation-problème (Meirieu, 1994) (*Problem Based Learning* ou **PBL** en anglais). Ce type d'apprentissage est issu du courant constructiviste selon lequel : 1) les connaissances sont construites, 2) l'apprenant est au centre du processus d'apprentissage et, 3) le contexte d'apprentissage joue un rôle déterminant (Deschênes et al., 1996). Dans nos travaux, la dimension coopérative pour la résolution des situations-problèmes est mise en avant : le concepteur peut ainsi baser le franchissement de l'obstacle par les apprenants sur toute forme de coopération ou de coordination entre apprenants et tuteurs (partage des responsabilités, résolution de conflits, échange de ressources, etc).

Pour l'instant, notre travail a porté principalement sur la mise au point d'un langage dédié à la mise en oeuvre de modèles de conception pour les situations-problèmes coopératives et ce, indépendamment de toute réflexion d'ordre méthodologique concernant la mise en place d'une méthode de design pédagogique adaptée.

Nous avons choisi de baser notre langage sur le langage de modélisation UML (*Unified Modeling Language*) (OMG, 2003). Les justifications sont multiples (Laforcade, 2004) : valeurs ajoutées des notations pour la communication et la compréhension (Costaglia et al., 2002), (Ferrucci et al., 2002) ; usage d'un standard ayant déjà fait ses preuves dans d'autres domaines industrialisés que la formation à distance, aspect « passerelle » du formalisme UML vers d'autres langages, etc. Nous verrons également dans une prochaine section que l'étude des langages de modélisation éducatifs existants préconise et s'appuie de manière implicite sur l'usage d'UML pour la conception de certains aspects d'un scénario pédagogique.

Toutefois, UML est un langage très général de conception de logiciels et les éléments de ce langage ne sauraient être simplement transposés pour d'autres usages sans une réflexion sur la sémantique des éléments de ce langage. Une solution consiste à spécialiser UML pour notre domaine des situations-problèmes coopératives *via* le mécanisme de méta-modélisation des profils UML. Ces profils correspondent au regroupement d'extensions du langage UML (d'un point de vue notation et sémantique) et de règles.

Les résultats de notre travail sont alors présentés ici comme l'élaboration d'un langage basé UML pour la modélisation de situations-problèmes coopératives. Toutefois, ce travail peut et doit être observé comme

un travail exploratoire sur les apports potentiels d'UML en termes d'usages/techniques/outils capitalisables pour la modélisation de situations d'apprentissage en général (et pas seulement limités aux PBL).

La section 2 présente le contexte de nos travaux : le processus de conception de formations à distance et la place des langages et modèles actuels. Ceci nous permet de mieux positionner notre contribution du langage CPM présenté dans la section 3 suivante. La section 4 présente différentes expérimentations autour de notre proposition : des travaux d'outillage du langage CPM, la mise à l'essai du langage sur un cas d'étude concret et la transformation de modèles de CPM vers IMS-LD.

2. Modèles et langages pour le design pédagogique

Afin de pouvoir positionner nos travaux, il nous paraît important de pouvoir faire référence au cadre précis de mise en œuvre de formations à distance pour lequel ces travaux ont été menés (section 2.1). Ceci nous permettra de situer les différentes propositions actuelles de langages/modèles/standards de description pédagogique et de dresser un constat sur lequel nous nous appuierons pour justifier notre travail exploratoire sur l'utilisation du formalisme UML pour la description de situations d'apprentissage (section 2.2).

2.1. Processus de conception et de mise en œuvre d'une formation à distance

Deux processus ou démarches sont généralement identifiés ([Vantroys, 2003](#)), ([IMS, 2003b](#)). Le premier porte le nom de *processus de conception*. Il est itératif et a pour objectif de produire la formation sous la forme d'un ensemble de modules ou *unités d'apprentissages* spécifiques à une plate-forme de formation à distance. Cette production engage l'ensemble de l'équipe pluridisciplinaire de conception composée d'enseignants (représentant également l'ensemble des différents métiers des SHS susceptibles de participer à la conception d'une formation : pédagogues, didacticiens, sociologues, etc.), de fournisseurs de ressources pédagogiques, de développeurs de composants, etc. (voir dans ([Vantroys, 2003](#)) le détail de ces rôles) et plus particulièrement de l'ingénieur pédagogique qui est en quelque sorte l'expert gérant les interactions entre l'ensemble des acteurs tout au long du processus. Ce processus suit six phases principales résumables ainsi :

- Expression initiale des besoins : l'enseignant et son équipe SHS décrivent de manière informelle la tâche.
- Analyse et conception : l'ingénieur pédagogique connaissant les particularités de la plate-forme de destination, en collaboration avec l'équipe précédente, formalise le scénario pédagogique.
- Implémentation : réalisation de l'ensemble des composants techniques et métiers (ressources pédagogiques) nécessaires au module de formation.
- Déploiement : mise en place sur la plateforme de formation à distance.
- Test : vérification du comportement de la plate-forme.

Ainsi, le scénario pédagogique se définit comme « le résultat manipulable de la modélisation d'une situation d'apprentissage » ([Daele et al., 2003](#)) ou plus précisément comme « la description du déroulement d'une situation d'apprentissage en termes de rôles, d'activités et d'environnement nécessaire à sa mise en œuvre, mais aussi en termes de connaissances manipulées » ([Pernin, 2003](#)). Nous considérons alors ce scénario comme l'un des éléments constitutifs des unités d'apprentissage et donc comme l'un des résultats de la conception de situation d'apprentissage. Sur la base du détail des phases précédentes, nous considérons dans nos travaux ce scénario pédagogique comme indépendant des plates-formes qui accueilleront la formation. Nous percevons donc plus précisément le scénario comme le résultat de la

phase de conception (évoluant bien sûr au gré des itérations) préliminaire aux phases d'implémentation/déploiement/test qui, elles, sont dépendantes d'une plate-forme spécifique.

Le second processus dit d'utilisation concerne les différentes phases relatives à l'exécution du module de formation :

- Instanciation : détermination des différentes personnes qui prendront part à la formation ainsi que de leurs rôles ; constitution des groupes d'apprenants.
- Exécution de la formation sur la plate-forme.

2.2. Manques actuels et besoins

Nous avons étudié dans (Laforcade, 2004) différents modèles et langages de modélisation pour la conception de formations : langage à base de méta-données (LOM, 2002), ontologies éducatives (travaux de l'équipe de Mizoguchi (Mizoguchi et Bourdeau, 2004), (Inaba et al., 2004)), les langages de modélisation pédagogique – EML (Educational Modeling Language) (Koper, 2002), (Rawlings et al., 2002), (IMS, 2003b) et les notations graphiques utilisées dans l'ingénierie pédagogique (notation MOT (Paquette, 1999) de la méthode MISA (Paquette et al., 1997)).

Cette étude nous a conduit à remarquer que :

- ces langages interviennent généralement tardivement dans la conception (on parle alors de conception avancée ou conception détaillée) et se basent plutôt sur les résultats d'analyse produits par les équipes pluridisciplinaires de conception ;
- ils cherchent à organiser et à structurer les contenus (ressources ou objets d'apprentissage) et les conteneurs (scénario pédagogique, unité d'apprentissage) dans le but de répondre à des besoins de réutilisation, assemblage, interopérabilité, etc. ;
- ils font l'objet de standards et de normes (LOM, 2002), (IMS, 2003b) ;
- ce sont pour la plupart des langages formels (au sens usuel en informatique, c'est-à-dire basés sur des grammaires formelles) dont les modèles produits sont concrètement spécifiés sous des formes dérivées du langage XML afin d'être facilement interprétables par la machine ; les modèles produits n'adressent donc pas les « humains » et ne cherchent donc pas à aider l'équipe pluridisciplinaire dans son travail de conception ;
- la notation MOT accompagnant la méthode d'ingénierie pédagogique MISA permet de décrire des unités d'apprentissage proches de celles des EML par le biais d'une approche issue de l'ingénierie des connaissances ; la notation présente les valeurs ajoutées d'un langage graphique mais se limite à un seul type de diagramme manipulant des primitives graphiques de type « boîtes » et « associations ». Actuellement, la version « MOT+ » de cette notation permet de décrire graphiquement des scénarios IMS-LD en manipulant les concepts définis dans le niveau A de la spécification (*method, play, act, role, role-part, learning-activity, support-activity, ...*).

Sur la base des différentes étapes du processus de conception décrit en section 2.1, nous constatons alors que :

- pour l'étape d'expression initiale des besoins : les modèles sont souvent informels et en langage naturel afin de garantir un dialogue entre les différents intervenants (enseignants, pédagogues, ou encore didacticiens) ;
- pour l'étape d'analyse/conception : les modèles produits sont moins abstraits ; l'utilisation du langage UML pour l'analyse est préconisée par des spécifications comme

IMS-LD (il s'agit plus précisément des diagrammes d'activités utilisés pour modéliser le découpage du scénario en actes et activités) (IMS, 2003a) ;

- pour l'étape de conception avancée : les différents langages que nous avons étudiés (méta-données, ontologies et EML) interviennent à cette étape ;
- l'approche IC pour le design pédagogique de MISA/MOT se positionne sur l'ensemble du processus de conception.

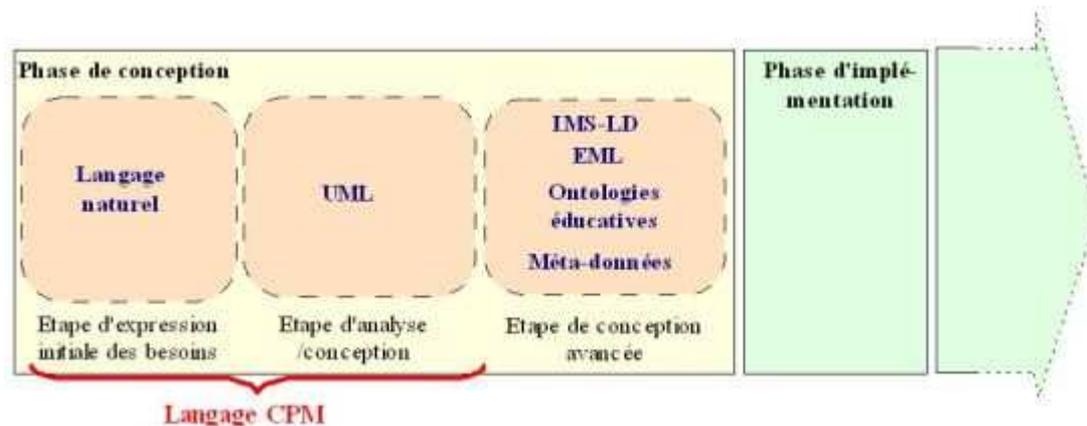


Figure 1 : positionnement des langages existants utilisés pour la conception de situations d'apprentissage et positionnement du langage CPM

Vis-à-vis du constat précédent et du positionnement actuel du formalisme d'UML dans la conception de situations d'apprentissage à distance, nous avons alors positionné notre langage basé sur UML en amont des langages plus formels existants.

En comparaison avec MISA/MOT, notre approche est basée sur UML et donc l'approche objet issue du domaine du génie logiciel tandis que MISA/MOT s'appuie sur le domaine de l'ingénierie des connaissances et sur une approche de représentation non-objet des concepts manipulés. De plus, l'aspect méthode est centrale dans MISA (la notation est secondaire) alors que nos travaux se focalisent sur l'élaboration d'un langage graphique basé UML (la notation) et non sur l'aspect méthode accompagnant ce langage (secondaire).

3. Le langage CPM

Le langage CPM (*Cooperative Pbl Metamodel*) est une proposition de langage basé sur UML pour la modélisation amont de situations-problèmes coopératives. Dans cette section nous présentons comment nous avons élaboré ce langage.

3.1. Vue générale

CPM est une extension du langage UML conçu par la technique de méta-modélisation des profils UML. Comme le langage UML, CPM est alors composé de :

- une syntaxe abstraite : représentée par le méta-modèle CPM qui définit les concepts et leurs relations ;
- une syntaxe concrète : représentée par le profil CPM qui définit la notation des concepts et de leurs relations ainsi que leur utilisation dans les diagrammes UML ;
- une sémantique : celle-ci est définie au niveau de la terminologie dans le méta-modèle (sous la forme de contraintes OCL et de règles en langage naturel) comme au niveau de la

notation (sous la forme de propositions d'usage des diagrammes).

Nous avons choisi la démarche visant à définir un méta-modèle préliminaire à l'élaboration du profil afin de clairement séparer la terminologie de la notation.

3.2. Le méta-modèle CPM : la syntaxe abstraite

La difficulté de notre langage est de proposer une syntaxe répondant à des besoins terminologiques variables étant donné les positionnements différents des modèles à élaborer. En effet, les modèles pour l'étape d'expression initiale des besoins sont très proches du modèle pédagogique théorique des PBL tandis que les modèles de conception plus avancée correspondent davantage à des modèles de scénario pédagogique comme ceux de la spécification IMS-LD. La terminologie employée diffère donc selon l'étape associée au modèle ; il en sera de même pour la notation utilisée dans les modèles.

Nous avons étudié tout d'abord l'apprentissage par situations-problèmes coopératives afin d'extraire les principaux concepts nécessaires à leur mise en œuvre pédagogique dans les modèles très amont (sujet, obstacle, rôle authentique, etc.). Nous avons ensuite étudié différents modèles d'informations provenant de divers domaines (modèle conceptuel du support à l'activité de DARE ([Bourguin, 2000](#)), SIMULIGNE ([Mbala, 2003](#)), modèle coopératif ([Guareis et al., 2000](#)), le méta-modèle *Software Process Engineering Metamodel* ([OMG, 2002](#)), et bien sûr le modèle conceptuel de la spécification d'IMS-LD ([IMS, 2003b](#)) afin de nous aider à élaborer le méta-modèle CPM ou à valider par analogie les concepts et les relations que nous proposons dans CPM.

Nous retenons de l'analyse de ces différents modèles trois caractéristiques essentielles pour notre méta-modèle CPM :

- proposer des éléments pour la **structuration hiérarchique** des activités d'une PBL coopérative ;
- proposer des éléments pour décrire l'objet de la PBL, c'est-à-dire les objectifs pédagogiques mais également tous les autres **aspects pédagogiques** des PBL (obstacles, ressources, contraintes, pré-requis, connaissances, etc.) ;
- proposer des éléments permettant de décrire les **aspects sociaux** des PBL (rôles, activités collaboratives, restrictions, droits, différents modes de coopération, etc.).

Le méta-modèle CPM repose sur deux principaux paquetages : le paquetage de fondation, qui représente un sous-ensemble du méta-modèle d'UML (ce sous-ensemble est en réalité le *Core* du méta-modèle d'UML), et le paquetage des extensions qui regroupe tous les concepts, relations et contraintes de notre langage (voir [Figure 2](#)). Cette approche de construction (similaire à celle du méta-modèle SPEM défini par l'[OMG, 2002](#)), a pour avantage de rendre explicite les relations existant entre les éléments définis par l'OMG dans le MOF^[1] (équivalent au Core d'UML) avec les nouveaux éléments spécifiques à CPM.

Le paquetage des extensions est décomposé à son tour en quatre sous-paquetages interdépendants afin de faciliter la présentation de la terminologie : paquetage des éléments de base, paquetage pédagogique, paquetage structurel, paquetage social (en référence aux trois besoins en vues/aspects identifiés précédemment).

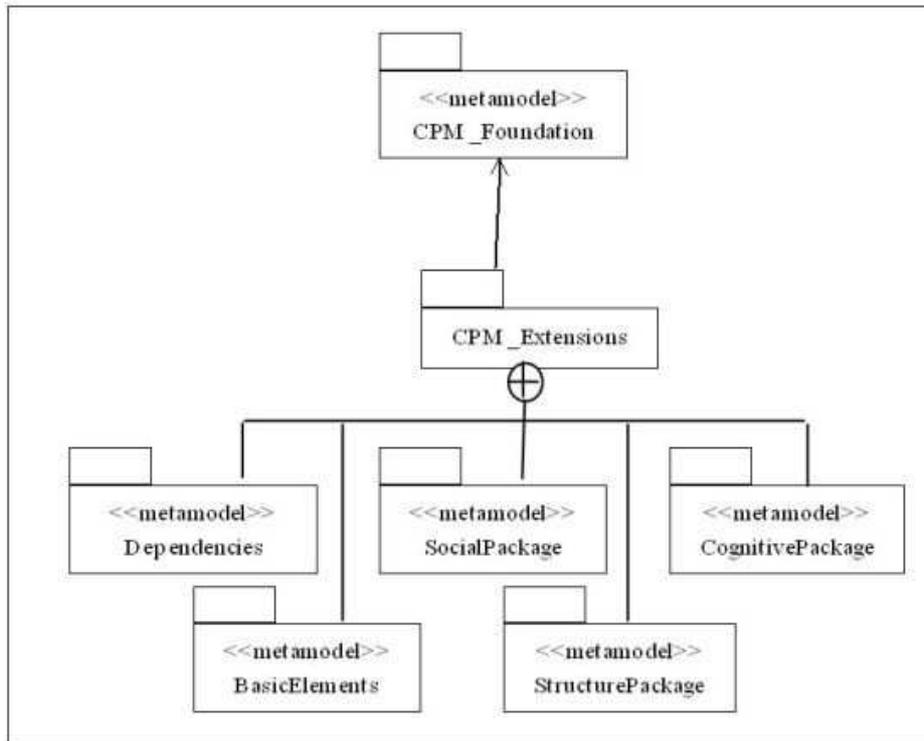


Figure 2 : les paquets formant le méta-modèle CPM

La sémantique des différents concepts et relations proposés est exprimée via des contraintes OCL, lorsque cela est possible, sinon en langage naturel ((Laforcade, 2004) – p156-177).

A titre d’illustration, la Figure 3 présente deux extraits : tout d’abord à gauche un extrait du paquetage structurel et à droite un extrait du paquetage social. Les concepts grisés font référence à des éléments définis dans le paquetage *CPM_Foundation*.. Volontairement, les concepts présentés dans cette figure sont assez généraux afin de correspondre à tout type de situation d’apprentissage.

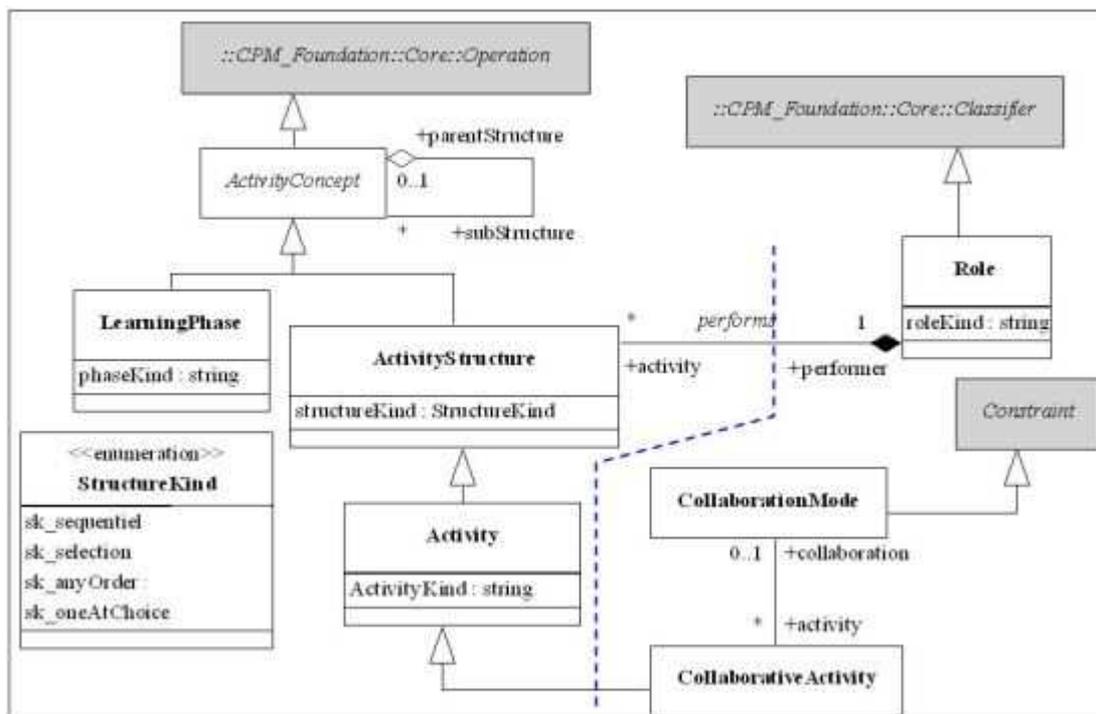


Figure 3 : extraits des paquetages structurels et sociaux du méta-modèle CPM

ActivityConcept représente le concept abstrait d'activité et permet de définir la relation de décomposition hiérarchique pour tous les types d'activités. *LearningPhase* est une spécialisation d'*ActivityConcept* qui permet de définir la décomposition externe du scénario de la situation d'apprentissage ; c'est-à-dire le découpage en séquences du scénario indépendamment des activités jouées par des rôles particuliers. En comparaison avec la spécification IMS-LD, *LearningPhase* peut correspondre au concept d'acte. Toutefois, alors que pour IMS-LD il n'y a qu'un seul niveau de décomposition du scénario, *LearningPhase* permet de définir autant de niveaux que nécessaire ; il hérite en effet de l'association *subStructure* d'*ActivityConcept* et le méta-attribut *phaseKind* permet d'indiquer le type de la phase. Par exemple, le scénario d'une situation-problème pourra être décomposé en actes, puis chaque acte en scènes, et finalement chaque scène contiendra les activités. *ActivityStructure* et *Activity* sont également des spécialisations d'*ActivityConcept* et représentent respectivement un ensemble d'activités ou une activité précise réalisée par un rôle particulier. Les structures d'activités peuvent être de différents types (précisé via le méta-attribut *structureKind*). Un *Role* permet de spécifier les rôles situés joués sur la globalité de la situation-problème et liés à la situation authentique qu'elle met en scène ; le type du rôle (apprenant, tuteur, etc.) est spécifié via le méta-attribut *roleKind*. Une *CollaborativeActivity* est une spécialisation d'*Activity* pour désigner les activités collaboratives de la situation-problème. Ces activités, bien que collaboratives, sont toutefois jouées par un seul rôle (à cause de la relation *performs*). Afin de préciser ce lien de collaboration entre plusieurs activités, celles-ci devront être associées à un même *CollaborationMode*. A titre d'exemple, si une activité collaborative AC implique conceptuellement la participation de deux apprenants A1 et A2 alors il faudra définir selon la sémantique de notre méta-modèle deux concepts d'activités collaboratives AC1 et AC2 réalisées respectivement par A1 et A2 ; AC1 et AC2 devront alors être également reliées par une même contrainte *CollaborationMode*

La coopération n'apparaît pas explicitement sous la forme de concept mais sera décrite au niveau des modèles produits via le partage des rôles, le partage des ressources, la structuration des activités en activités individuelles puis collectives et l'échange de ressources.

Il est possible de distinguer ou classer les 35 différents concepts du langage CPM selon leurs usages dans les différents modèles produits dans la phase de conception. Tout d'abord, certains concepts représentent le vocabulaire nécessaire à l'étape d'expression initiale des besoins (leurs analogies avec la terminologie employée pour la définition de PBL est directe : **Subject**, **Objective**, **Resource**, **Obstacle**, **SuccessCriterion**, etc.). D'autres concepts, à l'opposé, conviennent davantage à un usage de conception détaillée pour lequel le vocabulaire doit pouvoir correspondre avec la terminologie d'autres langages existants comme IMS-LD (voir [Table 1](#)).

Un exemple de différence important entre CPM et IMS-LD est, par exemple, celui du *role-part* d'IMS-LD (qui permet de relier les rôles avec les activités qu'ils jouent dans le contexte d'un *act* donné) qui n'a pas d'équivalence dans le langage CPM puisque toute activité ne peut être réalisée que par un unique rôle (voir l'association entre **Role** et **ActivityStructure** dans l'extrait du méta-modèle CPM dans la [Figure 3](#)).

| Terminologie d'IMS-LD | Terminologie dans CPM |
|---|---------------------------------|
| method | LearningProcess |
| play, act | LearningPhase |
| activity, learning activity, support activity | Activity |
| activity-structure | ActivityStructure |
| role, learner, staff | Role |
| role-part | ∅ |
| environment, learning object, service | Resource |
| Learning objective | Objective, Postcondition |
| prerequisite | StaticPBLElement, Precondition |
| Condition et property | PBLConstraint, StaticPBLElement |
| notification | Resource, PBLConstraint |

Table 1 : correspondance entre la terminologie d'IMS-LD et celle de CPM

3.2. Le profil CPM : la syntaxe concrète

Un profil UML est un ensemble cohérent d'extensions du méta-modèle d'UML dans le but de le spécialiser pour un domaine spécifique (OMG, 1999). Le profil CPM est alors une spécialisation d'UML dédiée à la modélisation de situations-problèmes coopératives.

Comme tout profil, le profil CPM est composé entre autres de :

- stéréotype : défini pour une méta-classe spécifique du méta-modèle d'UML, le stéréotype crée une nouvelle méta-classe basée sur la méta-classe UML existante. Il fournit ainsi un moyen de classer les instances de cette méta-classe de base et peut également aussi spécifier des contraintes additionnelles ou des valeurs marquées requises.
- définition de valeurs marquées : définie pour une métaclasse spécifique du méta-modèle d'UML comme associée à un stéréotype, la définition d'une valeur marquée agit comme un attribut d'une méta-classe UML, permettant ainsi l'attachement arbitraire d'informations à une instance.
- notation : la notation d'UML peut être personnalisée *via* la définition d'icônes associés aux stéréotypes.

La définition de stéréotypes est dirigée également par les diagrammes UML que l'on veut privilégier au niveau de la notation du profil. Ainsi, sur la base d'une analyse préliminaire des usages traditionnels des diagrammes UML en génie logiciel comme en génie éducatif, nous avons retenu l'utilisation des diagrammes de classe (pour modéliser les différents aspects statiques des modèles de design pédagogique quel que soit le niveau dans le processus de conception), de cas d'utilisation (pour décrire en phase d'expression initiale les besoins en termes d'activités de haut niveau d'abstraction ainsi que les rôles les réalisant), d'états (pour préciser la dynamique des différents concepts manipulés dans une situation-

problème et évoluant entre différents états : ressources, représentations mentales des apprenants, etc.) et d'activités (pour décrire l'organisation des activités, la structure des scénarios pédagogiques, la coopération entre rôles via la gestion des flux d'activités individuelles/collaboratives et la gestion des flux de ressources).

D'autres diagrammes sont utilisables (diagrammes d'objets, de séquence et de collaboration) mais aucun de leurs éléments de modélisation UML n'ont été étendus (pas de correspondance ou sémantique définie directement en relation avec les concepts CPM).

La [Table 2](#) présente un extrait des définitions des stéréotypes pour le profil CPM. On retrouve dans cet extrait trois des concepts présents dans l'extrait du méta-modèle CPM ([Figure 3](#)). A chaque stéréotype correspondent des méta-classes UML de référence qui vont être « étendues » par ce nouveau concept. Une première correspondance entre le méta-modèle CPM et le profil CPM suit la règle suivante : à tout concept défini dans le paquetage CPM_Extensions correspond un stéréotype défini sur la base de la méta-classe UML dont il hérite directement ou indirectement dans le méta-modèle CPM (méta-classes de référence en gras dans la figure ci-dessous). D'autres classes du sous-ensemble du méta-modèle d'UML sont proposées toutefois comme alternatives à la classe de base afin de pouvoir utiliser les concepts du langage CPM dans des diagrammes UML différents et ainsi augmenter les possibilités de description visuelle des modèles pour la conception de situations-problèmes coopératives. Ainsi, par exemple, **LearningPhase** est un stéréotype défini pour la méta-classe UML de référence *Operation* (les opérations sont utilisées dans les diagrammes de classe pour représenter des comportements ou méthodes d'un *classifier*) mais également pour les méta-classes *ActionState* et *SubActivityState* (pour être utilisé dans les diagrammes d'activités), *UseCase* (pour les diagrammes de cas d'utilisation) et *Classifier* (pour une nouvelle utilisation dans les diagrammes de classe).

Il est également possible d'adjoindre des icônes à certains stéréotypes afin de simplifier leur interprétation sémantique dans les diagrammes où ils sont utilisés.

| Stéréotype | Méta-classe | Icône |
|---------------|--|---|
| LearningPhase | Core::Operation ActivityGraphs::ActionState ActivityGraphs::SubactivityState UseCases::UseCase Core::Classifier |  |
| Activity | Core::Operation ActivityGraphs::ActionState ActivityGraphs::SubactivityState UseCases::UseCase Core::Classifier |  |
| Role | UseCases::Actor ActivityGraphs::Partition |  |

Table 2 : extraits de la table de définition des stéréotypes du profil CPM

De manière similaire aux stéréotypes, la définition des valeurs marquées fait l'objet d'une correspondance directe avec le méta-modèle CPM : tout méta-attribut rattaché à un concept du paquetage CPM_Extensions fait l'objet d'une définition de valeur marquée rattachée au stéréotype de même nom et prenant un paramètre de type *String*. La table suivante est également un sous-extrait de la table des 15

définitions de valeurs marquées du profil CPM.

| Définition valeur marquée | Type valeur | Sur stéréotype |
|---------------------------|-------------|----------------|
| phaseKind | String | LearningPhase |
| roleKind | String | Role |
| activityKind | String | Activity |

Table 3 : extraits de la table des définitions de valeurs marquées du profil CPM

Toutes les associations entre les classes du méta-modèle CPM doivent également trouver une correspondance avec les différentes possibilités de relation proposées par le méta-modèle d'UML ((Laforcade, 2004) p.186). Par exemple, l'association entre **Role** et **ActivityStructure** du méta-modèle CPM s'exprime via l'association entre *Classifier* et *Feature* définie dans le méta-modèle d'UML pour la sémantique des diagrammes de classe.

4. Expérimentations

Nous avons entrepris trois travaux d'expérimentation de notre proposition. La première expérimentation vise à fournir un environnement auteur, sous la forme d'un outil logiciel, supportant le langage CPM et permettant l'élaboration de modèles CPM (section 4.1). La deuxième expérimentation concerne une mise à l'essai pratique de l'élaboration de modèles CPM dans le cadre de la situation-problème SMASH (section 4.2). Cette mise à l'essai s'appuie sur l'environnement auteur issu de la première expérimentation et a pour principal objectif d'explorer les différents usages possibles des modèles CPM quant à la description/analyse/conception de situations-problèmes. Finalement, la troisième expérimentation cherche à étendre les usages des modèles CPM en proposant une génération automatique de modèles IMS-LD sur la base des informations contenues dans les modèles CPM (section 4.3).

4.1. Prototype d'environnement auteur

Nous avons « outillé » le profil CPM grâce à un Atelier de Génie Logiciel UML existant : *Objecteering* (le seul outil supportant toutes les propriétés et caractéristiques des profils UML).

Ceci a permis de répondre aux différents objectifs suivants :

- implémenter le langage CPM : via l'outil *Objecteering Profile Builder* ;
- vérifier l'élaboration de modèles conformes au langage CPM : via l'outil *Objecteering Modeler* ;
- prototyper un environnement-auteur mieux adapté à des ingénieurs pédagogiques connaissant les bases de la modélisation UML mais non-experts (voir [Figure 4](#)) ; grâce au langage propriétaire J (langage objet interprété dédié à la manipulation des éléments de modélisation et du méta-modèle d'UML) nous avons pu développer un ensemble de fonctionnalités personnalisant et adaptant l'interface de l'outil pour faciliter la création, le suivi et la manipulation de modèles CPM : recherche guidée d'éléments, ajout d'éléments dirigée par des boîtes de dialogue, etc.
- expérimenter la plus-value de disposer de modèles graphiques et interprétables par la machine : vérification de la cohérence entre la sémantique du langage CPM et la notation, transformation de modèles (projection de modèle CPM vers IMS-LD – présentée en section 4.3).

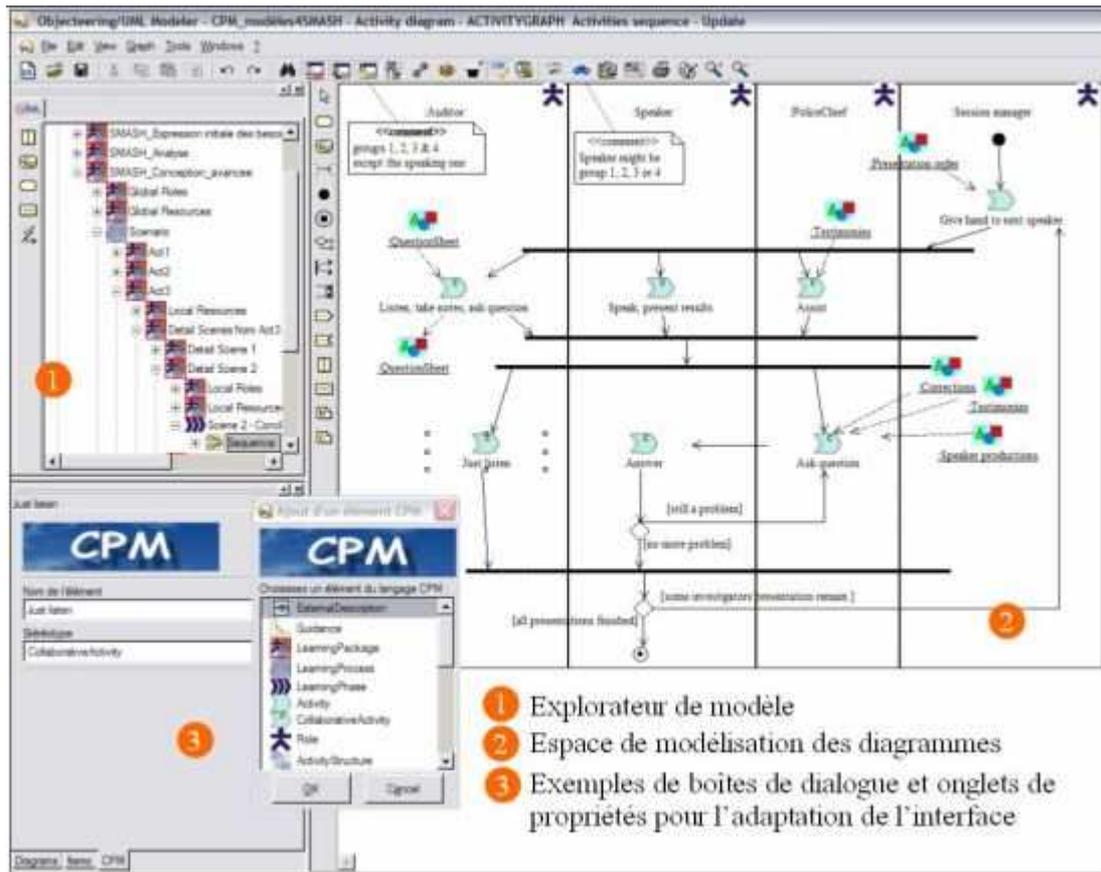


Figure 4 : capture d'écran du prototype d'environnement-auteur pour le langage CPL

4.2. Le cas d'étude SMASH

SMASH (scénario à l'origine de (Actis, 2005), puis retravaillé dans (Oudot, 2003) et finalisé dans (Laforcade, 2004)) est une situation-problème qui s'adresse aux enfants de 8 à 10 ans. Ils vont devoir jouer le rôle d'inspecteurs de police enquêtant sur la reconstitution d'un accident de vélo. Par le biais de témoignages différents et convenablement adaptés, les différents groupes d'enquêteurs doivent arriver à des conclusions différentes quant au responsable de l'accident. Cet obstacle ne sera surmonté que lorsqu'ils auront échangé/discuté leurs points de vues (conflit socio-cognitif) et acquis suffisamment de connaissances sur le code de la route, le comportement sur la route, etc., autant d'éléments correspondant aux véritables objectifs pédagogiques de la tâche.

Nous avons mis à l'essai le langage CPM sur le cas d'étude SMASH afin de réaliser un ensemble de modèles CPM pour SMASH (le lecteur pourra se reporter à (Laforcade, 2004) - section 9.2 - pour une illustration détaillée de ces modèles). Ce travail a été réalisé en nous appuyant sur des documents traditionnels (tableaux, diagrammes de Gantt, textes narratifs) d'analyse et de conception de SMASH élaborés par des professeurs des écoles avec lesquels nous avons collaboré.

Au final, nous avons pu modéliser une grande variété de diagrammes, pour des niveaux d'abstraction variables, pour des vues différentes (sociales, structurelles, pédagogiques, etc.) et des usages différents (analyse de la tâche en amont ou encore spécification précise du scénario pédagogique en aval) : voir Table 4 ci-après. Ceci a permis d'illustrer et d'explorer la richesse d'expression héritée d'UML, ce qui valide notre principal objectif : fournir un langage de modélisation visuel riche permettant de décrire un large champ de modèles pour la conception de PBL. Toutefois, nous avons également relevé des problèmes liés au rattachement à UML ainsi qu'au manque actuel d'une méthode adaptée à notre langage : il est parfois difficile de savoir quel diagramme privilégier pour décrire tel aspect d'une situation-problème, de même il est difficile de garantir la cohérence sémantique inter-diagrammes entre

les éléments de modélisation d'un même modèle.

Nous pensons que la proposition d'une méthode adaptée au langage CPM et outillée pourra surmonter ces obstacles et permettra alors de mieux guider les concepteurs et l'ingénieur pédagogique dans l'utilisation du langage. Mais pour cela, une étape importante de validation devra concerner, non plus le langage CPM, mais les modèles construits avec celui-ci. Ce travail expérimental de validation doit être mené par des ingénieurs pédagogiques connaissant les bases d'UML (le public-cible de notre proposition) sur d'autres cas d'études. Ces résultats permettront de mieux cadrer les usages du langage et donc faciliteront la mise en place d'une méthodologie adaptée.

| Usage | Expression initiale des besoins | Analyse | Conception |
|--------------------------------|--|---|---|
| Diagramme de cas d'utilisation | <ul style="list-style-type: none"> - découpage en activités - identification des rôles | | |
| Diagramme d'états | | | - description des différents états d'une ressource |
| Diagramme de classes | <ul style="list-style-type: none"> - définition des objectifs - définition de la tâche préliminaire - définition de l'obstacle - définition des critères de succès - définition des rôles | <ul style="list-style-type: none"> - objectifs détaillés - représentation des connaissances sur la situation authentique - analyse externe des activités | <ul style="list-style-type: none"> - spécification détaillée des rôles globaux et locaux - spécification détaillée des ressources |
| Diagramme d'activités | <ul style="list-style-type: none"> - définition de la tâche globale - définition des critères de succès selon les activités | <ul style="list-style-type: none"> - fiches de tâches individuelles et collectives - analyse interne des activités - détail des structures d'activités (séquence, choix, etc.) | <ul style="list-style-type: none"> - spécification du scénario en niveaux (actes/scènes/etc.) ; puis en activités par niveau - spécification des activités collaboratives et des actions de coordination (échange de ressources, synchronisation, etc.) |

Table 4 : les différents usages possibles des diagrammes UML selon le positionnement du modèle dans la phase de conception

4.3. Transformation de modèles : lien avec les standards

Nous avons expérimenté l'application d'une technique exploitant la possibilité d'interpréter par la machine des modèles UML conçus au sein d'un Atelier de Génie Logiciel UML (Lafordade, 2005). Cette technique a pour but de transformer un modèle UML vers un autre type de modèle. La transformation est réalisable grâce aux éléments spécifiques définis dans les profils UML : les stéréotypes et les définitions de valeurs marquées. Ces éléments informent et dirigent la collecte d'informations contenues dans le modèle UML. Nous avons appliqué cette technique dans le cadre de la spécification de scénarios

d'apprentissage (Figure 5) sur la base des exemples d'utilisation d'UML proposés par le consortium IMS dans (IMS, 2003a).

Il est ainsi actuellement possible de transformer des diagrammes d'activités modélisant des scénarios d'apprentissage à l'aide du langage CPM vers des modèles XML conformes au standard de la spécification actuelle d'IMS-LD.

Nous donnons dans la Table 5 ci-après les correspondances précises entre les concepts IMS-LD apparaissant dans les diagrammes d'activités UML (concepts du niveau A utilisés dans le document (IMS, 2003a)) et les concepts CPM.

| Concept IMS-LD | Concept CPM | | | |
|--------------------|------------------|-----------------------------------|------------------------------|---------------------|
| | Méta-classe UML | Stéréotype | Définition de valeur marquée | Valeur du paramètre |
| learner | Partition | Role | roleKind | learner |
| staff | Partition | Role | roleKind | staff |
| support-activity | ActionState | Activity ou CollaborativeActivity | | |
| learning-activity | ActionState | Activity ou CollaborativeActivity | | |
| activity-structure | SubactivityState | ActivityStructure | structureKind | |

Table 5 : correspondances IMS-LD ↔ CPM

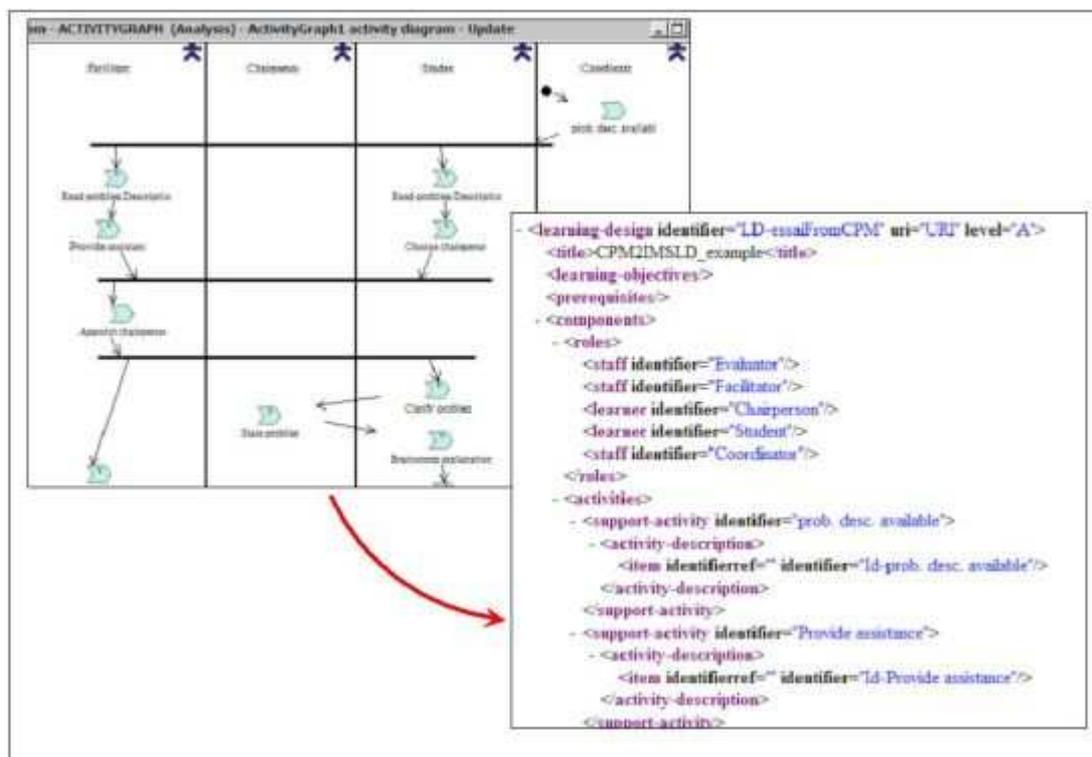
Les deux concepts d'activité d'IMS-LD (*support-activity* et *learning-activity*) sont déduits selon le type du rôle attribué à la partition dans laquelle se trouve l'activité concernée exprimée avec CPM : une activité positionnée dans la partition des activités d'un rôle d'apprenant est une *learning-activity*.

Les concepts de *method* et *play* ne sont pas traités car un diagramme d'activités ne permet de spécifier que les actes pour un *play* donné. Le découpage en actes nécessite pour sa part la mise en œuvre d'un algorithme approprié. Celui-ci est construit sur la base de remarques déduites de l'observation et de l'analyse des différences entre les diagrammes d'activités et des modèles XML correspondants proposés dans (IMS, 2003a) :

- l'utilisation des barres de synchronisation (*fork* et *join*) permet de modéliser le parallélisme des activités implicitement mentionné dans la narration ;
- les flèches traversant les couloirs expriment un besoin de précédence entre les deux activités qu'elles relient ;
- les cadres entourant des activités successives réalisées par le même rôle font référence aux *activity-structure* spécifiées dans le document XML correspondant.

Ce travail de transformation de modèles doit être approfondi afin d'élargir le nombre de concepts d'IMS-LD couverts par cette transformation automatique (nous travaillons actuellement sur la description des

ressources et environnements du niveau A et envisageons également des possibilités quant aux concepts de propriétés et de notification des niveaux B et C). Dans ce but, d'autres diagrammes CPM devront servir comme source d'informations (comme par exemple des diagrammes de classes décrivant les objectifs et les pré-requis de la situation d'apprentissage.



En revanche, ces travaux d'expérimentation montrent également, vis-à-vis de notre propre langage CPM, que de nombreuses informations capturées dans les modèles CPM n'ont pas de correspondance avec IMS-LD ; ceci est dû principalement au positionnement du langage CPM plus amont dans le processus de conception ainsi qu'au choix des situations-problèmes comme type de situation d'apprentissage étudié.

5. Conclusion

Nous commençons par dresser un bilan/discussion de notre contribution du langage CPM. Ensuite nous présentons des perspectives ouvertes par notre travail exploratoire à court terme comme sur le long terme.

5.1. Bilan/discussions

Nous avons proposé un langage facilitant l'élaboration de modèles pour la conception de situations-problèmes (PBL) sur des plates-formes de formation à distance. L'étude des différents langages correspondant en partie à cet objectif a permis de mettre en évidence i) le manque actuel de modèle et de langage pour l'étape d'expression initiale des besoins, ainsi que ii) la nécessité d'utiliser UML pour décrire des modèles abstraits (étape d'analyse) servant de base à la spécification formelle réalisée par les EML comme IMS-LD (étape de conception détaillée). Ce constat nous a alors amené à positionner notre proposition comme un langage de modélisation graphique spécialisant UML pour la conception de PBL en amont des langages de type EML : il couvre alors la phase de conception pour les étapes d'expression initiale des besoins, d'analyse et de conception (la [Table 6](#) établit une comparaison synthétique entre CPM et IMS-LD).

Le langage CPM proposé permet alors de décrire les différents modèles de la PBL pour les phases amont de conception sur deux niveaux :

- horizontalement : les modèles sont différents (en termes d'objectifs, d'abstraction, de contenus et d'usages) selon les étapes dans la phase de conception (expression initiale des besoins, analyse, conception avancée) ;
- verticalement : les modèles capturent différents points de vue ou perspectives sur la même situation d'apprentissage (pédagogique, didactique, social).

| | CPM | IMS-LD |
|-------------------------------|--|--------------------------------------|
| Langage | | |
| Type | Semi-formel (graphique) | Formel |
| Définition de la Terminologie | Méta-modèle CPM | Modèle d'information IMS-LD |
| Définition de la notation | Profil UML CPM | |
| Public cible | Ingénieur pédagogique spécialiste d'UML | Ingénieur pédagogique formé à IMS-LD |
| Modèles/instances | | |
| Phases concernées | Expression initiale des besoins, analyse et conception | Conception avancée |
| Type | Modèles UML | Modèles XML |
| Public cible | Equipe pluridisciplinaire chargée de la conception | Machine |
| Outillage | | |
| | Oui (Objecteering) | Divers prototypes |

Table 6 : comparaison synthétique entre les langages CPM et IMS-LD

Le langage CPM propose une syntaxe (terminologie et notation) et une sémantique adaptées pour produire les différents modèles utiles lors de la phase de conception. Un prototypage d'environnement-auteur pour le langage CPM a été proposé sur la base d'une implantation du profil CPM dans un AGL UML existant et d'une personnalisation/adaptation de l'interface de l'AGL pour faciliter la création et le suivi de modèles CPM. Ce type d'environnement concerne plus particulièrement l'ingénieur pédagogique pour lequel un niveau de connaissance suffisant d'UML est nécessaire.

Nous avons effectué différents travaux dans le but d'identifier et de décrire les apports de la (méta)-modélisation UML quant à la conception de situations d'apprentissage. Nous sommes arrivés à la

conclusion que l'utilisation des profils UML permet une conceptualisation explicite des différentes connaissances sur la situation d'apprentissage en cours de spécification.

Cette conceptualisation explicite est tout d'abord visuelle/graphique ce qui permet une meilleure capitalisation des connaissances des différents intervenants dans la conception de la formation mais également améliore leur communication. Enfin, cette conceptualisation explicite l'est également pour la machine lorsque les modèles sont élaborés, manipulés et exploités par des AGL UML. Le prototype que nous avons expérimenté laisse entrevoir la possibilité de créer des environnements auteurs spécifiques exploitant les profils UML au travers d'interfaces adaptées et de fonctionnalités spécifiques permettant de viser un public-cible plus large que les seuls ingénieurs pédagogiques experts d'UML.

En effet, le langage s'adresse à l'ingénieur pédagogique selon l'hypothèse qu'il connaît les bases de la modélisation UML et qu'un outil adapté l'assistera dans sa réalisation de modèles CPM. En revanche les modèles CPM s'adressent à l'ensemble de l'équipe pluridisciplinaire de conception pour décrire, spécifier et documenter la PBL qu'ils conçoivent. Le caractère visuel des modèles produits permet d'abstraire la complexité de la PBL mais constitue également une représentation adéquate pour favoriser le dialogue, la compréhension et l'implication des différents intervenants (nous pensons qu'une connaissance approfondie d'UML n'est pas nécessaire pour comprendre un modèle CPM élaboré). Nous nuancerons donc la remarque de (Faerber, 2004) affirmant que « [...] pour analyser, manipuler, justifier et formaliser la démarche enseignante on se heurte à un obstacle important : le savoir de l'enseignant est pour une grande part heuristique, et cela est difficilement compatible avec la rigueur algorithmique d'UML ». Nous pensons qu'UML et surtout des langages dérivés d'UML comme CPM ont leur place dans la description/modélisation/formalisation de situations d'apprentissage, qui plus est dans le contexte déjà fortement industrialisé de la conception de formations à distance à très grande échelle (comme à l'(OUNL, 2005) ou à la (Teluq, 2005)) pour lesquels l'équipe de conception n'est pas réduite qu'à un unique enseignant.

Nous avons également présenté un travail expérimental de transformation de modèle vers IMS-LD qui permettra de dépasser le contexte de la conception amont de formations à distance et de relier les modèles CPM aux plates-formes de formation à distance. Cette transformation de modèles est prometteuse ; elle permet d'étendre les usages des modèles produits grâce à la « passerelle » qu'est le formalisme UML : génération de code, exportation en XMI puis transformation externe des modèles UML, exécution directe de modèles UML, etc.

5.2. Perspectives

Nous présentons dans cette section différentes perspectives de recherche directement liées à notre proposition du langage CPM (court terme) ou indirectement liées à l'utilisation de la modélisation et de la méta-modélisation UML dans des contextes d'ingénierie des EIAH plus larges.

5.2.1 Expérimentation et méthodologie

Dans une perspective à court terme, le langage CPM devra être expérimenté plus avant sur de nombreux autres cas d'étude et en collaboration avec des spécialistes appartenant au public-cible du langage. En effet, tout langage basé sur UML hérite de sa pseudo indépendance aux différentes méthodes de conception (RUP, 2TUP, etc.), et de par ce fait, hérite également de la grande difficulté de modélisation que nous avons également rencontrée : il existe bien souvent de nombreux choix ou alternatives de représentation pour modéliser une vue de la situation d'apprentissage. La trop grande variété de diagrammes et de vues peut complexifier les modèles produits et ainsi aller à l'encontre de l'objectif souhaité. L'analyse de ces retours d'expérience nous permettra i) de mieux évaluer les capacités du langage à couvrir la phase de conception de situations d'apprentissage, ii) de mieux comparer/critiquer/positionner les différents langages et notations actuels (CPM vs IMS-LD vs MOT, etc.) et iii) d'identifier les besoins non couverts actuellement. Il est également possible que ces

expérimentations révèlent des limitations ou des capacités d'expression insuffisantes pour le langage CPM (c'est-à-dire la notation actuelle d'UML plus les icônes associées aux stéréotypes ajoutés) quant à la description de besoins spécifiques.

Il est également nécessaire de doter le langage CPM de méthodes, techniques et outils adaptés afin qu'il puisse réellement impliquer des concepteurs venant des SHS (comme les enseignants par exemple) dans la conception amont : une méthode d'ingénierie pédagogique spécifique à un langage basé UML tel que CPM est la clé de voûte qui garantira le succès de cette approche de la modélisation de situations d'apprentissage. Il est évident que des premières pistes de recherche consisteront à étudier les méthodes UML (RUP, 2TUP) comme les méthodes existantes d'ingénierie pédagogique (par exemple MISA).

5.2.2 Perspectives à long terme

A plus long terme, nous envisageons de nombreux travaux de recherche liés à l'utilisation directe comme indirecte de modèles pour les situations d'apprentissage élaborés par des techniques de méta-modélisation UML.

Vers une ingénierie dirigée par les modèles

Selon le constat qu'un modèle UML outillé est interprétable par la machine et que les profils permettent de guider les transformations, nous envisageons alors d'automatiser la transformation des modèles CPM d'expression initiale de besoins vers des modèles CPM d'analyse, puis, à leur tour, vers des modèles CPM de conception. Nous aurons ainsi des modèles qui guident et dirigent la conception, c'est-à-dire un contexte d'ingénierie des modèles propice également à la mise en place de méthodes d'ingénierie pédagogique adaptées à notre langage.

Notre travail actuel sur la transformation de modèles CPM laisse envisager la possibilité d'élaborer un profil UML spécifique à IMS-LD mais également la possibilité d'outiller ce profil afin qu'il facilite l'élaboration de modèles XML conformes au standard IMS-LD : un nouvel environnement-auteur basé UML pour IMS-LD est envisageable.

Une perspective supplémentaire vise également à étendre les usages des modèles CPM au niveau de la phase d'exécution de l'apprentissage : ces modèles seraient alors une source d'informations pour la régulation (suivi des activités, évaluation ou encore aide au tutorat) de l'apprentissage *effectif*. Un système adapté manipulant de tels modèles interprétables pourrait alors capter des actions des différents acteurs afin de modifier le modèle-machine en conséquence (rétro-action avec mise à jour du modèle et adaptation possible de l'environnement de formation). Il faut alors étudier les pistes concernant soit les modèles UML exécutables ([Mellor et Balcer, 2002](#)) soit l'exportation des modèles UML en XMI ([OMG, 2003b](#)) qui seront ensuite manipulés, transformés et intégrés dans un serveur d'applications.

Plus généralement, l'approche MDE (Model-Driven Engineering) a de nombreuses applications possibles dans le contexte de l'ingénierie des EIAH ([Renaux et al., 2005](#)), ([Ecole EIAH, 2005](#)).

Modèles pour échanger avec la communauté EIAH

Un usage particulier des travaux issus du Génie Logiciel et plus précisément du MDA (*Model-Driven Architecture*)/MDE concerne l'étude comparative/explicitation des modèles actuels pour les EIAH. Un travail de fond sur la définition de la sémantique de ces modèles permettrait de mieux les comprendre, les interpréter, les comparer et donc de mieux identifier les besoins d'un concepteur d'EIAH et de l'aider à réutiliser des modèles existants lui convenant. Un double travail sur l'explicitation de la sémantique des modèles (lien entre les modèles et leurs méta-modèles) et sur la réutilisation de ces modèles à des fins cognitives comme computationnelles (lien entre les modèles *via* leurs méta-modèles) est nécessaire pour se diriger vers un « tissage » des modèles existants. Ces deux actions de recherche permettront alors a/ de guider la conception d'EIAH et b/ de produire des outils facilitant la coopération entre chercheurs de disciplines différentes (SHS et informatique) en leur masquant la complexité des techniques sous-

jacentes. Les travaux actuels autour du QVT (*Queries-Views-Transformations*) et des DSL (*Domain Specific Language*) représentent des pistes pour nos futures recherches sur cet axe.

BIBLIOGRAPHIE

Références bibliographiques

BOURGUIN G. (2000). Un support informatique à l'activité coopérative fondé sur la Théorie de l'Activité : le projet DARE, Thèse de doctorat de l'Université des sciences et technologies de Lille. N : 2753.

COSTAGLIOLA G., DE LUCIA A., OREFICE S., POLESE G. (2002). A classification framework for the design of visual languages, *Journal of Visual Languages and Computing*, 13: 573-600.

DAELE A., BRASSARD C., ESNAULT L., O'DONOGHUE M., UYTTEBROUCK E., ZEILEGER R. (2003). Wp2. conception, mise en oeuvre, analyse et évaluation de scénarios pédagogique recourant à l'usage des technologies de l'information et de la communication, Rapport recre@sup. <http://tecfa.unige.ch/proj/recreasup/rapport/WP2.pdf>

DESCHENES A.J., BILODEAU H., BOURDAGES L., DIONNE M., GAGNE P., LEBEL C., RADADONATH A. (1996), Constructivisme et formation à distance, *Revue DistanceS*, Vol 1, num 1.

DILLENBOURG, BAKER *et al.* (1995), The Evolution of Research on Collaborative Learning, P. Reimann & H. Spada Ed.

FAERBER (2004). Caractérisation des situations d'apprentissage en groupe, *Revue STICEF*, Volume 11, 2004, ISSN : 1764-7223, mis en ligne le 30/12/2004, <http://sticef.org>

FERRUCCI F., TORTORA G., VITELLO G. (2002), Exploiting Visual Languages In Software Engineering, *Handbook of Software Engineering and Knowledge Engineering*, in Singapore World Scientific Publishing Co. editor.

GUAREIS DE FARIAS C. R., FERREIRA PIRES L., VAN SINDEREN M. (2000). A conceptual model for the development of CSCW systems, *Fifth International Conference on the Design of Cooperative Systems (COOP 2000)*, pages pp. 189_204, Sophia Antipolis, France, 2000.

IMS (2003a), IMS Learning Design Best Practice and Implementation Guide, Technical report, January.

IMS (2003b), IMS Learning Design Version 1.0 Final Specification, Technical report, February.

INABA A., MIZOGUCHI R. (2004). Learning Design Palette: An Ontology-aware Authoring System for Learning Design, *In Proc. of International Conference on Computers in Education (ICCE2004)*, Melbourne, Australia.

KOPER R. (2002), *Educational modeling language: adding instructional design to existing specifications*, Technical report, Educational Expertise Technology Centre, Open University of the Netherlands.

LAFORCADE P., BARBIER F., NODENOT, T., SALLABERRY, C. (2003), Profiling Co-operative Problem-Based Learning Situations, *Proceedings of the 2nd IEEE International Conference on Cognitive Informatics (ICCI'2003)*, London, UK, 18-20 august, IEEE Computer Society Press.

LAFORCADE P., BARBIER, F. (2004), UML-based Modeling of Educational Components for Cooperative Problem-based Learning Situation Design *in Instructional Technologies: Cognitive Aspects of Online Programs*, pages 165-191. Idea group Inc.

LAFORCADE P. (2004), *Modélisation et méta-modélisation UML pour la conception et la mise en*

œuvre de situations-problèmes cooperatives, Ph.D. Thesis, University of Pau et des Pays de l'Adour, http://www.univ-pau.fr/~laforcad/publications/Pierre_Laforcade-These.pdf

LAFORCADE P. (2005), Approche par transformation de modèles pour la conception d'EIAH - Illustration entre les langages CPM et IMS-LD, *conférence sur les Environnements Informatiques pour l'Apprentissage Humain*, LIRMM-Montpellier, 25-27 mai 2005.

LOM (2002), *LOM (Learning Object Metadata) working draft v4.1*, <http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html>

MBALA Hikolo A. (2003), *Analyse, conception, spécification et développement d'un système multiagents pour le soutien des activités en formation à distance*, Thèse de doctorat de l'Université de Franche-Comté. N_977, 2003.

MEIRIEU Ph. (1994), *Apprendre... oui, mais comment ?*, Collection Pédagogies, ESF Edition.

MELLOR S. J., BALCER, M. J. (2002), *Executable UML - A Foundation for Model-Driven Architecture*, Number ISBN : 0201748045. Addison-Wesley Pub Co ; 1st edition May-15-2002.

MIZOGUCHI R., BOURDEAU J. (2000), Using ontological engineering to overcome AIED problems, *International Journal of Artificial Intelligence in Education*, Vol.11(num 2): pp.107_121.

NODENOT T., LAFORCADE P., MARQUESUZAA C., SALLABERRY C. (2003), Knowledge Modeling of Co-operative Learning Situations: Towards a UML profile, *Proceedings of the 11th International Conference on Artificial Intelligence in Education (AIED'2003)*, Sydney, Australia, 20-24 July, International AI-ED Society.

OMG (1999), White Paper on the Profile mechanism v.1.0, Analysis and Design Platform Task Force. Report ad/99-04-07.

OMG (2002), *Software Process Engineering Metamodel (SPEM)*, Technical Report formal/2002-11-14, November 2002.

OMG, (2003), *Unified Modeling Language v1.5*, Report formal/03-03-01.

OMG (2003b), XML Metadata Interchange (XMI), v2.0, Report formal/03-05-02, May 2003.

OUDOT C. (2003), *Modélisation UML d'une situation d'apprentissage coopérative*, rapport de DESS Sciences Cognitives.

PAQUETTE G., CREVIER F., AUBIN C. (1997), Méthode d'ingénierie d'un système d'apprentissage (MISA), *Revue Informations In Cognito*, (8).

PAQUETTE G. (1999), Meta-knowledge Representation for Learning Scenarios Engineering, *In Proceedings of AI-Ed'99, AI and Education, open learning environments*, S. Lajoie et M. Vivet (Eds), IOS Press.

PAQUETTE G. (2004), *Educational Modeling Language: from an Instructional Engineering Perspective*, in an article to be published in a forthcoming handbook.

PERNIN J-P. (2003), *Critères pour une typologie des langages de modélisation pédagogique*. Présentation au gdr³ EIAH, Paris.

RAWLINGS A., VAN ROSMALEN P., KOER R., RODRIGUEZ-ARTACHO M., LEFRERE P. (2002), *Survey of educational modelling languages (EMLs)*, Technical report, September 19st.

RENAUX E., CARON P-A., LE PALLEC X. (2005), Learning Management System component-based design: a model driven approach, *Montréal Conference on e-Technologies (MCETECH'05)*, January 19-

25, Montréal, Canada.

TCHOUNIKINE P. (2002), Pour une ingénierie des environnements informatiques pour l'apprentissage humain, *Revue Information Interaction Intelligence* (www.revue-i3.org), volume 2(n_1) : pages 59-93, 2002.

VANTROYS Th. (2003), Du langage métier au langage technique, une plate-forme flexible d'exécution de scénarios pédagogiques, Thèse de Doctorat de l'Université des Sciences et Technologies de Lille.

Références à des sites Internet

ACTIS , “Smash Scenario”, ACTIS Ltd. <http://www.projectboxes.co.uk/catalogue/scenarios.html#smash> (consulté en mars 2005)

Troisième École thématique du CNRS sur les EIAH (Modèles, Architectures Logicielles et Normes pour le Développement et l'Intégration des EIAH)

http://www-rtp39.imag.fr/Autrans_2005/ProgrammeEcole/ProgrammeAutrans2005.htm (consulté en juillet 2005)

Open Universiteit Nederland <http://www.ou.nl/> (consulté en mars 2005)

Télé-université du Québec. <http://www.teluq.quebec.ca/> (consulté en mars 2005)

^[1] Meta-Object Facility : l'unique méta-méta-modèle, en d'autres termes, le langage permettant la définition de langages de modélisation ; par exemple, le méta-modèle d'UML est théoriquement une instance de ce modèle du plus haut niveau d'abstraction.

■ A propos des auteurs:

Pierre Laforcade est actuellement enseignant-chercheur au laboratoire LIUM de l'Université du Maine. Ses travaux de doctorats se sont déroulés au sein du laboratoire LIUPPA de l'Université de Pau et des Pays de l'Adour. Pendant sa thèse, il s'est intéressé au potentiel du formalisme UML (et de sa spécialisation via la méta-modélisation) pour la conception et la mise en oeuvre de situations-problèmes coopératives. Ses travaux concernent toujours actuellement l'ingénierie des EIAH avec pour spécialité l'aspect Génie Logiciel.

Adresse : IUT de Laval - Département INFO, 52 Rue des drs Calmette et Guérin, F-53020 Laval Cedex 9

Courriel : Pierre.Laforcade@univ-lemans.fr

Toile : <http://www-lium.univ-lemans.fr/~laforcad/>

Thierry Nodenot est enseignant-chercheur au laboratoire LIUPPA de l'Université de Pau et des Pays de l'Adour et mène ses recherches au sein de l'équipe IDEE. Il a encadré les travaux du doctorat de Pierre Laforcade avec Christian Sallaberry sous la responsabilité de Franck Barbier et poursuit ses recherches dans le domaine de l'ingénierie de modèles en EIAH.

Adresse : Laboratoire LIUPPA, Institut Universitaire de Technologie, 3 avenue Jean Darrigrand, 64115 BAYONNE CEDEX

Courriel : Thierry.Nodenot@iutbayonne.univ-pau.fr

Christian Sallaberry est enseignant-chercheur au laboratoire LIUPPA de l'Université de Pau et des Pays de l'Adour et mène ses recherches au sein de l'équipe IDEE. Il a encadré les travaux du doctorat de Pierre Laforcade avec Thierry Nodenot sous la responsabilité de Franck Barbier et travaille actuellement sur un projet de modélisation de ressources documentaires et de scénarios d'usages sur la base de critères spatiaux et temporels.

Adresse : UFR Droit, Economie et Gestion, Institut d'Administration des Entreprises, Université de Pau et des Pays de l'Adour, BP 575

Courriel : Christian.Sallaberry@univ-pau.fr

Référence de l'article :

Pierre Laforcade, Thierry Nodenot, Christian Sallaberry, Un langage de modélisation pédagogique basé sur UML, *Revue STICEF*, Volume 12, 2005, ISSN : 1764-7223, mis en ligne le 08/11/2005, <http://sticef.org>

© Revue Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation, 2005