



THESE

pour obtenir le grade de Docteur de l'université du Maine

Spécialité : Informatique

Méta-modélisation pour décrire et instrumenter une situation d'apprentissage de Pédagogie par Projet Collectif

Présentée et soutenue publiquement le 07 Janvier 2009 par

Firas ABDALLAH

Jury

Président du jury et Rapporteur :

M^{ME} MONIQUE GRANDBASTIEN Professeur à l'Univesité de Nancy 1

Rapporteur :

M. JEAN-LUC DEKEYSER Professeur à l'Université de Lille 1

Examineurs :

M^{ME} DOMINIQUE PY Professeur à l'Université du Maine

M. SÉBASTIEN GEORGE MCF à l'INSA de Lyon

Encadrants :

M^{ME} CLAUDINE TOFFOLON MCF à l'Université du Maine

M. BRUNO WARIN MCF à l'Université du Littoral

Directrice :

M^{ME} DOMINIQUE PY Professeur à l'Université du Maine

Laboratoire d'Informatique de l'Université du Maine (LIUM), Le Mans, 2009

A mes parents...

Table des matières

Table des matières	i
Résumé	5
1 Introduction	7
1.1 Contexte de recherche	7
1.2 Problématique	9
1.3 Objectifs	13
1.4 Méthodologie	14
1.5 Plan de la thèse	17
2 Pédagogie par Projet Collectif (PPC)	19
2.1 Courants de pensée psycho-pédagogiques	19
2.1.1 Le béhaviorisme	20
2.1.2 Le cognitivisme	21
2.1.3 Le constructivisme	22
2.1.4 Le socio-constructivisme	22
2.2 Pédagogie par Projet Collectif (PPC)	23
2.2.1 Situation de la Pédagogie par Projet Collectif	23
2.2.2 Définition de la Pédagogie par Projet Collectif	26
2.2.3 Caractéristiques de la Pédagogie par Projet Collectif	32
2.2.4 Pédagogie par Projet Collectif et éducation interdisciplinaire	36
2.2.5 Efficacité de la Pédagogie par Projet Collectif	36
2.2.6 Relation des enseignants avec les apprenants dans la PPC .	37
2.3 Mepulco-Université	40
2.3.1 Introduction	40
2.3.2 Présentation de la méthode Mepulco-Université	40
2.4 Conclusion	41
3 Etude des langages de scénarisation pédagogique	43

3.1	Introduction	43
3.2	La scénarisation des activités pédagogiques	44
3.2.1	Qu'est ce qu'un scénario pédagogique	44
3.2.2	Conception d'un scénario pédagogique	46
3.3	Etude des langages de modélisation pédagogiques existants	48
3.3.1	Introduction	48
3.3.2	IMS-LD	50
3.3.3	LDL	59
3.3.4	CPM	61
3.3.5	Approche de production de modèles d'expression de scénarios pédagogiques	64
3.4	Conclusion	65
4	Méta-modélisation de la Pédagogie par Projet Collectif	67
4.1	Introduction	67
4.2	Approche, utilisation et méthodologie	68
4.2.1	Approche de modélisation proposée	68
4.2.2	Utilisation du méta-modèle PPC	69
4.2.3	Méthodologie suivie pour élaborer le méta-modèle PPC	69
4.3	Elaboration du méta-modèle Pédagogie par Projet Collectif	70
4.3.1	Méta-modèle conceptuel général	71
4.3.2	Méta-modèle d'une étape de la PPC	77
4.3.3	Méta-modèle d'une tâche de la PPC	79
4.4	Exemple d'un scénario PPC	81
4.4.1	Cadre général	81
4.4.2	Présentation du scénario	82
4.5	Réalisation technique	83
4.6	Conclusion	85
5	Aide au choix d'un environnement d'apprentissage dédié à la Pédagogie par Projet Collectif	87
5.1	Introduction	87
5.2	Plates-formes de formation	89
5.2.1	E-learning à la base de la plate-forme	89
5.2.2	Définition d'une plate-forme de e-formation	89
5.2.3	Types de plates-formes de formation	90
5.2.4	Exemples de plates-formes de e-formation	92
5.2.5	Discussion	94
5.3	Les deux domaines du CSCW et du CSCL	95
5.3.1	CSCW (Computer-Supported Cooperative Work)	95
5.3.2	CSCL (Computer-Supported Collaborative Learning)	99
5.3.3	Liens entre CSCW et CSCL	100

5.4	Aider au choix d'un système d'apprentissage : Le modèle de fonctionnalités	103
5.4.1	Le Modèle proposé	105
5.4.2	Mise en oeuvre du modèle de fonctionnalités PPC	108
5.5	Conclusion	110
6	Opérationnalisation d'un scénario PPC	113
6.1	Introduction	113
6.2	IDM (Ingénierie Dirigée par les Modèles)	114
6.2.1	Les quatre niveaux de l'IDM	115
6.2.2	Concepts fondamentaux de l'IDM	118
6.3	Le MDA (<i>Model Driven Architecture</i>)	120
6.4	Introduire MDA dans l'activité de scénarisation et opérationnalisation pédagogique	123
6.5	Transformation du scénario PPC en un scénario PPC opérationnalisable par Moodle	124
6.5.1	Le langage de transformation ATL (<i>Atlas Language Transformation</i>)	124
6.5.2	Etapes à faire pour implémenter les règles de transformations ATL	128
6.6	Règles de transformations depuis le méta-modèle PPC vers le méta-modèle Moodle	133
6.6.1	Règles entre Projet, Etape, et Tache du côté méta-modèle PPC et Course du côté méta-modèle Moodle	134
6.6.2	Règles entre Acteur, Role du côté méta-modèle PPC et User, Role du côté méta-modèle Moodle	138
6.6.3	Règle entre Production du côté méta-modèle PPC et Files du côté méta-modèle Moodle	143
6.6.4	Règle entre Ressource du côté méta-modèle PPC et Ressource du côté méta-modèle Moodle	144
6.6.5	Règle entre Domaine, et Prerequis du côté méta-modèle PPC et Ressource du côté méta-modèle Moodle	144
6.6.6	Règle entre Outil du côté méta-modèle PPC et Activity du côté méta-modèle Moodle	146
6.6.7	Règle entre Metrique du côté méta-modèle PPC et Quiz du côté méta-modèle Moodle	147
6.6.8	Résultats	148
6.7	Outil de Transformation d'un scénario PPC (OTCPPC)	149
6.7.1	Lancer des transformations ATL depuis Java	150
6.7.2	Présentation de l'outil OTCPPC	151
6.8	Conclusion	153

7 Conclusion générale	157
7.1 Bilan des travaux réalisés	157
7.1.1 Méta-modèle dédié à la Pédagogie par Projet Collectif (PPC)	158
7.1.2 Modèle de fonctionnalités PPC	158
7.1.3 Outil de Transformation de sCénario PPC (OTCPPC)	159
7.2 Apports de la thèse	159
7.3 Perspectives	160
7.4 Conclusion	161
Bibliographie	163
A Annexe A : Description du scénario PPC élaboré par ModX	183
B Annexe B : Implémentation des règles de transformation ATL	191
B.1 Code KM3 de la création du méta-modèle PPC	191
B.2 Code KM3 de la création du méta-modèle Moodle	191
B.3 Code ATL de la création des règles de transformation entre le méta- modèle PPC et le méta-modèle Moodle	191
Table des figures	205
Liste des tableaux	208

"Si vous trouvez que l'éducation coûte cher, essayez l'ignorance."
Abraham Lincoln

Remerciements

Ce mémoire de thèse est le fruit de quatre années d'un long, dur, fatiguant mais marquant et enrichissant travail. Ce travail n'aurait pu être dans cet état sans l'aide et les conseils de nombreuses personnes.

Je remercie Monique Grandbastien, Professeur à l'université Henri Poincaré Nancy, de m'avoir fait l'honneur d'accepter d'être rapporteur de cette thèse.

Je remercie Jean-Luc Dekeyser, Professeur à l'université de Lille 1, de m'avoir fait l'honneur d'accepter d'être rapporteur de cette thèse.

Je remercie Sébastien George, Maître de conférences à l'INSA Lyon, d'avoir accepté de participer à mon jury.

Je remercie Dominique Py, Professeur à l'université du Maine, d'avoir accepté de diriger cette thèse. Sa rigueur et ses conseils m'ont permis d'approfondir certains aspects de mon travail.

Un immense merci à Claudine Toffolon, Maître de conférences à l'université du Maine, et Bruno Warin, Maître de conférences à l'université du Littoral, qui ont assuré l'encadrement scientifique de cette thèse. Ils m'ont donné le goût de la recherche et le travail collectif et m'ont apporté aussi la méthodologie de travail. Je leur témoigne ici toute ma reconnaissance et remerciement.

Je n'oublie jamais mes collègues à l'IUT de Laval pour leur support pendant une grande partie de ce travail. Ils ont contribué de près ou de loin à l'aboutissement de cette thèse : Hassina, Boubekour, Noa, Diem, Alain, et Amine.

Je remercie les membres du projet REDiM : Christophe Choquet de m'avoir ouvert de nouveaux horizons scientifiques à chaque fois que je lui posais des questions, Pierre Laforcade, pour ses conseils dans le domaine d'IDM, Sébastien Iksal, Lahcen Oubahssi, Raphaëlle Cretin-Pirolli, Philippe Cottier, Vincent Barré, pour leur aide, leurs conseils, comme les échanges scientifiques lors des réunions. Je les remercie également pour leur bonne humeur, leur humour et leur soutien.

Merci au personnel et à tous les enseignants travaillant à l'IUT de Laval.

C'est le tour des amis, vous avez contribué à votre manière à m'épauler dans ce chemin risqué ; alors dans le désordre et avec les oublis : Merci à Chawki, Leda, Mazen, Mohammad. . .

L'amour est un sentiment noble qui permet à l'être humain de mieux vivre et mieux produire. Merci à mon amour, Jasmin grâce à ton soutien j'ai pu aboutir ce premier projet.

Un gros merci à mon oncle Ali, Claire, Rémi et Léa de m'avoir accueilli et entouré dans leur maison pendant de grandes périodes de cette thèse.

Un tendre merci à ma tante Wissal de m'avoir soutenu affectivement et accompagné dans les moindres détails.

Merci à mes oncles Abdallah et Khaled et mon beau frère Hassan qui m'ont soutenu pendant toute cette période.

C'est un énorme remerciement que j'adresse maintenant à toute ma famille. Vous avez su à votre manière, par vos paroles et vos gestes, m'encourager et m'accompagner dans mon coeur dans tous les moments de la thèse. Merci à mes soeurs Manal et Manar avec qui je partageais mes soucis et mes inquiétudes, à mon adorable mère, source de tendresse et de patience, à mon père, source de sagesse et meilleur exemple pouvant être suivi pour achever une thèse. Sans ton support mon père je n'aurais jamais pu réussir ce travail!

Résumé

Le sujet abordé par cette thèse relève du domaine de l'ingénierie des Environnements Informatiques pour l'Apprentissage Humain (EIAH). D'un point de vue pédagogique, nous nous intéressons à la spécification de situations d'apprentissage dans un contexte de Pédagogie par Projet Collectif (PPC) en milieu universitaire. La PPC est une méthode de la pédagogie active "regroupant" les deux méthodes actives : Pédagogie par Projet (PP) et Apprentissage Collectif (AC). D'un point de vue informatique, nous nous intéressons au cycle de vie complet de développement du logiciel. Plus précisément depuis la spécification des besoins pédagogiques jusqu'à l'opérationnalisation dans un EIAH. Dans cette thèse, nous définissons un processus de conception et de médiatisation de situations d'apprentissage dans un contexte de Pédagogie par Projet Collectif (PPC). Ce processus est un processus d'ingénierie qui met l'enseignant au coeur du travail à toutes les étapes du cycle de vie. Il intègre la nécessaire compréhension partagée entre l'enseignant et l'expert de la plate-forme (ingénieur pédagogique) aussi bien lors du choix de plate-forme que de l'opérationnalisation du scénario PPC sur la plate-forme.

Nous centrons notre processus autour de la notion de scénarios permettant de mettre en oeuvre des situations d'apprentissage dans un contexte de Pédagogie par Projet Collectif (PPC) et nous adoptons une démarche d'Ingénierie Dirigée par les modèles (IDM) autour de ces scénarios. Cette approche est proposée dans le but de favoriser la conciliation des deux points de vue : pédagogique et technique. Pour ce faire nous adoptons une approche d'expression centrée sur une méta-modélisation spécifiquement dédiée à la Pédagogie par Projet Collectif (PPC). Notre recherche en EIAH nous a permis de constater une insuffisance des langages de scénarisation pédagogique dans un contexte de Pédagogie par Projet Collectif (PPC). Ainsi définir un méta-modèle dédié à la Pédagogie par Projet Collectif (PPC) peut s'avérer plus pertinent pour l'enseignant qui évolue alors dans une culture pédagogique qui lui est propre. Il permet ainsi de pallier à l'inadéquation de la description inhérente à une approche généraliste. Nous proposons d'utiliser notre méta-modèle PPC pour aider l'enseignant concepteur à élaborer ses scénarios d'apprentissage. Dans la dernière phase nous choisissons la

stratégie de la réutilisation de plates-formes existantes et déjà opérationnelles plutôt que celle du développement ex-nihilo d'une nouvelle plate-forme, fût-elle à base de composants réutilisables. Cependant l'extrême complexité et diversité des technologies et compétences à maîtriser rendent difficile la réutilisation d'une plate-forme dans un contexte nouveau, notamment faute de capitalisation suffisante. Ces problèmes de réutilisation sont d'ailleurs une problématique centrale dans les activités du génie logiciel, mais ils pointent sans doute avec plus d'acuité dans le domaine nouveau et fondamentalement transdisciplinaire des EIAH. Pour cela, afin de maximiser la pertinence de l'opérationnalisation finale, nous proposons d'aborder le problème par une phase de choix de plates-formes à partir d'un modèle de fonctionnalités que nous avons défini et dédié à la PPC. Ce modèle de fonctionnalités agit alors comme un modèle qualité spécifiquement défini pour la PPC. L'enseignant concepteur est guidé pour choisir une plate-forme la mieux adaptée à ses besoins formulés à travers son scénario. Il peut être amené à choisir une plate-forme parmi plusieurs mises à sa disposition ou bien évaluer dans quelle mesure une certaine plate-forme est bien adaptée à son besoin (scénario). Ensuite nous proposons une démarche par transformation de modèles depuis le méta-modèle de la situation d'apprentissage de la PPC vers celui de la plate-forme cible qui a été sélectionnée. Cette dernière phase conclut notre démarche et permet de satisfaire notre hypothèse fondatrice de départ qu'une situation d'apprentissage de type PPC peut être médiatisée par transformation du scénario PPC en un scénario PPC opérationnalisable par la plate-forme cible suivant une approche de type MDA (*Model Driven Architecture*). Cette approche basée sur les modèles, met au centre la technique de transformation de modèles. Cette technique favorise la traçabilité entre les différents modèles produits à différents niveaux d'abstraction. Nous avons testé nos travaux sur la plate-forme Moodle, utilisée à l'IUT de Laval. Ce cas d'étude nous a amené de plus à développer un outil informatique à l'usage des enseignants-concepteurs destiné à faciliter la transformation d'un scénario PPC en scénario PPC opérationnalisable sur une plate-forme cible quelconque, Moodle dans notre cas.

Chapitre 1

Introduction

1.1 Contexte de recherche

Le travail de cette thèse s'inscrit dans le domaine de l'ingénierie des Environnements Informatiques pour l'Apprentissage Humain (EIAH) spécifié [197] comme *l'ensemble des travaux visant à définir des éléments de méthodes et de techniques reproductibles et/ou réutilisables facilitant la mise en place (conception - réalisation - expérimentation - évaluation - diffusion) d'environnements de formation ou d'apprentissage (dans leur articulation avec les dispositifs informatiques d'aujourd'hui) en permettant de dépasser le traitement ad hoc des problèmes*. L'objectif de cette ingénierie [197], [198] étant d'offrir, aux équipes chargées de la mise en place d'un EIAH, un ensemble de méthodes et techniques capitalisées, c'est-à-dire identifiées et décrites [126], pour les guider dans le processus de mise en place de leur EIAH. Nous constatons que dans ce cadre l'évolution des Technologies de l'Information et de la Communication (TIC) notamment celles liées à l'Internet amènent à formuler non seulement de nouveaux besoins en termes pédagogiques : nouvelles pédagogies, nouveaux programmes de formation, évolution du public, . . . mais aussi de nouveaux modes de travail : travail coopératif, travail en mode projet, gestion distribuée de projets.

La recherche en EIAH est pluridisciplinaire : chercheurs des sciences humaines (didacticiens, pédagogues ou autres acteurs des sciences de l'éducation) et chercheurs en informatique sont intimement liés à la conception d'un EIAH [197]. Le travail sur la conception d'un EIAH ne se fait pas en étudiant tout d'abord les sciences humaines (didactiques, sciences de l'éducation, etc.), et puis en spécifiant l'artefact informatique qu'il s'agirait ensuite pour l'informaticien de réaliser. Réciproquement, il ne s'agit pas aussi que l'informaticien conçoive un système fondé sur une nouvelle technologie puis cherche ensuite comment l'utiliser en éducation. *"Il ne s'agit pas de considérer l'éducation puis la technologie ou la technologie puis*

l'éducation, mais la recherche en EIAH" [197], [198].

Malgré la grande variété d'EIAH (hypermédia, système à base de connaissances, systèmes tuteur intelligent, plate-forme de formation ouverte et à distance) la conception d'EIAH nécessite dans tous les cas de prendre en compte une multitude de problèmes inter-reliés [197] :

- *analyse didactique du contexte (analyse des conceptions des apprenants, des obstacles à l'évolution de ces conceptions, etc) ;*
- *identification des objectifs d'apprentissage ;*
- *étude du contexte d'insertion de l'activité ;*
- *conception et spécification de l'activité proposée ;*
 - *conception de la tâche à réaliser ;*
 - *identification des acteurs et de leurs rôles ;*
 - *prise en compte des dispositions personnelles, des intentions et attitudes des acteurs ;*
 - *articulation acteurs/outils ;*
- *étude des comportements émergents, étude de l'activité réelle et de l'usage effectif des outils ;*
- *évaluation et affinement.*

Selon Tchounikine [198], les intérêts de l'existence de l'ingénierie EIAH sont : l'amélioration de la définition et l'échange de résultats, la proposition d'un cadre spécifique pour certains travaux, et la modification de l'économie de la recherche. Nous inscrivons notre travail dans le cadre de l'ingénierie et plus spécifiquement sur les aspects conception et opérationnalisation de scénario pédagogique où les enseignants deviennent acteur de leur propre conception. Nous désirons appliquer des situations d'apprentissage situées dans un contexte de **Pédagogie par Projet Collectif** (PPC) telle qu'elle est déjà initiée dans certaines formations universitaires [69]. La Pédagogie par Projet Collectif (PPC) est issue du courant socio-constructiviste selon lequel : 1) les connaissances sont construites, 2) l'apprenant construit sa façon de comprendre et d'apprendre et, 3) l'apprentissage va du social vers l'individu [117], [16], [8].

Lebrun [130] considère que la PPC est une méthode de la pédagogie active "regroupant" les deux méthodes : Pédagogie par Projet (PP) et Apprentissage Collectif (AC) ¹. Une situation d'apprentissage dans un contexte de PPC présente donc un certain nombre de caractéristiques [167], [87], [130], [193] : 1) Elle est initiée à partir d'une thématique concrète de la vie (besoin, manque, conflit socio-cognitif, souhait d'arriver quelque part, d'aboutir à quelque chose, etc), soit par l'enseignant, soit par les apprenants. 2) Elle engage les apprenants dans des projets complexes ancrés dans la réalité à travers lesquels ils développent et appliquent des compétences et du savoir. 3) Elle identifie et formule des buts par rapport auxquels les enjeux et les résultats des processus d'acquisition des connaissances

¹Lebrun utilise le terme "Apprentissage Coopératif"

ne sont ni prédéterminés, ni entièrement prévisibles. 4) Les apprenants qui sont mis dans une situation de travail de groupe interviennent comme parties prenantes avec des responsabilités. 5) La situation incite les apprenants à consulter de nombreuses sources d'information et de disciplines afin de résoudre des problèmes (ce qui les amène à développer des compétences transversales). 6) Elle induit un ensemble de tâches dont la réalisation amène les apprenants à s'impliquer, à interagir et à jouer un rôle actif. Cette pédagogie pose des problèmes spécifiques liés d'une part aux modes de travail mis en place : gestion de projet, apprentissage coopératif, . . . et d'autre part à l'articulation de ces modes de travail avec les compétences métiers issues du domaine spécifique de la formation dispensée.

D'un point de vue informatique, nous nous intéressons à la conception et l'opérationnalisation de scénario pédagogique dans un EIAH et plus particulièrement à la mise en oeuvre d'une démarche d'Architecture Dirigée par les Modèles (MDA) pour concilier les deux approches : pédagogie et technique. Etant donné le nombre important de plates-formes déjà existantes et les difficultés que représentent le développement d'une nouvelle plate-forme, notre travail cherche à réutiliser et capitaliser ce que les plates-formes existantes proposent. La thèse se situe donc dans les travaux de choix et d'adaptation de plate-forme existante et non pas dans ceux visant la mise en place d'une nouvelle plate-forme.

1.2 Problématique

Tchounikine [198] définit un EIAH (cette fois ci pris comme système et non comme domaine ou discipline) comme un "*un environnement informatique conçu dans le but de favoriser l'apprentissage humain*". La production d'un EIAH nécessite une recherche multidisciplinaire : sciences de l'éducation, pédagogie, psychologie cognitive, didactique, linguistique, communication, informatique, etc.

L'approche traditionnelle d'ingénierie des dispositifs instrumentés, appelée aussi approche documentaliste, est essentiellement concentrée sur les mécanismes permettant de faciliter la (ré)ingénierie, la réorientation, et l'utilisation efficace des ressources numériques existantes pour supporter l'apprentissage et l'enseignement (en fonction de la plate-forme choisie). Une nouvelle approche, appelée approche centrée sur les activités, recommande à partir d'un scénario d'apprentissage, d'aboutir à un système d'apprentissage comme un site web par exemple [166].

Certains auteurs ne font aucune distinction entre scénario pédagogique et scénario d'apprentissage², d'autres, non seulement les différencient mais également en font une distinction avec le scénario de formation [161]. Ces auteurs considèrent qu'un scénario pédagogique est la "*conjonction d'un scénario d'apprentissage et d'un scénario de formation qui lui est associé avec l'expression des liens qui les*

²Pour notre part, nous utilisons indifféremment les deux expressions "scénario pédagogique" et "scénario d'apprentissage"

unissent" [161]. Un scénario d'apprentissage est l'"ensemble des activités destinées aux apprenants et organisées en un tout cohérent ; à ces activités, on greffe les instruments offerts comme supports aux activités (instruments-intrants) et les instruments à être réalisés par les apprenants (produits)" [161]. Un scénario de formation est l'"ensemble des activités destinées au formateur et organisées en un tout cohérent ; à ces activités, on greffe les instruments offerts comme supports aux activités (instruments-intrants) et les instruments à être réalisés par le formateur (produits)" [161].

Selon Pernin et Lejeune, un scénario d'apprentissage *"représente la description, effectuée a priori ou a posteriori, du déroulement d'une situation d'apprentissage ou unité d'apprentissage visant l'appropriation d'un ensemble précis de connaissances, en précisant les rôles, les activités ainsi que les ressources de manipulation de connaissances, outils et services nécessaires à la mise en oeuvre des activités [164].* Pernin *et al.* [164], [166], [165] caractérise un scénario principalement par :

- sa granularité. Selon la granularité de la situation d'apprentissage souhaitée, il existe trois niveaux de granularité qui sont :
 - niveau décrivant des activités élémentaires telles que des exercices, des textes de lecture, des simulations, etc. . .
 - niveau décrivant l'enchaînement d'activités pour permettre de décrire l'organisation de la séquence de ces activités
 - et niveau de structuration pédagogique permettant de décrire les structururations d'unités de haut niveau tels que les cours, modules etc.
- son cycle de vie, étant défini comme l'évolution de son statut dans le temps, est composé de plusieurs phases :
 - la phase de conception initiale définissant un scénario abstrait sans tenir en compte des conditions de mise en oeuvre ;
 - la phase de contextualisation définissant les conditions d'exploitation d'un scénario abstrait dans un contexte précis ;
 - la phase d'exploitation définissant un scénario contextualisé prêt à être utilisé par différents acteurs (apprenants, enseignants, tuteurs, etc.) ;
 - la phase de retour d'usage servant à évaluer les résultats de la phase d'exploitation des scénarios afin de fixer les conditions de leur réutilisation ultérieure dans d'autres contextes.
- une description composée de plusieurs facettes (prescription, suivi-contrôle, connaissances) manipulées par un ou plusieurs types d'acteurs.

En fonction de la granularité (le niveau de la situation d'apprentissage), Pernin décrit trois niveaux de scénarios qui sont :

- **Scénarios de structuration pédagogique** : organiser les unités de structuration pédagogiques telles que les cours, modules, leçons, cursus, crédits.

- **Scénarios d’enchaînement d’activités** : organiser les activités d’un apprenant ou d’un groupe d’apprenants au sein d’une séquence pédagogique, d’une séance, etc.
- **Scénarios de réalisation de tâche** : décrire le déroulement d’une tâche précise (ex. parcourir un hyper-document, résoudre un exercice, manipuler une simulation).

Une partie de la communauté de recherche préconise des environnements où l’on permet aux acteurs pédagogiques : enseignants, concepteurs, ingénieurs pédagogiques, . . . de formuler eux-mêmes leurs scénarios d’apprentissage à travers un langage formel de description de scénario ou *Educational Modelling Language* (EML). L’enjeu est ensuite d’inclure ces scénarios dans une plate-forme qui "jouerait" ces scénarios. Dans cette approche, une proposition majeure, *Instructional Management Systems-Learning Design* (IMS-LD) a été faite par *IMS Global Learning Consortium* [112], [113], [114], [122].

Cependant, de nombreux travaux et expériences ont été menés et ont montré des limites à cette approche. Ainsi pour [43] "*Cependant malgré de nombreux travaux réalisés pour simplifier l’appréhension du standard, sa complexité le rend difficile à maîtriser*", de même pour [39] "*la tâche de transposition d’un modèle pédagogique demeure complexe pour l’enseignant qui n’est pas familiarisé avec les concepts IMS-LD*" ou encore [90] "*. . . mais cette description est complexe et nécessite une compréhension approfondie de l’EML utilisé*". Certaines limites sont dues aux hypothèses de base de ces langages. Modéliser un cours à partir de la métaphore théâtrale par exemple, ne permet pas de répondre à tous les besoins de scénarisation pédagogique malgré les intentions des auteurs de proposer un langage qui conviendrait à tous les types de pédagogies. On peut également évoquer parmi les critiques, la difficulté pour des enseignants non spécialisés dans ces langages, de formuler leurs besoins notamment dans le domaine de l’apprentissage collectif : "*It is not always easy to model a collaborative learning situation using IMS-LD level B and C*" [143]. Notre cible est les enseignants. Il s’agit de leur permettre de devenir eux-mêmes concepteurs de leur propre situation d’apprentissage. On parle dans cas "d’enseignant concepteur". Une voie de recherche est actuellement suivie avec la mise à la disposition d’outils comme *Reload* [176], *Coopcore* [56], . . . ou de plate-formes *Internet* comme *NetUniversité* [151] qui proposent une interface qui rend possible à des enseignants-concepteurs de formuler leurs scénarios sans avoir une connaissance parfaite du langage. Une autre voie consiste à proposer à ces EML des extensions plus adaptées aux situations pédagogiques rencontrées [133], [92].

Une approche différente à ces EML généralistes est de préconiser une méta-modélisation des situations d’apprentissage à un niveau "métier". Un méta-modèle est un langage de modélisation spécifique à un domaine (*Domain Specific Language*), il est utilisé pour exprimer les concepts communs des modèles appartenant

à un même domaine. Il est construit à partir de modèles informels, de recommandations dans un langage naturel ou semi-formel écrits généralement en UML (*Unified Modeling Language*). Cette approche a été utilisée par plusieurs chercheurs : le langage de modélisation CPM (*Cooperative Problem-based learning Meta-model*) est dédié à l'apprentissage par problème (*Problem Based Learning*). Le langage LDL (*Learning Design Language*) s'appuie sur un méta-modèle défini pour le CSCW [143]. Une autre solution existante, est de laisser les enseignants définir les concepts de base du modèle pédagogique qu'ils souhaitent dans un méta-modèle [45], [50] en fonction de leurs propres besoins au moyen d'un modèleur de méta-modèle comme ModX [149] ou d'édition de scénarios comme ECOS. Selon [50], cette approche permet de : 1) débarrasser le pédagogue de l'apprentissage nécessaire d'un langage de modélisation computationnel, EML de type IMS-LD, ou techno-centré, comme *Unified Modeling Language* (UML) 2) favoriser l'expression de situations pédagogiques riches et conformes aux objectifs des concepteurs 3) permettre l'échange et la communication entre concepteurs autour de modèles métiers qu'ils partagent. De tels modèles sont porteurs de caractéristiques du contexte pédagogique de ses concepteurs. Ils reflètent une démarche en des termes propres à une expertise pédagogique ciblée et spécifique.

Il existe des centaines de plates-formes commerciales et "Open Source" ³, mais aucune de ces plates-formes ne semble être adaptée à une situation d'apprentissage dans un contexte de la PPC [87], [17]. Ce phénomène est d'ailleurs vrai pour n'importe quelle pédagogie. Le Collectif AS Plateforme [75] observe que l'extrême complexité et diversité des technologies et compétences à maîtriser rend difficile non seulement le développement et la mise en oeuvre d'un EIAH mais aussi, faute de capitalisation suffisante, sa réutilisation dans un contexte nouveau. Ces problèmes de développement et réutilisation sont d'ailleurs une problématique centrale dans les activités du génie logiciel [192], [96], mais ils pointent sans doute avec plus d'acuité dans le domaine nouveau et fondamentalement transdisciplinaire des EIAH.

De plus, à la suite de [43], [44] nous observons que les enseignants quand ils utilisent une plate-forme ne l'utilisent pas toujours telle qu'elle a été prescrite ou "bricolent" leur propre environnement numérique de formation en utilisant un composite d'outils : wiki, messagerie, En vue de la transformation du scénario de la situation d'apprentissage donnée pour qu'il soit opérationnalisable par la plate-forme existante, une des avancées serait de favoriser la compréhension partagée de modèles représentatifs. Ces derniers sont issus de la pédagogie, et d'une manière générale des Sciences Humaines et Sociales (SHS), et de l'informatique. Notre stratégie consiste donc, à assister l'enseignant pour implémenter son scénario PPC dans la plate-forme cible choisie, plutôt que développer une nouvelle plate-forme. Notre travail cherche à réutiliser et capitaliser ce que les

³<http://thot.cursus.edu/>

plates-formes existantes proposent.

Nous pensons qu'une approche de type MDA (*Model Driven Architecture*) [94] du génie logiciel en mettant au centre le modèle, la méta-modélisation, les transformations et les niveaux d'abstraction peut contribuer à une telle avancée. Cette approche devant faciliter le dialogue entre l'enseignant, détenteur des connaissances de situation d'apprentissage et l'ingénieur pédagogique, détenteur des connaissances de la plate-forme. Nous posons comme hypothèse fondatrice qu'une situation d'apprentissage de type PPC peut être médiatisée par transformation d'un scénario PPC en un scénario PPC opérationnalisable par la plate-forme cible suivant une approche de type MDA (*Model Driven Architecture*). Cette approche basée sur les modèles, met au centre la technique de transformation de modèles. Cette technique permet la traçabilité entre les différents modèles produits dans différents niveaux d'abstraction.

- La première question qui se pose est de déterminer la place et la forme de cette méta-modélisation dans la définition d'une situation d'apprentissage de PPC. Comment définir un méta-modèle dédié à la Pédagogie par Projet Collectif pour décrire des situations d'apprentissage dans un contexte de PPC (autrement dit des scénarios PPC)? Le méta-modèle PPC sera composé d'une syntaxe et d'une sémantique. La syntaxe définira les sujets (concepts/entités) et les verbes (liens entre les concepts), la sémantique donnera l'interprétation aux sujets et verbes [126].
- La seconde question qui se pose est comment aidons-nous l'enseignant-concepteur à exprimer ses besoins, et à choisir une plate-forme existante afin de médiatiser les scénarios PPC?
- La troisième question qui se pose est celle du processus pour passer de la situation d'apprentissage méta-modélisée à la médiatisation sur une plate-forme donnée? Autrement dit, comment transformer un scénario PPC en un scénario PPC opérationnalisable par la plate-forme cible?

1.3 Objectifs

Les objectifs visés par notre recherche concernent le choix de plates-formes en fonction de leurs aptitudes à supporter une Pédagogie par Projet Collectif et l'opérationnalisation d'un scénario PPC dans les plates-formes choisies. Nous souhaitons définir un processus générique d'ingénierie qui intègre la nécessaire compréhension partagée entre l'enseignant-pédagogue et l'expert de la plate-forme (ingénieur pédagogique) lors du choix de cette plate-forme et de l'opérationnalisation d'un scénario PPC sur celle-ci. Nous montrons qu'un tel processus peut s'articuler sur la notion de méthode comme moyen de scénarisation pédagogique et sur une approche d'Architecture Dirigée par les Modèles (MDA) comme moyen

privilegié de compréhension partagée.

Le premier objectif concerne la méta-modélisation de situations d'apprentissage PPC. Nous visons la définition d'un métamodèle de référence pour la PPC et son application à des situations d'apprentissage PPC. Le deuxième concerne la proposition d'un modèle de fonctionnalités pour guider le choix des enseignants concepteurs pour une plate-forme la mieux adaptée à leurs besoins formulés à travers leur scénario PPC. L'enseignant peut être amené à choisir une plate-forme parmi plusieurs mises à sa disposition ou bien évaluer dans quelle mesure une plate-forme donnée est bien adaptée à son besoin (scénario). Le troisième objectif concerne la production de modèle de situation d'apprentissage par transformation d'un méta-modèle dédié à la PPC. Pour ce faire, nous nous proposons de définir un processus outillé informatiquement qui s'appuiera sur la méta-modélisation définie pour assister l'enseignant et l'ingénieur-pédagogique à médiatiser la situation d'apprentissage.

Le processus d'ingénierie que nous proposons est illustré par la figure 1.1. L'enseignant concepteur utilise le méta-modèle dédié à la Pédagogie par Projet Collectif (PPC) pour élaborer son scénario PPC qu'il souhaite mettre en oeuvre. Dans une seconde étape, l'enseignant-concepteur utilise le modèle de fonctionnalités PPC afin de choisir la plate-forme qui correspond le plus aux besoins formulés dans son scénario d'apprentissage PPC. Enfin, il utilise l'Outil de Transformation de sCénario PPC (OTCPPC) proposé permettant d'appliquer les règles de transformation de modèles pour transformer son scénario PPC en un scénario PPC opérationnalisable par la plate-forme choisie. Ce dernier est fourni à l'ingénieur pédagogique, chargé de le déployer dans la plate-forme cible, afin d'être "joué" par les acteurs de la Pédagogie par Projet Collectif (PPC).

1.4 Méthodologie

Dans cette section, nous décrivons notre méthodologie et processus de travail (figure 1.1). Tout d'abord, nous proposons un méta-modèle dédié à la Pédagogie par Projet Collectif (PPC). Pour le construire, nous adoptons le point de vue de [200] qui se base, d'un côté sur une étude théorique (bibliographique) pour récolter tous les éléments et concepts d'une Pédagogie par Projet Collectif. D'un autre côté et pour assurer l'aspect réaliste du méta-modèle, ce point de vue se base sur des études empiriques (ex : Mepulco-Université, Licence Professionnelle à l'IUT de Laval etc.) Le méta-modèle résultant (méta-modèle PPC) permet à son tour d'interpréter, d'un côté, l'étude théorique achevée, et d'un autre côté, de catégoriser et de structurer les phénomènes (expériences) étudiés figure 1.2.

Pour faire l'étude théorique, nous nous basons sur des publications scientifiques académiques et concernant l'étude pratique, nous la soutenons par l'expertise acquise par l'analyse et l'observation sur plusieurs années de la pratique de Péda-

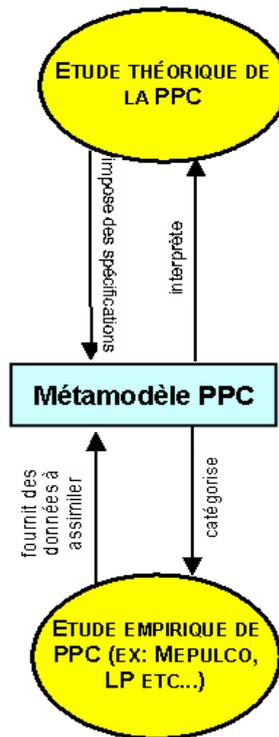


FIG. 1.2: Approche pour proposer le méta-modèle PPC

besoin (scénario).

La plate-forme Moodle, utilisée à l'IUT de Laval, est choisie comme cas d'étude pour montrer comment opérationnaliser dans une plate-forme donnée un scénario PPC élaboré par le méta-modèle PPC ⁴. Pour cela, nous élaborons (figure 1.1) le méta-modèle fonctionnel de la plate-forme Moodle dans le but de définir troisièmement des règles de transformation depuis le méta-modèle PPC vers son méta-modèle. Les activités de transformation sont réalisées en tant que processus automatisés qui prennent un ou plusieurs modèles sources en entrée et produisent un ou plusieurs modèles de production cible en suivant un ensemble de règles de transformation. Ce processus étant référé comme Modèle de Transformation [185]. La transformation de modèle se fait par "*mapping*" entre un modèle initial et un modèle cible. Chaque modèle est décrit par un méta-modèle, qui recense les caractéristiques de ce modèle. Le "*mapping*" est alors défini comme une traduction entre le méta-modèle initial et le méta-modèle cible.

Nous développons des règles de transformations pour proposer à l'enseignant-concepteur un Outil de Transformation de sCénario PPC (OTCPPC) permettant

⁴Nous avons choisi Moodle car c'est l'une des plates-formes adaptées à la pédagogie socio-constructiviste des plus connues

de transformer le scénario PPC en un scénario PPC opérationnalisable par la plate-forme cible.

Le scénario PPC opérationnalisable est fourni à l'ingénieur pédagogique, chargé de le déployer dans la plate-forme cible.

1.5 Plan de la thèse

Ce mémoire de thèse est décomposé en 7 chapitres.

Le deuxième chapitre est consacré à l'étude théorique de la Pédagogie par Projet Collectif (PPC). La situation, la définition et les caractéristiques de cette méthode pédagogique sont décrites. Nous continuons ce chapitre par la présentation d'une pratique de la PPC appliquée à l'Université du Littoral Côte d'Opale : **Mepulco-Université**. La présentation de cette pratique permet d'illustrer concrètement l'étude théorique.

Dans le troisième chapitre, nous étudions le concept de scénario pédagogique et son cycle de vie dans un contexte informatique. Nous présenterons ensuite les principaux langages de modélisation pédagogique EML (*Educational Modeling Language*) et les approches de scénarisation proposées bien sûr en les discutant du point de vue de la Pédagogie par Projet Collectif (PPC). Ceci nous permet de mettre en évidence les limites actuelles pour élaborer des scénarios pédagogiques dans un contexte de la Pédagogie par Projet Collectif (PPC).

Face à ces limites et vis-à-vis des conclusions de l'étude réalisée dans les chapitres précédents, nous présentons dans le quatrième chapitre notre première contribution dans la thèse consistant à proposer un méta-modèle dédié à la Pédagogie par Projet Collectif (PPC). Nous expliquons aussi l'approche de modélisation choisie, l'utilisation de ce méta-modèle et la méthodologie suivie pour l'élaborer. Pour illustrer son utilisation, nous présentons un scénario PPC exemple qui a été élaboré à l'IUT de Laval pour la première année en informatique à partir de ce méta-modèle. Nous terminons ce chapitre en élaborant le méta-modèle PPC et le scénario PPC conforme en utilisant un outil d'élaboration de méta-modèles et des modèles correspondants. Ce outil, ModX, est développé au laboratoire de TRIGONE à Lille.

Le cinquième chapitre concerne l'étude du cadre technique de nos travaux : les plates-formes de formations et les plates-formes de travail collectif. Tout d'abord, nous commençons par présenter brièvement des environnements supports existants. Notre deuxième contribution dans la thèse consiste à élaborer un modèle de fonctionnalités aidant à choisir une plate-forme pour une situation d'apprentissage dans un contexte de la PPC. Ce modèle de fonctionnalités est utilisé pour élaborer une typologie des plates-formes de formation et de travail collectif en fonction de la Pédagogie par Projet Collectif. Cette typologie va permettre à l'enseignant concepteur de la PPC de catégoriser les plates-formes avec lesquelles il

a la possibilité de travailler. Cela va lui permettre de choisir la plate-forme qui sert au plus près les objectifs du scénario pédagogique qu'il souhaite mettre en oeuvre.

Dans le sixième chapitre, nous expliquons l'approche IDM (Ingénierie Dirigée par les Modèles) et sa relation avec les EIAHs. Cette étude est suivie d'une troisième contribution répondant à l'opérationnalisation d'un scénario PPC. Nous élaborons et développons des règles de transformation entre le méta-modèle PPC et celui de la plate-forme Moodle étant choisie comme étude de cas. Nous expliquons dans ce chapitre le langage de transformations ATL (*Atlas Language Transformation*) utilisé afin d'élaborer et de développer les règles de transformation proposée. Nous terminons ce chapitre par la proposition d'un outil avec une interface graphique qui sera utilisée par l'enseignant concepteur de la PPC afin de transformer son scénario PPC en scénario PPC opérationnalisable par la plate-forme cible.

Le mémoire de thèse se termine par un chapitre de conclusions et de perspectives (chapitre 7).

Dans les annexes nous proposons divers éléments évoqués ou présentés plus succinctement dans l'élaboration du scénario PPC avec l'outil ModX, et dans les règles de transformations entre le méta-modèle PPC et celui de Moodle. Des extraits de code d'implémentation de règles de transformation sont également proposés en annexe.

Chapitre 2

Pédagogie par Projet Collectif (PPC)

Dans le chapitre précédent, nous avons évoqué que la Pédagogie par Projet Collectif (PPC) était au centre de nos travaux de recherche. Un des objectifs de nos travaux est de répondre aux limites actuelles en terme de langage de scénarisation pédagogique dans un contexte de Pédagogie par Projet Collectif (PPC). Nous avons spécifié dans le chapitre 1 que nous adoptons une approche d'expression centrée sur une méta-modélisation dédiée à la Pédagogie par Projet Collectif (PPC). Pour proposer un méta-modèle dédié à la PPC, il est nécessaire de fixer un cadre théorique à la Pédagogie par Projet Collectif (PPC). Pour cela, nous commençons tout d'abord par présenter les courants de pensée psycho-pédagogiques existants. Notamment le socio-constructivisme, courant auquel appartient la Pédagogie par Projet Collectif (PPC) (section 2.1). Nous nous attachons ensuite à faire un état de l'art de la Pédagogie par Projet Collectif (PPC) (section 2.2). Nous présentons une application pratique de la PPC à l'Université du Littoral Côte d'Opale : **Mepulco-Université** (section 2.3). Ceci nous permet d'illustrer concrètement l'étude théorique.

2.1 Courants de pensée psycho-pédagogiques

Dans le cadre de notre thèse, nous nous intéressons à l'étude de la **pédagogie** qui désigne l'activité que l'enseignant déploie pour enseigner. Elle est définie comme le principe des actions d'éducation, d'enseignement et de formation [150]. Nous concluons que : enseigner, apprendre, instruire et former, relèvent à la fois de l'activité pédagogique et de l'ordre des idées qui en guide cette activité. La pédagogie a pour objet la part systématique de l'activité humaine conduisant

les actions d'éducation et de formation. Elle ne s'occupe pas de l'enseignant, ni du savoir, ni de l'apprenant mais de l'**activité** qui les réunit. Altet [7] définit la pédagogie comme "*le champ de la transformation de l'information en savoir par la médiation de l'enseignant, par la communication, par l'action interactive dans une situation éducative donnée.*" Le Savoir ne devient Connaissance que par la démarche personnelle d'apprentissage de l'apprenant.

Les courants de pensée psycho-pédagogiques sont une des manières de catégoriser les pédagogies en s'appuyant principalement sur des données de la psychologie [16], nous étudions les différents courants de pensée psycho-pédagogiques : le behaviorisme, le cognitivisme, le constructivisme, et le socio-constructivisme. Ce dernier et dans le cadre de notre thèse, nous intéresse le plus parce qu'il est celui auquel appartient la Pédagogie par Projet Collectif (PPC).

2.1.1 Le behaviorisme

Fondé par Watson en 1913 [211], le behaviorisme est connu littéralement comme la science du comportement. La pédagogie centrée sur le comportement, issue des découvertes d'Ivan Pavlov en 1902 sur le conditionnement est la base du courant behavioriste fondé sur la métaphore du "remplissage" d'un récipient vierge que serait l'apprenant. Waston fonde sa théorie sur la relation stimulus-réponse et sur la notion d'adaptation. En fait, dès le milieu du XIXe siècle, le philosophe français Théodule Ribot a étudié le behaviorisme en se livrant à des recherches sur la psychologie expérimentale [8].

Cependant, c'est Henri Piéron qui, dans sa leçon préparatoire à l'Institut des hautes études en 1908, introduit le premier la notion de comportement. Le comportement pour un behavioriste est "*l'ensemble des réactions adaptatives objectivement observables, qu'un organisme exécute en riposte aux stimuli, eux aussi objectivement observables, provenant du milieu dans lequel il vit*" [8].

Ce courant est appliqué par la pédagogie traditionnelle. Les courants de pensée psycho-pédagogiques behavioristes appelés aussi comportementalistes [16] ont toujours été dominants et encore aujourd'hui très répandus et facilement reconnaissables dans l'enseignement à l'aide des technologies de l'information et de la communication (TIC). Les exercices en ligne pour la mise en pratique de contenus théoriques, les tutoriels et les exposés magistraux enregistrés en mode vidéo et audio, accessibles sur une page Web en sont des exemples types [16]. L'introduction de l'informatique dans l'enseignement a d'abord et, naturellement, été de nature behavioriste. En fait, le modèle sur lequel se fonde le behaviorisme, étant un modèle comportemental et ne traitant pas les représentations mentales, est confortable dans l'optique d'une introduction de la machine numérique. L'enseignement est alors la création de nouveaux conditionnements. Le behaviorisme est la cause de la naissance de plusieurs formes d'enseignement expérimental où l'apprentissage est par essais et erreurs dans lequel l'erreur est vécue comme ex-

périence utile [203].

Skinner [187], psychologue du courant behavioriste, développe l'enseignement programmée, dans lequel l'enseignant doit avoir prévu les réactions de l'apprenant pour procéder à un renforcement négatif ou positif. Pour Skinner, l'apprentissage correspond à une association entre un évènement du monde et une réponse motrice. Son conditionnement opérant postule que l'apprentissage peut être obtenu en utilisant des renforcements positifs (ou récompenses) et des renforcements négatifs (ou punitions). L'individu adapte son comportement afin d'éviter les renforcements négatifs et d'augmenter les renforcements positifs. Skinner débouche sur l'idée de machines à enseigner permettant l'exécution d'enseignements modélisés sous formes d'algorithmes linéaires.

L'apprenant doit démontrer qu'il peut donner une réponse correcte à un stimulus donné et reproduire, par imitation, une série de comportements en les associant les uns aux autres. La responsabilité de l'atteinte des objectifs d'apprentissage appartient au professeur. Le modèle sur lequel le béhaviorisme se fonde, est un modèle seulement comportemental qui rejette l'étude des représentations mentales, et qui simplifie l'étude pour la limiter essentiellement au modèle stimulus-réponse.

2.1.2 Le cognitivisme

Le cognitivisme fait partie des sciences cognitives qui a pour objet l'étude et la compréhension des systèmes capables de traiter des connaissances. Il a fait son apparition dans les années 1950 en opposition au béhaviorisme [37]. Ce courant est né en 1956 en même temps que l'Intelligence Artificielle [58]. Il a été inventé par Muller et Bruner [37]. Son objet est l'étude du fonctionnement de l'activité intellectuelle [8]. L'apprentissage ne peut être limité au conditionnement déjà enregistré, mais à l'inverse il est envisagé comme nécessitant un traitement complexe de l'information reçue par l'apprenant. Une attention toute particulière est portée sur les stratégies utilisées par l'apprenant dans la réalisation de ses activités d'apprentissage [16]. Ce qui est important ici, ce n'est pas la performance (les comportements observables) mais plutôt les stratégies utilisées pour acquérir la connaissance. L'évaluation est plutôt orientée sur la construction progressive de la connaissance. La rétroaction est fournie par le professeur et concerne tant les connaissances acquises que les stratégies employées pour réaliser la tâche.

Les psychocognitivistes s'intéressent plus particulièrement au traitement de l'information chez l'apprenant. Ils cherchent à expliquer l'apprentissage en décrivant les processus mentaux à l'origine de la gestion de la construction des connaissances (analyse, représentations, raisonnement, validation, etc.). Ils considèrent que l'apprenant perçoit les informations qui lui proviennent du monde extérieur, les reconnaît, les stocke dans sa mémoire, puis les récupère lorsqu'il en a besoin pour mieux comprendre son environnement ou résoudre des problèmes. Ils consi-

dèrent que la mémoire possède une structure propre, dans le sens que l'information est organisée et nécessite un recours à des stratégies pour cette organisation. Les psychocognitivistes se concentrent sur l'étude des états mentaux et se divisent en deux courants qui sont le symbolisme et le connexionnisme.

Le symbolisme considère que le cerveau de l'être humain fonctionne comme l'ordinateur ; ce qui explique le recours à l'étude de programmes informatiques symboliques qui peuvent conduire à l'amélioration du fonctionnement du cerveau. L'information traitée est modélisée comme un filtrage des stimuli, puis une formalisation et une représentation mentale et enfin une computation. Ce courant reste limité car le traitement symbolique ne représente pas toute l'activité du cerveau.

Les connexionnistes, à la fin des années 1980, proposent une autre interprétation du cognitivisme. Le cerveau ne fonctionne plus sur un modèle de calcul logique sériel, mais plutôt sur un modèle d'un calcul parallèle [37].

2.1.3 Le constructivisme

Le constructivisme se manifeste tout particulièrement dans la psychologie de Piaget [168], il se manifeste aussi dans certaines recherches de la psychologie sociale, et dans l'école de Palo Alto [8]. Suivant ce courant l'apprenant construit sa façon de comprendre et d'apprendre. Ce courant fondé sur la maturation biologique et la construction progressive de l'individu par l'expérience, par opposition à l'intelligence innée. Par suite, construire un artefact externe amène un apprenant à être plus actif intellectuellement. Cette théorie induit une pédagogie basée sur l'interaction entre l'apprenant et son environnement, versus la transmission d'un savoir pré-construit d'un professeur à un apprenant. Le constructivisme est un courant qui est à la base de la conception de beaucoup de systèmes pédagogiques informatisés existants parce que le support numérique, qui a un caractère dynamique, permet l'application de ses concepts [58]. Les courants constructivistes semblent [158], être devenus les théories éducatives prédominantes. De ces courants naissent la pédagogie active que nous définissons dans la partie suivante.

2.1.4 Le socio-constructivisme

Ce courant met en évidence l'importance des interactions sociales dans le développement de l'individu [209]. L'apprentissage est avant tout un acte social. L'apprentissage va du social vers l'individu ; il se fait en groupe par le biais d'interactions entre pairs. Le processus d'apprentissage est autant individuel que collectif et est la conséquence de deux types de négociation [209]. Le premier avec soi-même par l'interaction entre les structures cognitives élaborées par l'apprenant et la réflexion cognitive qui s'en suit sur ses propres connaissances. Le deuxième avec les personnes de la communauté ou du milieu. Les apprenants ne sont pas des

récepteurs de l'information communiquée, mais doivent être considérés comme des constructeurs actifs de la connaissance dans un contexte de communauté d'apprentissage [209], [168]. Les interactions avec d'autres apprenants sont fondamentales dans le rééquilibrage des principes assimilation/accommodation du modèle socio-constructiviste [168]. Cette interaction fait naître le conflit socio-cognitif [189] et permet la résolution des problèmes de manière collaborative. Ces interactions, intenses et structurées, permettent d'établir une compréhension commune.

L'étude de ces différents courants de pensée psycho-pédagogiques va nous permettre de mieux comprendre la différence entre la pédagogie traditionnelle et la pédagogie active à laquelle appartient, comme nous allons voir dans la suite, la Pédagogie par Projet Collectif (PPC).

2.2 Pédagogie par Projet Collectif (PPC)

2.2.1 Situation de la Pédagogie par Projet Collectif

Pédagogie active et Pédagogie traditionnelle

Différentes pédagogies sont conformes à un ou plusieurs courants de pensée psycho-pédagogiques. Nous distinguons en général deux types de pédagogies : les pédagogies dites "de la transmission, de la connaissance ou de l'empreinte" [7] appelées souvent "pédagogies traditionnelles" qui naissent du béhaviorisme ou du cognitivisme et les "pédagogies actives" naissant du constructivisme et du socio-constructivisme.

Les pédagogies traditionnelles sont centrées sur la transmission des savoirs constitués. Parfois, elles sont nommées "pédagogies transmissives" centrées sur le contenu. A ces pédagogies s'opposent celles des pédagogies dites "pédagogies actives" qui sont un ensemble de propositions et d'actions originales se constituant comme courant innovateur à la fin du XIXe siècle [150]. La pédagogie active est connue sous différentes acceptations : pédagogies nouvelles, éducation progressive, école moderne, fondées sur des méthodes dites "naturelles", "actives", etc.

Dans le cadre de notre thèse, nous choisissons d'utiliser les deux termes : "pédagogies traditionnelles" et "pédagogies actives" pour distinguer ces deux courants pédagogiques. Et plus particulièrement, nous sommes intéressés par les pédagogies actives qui sont définies comme suit : "*Dans la pédagogie active, c'est l'apprenant qui est actif. Il construit son savoir à partir de situations de recherche : avoir à réaliser une tâche, produire quelque chose l'amène à se poser des questions et à découvrir certaines règles, ou gestes dont il a besoin. Il faut pour cela lui faciliter la tâche. L'enseignant (le professeur) est donc celui qui organise la situation d'apprentissage et met à disposition des ressources, c'est le guide et le facilitateur. L'apprenant doit à partir de là agir, tâtonner, faire des hypothèses, les vérifier...*" [85] Selon Meyers et Jones [177], la pédagogie active est compo-

sée de quatre éléments principaux : lire, réfléchir, écrire, et parler/écouter. Elle permet aux apprenants de bien conceptualiser et internaliser le matériel présenté dans la classe.

Lebrun [130] caractérise les pédagogies actives par :

- le caractère personnel de l'apprentissage parce que chaque personne apprend les choses différemment ;
- le rôle catalyseur des connaissances antérieures, ce qui explique la nécessité des prérequis pour qu'un apprenant commence une étape ou une tâche dans un projet ;
- l'importance des "ressources" à disposition, les cours sont transmis sous forme de supports (CD-ROM, etc.), des langages de programmation nécessitent l'installation de quelques environnements ;
- le rôle du contexte et de l'expérience concrète ;
- les compétences de haut niveau à exercer ;
- la "démarche de recherche" dans l'apprentissage ;
- le changement conceptuel (prise de conscience, déséquilibre, reformulation) ;
- le caractère interactif et coopératif de l'apprentissage ;
- le lien entre projet personnel, professionnel, d'études ;
- l'importance d'une construction, d'une production ;
- le rôle de la réflexion sur l'apprentissage qui se passe (le caractère méta).

Les courants de pensée psycho-pédagogiques sont mis en oeuvre en suivant une méthode pédagogique correspondante.

Méthodes actives et méthodes traditionnelles

Une méthode est, étymologiquement, ce qui permet d'accéder au lieu que l'on recherche : c'est un chemin (odos, "route") qui permet de traverser (meta, "travers") [150]. "*La notion de méthode a pu faire et fait toujours l'objet de définitions variables, parfois fort contrastées, au point qu'une étude à part entière pourrait être consacrée à un inventaire ordonnée de ses différentes acceptions*" [36]. Nous choisissons une définition générale réunissant les éléments le plus souvent rencontrés chez différents auteurs. Les éléments constitutifs d'une méthode sont selon [36] :

- un ensemble de moyens ;
- pour atteindre un ou plusieurs buts inscrits dans une visée ;
- à travers des actions organisées et distribuées dans le temps.

Une méthode pédagogique n'est pas seulement un assemblage de techniques et de procédés à appliquer pour produire toujours les mêmes effets [36]. Les méthodes en pédagogie constituent un cadre pour penser et réaliser la pratique éducative en articulant les moyens et les buts dans une organisation à la fois spatiale et temporelle.

Les méthodes traditionnelles, correspondant à la pédagogie traditionnelle, relèvent

d'une philosophie de l'éducation fondée sur l'attention et l'application de l'apprenant [8]. Aux efforts et capacités de l'apprenant à apprendre correspondent les capacités du professeur à exposer, à expliquer et à vérifier les savoirs acquis et les faire réviser. Ces méthodes se caractérisent par leur application dans des classes homogènes, selon les mêmes modalités avec tout le groupe, avec une responsabilité technique appartenant au professeur et une responsabilité morale appartenant à l'apprenant. Les principales caractéristiques des méthodes traditionnelles sont proposées par [8] et [36]. Les méthodes traditionnelles sont celles :

- de l'enseignement frontal c'est-à-dire un professeur face à une classe en rangs alignés. Il s'occupe de transmettre le savoir directement aux apprenants qui sont censés l'enregistrer et l'accumuler ;
- mettant en oeuvre des moyens les plus souvent utilisés dans la tradition pédagogique (tableau noir, dispositifs ou exercices dits "classiques", évaluation sous forme d'interrogations écrites, notation chiffrée) ;
- de l'enseignement collectif, développant l'attention, la mémoire, le raisonnement ;
- du dialogue sous forme d'interrogation-réponses- entre le professeur et les apprenants et exceptionnellement des apprenants entre eux ;
- du silence, de l'obéissance, de l'autorité magistrale, renforcée par les récompenses, les classements par mérite, ou les sanctions.

Les méthodes actives, correspondant à la pédagogie active, sont nées en réaction contre les méthodes traditionnelles dans le but de comprendre et aider les apprenants en difficulté. Des noms de grands pédagogues restent attachés à leur mise en oeuvre, comme ceux de Decroly, Claparède, Ferrière, Freinet, ou Dewey et Cousinet, et plus récemment Rogers [8]. L'expression "méthodes actives" résulte d'un raccourci ; elle signifie aujourd'hui méthodes (et techniques) pédagogiques qui utilisent et/ou qui provoquent l'activité de l'apprenant [7]. Les types de méthodes pédagogiques dites actives selon [130] sont :

- L'Apprentissage Par résolution de Problèmes (APP) ;
- L'Apprentissage Collectif (AC)¹ ;
- La Pédagogie du Projet ou par le Projet (PP).

Ces méthodes ne sont pas exclusives : l'apprentissage par résolution de problèmes peut recourir à certaines formes collectives (travail en équipe) et déboucher sur la production d'un projet personnel.

Puisque dans le cadre de notre thèse nous désirons appliquer la Pédagogie par Projet dans un contexte collectif ², nous étudions alors les deux méthodes actives : Pédagogie par Projet (PP) et Apprentissage Collectif (AC) constituant ce que l'on appelle Pédagogie par Projet Collectif (PPC) (figure 2.1).

¹Lebrun utilise le terme "Apprentissage Coopératif"

²Dans le cadre de notre travail nous retenons le terme de "collectif" pour parler de l'apprentissage collaboratif et coopératif dans la mesure où nous considérons les deux modes d'apprentissage

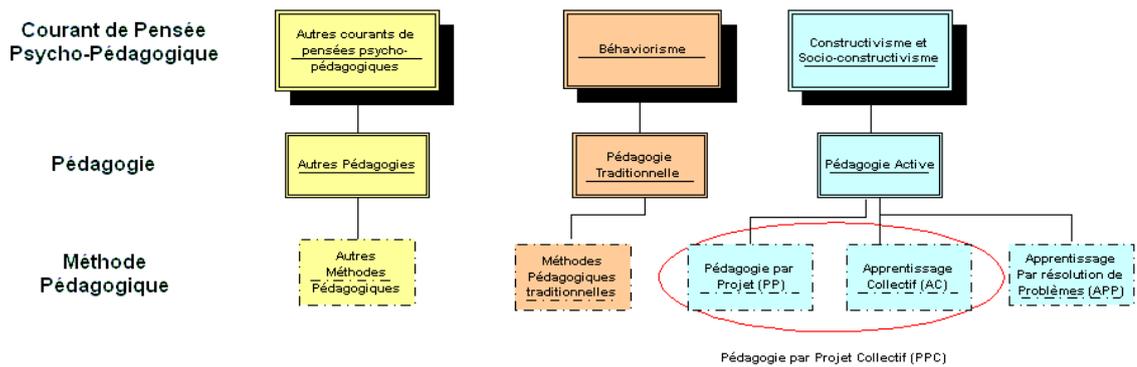


FIG. 2.1: Situation de la Pédagogie par Projet Collectif

2.2.2 Définition de la Pédagogie par Projet Collectif

Certains chercheurs considèrent que la Pédagogie par Projet est de nature collective n'existant que dans un contexte social du groupe [167] [87] [134]. Leroux [134] dit que cette méthode a fait ses preuves depuis longtemps pour permettre l'apprentissage collectif. "Les apprenants travaillent collectivement en petites équipes à partir d'un cahier de charge" [87]. Un projet peut être individuel ou collectif [77]. "Un projet peut être strictement individuel, mais cette situation n'est pas favorable à son développement et reste très exceptionnelle" [7]. Dans la plupart des cas, il est le fruit d'un travail collectif. Boutinet [34] souligne que le projet "qu'il soit individuel ou collectif, il est destiné à s'ouvrir à un espace de négociation dont il ne peut faire l'économie, cet espace favori sera l'émergence d'un imaginaire socialisé capable d'affronter le conseil et la critique d'autrui. En général, les projets comportent plusieurs apprenants travaillant ensemble dans des équipes. Selon Perrenoud [167], un projet oblige à coopérer et collaborer en développant des compétences correspondantes : savoir écouter, formuler des propositions, négocier des compromis, prendre des décisions et s'y tenir. En plus, il permet aux apprenants d'offrir ou demander de l'aide, savoir répartir les tâches et les coordonner ; savoir évaluer en commun l'organisation et l'avancement du travail, gérer ensemble des tensions, des problèmes d'équité ou de reconnaissance, des échecs. Perrenoud [167] va au-delà des compétences ; dans un projet les apprenants prennent conscience de l'importance de l'intelligence collective ou distribuée, de la capacité du groupe de se fixer des buts difficiles à atteindre individuellement. Pratiquement, nous remarquons que la Pédagogie par Projet est souvent appliquée dans un contexte collectif ; L'IUT de Bayonne [141], l'Ecole de Technologie Supérieure (ETS) [173], l'IUT Paul Sabatier, [48], et l'IUT de Calais [196] ont expérimenté des dispositifs de formation à base de projets impliquant des groupes d'apprenants.

Nous ne disons pas que le projet ne peut pas être individuel [87] mais pour nos travaux de recherche, nous nous intéressons à la Pédagogie par Projet dans un cadre collectif. Certains voient la Pédagogie par Projet comme une socio-pédagogie [87]. L'aspect social ici explique l'association presque permanente de l'Apprentissage Collectif avec la Pédagogie par Projet. Nous avons simplement voulu ici mettre l'accent sur les définitions qui émergent le plus dans la littérature traitant la Pédagogie par Projet (PP) et l'Apprentissage Collectif (AC).

Pédagogie par Projet

La Pédagogie par Projet a été initié par le philosophe américain Dewey [66] qui a parlé du "*learning by doing*" (Apprendre par l'action) au début du XXème siècle. La Pédagogie par Projet a été étudiée aussi aux Etats-Unis par Kilpatrick [120]. Cette méthode pédagogique a été étudiée en Europe par certains pédagogues comme Makarenko (1888-1939), Freinet (1896-1966) et le Groupe Français d'Education Nouvelle (1982) [194]. Dans les années 90 avec la croissance rapide de la télématique, la Pédagogie par Projet a été redéfinie pour aider les enseignants dans des tâches comme l'incorporation d'une évaluation authentique, le guidage des apprenants dans des choix de la vie, et la fourniture des expériences qui conviennent aux différents intérêts et capacités des apprenants.

Un projet est alors défini comme une activité qui possède un but précis engageant des personnes et se déroulant dans un environnement social pour résoudre des problèmes réels [120]. Hubert [110] définit le projet comme étant "*une activité pratique planifiée et signifiante, impliquant des recherches, la résolution de problèmes et menant à la réalisation d'un produit concret qui sera socialisé. Cette action appelée "projet" transformerait le milieu, mais aussi l'identité de ses auteurs en produisant des compétences nouvelles*". Il y aura de la pédagogie par projet si, le projet aboutissant sur une production concrète, a pour but de contrôler des problèmes réels et de s'accompagner d'une modification du statut de l'apprenant pour le rendre plus égalitaire à celui de l'enseignant dans la responsabilité de la gestion du projet.

Le terme anglosaxon correspondant à la Pédagogie par Projet est "*Project Based Learning*" (PBL). Le PBL est une méthode qui organise l'apprentissage autour des projets. Ces derniers, basés sur des questions ou des problèmes de défi, impliquent les apprenants dans la conception, la résolution des problèmes, la prise de décision, ou dans des activités de recherche [202].

L'institut de l'éducation (*Buck Institute for Education*, 2003) a donné une définition "standard" de PBL [174] "*a systematic teaching method that engages students in learning knowledge and skills through an extended inquiry process structured around complex, authentic questions and carefully designed products and tasks.*" Cette définition considère le PBL comme une méthode qui engage les apprenants

dans l'apprentissage de la connaissance et de nouvelles compétences, par l'intermédiaire des processus de recherche structurés autour des questions authentiques et des tâches bien déterminées. Cette méthode lie les activités d'apprentissage qui la caractérise avec plusieurs recherches en psychologie cognitive aboutissant à étudier comment les gens apprennent "*How People Learn*" [174].

Synteta [193] traduit l'expression de "*Project Based Learning*" en français par "*apprentissage par projets*" (pour nous c'est la pédagogie par projet) et la définit par :

- Des expériences d'apprentissage engageantes qui incluent les apprenants dans des projets complexes ancrés dans la réalité et à travers lesquels ils développent et appliquent des compétences et du savoir.
- Un apprentissage qui incite les apprenants à avoir recours à de nombreuses sources d'information et disciplines afin de résoudre des problèmes.
- Un apprentissage dont les buts sont identifiés et formulés mais par rapport auxquels les enjeux et les résultats des processus d'acquisition des connaissances ne sont ni prédéterminés, ni entièrement prévisibles.
- Des expériences à travers lesquelles les apprenants apprennent à gérer et à manipuler des ressources telles que le temps et divers matériaux.

Selon George [87], l'essentiel dans cette méthode est que les apprenants travaillent collectivement en petites équipes à partir d'un cahier de charges visant une production effective.

La pédagogie par projet se compose de tâches complexes posant des problèmes aux apprenants. Ces tâches peuvent être la formulation de questions de recherche cohérentes, la définition d'un projet de recherche, la recherche de ressources, l'estimation du temps nécessaire à la réalisation d'une autre tâche, la gestion de ce temps, la collaboration et le *feed-back* ainsi que le suivi du projet [183]. Lors de ce travail, il y a nécessité d'une véritable gestion de projet (gestion du temps, gestion des tâches) [87]. Pour gérer un projet pédagogique, un enseignant peut l'"orchestrer" [183] en plusieurs scénarios séquentiels qui peuvent, à leur tour, être décomposés en plus petites unités ou phases afin d'amener l'apprenant à se concentrer sur des problèmes plus simples et de lui fournir des outils facilitant la tâche.

Pour nous, la Pédagogie par Projet (PP) est une méthode de la pédagogie active possédant un but précis de construction de la connaissance et de la production concrète. Elle engage les apprenants dans un projet composé des étapes et des tâches à accomplir dans le temps tout en utilisant de nombreuses sources d'information, des ressources, des outils et un cahier de charge.

Apprentissage Collectif (AC)

Parce que ce qui nous intéresse c'est le travail en mode projet en groupe de manière collaborative et coopérative, nous utilisons la notion de collectif. Ainsi

nous adoptons le point de vue de George [87] consistant à utiliser l'expression "Apprentissage Collectif" englobant la collaboration et la coopération parce qu'il s'agit pour nous de l'"Apprentissage Coopératif" et de l'"Apprentissage Collaboratif".

Les chercheurs différencient la coopération et la collaboration [87] [17] [106] [35] [159]. La démarche coopérative, plus structurée, réserve à l'enseignant un plus grand contrôle sur l'apprentissage. Par contre, la démarche collaborative propose aux apprenants plus de liberté. La tâche collaborative, contrairement à la tâche coopérative, n'est pas morcelée entre les participants. Tandis que dans une tâche coopérative, et au sein d'une équipe, chaque membre individuellement est responsable de réaliser une sous-tâche. La tâche (coopérative) est complète lorsque tous les membres de l'équipe mettent en commun leurs réalisations [105].

Si nous reprenons la racine du mot collaborer, ("co-labore"), il signifie travailler (labore) ensemble (co) [16] [87]. L'origine du mot coopérer, ("co-operari"), signifie opérer (operari) ensemble (co). Nous trouvons qu'il y a une différence entre les deux mots "Opérer" et "Travailler"; "Opérer" veut dire produire un résultat et "Travailler" ne produit pas nécessairement un résultat.

L'expression "Apprentissage Collaboratif" a une origine anglaise [87]. Les apprenants sont supposés être responsables et posséder des aptitudes sociales. Panitz [159] considère l'apprentissage collaboratif comme une philosophie personnelle et non seulement une méthode pédagogique.

L'apprentissage collaboratif est "*une situation dans laquelle deux ou plusieurs personnes apprennent ou essayent d'apprendre quelque chose ensemble*" [67]. Pour Kaye (1992) cité dans [16], apprendre de manière collaborative, implique que chaque membre du groupe prend des responsabilités et joue un rôle au sein du groupe. Comme la collaboration suppose également une association, chacun accepte alors que le groupe soit plus efficace dans l'ensemble qu'un seul individu. Les apprenants sont responsables de leur apprentissage ainsi que de celui des autres [88]. Dans l'apprentissage collaboratif, l'apprenant a beaucoup de liberté; en fait les tâches ne sont pas très dirigées et ce sont les apprenants qui contrôlent et gèrent leur travail de groupe.

Nous avons retenu dans [106] une définition qui nous semble intéressante et claire dans le sens où elle tient compte de toutes les facettes de l'apprentissage collaboratif parce qu'elle donne des dimensions individuelles et collectives. L'apprenant développe ses idées et ses travaux personnels pour les valider avec les autres qui lui donnent le soutien et la motivation. Cette définition parle de la communication et la coordination, nécessitant des talents de négociation, entre les apprenants. Elle parle aussi de la responsabilité individuelle et de l'autonomie dans l'apprentissage. Enfin, elle n'oublie pas de lier les objectifs personnels de chaque apprenant au but commun du groupe :

"L'apprentissage collaboratif est une démarche active et centrée sur l'apprenant."

Au sein d'un groupe et dans un environnement approprié, l'apprenant exprime ses idées, articule sa pensée, développe ses propres représentations, élabore ses structures cognitives et fait une validation sociale de ses nouvelles connaissances. La démarche collaborative reconnaît les dimensions individuelle et collective de l'apprentissage, encourage l'interaction et exploite les cognitions réparties au sein de l'environnement. Le groupe, acteur principal et ressource première de la collaboration, joue un rôle de soutien et de motivation. Il contribue à l'atteinte, par chaque apprenant, d'un but commun et partagé. La collaboration qui s'y développe est faite de communication entre apprenants, de coordination de leurs actions et d'engagement de chacun face au groupe.

L'apprenant s'engage à travailler avec les membres du groupe en vue de la réalisation du but commun tout en conciliant ses intérêts et ses objectifs personnels. Il collabore dans le cadre des interactions de groupe en partageant ses découvertes. Les échanges avec le groupe et la réalisation d'une tâche collective lui permettent de partager ses découvertes, de négocier le sens à donner à son travail et de valider ses connaissances nouvellement construites. Dans sa démarche, il fait preuve d'autonomie et assume la responsabilité de son apprentissage tout comme il se sent responsable de l'atteinte du but qu'il partage avec tous.

Le groupe est un catalyseur de l'apprentissage. En se donnant un but, en s'engageant dans la réalisation collective d'une tâche, le groupe, en tant que groupe, apprend et construit des connaissances. Ce faisant, il permet à chaque apprenant de se confronter aux autres et il nourrit les apprentissages de chacun. Dans la démarche collaborative, les apprenants collaborent aux apprentissages du groupe alors que le groupe collabore à ceux des apprenants."

L'origine de l'expression "Apprentissage Coopératif" est essentiellement américaine avec les écrits de Dewey [65] mettant l'accent sur la nature sociale de l'apprentissage. Cet apprentissage se caractérise par une interdépendance positive des objectifs des apprenants, avec des responsabilités individuelles [116]. L'apprentissage coopératif est un procédé éducatif dans lequel des petits groupes de 3 à 5 apprenants, constitués intentionnellement, travaillent inter-dépendamment sur une tâche bien définie et structurée. Les apprenants sont responsables de leurs performances. Les rôles des apprenants doivent être assignés, par l'enseignant, de manière à être interdépendants.

L'établissement d'une différence entre une démarche coopérative et une démarche collaborative [35], [159], [106] est fait en fonction du degré d'autonomie des apprenants, de leur maturité et de leur capacité à exercer un contrôle sur leur apprentissage. En fait, parce qu'elle est plus structurée et plus encadrante, la démarche coopérative est davantage utilisée auprès d'apprenants qui n'ont pas élaboré un vaste répertoire de stratégies d'apprentissage. Elle est plus adaptée aux apprenants qui sont moins autonomes ou qui n'ont pas acquis suffisamment de maturité cognitive. A l'inverse, la démarche collaborative, empreinte de plus de souplesse,

convient davantage aux personnes ayant atteint un degré de maturité et d'autonomie plus élevé et pouvant puiser dans leurs expériences antérieures différentes stratégies en lien avec l'apprentissage. Des chercheurs sont portés à considérer la démarche coopérative comme une introduction et une étape préparatoire à la démarche collaborative [16].

Contrairement à la collaboration, la coopération suppose davantage une division et une distribution de tâches entre les membres d'un groupe de travail [106]. Selon [106], cette répartition des tâches nuit aux interactions entre les membres du groupe, car chacun peut réaliser sa tâche indépendamment du groupe. Concernant la collaboration, elle nécessite une coordination et une synchronisation de l'activité de chacun, en vue de la réalisation du but ou du projet commun. Les interactions sociales y sont plus nombreuses puisqu'il faut constamment négocier, partager et échanger pour construire des solutions à la tâche collective à résoudre. Comme nous l'avons signalé au début, nous adoptons le point de vue de George [87] consistant à utiliser l'expression "Apprentissage Collectif" permettant aux apprenants de coopérer et collaborer d'une façon entrelacée dans le temps. L'apprentissage coopératif et l'apprentissage collaboratif sont des formes possibles d'apprentissage collectif [87].

D'un autre côté, Johnson, Johnson et Smith [115] ont caractérisé l'apprentissage collectif par cinq éléments de base :

Interdépendance positive : L'interdépendance positive se rapporte à la création d'une atmosphère d'apprentissage dans laquelle la réussite du groupe dépend de celle de chaque individu dans ce groupe. Assigner simplement une tâche au groupe n'est pas suffisant. Les rôles des membres de groupe doivent être structurés pour stimuler l'interdépendance. Par exemple, Karl Smith donne une technique simple qui consiste à distribuer un seul exemplaire d'un document au lieu d'en faire une copie pour chacun des membres. De cette manière, le groupe est obligé de partager le document, ce qui favorise l'interdépendance entre les membres et par suite la collaboration.

Interaction tête-à-tête : L'interaction tête-à-tête essaye d'engager l'apprenant dans l'explication de son apprentissage à ses collègues. L'idée est d'obliger les apprenants à enseigner l'un l'autre.

Responsabilité individuelle : La responsabilité individuelle aborde la question d'évaluer le travail individuel de l'apprenant dans l'effort de groupe. Elle va plus loin que l'évaluation individuelle. La rétroaction au groupe entier des performances individuelles est une partie critique de responsabilité individuelle. Un exemple est d'inviter aléatoirement un membre de l'équipe à présenter le travail du groupe. Ceci crée la pression sur le groupe de s'assurer que chaque membre du groupe comprend le travail effectué par le groupe.

Compétences de travail collectif : Les compétences de travail collectif se rapportent à la nécessité d'enseigner aux apprenants comment fonctionner dans un

groupe. Ils devraient avoir une compréhension de la dynamique de groupe, de management de conflit, et d'autres compétences sociales nécessaires pour fonctionner efficacement dans un groupe.

Comportement du groupe : il est important de faire engager les apprenants dans un exercice d'auto-évaluation.

Ces cinq éléments aide les apprenants à : donner, recevoir, clarifier, organiser l'information, et également à recevoir la rétroaction, examiner les idées, voir leurs inter-connexions, créer, reconnaître les hypothèses, etc. . .

Les mécanismes inhérents aux systèmes cognitifs individuels ne disparaissent pas dans l'apprentissage collectif car la collaboration et la coopération ne suppriment pas les systèmes cognitifs individuels [108]. Les processus cognitifs et sociaux sont intimement liés, et la causalité entre ces deux catégories de processus est circulaire [108]. Les processus cognitifs qui rendent l'apprentissage collectif efficace ou effectif sont intimement liés aux interactions qui apparaissent entre les individus engagés dans l'apprentissage [212].

2.2.3 Caractéristiques de la Pédagogie par Projet Collectif

Une situation d'apprentissage dans un contexte de PPC présente un certain nombre de caractéristiques. Nous avons retenu les caractéristiques citées dans la thèse de Sébastien George [87] en les enrichissant des analyses de [167], [202], [130], [134], [77] et de nos réflexions :

Initiée à partir d'une thématique concrète de la vie

Elle est initiée à partir d'une thématique concrète de la vie (besoin, manque, souhait d'arriver quelque part, d'aboutir à quelque chose, etc), soit par l'enseignant, soit par les apprenants.

Selon Ward *et al.* [210], les thèmes des projets doivent provoquer la curiosité pour fournir une raison pour l'apprentissage, aider les apprenants à découvrir ce qu'ils font et ne connaissent pas, assurer que les apprenants acquièrent plus de ce qu'ils savaient, créer un besoin et un désir pour acquérir la compétence et la connaissance, mener naturellement à la recherche interdisciplinaire, bâtir de rigides communautés d'apprenants, et mener à la collaboration et la coopération dans le sens le plus fort qui est basé sur la volonté et le désir de réussir. Par exemple, un groupe d'apprenants souhaitant réussir un projet de création de site web pour son association, ou bien ayant un but de produire un logiciel pour le vendre etc. . .

Valeur affective pour l'apprenant

Une situation d'apprentissage dans un contexte de PPC doit avoir un sujet d'étude ou de production ayant une valeur affective pour l'apprenant. Elle engage

affectivement l'apprenant ; cet engagement affectif dans la tâche est primordiale. L'adhésion des apprenants au projet est essentielle. Ils doivent s'approprier le projet qui ne doit pas rester celui de l'enseignant. Ce dernier doit agir comme un médiateur et non pas comme distributeur de savoir ; "*il n'est pas là pour réaliser le projet à la place des apprenants*" [134]. Il intervient dans le projet de sa propre initiative ou suite à la demande de l'apprenant. Il évalue le projet après une présentation publique et donne une notation. Les expériences pratiques de la Pédagogie par Projet Collectif menées par Polman [170] montrent que si les apprenants ne se sentent pas engagés, c'est l'échec. La motivation des apprenants est un facteur clé de la réussite du projet. Elle leur permet de travailler d'une façon plus autonome ainsi que de prendre des initiatives. Il faut donc que le sujet du projet corresponde aux centres d'intérêts des apprenants de façon à ce qu'ils considèrent le projet comme un défi qui les encourage à le relever.

Oeuvre commune

Le projet doit nécessiter une oeuvre commune qui devient un but en soi même. Bruner [125] considère que les oeuvres communes sont importantes dans le travail en groupe des apprenants. Selon lui, "*ces oeuvres collectives à la fois créent et maintiennent la solidarité du groupe ; ... elles sensibilisent à la division du travail nécessaire pour produire des biens*". De cette façon, les apprenants ne cherchent pas à satisfaire les consignes de l'enseignant mais à atteindre l'oeuvre commune.

Techniques de gestion de projet

La situation d'apprentissage de la Pédagogie par Projet Collectif (PPC) se caractérise par le recours aux techniques traditionnelles de gestion de projet. Un projet nécessite de gérer son temps en le décomposant en étapes décomposées elles mêmes en tâches ou actions. La planification des étapes et des tâches est primordiale pour donner un cadre temporel au projet. Les étapes sont définies normalement par l'enseignant mais le calendrier des tâches est négocié avec les apprenants [87].

Des chercheurs utilisent une démarche proche des projets industriels [172]. Cette approche peut être qualifiée de démarche technico-économique [87]. Legrand [131] insiste sur la nécessité d'une planification du projet qui ne soit pas trop rigide. L'activité ou la tâche "*doit être planifiée de façon suffisamment souple pour laisser place à des réorientations chaque fois que cela paraîtra nécessaire après un débat et une prise de décision collégiale*" [131].

Rôle actif des apprenants

La situation d'apprentissage de la PPC doit induire un ensemble de tâches dans lesquelles tous les apprenants peuvent s'impliquer et jouer un rôle actif, qui peut varier en fonction de leurs moyens et intérêts. Les apprenants doivent être amenés à pratiquer la négociation permettant l'explicitation et la socialisation de l'intention d'action et aboutissant à un consensus explicite entre eux et l'enseignant. En plus, une situation d'apprentissage de la PPC incorpore beaucoup plus d'autonomie de l'apprenant, plus de choix, de liberté dans le temps de travail, et de responsabilité que la pédagogie traditionnelle.

Dans un projet, chaque apprenant risque d'être emporté par des options collectives [167] qu'il ne comprend pas ou ne partage pas. Une PPC doit favoriser un double apprentissage :

- l'apprentissage de l'autonomie par rapport au groupe, permettant à l'apprenant de se ménager des zones dans lesquelles il reste maître de son action ;
- l'apprentissage des pratiques de se faire entendre dans un groupe et d'influencer et participer dans les décisions collectives.

Développement des compétences et du savoir

Une PPC doit amener à heurter des obstacles inattendus et à découvrir de nouveaux horizons de la science et de la culture. Elle doit faire appel à diverses connaissances à intégrer et à des compétences d'anticipation, d'innovation, et de création. Elle engage les apprenants dans des projets complexes ancrés dans la réalité à travers lesquels ils développent et appliquent des compétences et du savoir. De cette façon, les apprenants découvrent des mondes "*insoupçonnés*" [167] offrant des entrées dans les savoirs parce qu'ils donnent à voir des pratiques et des compétences qui mobilisent des savoirs. Nous pouvons dire que la PPC permet l'acquisition personnelle de savoirs et savoir-faire nouveaux [130].

Production concrète

Un projet doit aboutir à la production concrète d'un objet au sens large (texte, journal, maquette, carte, expérience scientifique, création artistique ou artisanale, etc.) La Pédagogie par Projet Collectif ne se limite pas au produit final, mais elle se centre sur le processus ou la démarche dynamique [130]. L'application de cette méthode nécessite l'étude de besoin du projet, son analyse, sa conception, son cahier des charges, et sa réalisation [77].

Les apprenants sont impliqués dans une recherche constructive face à un défi

La PPC doit impliquer les apprenants dans une recherche constructive. Cette dernière est un processus à but dirigé qui implique l'investigation, la construction de la connaissance, et la résolution de problème. Les investigations peuvent être

une conception, une prise de décision, un problème trouvé, une résolution d'un problème, une découverte, ou un processus de modélisation. Mais, afin d'être considéré comme une PPC, les tâches centrales du projet doivent produire les apprentissages attendus par la transformation et la construction de la connaissance chez les apprenants [202]. Si les tâches centrales du projet ne représentent aucune difficulté à l'apprenant et peuvent être effectuées en utilisant des informations ou compétences déjà acquises, nous sommes alors dans le cas d'un exercice, et non pas d'une PPC.

Evaluation du projet à plusieurs niveaux

D'une façon générale, avec son approche innovatrice sur l'apprentissage, la PPC exige une approche innovatrice concernant l'évaluation. Elle exige une évaluation diverse et fréquente, y compris l'évaluation de l'enseignant concepteur ³, l'évaluation des apprenants, l'auto-évaluation et la réflexion [193]. "*On percevra, dans chacune des étapes (du projet), la place et l'importance de l'évaluation...*" [130]. L'évaluation d'une étape du projet permet d'un côté l'évaluation et de l'autre la correction de travail des apprenants. Selon Matsuura [145], évaluer un projet c'est évaluer le produit final, les apprenants, et le processus de travail (*Product, Process, People*).

Le projet se termine par une présentation du résultat devant un jury après la réception du ou des produits réalisé(s) [87] [134]. Cette présentation est faite de différentes manières en fonction de la nature du projet et se termine par une notation finale. L'étape de réception de la production des apprenants est importante "*au plan des apprentissages, la réception est une occasion exceptionnelle du point de vue méta-cognitif. Il y a en effet à ce moment là obligation de retour, de reconceptualisation, de reformulation, d'expression de ce que chacun a produit*" [208]. Le projet doit être vu par la collectivité pour qu'il y ait un sens. Cette collectivité apprécie le projet en évaluant la production attendue [131]. Cette évaluation permet aux apprenants d'avoir un retour sur le travail achevé permettant "*un éclairage nouveau sur le travail réalisé dans le projet*" et une obligation d'argumentation des choix effectués [87].

George [87] insiste sur la distinction de l'évaluation du projet de celle des objectifs d'apprentissage. Les objectifs d'apprentissage peuvent être atteints correctement sans avoir terminé le projet, l'inverse est également possible. Dans ce cas, c'est à l'enseignant de juger si les objectifs d'apprentissage qu'il avait fixés ont été atteints ou non.

Le travail dans un projet offre aux apprenants de faire une auto-évaluation "*spon-tanée*" ou "*sollicitée*" [167]. L'auto-évaluation alimente une forme de lucidité qui peut guider de nouveaux apprentissages, mais aussi, tout simplement, permettre

³acteur responsable de structurer le projet en étapes et de les définir ; il est étudié en détail dans le chapitre 4

à chacun de cerner et découvrir ses points forts et ses points faibles afin de mieux choisir ses interventions et son rôle en conséquence.

2.2.4 Pédagogie par Projet Collectif et éducation interdisciplinaire

Dans cette sous-section, nous étudions l'importance de la Pédagogie par Projet Collectif (PPC) dans la formation des apprenants ayant des compétences transversales. Ces compétences leur permettent de travailler avec n'importe qui appartenant à n'importe quel domaine scientifique.

Les disciplines segmentées divisent la connaissance en hiérarchies utiles et organisent des faits et des théories qui dirigent la recherche et apportent l'ordre à notre compréhension [199], [210]. Une des faiblesses principales des disciplines est qu'elles limitent parfois la conception de telle façon qu'un apprenant devient un expert dans sa discipline et voit les choses d'un seul point de vue, mais ne peut pas parler à d'autres (appartenant à d'autres disciplines). Il y a des problèmes qui se posent à notre société qui nécessitent l'étude de diverses disciplines. Résoudre la pollution environnementale, l'épuisement de ressources, et la faim du monde exigent des expertises dans les domaines économiques, mathématiques, physiques, biologiques, chimiques, médicaux, politiques, et sociologiques. Selon Meier *et al.* [147], les apprenants enseignés avec le système à base typiquement disciplinaire n'ont pas pu résoudre les problèmes qui exigent de leur part de faire des relations et des rapports entre les concepts et le contenu.

En revanche, l'enseignement interdisciplinaire commence par une matière, un thème, un problème, ou **un projet** qui exige la participation d'apprenant et la connaissance des disciplines multiples afin d'atteindre une résolution [61], [89], [147], [178], [199], [210]. L'enseignement interdisciplinaire emploie les applications appropriées qui motivent les apprenants à rechercher les faits dont ils ont besoin. Ce type d'enseignement encourage les apprenants à rechercher de nouvelles solutions aux problèmes appropriés en utilisant la connaissance et les ressources disponibles. Il augmente chez eux la réflexion et les compétences de résolution des problèmes tout en augmentant leurs capacités créatrices.

La Pédagogie par Projet Collectif (PPC) permet d'appliquer l'enseignement interdisciplinaire parce que le concept central de la PPC est que les apprenants apprennent en essayant de résoudre des problèmes réels [210].

2.2.5 Efficacité de la Pédagogie par Projet Collectif

De nombreux chercheurs en sciences de l'éducation affirment que la PPC est une méthode efficace pour inciter les apprenants désengagés à devenir motivés et bien engagés [118]. Les apprenants suivant la PPC peuvent ne pas réussir dans les tests à choix multiples comme ceux suivant une méthode de la pédagogie traditionnelle. Des études faites par Norman et Schmidt [153] indiquent une meilleure

conservation à long terme de la connaissance par les apprenants suivant la PPC. La PPC est capable de structurer la connaissance de sorte que l'acquisition soit optimisée, que les apprenants développent des compétences d'auto-apprentissage, et qu'il y ait une augmentation de la motivation pour apprendre [14], [210].

Helic *et al.* [103], [104] insistent sur l'efficacité de la PPC pour les apprenants et également pour les enseignants. Thomas [202] résume l'efficacité de la PPC :

- Gains dans l'accomplissement des apprenants de leurs tâches : Thomas affirme que la performance des diplômés a augmenté en appliquant la PPC. Cette méthode pédagogique a un effet positif sur l'acquisition de compétences importantes au niveau de la réflexion par les apprenants.
- Meilleure compréhension des thèmes : Boaler [30] a fait une étude, qui a duré trois ans rapportée dans la revue *Education Week*, montrant l'efficacité de la PPC. Nous résumons les résultats de cette étude par : "*Students taught with a more traditional, formal, didactic model developed an inert knowledge that they claimed was of no use to them in the real world.*" Au contraire, "*Students taught with a more progressive, open, project-based model developed more flexible and useful forms of knowledge and were able to use this knowledge in a range of settings*" [29].
- Changements dans la résolution collective des problèmes, dans les habitudes de travail, et dans d'autres comportements : dans leur expérience, Tretten et Zachariou [205] déclarent "*Students, working both individually and cooperatively, feel empowered when they use effective work habits and apply critical thinking to solve problems by finding or creating solutions in relevant projects. In this productive work, students learn and/or strengthen their work habits, their critical thinking skills, and their productivity. Throughout this process, students are learning new knowledge, skills and positive attitudes.*"

2.2.6 Relation des enseignants avec les apprenants dans la PPC

La Pédagogie par Projet Collectif (PPC) est une méthode qui transforme l'enseignement de "*teachers telling*" en "*students doing*" [193]. Les enseignants ne sont plus les dispensateurs de savoir. Ils ne sont plus des intermédiaires entre le savoir et les apprenants comme dans les méthodes de la pédagogie traditionnelle (figure 2.2). Mais ils agissent comme des médiateurs, ils sont derrière les apprenants, face à la tâche, pour les aider (figure 2.3) ⁴. Ils sont les concepteurs d'un dispositif d'apprentissage visant l'émergence des projets en régulant l'activité et gérant les conflits.

⁴Figures adaptées de <http://www.ac-versailles.fr/ppcp/fichiers/pedagogie.pdf>

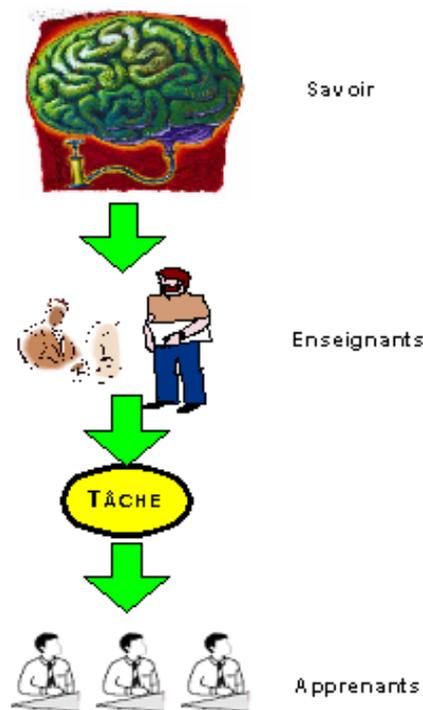


FIG. 2.2: Rôle des apprenants et des enseignants dans les méthodes de la pédagogie traditionnelle

Rôle de l'enseignant

L'enseignant doit être un facilitateur ou un consultant du projet. Dans ce sens, son rôle est celui "d'un régulateur et d'un informateur intervenant à la demande ou de sa propre initiative au fur et à mesure de l'avancement" [131]. Il ne doit pas prendre des décisions à la place des apprenants, mais il doit les aider à atteindre leur objectif final, en les poussant à travailler et réfléchir sur certains points précis, à se focaliser sur le processus de travail et non pas seulement sur la production finale, à prendre du recul par rapport à leur réalisation [87], [134].

La littérature sur la PPC indique l'importance de l'enseignant dans le succès de cette méthode pédagogique. Cette dernière exige des changements chez lui dans l'enseignement, la direction de l'apprentissage, la transmission de la connaissance, la surveillance de l'enseignement, et l'évaluation de l'apprentissage [89], [210]. L'enseignant facilite le développement des projets et agit en tant que conseiller expert. Il doit agir en tant que modèle pensant à haute voix avec les apprenants, et pratiquer le comportement qu'il souhaite voir pratiquer par les apprenants [210]. Il doit garder la dynamique de l'apprentissage, s'assurer qu'aucune phase de l'apprentissage n'est négligée, observer la connaissance de l'apprenant uniformément

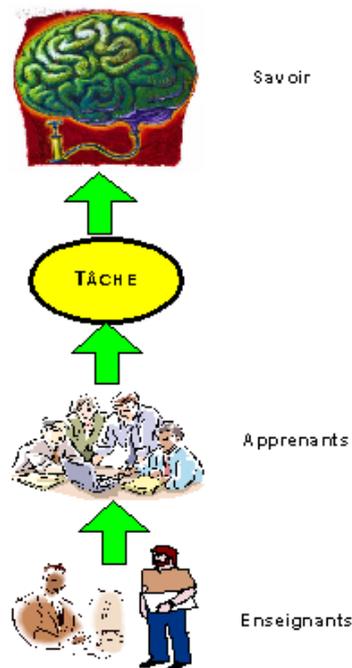


FIG. 2.3: Rôle des apprenants et des enseignants dans la Pédagogie par Projet Collectif

et attentivement, et maintenir l'implication de tous les apprenants dans l'apprentissage [14].

Rôles des apprenants

Le rôle des apprenants est de chercher comment résoudre les problèmes posés dans le projet et de prendre des décisions plutôt que d'être des auditeurs passifs. Le projet doit être considéré comme un défi pour l'apprenant afin de lui donner la motivation ; notion sur laquelle est fondée cette méthode pédagogique. L'apprenant doit apprendre par essais, erreurs et corrections. L'erreur est alors dans la norme, elle fait partie du processus [171].

Après l'exécution d'un projet, les apprenants sont invités à créer une auto-évaluation du projet, comme l'écriture d'un méta-rapport, ce qui leur permet de se concentrer sur leur apprentissage et de voir leur progrès. L'auto-évaluation leur donne un sens de responsabilité de leur apprentissage [193].

2.3 Mepulco-Université

2.3.1 Introduction

Dans le cadre de l'étude de la Pédagogie par Projet Collectif, nous proposons de présenter la pratique du groupe MEPULCO du LIL (laboratoire informatique du littoral) qui a élaboré une méthode d'application de la PPC "Mepulco-Université". Ce groupe a encadré depuis 2002 à nos jours plusieurs dizaines de projets (<http://mepulco.net>). Les projets ont été proposés à des apprenants ayant le niveau de Bac +2 à Bac +5 et appartenant aux cursus officiels de diplômes informatiques délivrés à l'Université du Littoral Côte d'Opale. La plupart des projets étaient des projets d'informatique ou de systèmes d'information, mais la méthode Mepulco-Université pourrait s'appliquer à d'autres disciplines.

Le groupe MEPULCO a fait un bilan des difficultés rencontrées ou bien des insatisfactions qui l'ont amené à mettre en place cette pratique présentée ci-après. Ces insatisfactions selon le groupe MEPULCO sont [196], [195] :

- Peu de documentation, gage de qualité du logiciel informatique, était produite. Cet aspect du travail semblait toujours secondaire pour les apprenants.
- La reprise des travaux par un autre groupe était par conséquent quasi impossible. Il n'était pas rare de voir un même sujet donné plusieurs années de suite sans que soient exploités les résultats de l'année écoulée.
- L'organisation du travail était difficile à mettre en place. La notion de planification était inexistante. La division des responsabilités était souvent confuse. Il était difficile de cerner la part de travail de chacun dans le groupe.
- La documentation du projet était répartie entre les apprenants sans organisation spécifique.

Mepulco-Université est basée sur les expériences de ses membres et tiennent en compte des avancées constantes dans les Technologies de l'Information et de la Communication (TIC). Elle consiste en un ensemble de principes à respecter concernant les échanges entre apprenants et le support, si possible électronique, à ces échanges. Cette pratique a été utilisée sur plusieurs années et a montré son efficacité auprès des apprenants et également les enseignants [196], [195].

2.3.2 Présentation de la méthode Mepulco-Université

Mepulco-Université propose aux apprenants une démarche de réalisation de projets et aux enseignants une assistance à l'encadrement de projets. Elle est proposée sous forme de deux kits : le premier à destination des enseignants et le deuxième à destination des apprenants. *"Le kit contient le matériel méthodologique nécessaire à une gestion de projets en milieu universitaire. Il décrit l'organisation à mettre en place : la démarche (les étapes) du projet en fonction de sa catégorie, l'organisation et le déroulement des réunions, et la répartition des rôles au sein*

de l'équipe. Il donne également des indications pour la rédaction du rapport et la préparation de la soutenance qui clôt le projet." [196], [195]. Le kit contient des dossiers pilotes, des exemples de réalisation, une norme documentaire de rédaction de rapport technique à respecter.

Chaque projet, mené selon les recommandations du kit, utilise un ensemble de techniques et d'outils de réalisation de projets appliqué par les apprenants et un ensemble de techniques et outils de suivi de projets à destination des enseignants afin de les aider dans le suivi et l'encadrement de la réalisation.

Mepulco applique la Pédagogie par Projet Collectif (PPC) en respectant toutes ses caractéristiques. Elle propose des projets réels aux apprenants (développement des sites web, logiciels, etc.), elle engage les apprenants dans les différentes tâches du projet pour leur permettre de s'appropriier le projet. Elle les oblige à documenter les étapes et les tâches du projet dans une oeuvre commune en suivant une organisation spécifique. Les techniques de gestion de projet sont utilisées dans la méthode Mepulco-Université afin d'aider les apprenants à accomplir leurs tâches nécessaire pour une production concrète. Certains projets ayant une finalité "industrielle" c'est-à-dire avec un client du monde industriel. A la fin du projet, les apprenants présentent collectivement leur travail et le produit résultant de ce travail devant un jury composé de leur tuteur et d'un groupe d'enseignants.

Les principes préconisés par la méthode d'application de la PPC Mepulco-Université et exposés dans [196], [195] sont les suivants :

- La répartition des responsabilités.
- La sollicitation régulière de l'équipe.
- Le respect d'une méthode qualité de type PDCA (Plan Do Check Act) étant respectée dans la PPC [36].
- La concrétisation d'un espace de communication et de gestion des contenus produits ou utiles.
- La capitalisation finale.

On trouvera plus de détails sur le site du groupe Mepulco (<http://mepulco.net>), notamment dans le lien "*Informations & Liens*", rubrique "*Méthode Mepulco-Université*".

2.4 Conclusion

Dans ce chapitre nous avons tout d'abord défini et présenté les différents courants de pensée psycho-pédagogiques existants (section 2.1). Nous avons ensuite consacré toute la section 2.2 à la situation de la Pédagogie par Projet Collectif (PPC) comme une méthode de la pédagogie active appartenant au socio-constructivisme. Ceci nous a permis de définir cette méthode comme un regroupement de deux méthodes actives : la Pédagogie par Projet et l'Apprentissage Collectif. Ensuite, nous avons décrit ses caractéristiques, sa relation avec l'édu-

cation interdisciplinaire, son efficacité et la relation entre les enseignants et les apprenants l'appliquant.

Pour donner un aspect pratique à l'explication théorique de la section 2.2, nous avons introduit (section 2.3) la pratique appliquée de cette méthode à l'Université du Littoral Côte d'Opale : **Mepulco-Université**. Cette dernière a pu être expérimentée et validée [196], [195].

La situation d'apprentissage dans un contexte de la PPC a été traitée dans ce chapitre sans s'intéresser à sa transposition dans un EIAH. Elle a été étudiée théoriquement et pratiquement pour permettre de dégager des contraintes pour encadrer la réalisation de support à ces situations dans un contexte médiatisé, d'analyser ces besoins afin d'en proposer des solutions pertinentes pour satisfaire au mieux les besoins qui seront énoncés dans le chapitre suivant.

Chapitre 3

Etude des langages de scénarisation pédagogique

3.1 Introduction

L'approche traditionnelle d'ingénierie des dispositifs instrumentés, appelée aussi approche documentaliste, est essentiellement concentrée sur les mécanismes permettant de faciliter la (ré)ingénierie, la réorientation, et l'utilisation efficace des ressources numériques (documents) existantes pour supporter l'apprentissage et l'enseignement (en fonction de la plate-forme choisie). Une nouvelle approche, s'appelant approche centrée sur les activités, recommande à partir d'un scénario d'apprentissage, d'aboutir à un système d'apprentissage [166]. Activités et rôles sont centraux dans cette nouvelle approche. Cette approche se focalise sur ce que l'apprenant doit réaliser plutôt que sur les objets qu'il doit manipuler [146]. Cette nouvelle approche a eu comme conséquence de focaliser la communauté EIAH sur la notion de scénario. Selon cette nouvelle approche, la description de situations d'apprentissage nécessite la conception et la mise en oeuvre de scénarios pédagogiques. Nous désirons appliquer des situations d'apprentissage dans un contexte de Pédagogie par Projet Collectif (PPC), nous étudions l'élaboration des scénarios pédagogiques dans ce contexte. Pour élaborer des scénarios dans notre contexte de la PPC, nous soulevons le problème de leur expressivité dans des modèles particuliers.

Nous rappelons que l'un des objectifs de cette thèse concerne la proposition d'un méta-modèle dédié à l'élaboration de modèles pour aider à la scénarisation de la Pédagogie par Projet Collectif (PPC). L'objectif de ce chapitre est d'étudier les langages existants actuellement pour décrire tout ou une partie de scénarios de situations d'apprentissage. Cette étude va argumenter notre choix de méta-

modélisation dédiée à la PPC. Nous notons ici que nous voulons désigner par le terme "langage" tout ce qui peut exprimer ou décrire les éléments d'une situation d'apprentissage. Un langage de scénarisation pédagogique peut être formel, semi-formel, écrit généralement en UML (*Unified Modeling Language*) ou même informel, scénario exprimé textuellement.

La conception d'un scénario pédagogique s'inscrit dans un processus global de design pédagogique dont l'une des phases essentielles est l'élaboration du modèle permettant l'expression du scénario en question. Le design pédagogique est défini comme l'ensemble des théories et des modèles permettant de comprendre, d'améliorer et d'appliquer des méthodes d'enseignement favorisant l'apprentissage [160]. Le design pédagogique est une forme d'ingénierie visant l'amélioration des pratiques pédagogiques. La pratique du design pédagogique conduit à un ensemble de plans et de devis regroupant un ensemble de prescriptions qui favorisent l'apprentissage plutôt qu'une description des processus d'apprentissage. Ces plans et devis se déploient sur plusieurs niveaux parce qu'ils dépendent de l'ampleur du problème de formation. Ces différents niveaux peuvent être : un programme de formation, un plan de formation, un cours, une activité d'apprentissage, et un module. C'est à l'intérieur de ce processus que s'insère la création d'un scénario pédagogique. Un modèle pédagogique est donc une description à un niveau abstrait d'une situation d'apprentissage. C'est sur ce modèle que doit être fondé un scénario pédagogique [73].

Dans ce chapitre, nous nous attacherons d'abord (section 3.1) à définir le concept de scénario pédagogique. Nous présenterons ensuite (section 3.2) les principaux langages de modélisation pédagogique EML (*Educational Modeling Language*) et les approches de scénarisation proposées. Enfin, nous terminons ce chapitre par une conclusion (section 3.3) nous amenant à la proposition d'un méta-modèle dédié à la Pédagogie par Projet Collectif (PPC) utilisé dans le chapitre suivant. Un méta-modèle est un langage de modélisation spécifique à un domaine (*Domain Specific Language*), il est utilisé pour exprimer les concepts communs des modèles appartenant à un même domaine. Il est construit à partir de modèles informels, de recommandations dans un langage naturel ou semiformal écrits généralement en UML (*Unified Modeling Language*).

3.2 La scénarisation des activités pédagogiques

3.2.1 Qu'est ce qu'un scénario pédagogique

La notion de scénario prête à discussion dans le contexte éducatif. De la simple préparation de classe à la description détaillée du plan d'une séquence pédagogique, en passant par l'énoncé d'un travail proposé aux apprenants, nombreuses sont les productions des enseignants qui s'apparentent à un "scénario", c'est-à-dire à la prédiction/prescription du déroulement d'une future activité.

Les chercheurs en EIAH s'intéressant à cette notion considèrent souvent qu'un scénario pédagogique est une spécification d'une future activité pédagogique telle qu'elle se déroulera en présentiel ou à distance au moyen des outils informatiques [70]. Un scénario pédagogique décrit la façon particulière dont la méthode pédagogique est utilisée [63]. Nous avons dit dans le premier chapitre que parce que le travail ne justifie pas de distinguer les deux notions, nous utilisons dans la suite de la thèse indifféremment les deux expressions "scénario pédagogique" et "scénario d'apprentissage". Pour Paquette *et al.* [161], un scénario pédagogique est la "*conjonction d'un scénario d'apprentissage et d'un scénario de formation qui lui est associé avec l'expression des liens qui les unissent*". Un scénario d'apprentissage est l'"*ensemble des activités destinées aux apprenants et organisées en un tout cohérent ; à ces activités, on greffe les instruments offerts comme supports aux activités (instruments-intrants) et les instruments à être réalisés par les apprenants (produits)*". Un scénario de formation est l'"*ensemble des activités destinées au formateur et organisées en un tout cohérent ; à ces activités, on greffe les instruments offerts comme supports aux activités (instruments-intrants) et les instruments à être réalisés par le formateur (produits)*".

Pour Pernin *et al.* un scénario pédagogique "*représente la description, effectuée a priori ou a posteriori, du déroulement d'une situation d'apprentissage ou unité d'apprentissage visant l'appropriation d'un ensemble précis de connaissances, en précisant les rôles, les activités ainsi que les ressources de manipulation de connaissances, outils et services nécessaires à la mise en oeuvre des activités.*" [166].

Nous signalons que Pernin *et al.* [166] différencie entre un scénario prédictif "*établi a priori par un concepteur en vue de la mise en place d'une situation d'apprentissage, instrumentée ou non par les technologies numériques. . .*", un scénario prescriptif (prescription d'une activité d'apprentissage aux participants), et un scénario descriptif "*décrivant a posteriori le déroulement effectif d'une situation d'apprentissage, en y incluant en particulier les traces de l'activité des acteurs et leurs productions. . .*"

Pour Charlier *et al.* [49], le scénario pédagogique est défini comme "*le résultat du processus de conception d'une activité d'apprentissage, processus s'inscrivant dans un temps donné et aboutissant à la mise en oeuvre du scénario. Dans un scénario, on trouve donc des objectifs, une planification des activités d'apprentissage, un horaire, une description des tâches des étudiants, des modalités d'évaluation qui sont définis, agencés et organisés au cours d'un processus de design*".

Pour Schneider *et al.* [183], un scénario pédagogique est "*une séquence de phases dans lesquelles les apprenants ont des tâches à effectuer et des rôles spécifiques à jouer.*"

Koper et Tattersall [123] définissent le scénario pédagogique comme concrétisation du savoir didactique des enseignants lors de la réalisation d'un acte pédagogique,

un cours, une séquence, l'ensemble d'un parcours. Par exemple un scénario pédagogique exprimé en IMS-LD est constitué d'un package qui contient :

- un fichier XML "manifest" qui décrit la méthode, la pièce, les actes, les rôles, les activités, l'environnement, les propriétés, les conditions et ou les notifications de la spécification, qui indique en outre les ressources qui lui sont liées
- le groupe de documents ou les ressources mentionnées dans le fichier "manifest" XML [39]

Le scénario est considéré comme une spécification guidant le déroulement de l'activité dans l'environnement informatique pour lequel il a été conçu [70]. Le fait de passer de la simple spécification textuelle qui décrit l'activité vers son déroulement s'effectue par une série de transformations qui opérationnalisent le scénario dans l'environnement informatique cible choisi. Selon Durand [70], l'opérationnalisation du scénario consiste à :

- choisir les participants de l'activité en fonction des critères établis par le scénario ;
- attribuer aux participants les rôles prévus par le scénario ;
- sélectionner les services et contenus dont l'usage est prévu par le scénario dans une infrastructure d'exécution cible.

Généralement, le scénario pédagogique utilise le format XML représentant un système de notation arbitraire. Ce dernier sert à décrire avec des termes indépendants du contexte de son exécution la future activité prévue par le concepteur du scénario. La scénarisation pédagogique nécessite la construction d'un modèle conceptuel de l'activité pédagogique, qui est conforme à un métamodèle regroupant les concepts et les relations inter-concepts utiles à sa description [144].

Pour nous, un scénario pédagogique est un processus de conception d'une activité d'apprentissage permettant de décrire le déroulement d'une situation d'apprentissage dans le temps. Il précise la planification des tâches et leurs objectifs en les affectant aux apprenants. Il décrit les rôles des apprenants et des enseignants, les ressources et outils à utiliser pour la mise en oeuvre de l'activité d'apprentissage.

3.2.2 Conception d'un scénario pédagogique

Pernin [164] caractérise le scénario pédagogique par :

- sa granularité (le niveau de situation auquel il s'intéresse)
- son cycle de vie (l'évolution de son statut dans le temps)
- sa description composée de plusieurs facettes (prescription, suivi-contrôle, connaissances) manipulées par un ou plusieurs types d'acteurs.

En fonction de la granularité (le niveau de la situation d'apprentissage), Pernin décrit trois niveaux de scénarios qui sont :

- **Scénarios de structuration pédagogique** : organiser les unités de structuration pédagogiques telles que les cours, modules, leçons, cursus, crédits.
- **Scénarios d'enchaînement d'activités** : organiser les activités d'un apprenant ou d'un groupe d'apprenants au sein d'une séquence pédagogique, d'une séance, etc.
- **Scénarios de réalisation de tâche** : décrire le déroulement d'une tâche précise (ex. parcourir un hyper-document, résoudre un exercice, manipuler une simulation).

Il décompose le cycle de vie d'un scénario pédagogique en trois étapes :

- déterminer les connaissances et enjeux d'apprentissage et leurs règles d'acquisition afin de concevoir le scénario et la situation d'apprentissage. Pour cela, un scénario doit être exprimé dans un langage de modélisation pédagogique [113] ;
- mettre en oeuvre la situation d'apprentissage décrite par le scénario afin de l'observer dans un dispositif d'exécution [56] ;
- évaluer le scénario pour pouvoir mieux le réutiliser.

Pour conceptualiser un EIAH embarquant un scénario pédagogique, il faut bien prendre en compte la phase de modélisation du scénario dans le processus de conception. Le processus de conception de scénario pédagogique est composé [126], [207], [73] de six étapes :

- **Expression initiale des besoins** : Dans cette étape, l'enseignant initialise le processus par une perception des besoins à satisfaire indépendamment de la méthode qui sera utilisée pour concevoir la situation d'apprentissage qui sera décrite par le scénario. Il décrit de manière informelle les différentes activités du module en fixant des objectifs globaux ou locaux et éventuellement des prérequis pour réaliser le module. Il précise le type de contenu et les ressources pédagogiques nécessaires pour l'accomplissement du module. Cette analyse est menée a priori en amont.
- **Analyse et conception** : Cette étape a pour objectif la formalisation des phases préliminaires de développement du système de formation et de la situation d'apprentissage afin de rendre ce développement plus satisfaisant aux besoins énoncés à l'étape "Expression initiale des besoins". Suite à la description informelle, un ingénieur pédagogique connaissant la particularité de la plate-forme de destination, en collaboration avec l'enseignant, formalise le scénario pédagogique en le découpant en différentes phases reliées entre elles. L'expression de ce modèle est réalisée en utilisant un langage de modélisation pédagogique.
- **Implémentation** : Cette étape consiste à traduire la conception, exprimée par le langage de modélisation pédagogique, en code écrit dans un langage donné (XMI par exemple). Ce code sera interprété par la machine et par

suite le scénario pédagogique est opérationnel.

- **Déploiement** : Tous les éléments développés ou récupérés précédemment sont mis en place sur la plate-forme. Cette étape concerne l’administrateur de la plate-forme de formation. A la fin de cette étape, le module peut alors être référencé.
- **Tests** : Cette étape permet la validité de la conception dans la réalité d’apprentissage concrète et la vérification du comportement de la plate-forme, la cohérence des modèles et la bonne agrégation de tous les composants de la plate-forme. Ces tests sont réalisés par l’ingénieur pédagogique qui assure ainsi la validité d’un point de vue pédagogique et fonctionnel.
- **Evaluation** : cette étape consiste à évaluer le comportement des apprenants et des tuteurs afin de vérifier que le scénario pédagogique prévu initialement correspond bien aux besoins des différents acteurs et détecter les ajustements récurrents qui pourront par la suite être directement intégrés dans le module et ceci dans une démarche d’amélioration continue. Ce qui peut amener à revoir l’expression des besoins et l’analyse et la conception réalisées au départ.

Dans le cadre de nos travaux de recherche, nous considérons qu’une des étapes essentielles de ce cycle de vie est l’étape de conception (2). Cette étape relève d’un processus de modélisation dont l’objectif est l’identification du modèle d’expression d’un scénario [73]. Ce modèle identifie, à un niveau abstrait, la ou les primitives du domaine de l’expertise pédagogique à instancier, et la manière dont sont organisées ces primitives. Il permet une structuration des connaissances pédagogiques décrivant un système d’enseignement, de formation ou d’apprentissage via un scénario pédagogique explicite.

Dans la section suivante, nous étudions plusieurs propositions issues des travaux en EIAH traitant la scénarisation pédagogique.

3.3 Etude des langages de modélisation pédagogiques existants

3.3.1 Introduction

Ces dernières années ont vu émerger différentes normes, standards ou encore spécifications pour décrire les unités d’apprentissage. Le besoin de favoriser l’interopérabilité est derrière l’existence de ces normes et standards qui ont amené à avoir un objectif de consensualisation de la description des situations d’apprentissage. Par exemple, le LOM [139] (*Learning Object Metadata*) est né du besoin d’indexer les ressources pédagogiques. Le LOM est basé sur les langages d’indexation proposés par *Dublin Core*. La naissance de cette norme dans les années 90 avait pour but d’éviter de perdre les ressources et les cours stockés dans des

banques de ressources. Mais ce qu'on a reproché au LOM est qu'il réduit la description de l'unité d'apprentissage à une simple agrégation de ressources. C'est en répondant à cette insuffisance que le SCORM [5] (*Sharable Content Object Reference Model*) est né. Il est considéré comme plus centré sur la Formation A Distance (FAD) via le web car il différencie entre une ressource brute et une unité de structuration pour permettre le contrôle de l'activité de l'apprenant. Il a été critiqué car il permet une structuration des unités d'apprentissage basée seulement sur le contenu, ce qui fait qu'il est uniquement adapté à la pédagogie traditionnelle se basant sur les cours classiques [165].

Des travaux de recherche ont montré que les plates-formes de formation ne peuvent pas se contenter seulement de mettre à disposition des ressources d'apprentissage [126]. Les langages de modélisation pédagogique sont apparus suite à cette insuffisance. Leur but est de permettre de spécifier la gestion des ressources et les activités les utilisant. Ils doivent prendre en compte des notions complexes comme celles d'acteurs, rôles, structures d'activités. Ces langages ont permis de passer d'une vision statique centrée sur les ressources numériques à une vision plus dynamique centrée sur l'activité.

Dans le cadre des langages de modélisation pédagogique basées sur l'activité, le projet CEN/ISSS WS/LT (*Workshop on Learning Technology*) [175] avait comme but de comparer et analyser les candidats au titre de *Educational Modeling Language* (EML). La définition de référence pour l'EML est :

"An EML is a semantic information model and binding, describing the content and process within a "unit of learning from" a pedagogical perspective in order to support reuse and interoperability" [175].

Ces EML ou langages de modélisation pédagogique proposent un modèle d'information (les concepts, leurs relations et leur sémantique) ainsi que la correspondance (*binding*) de ce modèle vers une représentation formelle compréhensible par l'ordinateur (le plus souvent en XML). La plus petite unité définissant un modèle conceptuel est l'unité d'apprentissage (*unit of learning* aussi appelé *unit of study* ou *unit of instruction* selon le contexte ou les interlocuteurs). Nous sommes dans la granularité minimale lorsqu'il est impossible de réduire cette unité à une sous-partie de ses composants sans perdre la sémantique et l'efficacité recherchée vis-à-vis des objectifs pédagogiques.

Une unité d'apprentissage peut être de granularité variée : cours, modules de formation, leçons ainsi qu'un projet, ses étapes, et ses tâches etc. "A 'unit of learning' can be any instructional or learning event of any granularity, e.g. a course, a workshop, a lesson or an informal learning event" [122]. Dans un EML les unités d'apprentissage sont décrites selon l'approche pédagogique utilisée ce qui veut dire que l'EML est indépendant de ces approches pédagogiques. Les modèles désignés par les EML ne doivent pas contenir d'éléments contextuels (comme le type des médias utilisés) afin de les rendre réutilisables pour différentes configurations

techniques.

Dans le projet CEN/ISSS WS/LT deux candidats (PALO et EML-OUNL) ont retenu l'attention parmi six (CDF, EML-OUNL, LMML, PALO, Targeteam, TML / Netquest). Seulement le langage EML-OUNL correspondait au mieux à la définition d'un EML ci-dessus (PALO se limitant à la description de tâches individuelles). Par suite et dans la sous-section 3.3.2, nous présentons le langage IMS-LD : travail résultant de la standardisation faite par le consortium IMS du langage EML-OUNL. Ce langage appartient à l'approche dite neutre car il permet de mettre en oeuvre plusieurs méthodes pédagogiques et par suite permet de décrire n'importe quelle situation d'apprentissage quel que soit le contexte pédagogique.

Une autre approche, n'étant pas neutre, permet de mettre en oeuvre une méthode pédagogique particulière en décrivant une situation d'apprentissage appartenant à un contexte pédagogique particulier. Dans les sous-sections suivantes, 3.3.3 et 3.3.4, nous présentons des langages dédiés qui ont été proposés pour répondre à un domaine spécifique DSL (*Domain Specific Language*). Ce dernier est utilisé pour exprimer les concepts communs des modèles appartenant à un même domaine. Il est construit à partir des modèles informels, de recommandations dans un langage naturel ou semi-formel écrits généralement en UML (*Unified Modeling Language*). Nous étudions dans la sous-section 3.3.3 le langage LDL (*Learning Design Language*) s'appuyant sur un méta-modèle défini pour le CSCW (*Computer supported cooperative work*) [143]. Dans la sous-section 3.3.4, nous étudions le langage de modélisation CPM [126] (*Cooperative Problem-based learning Meta-model*) dédié à l'apprentissage par problèmes (*Problem Based Learning*). Dans la sous-section 3.3.5, nous expliquons l'approche proposée par [119], [73] consistant en une conceptualisation s'appuyant sur des modèles spécifiques au contexte pédagogique.

3.3.2 IMS-LD

Introduction

L'approche dominante sur ce qui se fait actuellement en termes de scénarisation pédagogique est centrée autour de la spécification IMS LD. Le consortium IMS créé en 1997 avait pour objectif de définir et promouvoir des spécifications destinées à faciliter l'interopérabilité des technologies de l'éducation. Parmi les différentes spécifications d'IMS se trouve Learning Design. Cette spécification est issue des travaux de l'Université Ouverte des Pays-Bas portant sur les langages de modélisation de l'apprentissage (*Educational Modelling Language*) [113]. Le but de cette spécification est d'intégrer les préoccupations de conception éducative (*Instructional Design*) dans le mouvement des standards internationaux. La spécification d'IMS-LD se base sur un modèle d'information de conception d'appren-

tissage intégrant les travaux d'EML-OUNL mais également d'autres spécifications existantes du consortium IMS : *IMS Content Packaging*, *IMS Meta-Data* et *IMS Simple Sequencing* [112] [113].

IMS-LD utilise la métaphore théâtrale [39], elle implique la présence de rôles, de ressources et le scénario pédagogique lui-même. IMS-LD intègre une collection de ressources dans une **méthode** (*Method*) spécifiant les aspects dynamiques de l'apprentissage. Un scénario est constitué de plusieurs **pièces** (*Play*) indépendantes les unes des autres et qui sont divisées en un ou plusieurs **actes** (*Act*) et conduites par plusieurs acteurs assumant différents **rôles** (*role_part*) à différents moments. Chaque rôle doit réaliser un certain nombre d'**activités** pour finir le processus d'apprentissage. De plus, tous les rôles doivent être synchronisés à la fin de chaque acte avant de traiter l'acte suivant.

La spécification IMS-Learning Design [112] fait appel à des concepts pédagogiques permettant de modéliser les unités d'apprentissage. IMS-LD possède une certaine flexibilité et par suite prend en compte une grande variété d'approches pédagogiques [39]. Il n'impose pas d'approche pédagogique particulière mais peut être utilisé avec un grand nombre de scénarios et d'approches pédagogiques. IMS-LD est appelé un méta-modèle pédagogique. Nous pouvons modéliser une activité d'apprentissage grâce à la description des différents rôles, activités, environnements, méthodes [113], propriétés, conditions et notifications. L'utilisation d'IMS-LD est importante dans la transformation des plans de cours en unités d'apprentissage décrites de manière formelle et pouvant être exécutées avec un joueur IMS-LD basé sur un moteur tel que Coppercore [56].

Présentation

IMS-LD fournit une infrastructure d'éléments capables de décrire des activités d'enseignement de manière formelle en XML. IMS-LD propose trois niveaux de description d'une activité pédagogique :

- Le niveau A décrit le vocabulaire au coeur de la spécification afin de concevoir des unités d'apprentissage supportant diverses pédagogies ; le niveau A inclut la définition de la "méthode" [39], des pièces, des actes, des partitions, des activités d'apprentissage et des environnements de tutorat. Ce niveau décrit le vocabulaire clé de la spécification afin de concevoir des unités d'apprentissage, il contient la description des éléments qui configurent IMS-LD et la coordination entre eux.
- Le niveau B ajoute au niveau A de nouveaux concepts permettant une personnalisation ainsi que des scénarios plus raffinés. Ces concepts sont ceux de propriétés variables (*properties*) et de conditions (*conditions*). Pour qu'une activité soit considérée comme achevée, il faut accomplir une propriété. En plus, l'accomplissement d'une activité peut influencer une propriété.

- Le niveau C ajoute des notifications (*notifications*) permettant d'étendre la dynamique des scénarios construits [126]. Ces notifications permettent outre la notification automatique (courrier électronique) le déclenchement d'une propriété ou une condition.

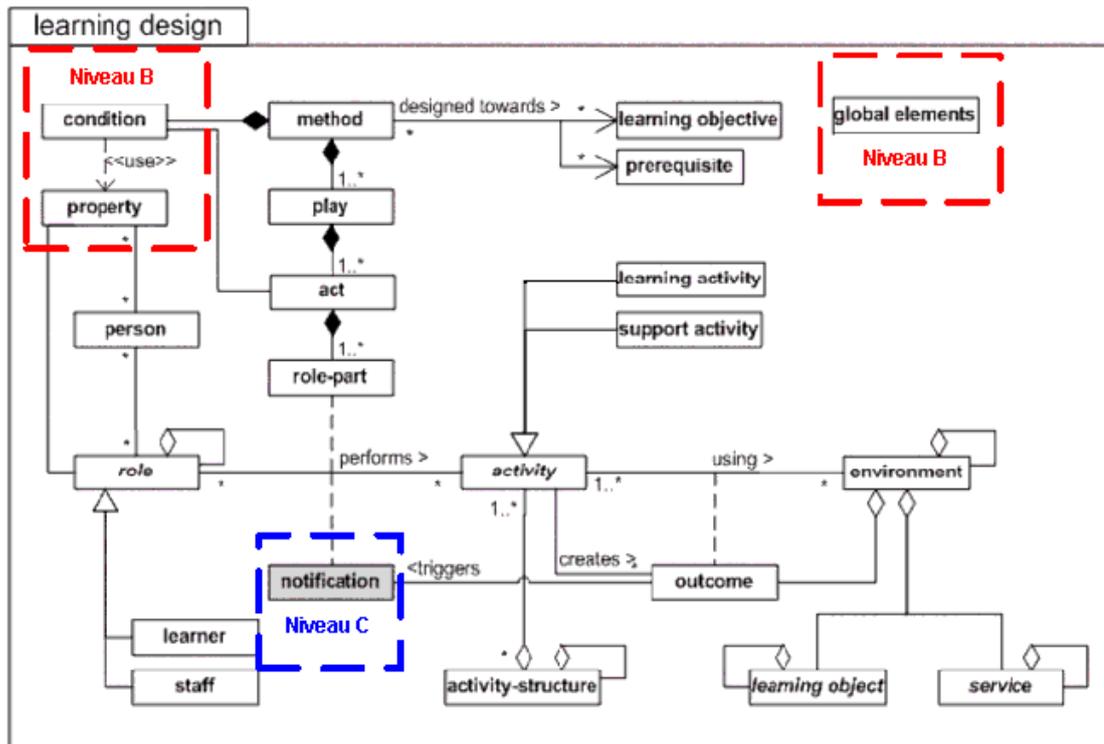


FIG. 3.1: Architecture de la spécification IMS-LD, Niveaux A, B, et C. Source [113] annotée

Dans la Figure 3.1, une personne (*person*) joue des rôles (*role*) dans le processus d'apprentissage/enseignement. Le rôle est soit un rôle d'apprenant (*learner*) ou d'encadrement (*staff*). La personne doit achever des résultats (*outcomes*) en jouant ce rôle et cela par la réalisation des activités d'apprentissage (*learning activity*) ou de soutien (*support activity*) plus ou moins structurées (structures d'activités (*activity-structure*)). Les activités d'apprentissage (*learning activity*) sont dirigées par des objectifs à atteindre. Elles sont constituées principalement d'une description du travail à réaliser et de liens avec un ou plusieurs environnements de travail. Tandis que les activités de support (*support activity*) peuvent par exemple être des activités de correction de devoir ou d'aide pour des apprenants en difficultés. Les activités peuvent être assemblées dans des structures d'activités

(*activity-structure*). La réalisation de ces activités se fait dans le contexte d'un environnement (*environment*) qui est constitué d'objets d'apprentissage (*learning object*) et de services (*services*) utilisés pendant la réalisation des activités associées à cet environnement [113].

L'élément méthode (*method*) permet de scénariser l'activité à partir des descriptions fournies par les objets définis dans le paragraphe précédent. C'est à ce niveau là que nous pouvons remarquer la singularité d'IMS-LD. Cette spécification utilise de la métaphore théâtrale pour décrire le déroulement de l'activité. Une méthode est constituée d'une ou de plusieurs pièces de théâtre (*play*) concurrentes (se réalisent parallèlement). Une *play* est constituée d'un ou plusieurs actes (*act*) séquentiels. Un acte est composé d'un ou de plusieurs *role-part* concurrents permettant l'association d'un rôle à une activité ou à une structure d'activité [113]. La méthode (*method*) détermine ou définit quel rôle assurera quelles activités à tel moment du processus. L'utilisation de la méthode est nécessaire pour atteindre des objectifs d'apprentissage (*learning objectives*) qui correspondent à la spécification de résultats pour l'apprenant. Cette méthode est décrite sur l'hypothèse de certains pré-requis (*prerequisites*) correspondant à la spécification des pré-requis d'entrée pour les apprenants.

Une méthode peut contenir des conditions (*condition*) ou règles permettant de raffiner la visibilité des activités et des environnements pour les personnes ou rôles. Ces conditions définissent des expressions booléennes sur les propriétés de ces personnes ou rôles. Une propriété peut être regroupée dans des groupes de propriétés (*property-groups*) qui sont un exemple de propriétés globales (*global elements*) pouvant être définies. Les propriétés peuvent être de différents types, locales comme globales pour les personnes comme pour les rôles.

Le mécanisme de notification (*notification*), appartenant au niveau C, permet également la précision de certains détails du processus de l'apprentissage, par exemple avertir l'enseignant de l'arrêt d'une activité.

Le résultat peut entraîner la génération d'une notification produisant une nouvelle activité disponible pour un rôle donné. Par exemple, la fin d'une activité d'un apprenant (atteindre un résultat) peut déclencher pour un autre apprenant ou même un enseignant une nouvelle activité en conséquence.

Il est possible de créer des rôles (*role*) de deux types : apprenant (*learner*) et encadrement (*staff*). *Learner* étant réservé aux rôles joués par les apprenants et *staff* aux managers de l'activité. Il est possible d'imbriquer les rôles [113].

Scénariser une situation de la PPC avec IMS-LD

Après cette étude non contextuelle de la spécification IMS-LD, nous essayons maintenant de l'étudier dans le contexte de la Pédagogie par Projet Collectif (PPC) et de voir jusqu'à quel degré cette spécification répond aux besoins de l'enseignant concepteur de la PPC. Pour mener cette étude, nous utilisons un

scénario PPC écrit par Hassina El-Kechaï dans le cadre de son Master recherche à l'Université du Maine comme cas d'étude. Ce scénario est un scénario PPC qui a été élaboré en utilisant la spécification IMS-LD. Il décrit un projet pour les apprenants ayant une finalité, le développement d'un site web. Ce projet est composé des étapes suivantes : Définition du projet, Conception Générale, Conception Détaillée, Réalisation du site, et Mise en ligne du site. Il avait des objectifs d'apprentissage globaux liés à l'apprentissage de la gestion de projets et des objectifs d'apprentissage globaux liés à l'apprentissage du travail collectif.

Dans ce qui suit, nous prenons des schémas XML de ce scénario correspondant au schéma IMS-LD ¹ tels qu'ils sont pour voir comment IMS-LD a été utilisé pour élaborer un scénario PPC. Nous notons que l'auteur du scénario n'est pas un simple enseignant concepteur de Pédagogie par Projet Collectif mais aussi un chercheur dans le domaine des EIAH.

Acteurs et rôles : Nous commençons par l'étude de la définition des acteurs dans le sens de rôles (figure 3.2). Nous savons que dans la PPC, il existe plusieurs types d'acteurs ² (ex : enseignant concepteur, tuteur, expert, apprenant, etc...) jouant plusieurs rôles. Dans IMS-LD, il n'existe que les concepts "*Learner*" et "*Staff*". Par exemple, pour définir le rôle "chef du projet", l'auteur du scénario est obligé d'utiliser le concept "*Learner*" ce qui n'est ni évident et ni logique pour un simple enseignant concepteur de la PPC. Pour définir le tuteur, le concepteur

```

<!-- ROLES -->
- <imsld:roles>
  <!-- ACTEURS HUMAINS APPARTENANT AUX DIFFERENTS PROCESSUS -->
  <!-- ACTEURS "LEARNER" APPARTENANT AUX DIFFERENTS PROCESSUS -->
- <imsld:learner identifier="R-learner">
  <imsld:title>Apprenant</imsld:title>
- <imsld:learner identifier="R1-learner">
  <imsld:title>Responsable du projet (Chef de projet)</imsld:title>
</imsld:learner>
- <imsld:learner identifier="R2-learner">
  <imsld:title>Responsable technique</imsld:title>
</imsld:learner>

```

FIG. 3.2: Définition des rôles dans la PPC en IMS-LD

et l'enseignant (figure 3.3), l'auteur du scénario a utilisé un seul concept "*Staff*" en spécifiant que l'acteur est tuteur (et/ou concepteur, et/ou enseignant) aux niveaux des attributs du concept (identificateur et titre) et non pas en utilisant des concepts différents (inexistence des concepts "concepteur", "tuteur", etc.). C'est vrai que la finalité est atteinte mais conceptuellement nous pensons que ce n'est

¹Modéliser un scénario pédagogique consiste à produire un fichier XML respectant le schéma IMS-LD

²les différents acteurs de la PPC sont étudiés dans le chapitre suivant

pas évident et facile pour un simple enseignant concepteur de la PPC.

Tâches et étapes d'un projet : Pour la création des tâches (figure 3.4) consti-

```
<!-- ACTEURS "STAFF" APPARTENANT AUX DIFFERENTS PROCESSUS -->
- <imsl:staff identifiant="R-tuteur">
  <imsl:title>Tuteur</imsl:title>
</imsl:staff>
- <imsl:staff identifiant="R-designer">
  <imsl:title>Concepteur</imsl:title>
</imsl:staff>
- <imsl:staff identifiant="R-enseignant">
  <imsl:title>Enseignant</imsl:title>
</imsl:staff>
</imsl:roles>
```

FIG. 3.3: Définition des rôles dans la PPC en IMS-LD

tuant les étapes du projet ³, l'auteur du scénario est obligé d'utiliser le concept "*Activity*" et plus précisément de choisir entre "*learning activity*" (activité d'apprentissage) et "*support activity*" (activité de soutien) alors que dans la Pédagogie par Projet Collectif (PPC) les tâches sont catégorisées comme tâches individuelles ou tâches collectives [87], [134] et une même tâche est vue comme une tâche de support et d'apprentissage en même temps. Le but d'un projet est d'accomplir des tâches pour apprendre, produire, et travailler ensemble (tâches de support). A ce niveau là, nous notons qu'IMS-LD est insuffisant pour répondre à la spécificité de la PPC. Un enseignant concepteur de la PPC ayant l'habitude de travailler avec les concepts : Projet, Etape et Tâche est obligé de les remplacer par les deux concepts "*learning activity*" et "*support activity*" avec IMS-LD, chose qui n'est pas intuitive à faire et nécessitant une bonne compréhension de la spécification IMS-LD.

L'aspect dynamique : Sans vouloir remettre en cause IMS-LD, nous lui reprochons d'un point de vue PPC de définir les concepts séparément (par exemple définir les environnements à part, puis les rôles et enfin les services) et utilise des identificateurs pour référencer leurs instances dans le concept "*method*" [10] intégrant une collection de ressources spécifiant l'aspect dynamique. L'enseignant concepteur de la PPC a besoin d'un langage contenant tous les composants de la PPC dans une hiérarchie simple et employant des textes libres pour décrire simplement et facilement chaque concept. Par exemple pour définir un projet, il vaut mieux définir ses étapes, les objectifs d'apprentissage de chaque étape, etc. puis pour chaque étape les tâches à faire et puis dans chaque tâche définir les acteurs concernés et leurs rôles, les outils à utiliser etc. Par contre, IMS-LD pour

³Nous allons voir en détail que le projet est constitué des étapes constituées des tâches dans le chapitre suivant

```

<!-- ACTIVITIES -->
- <imsld:activities>
  <!-- LEARNING ACTIVITIES -->
  <!-- ETAPE 1 -->
  <!-- ACTIVITE 1: ETUDE PREALABLE -->
- <imsld:learning-activity identifier="LA1">
  <imsld:title>Etude préalable</imsld:title>
- <imsld:learning-objectives>
  - <imsld:item identifierref=" identifier="LA1-obj">
    <imsld:title>Savoir identifier la cible (public) et les besoins, estimer le niveau d'équipement des futurs internautes et surtout définir les
    objectifs généraux et le type de site qu'il faudra mettre en oeuvre</imsld:title>
    </imsld:item>
  </imsld:learning-objectives>
  <imsld:environment-ref ref="S2-messagerie" />
  <imsld:environment-ref ref="S3-chat" />
  <imsld:environment-ref ref="S4-forum" />
  <imsld:environment-ref ref="LO1-Etude-prealable.html" />
  <imsld:activity-description>Cette étude, dite préalable ou préliminaire, peut s'appuyer sur divers éléments: Des analyses de faisabilité (qui
  justifient le lancement du projet en fonction des ressources de l'entreprise) Des études de marché (qui positionnent le futur site en fonction
  de l'existant) Eventuellement, des études de concurrence (qui donnent un panorama de ce qui a été déjà fait en la matière par d'autres
  entreprises</imsld:activity-description>
- <imsld:complete-activity>
  <imsld:user-choice />
</imsld:complete-activity>
</imsld:learning-activity>

```

FIG. 3.4: Définition des tâches de la PPC en IMS-LD

décrire l'enchaînement d'activités, a recours à la métaphore du théâtre utilisée de manière artificielle et qui impose un niveau de structuration différent de celui de la Pédagogie par Projet Collectif (activités d'apprentissage (*learning activity*) et de support (*support activity*) d'IMS-LD assemblées dans des structures d'activités (*activity-structure*) ont une structuration différente des tâches assemblées dans des étapes qui sont elles-mêmes assemblées dans un projet). Le projet est dynamique dans sa nature parce que le fait de définir des étapes avec un plan de tâches pour chaque étape, **dans une hiérarchie simple**, donne au projet l'aspect dynamique. C'est vrai que l'absence de références dans un langage contenant tous les composants de la PPC dans une hiérarchie simple signifie que les concepts doivent être répétés s'ils doivent être employés plus qu'une fois dans le projet [10]. Malgré la perte dans la concision, cela nous permet de gagner plus de compréhensibilité de la part des enseignants concepteurs de la PPC [10]. Par exemple figure 3.4, l'auteur est obligé de définir les activités "*activity*" (dans le sens de tâches de projet) et les structures d'activités "*activity-structure*" (dans le sens d'étapes) à part, et après mettre leurs références dans une métaphore théâtrale. L'utilisation de la métaphore théâtrale consiste à décrire le déroulement de l'activité. L'élément "*method*" contient différentes séquences (*play*). Chaque séquence est constituée d'un ou plusieurs actes (*act*). Ce n'est pas évident pour un enseignant-concepteur en PPC d'ajouter de nouveaux concepts dynamiques regroupant les références d'autres concepts (Projet, Etape et Tâche) : après la définition des tâches, des étapes etc... à part, l'enseignant concepteur est obligé de définir dans une méthode "*method*" puis une "*play*" les actes alors que le fait de seulement numéroter les étapes reflète un certain dynamisme.

```
<!-- ACTIVITY STRUCTURES -->
<!-- STRUCTURE 1 Conception détaillée du site (Etape3) -->
- <imsld:activity-structure identifiant="AS1-Conception-detaillée-site" structure-type="sequence">
  <imsld:title>Conception détaillée du site</imsld:title>
  <imsld:environment-ref ref="S2-messagerie" />
  <imsld:environment-ref ref="S3-chat" />
  <imsld:environment-ref ref="S4-forum" />
  <imsld:environment-ref ref="L08-Conception-maquette.html" />
  <imsld:environment-ref ref="L015-outcomes-ress-Plan-projet.doc" />
  <imsld:environment-ref ref="L016-outcomes-ress-Conception-generale.doc" />
  <imsld:environment-ref ref="L09-Conception-détaillée-site.zip" />
  <imsld:environment-ref ref="L017-outcomes-ress-Aspect-site.doc" />
  <imsld:learning-activity-ref ref="LA12" />
  <imsld:learning-activity-ref ref="LA13" />
  <imsld:learning-activity-ref ref="LA14" />
</imsld:activity-structure>
```

FIG. 3.5: Aspect dynamique

Discussion

IMS-LD est un standard où la place de la modélisation UML est prédominante [42]. La modélisation d'un scénario pédagogique consiste donc à produire un fichier XML respectant le schéma IMS-LD représentant le méta-modèle UML du standard. Cette tâche n'est pas simple pour un enseignant concepteur [42]. L'étude dans le contexte de la Pédagogie par Projet Collectif (PPC) que l'on a faite pour étudier l'expressivité d'IMS-LD et les recherches effectuées nous ont amené à conclure que LD n'est pas adapté ni suffisant pour le côté PPC, celui-ci relève à notre sens d'une approche qui nous semble trop généraliste parce que ces langages restent limités même avec leurs extensions "*these extensions still have some limitations*" surtout pour des situations d'apprentissage collectif "*we have found some limitations in reflecting learning experiences that are group-based*" [133].

En effet, un langage généraliste (comme IMS-LD et ses extensions) aura tendance à proposer des spécifications qui traiteront certes correctement le problème, mais pas de la manière la plus satisfaisante. Le scénario PPC que nous avons utilisé ci-dessus pour étudier IMS-LD dans le contexte de la PPC montre que pour un expert dans le domaine des EIAH, IMS-LD traite le problème et permet de scénariser une situation d'apprentissage de la PPC mais d'une manière parfois artificielle. "*It is not always easy to model a collaborative learning situation using IMS-LD level B and C*" [143].

Nous résumons les critiques à ce standard (IMS-LD) par :

- Premièrement, nous critiquons la difficulté pour l'enseignant concepteur non spécialisé dans ce langage de formuler ses besoins [39], [143], de modéliser "*dans un langage a priori très complexe et éloigné des pratiques enseignants.*" [42].
- Deuxièmement, nous critiquons la difficulté de traduire des situations d'apprentissage de nature collective (en ce qui nous concerne c'est la Pédagogie par Projet Collectif (PPC)) à l'aide du langage IMS-LD ; d'où la proposition des extensions citées ci-dessous pour répondre à l'aspect collectif de l'apprentissage mais qui nuisent à la standardisation voulue [42].
- Troisièmement, nous critiquons la difficulté de traduire des situations d'apprentissage de nature dynamique (définir les concepts séparément et utiliser des identificateurs pour référencer leurs instances dans le concept "*method*").

Pour répondre à ces critiques, des chercheurs ont proposé des extensions afin de mieux préciser le contexte pédagogique et mieux répondre à des intentions de conception dans un contexte particulier. Ces extensions portent sur la mise au point de gabarit ou de pattern [133], [107], des extensions appelées "*modèle d'organisation du tutorat*" décrivant une activité d'accompagnement des apprenants et spécifiant les rôles des acteurs participant à cette activité [91]. D'autres chercheurs ont choisi de proposer un méta-modèle métier ; ils ont proposé un langage de modélisation spécifique à un domaine (*Domain Specific Language*). Le méta-modèle résultant sert à exprimer les concepts communs des modèles appartenant à un même domaine. Il est construit à partir des modèles informels, de recommandations dans un langage naturel ou semi-formel écrits généralement en UML (*Unified Modeling Language*).

Pierre Laforcade [126] montre la difficulté de modéliser une situation d'apprentissage dans le contexte d'apprentissage par problème (*Problem Based Learning* PBL) en IMS-LD. Il propose la modélisation d'un scénario pédagogique de PBL à l'aide du langage CPM (*Cooperative PBL Meta-model*).

Christian Martel propose un méta-modèle LDL (*Learning Design Language*) [143] permettant d'articuler des aspects organisationnels et sociaux tels qu'ils sont supportés par une plate-forme de formation.

La difficulté d'IMS-LD, l'insuffisance des extensions proposées à ce standard, et la faisabilité et la validité de propositions de langages dédiés à des domaines (méta-modèles métiers) nous pousse à proposer un méta-modèle dédié à la Pédagogie par Projet Collectif (PPC). Avant d'expliquer l'élaboration de ce méta-modèle dans le chapitre suivant, nous étudions les deux langages : LDL (*Learning Design Language*) et CPM (*Cooperative PBL Meta-model*). L'étude de ces deux langages nous intéresse dans le sens qu'elle nous permet d'étudier et de comprendre des exemples existants de propositions des méta-modèles dédiés à des domaines précis.

3.3.3 LDL

Introduction

Le laboratoire SysCom⁴ (Systèmes Communicants) de l'Université de Savoie, en collaboration avec l'équipe ARCADE⁵ (Ateliers de Réalisation et de Conception d'Applications Destinées à l'Éducation) du CLIPS-IMAG de l'Université de Grenoble et la société Pentilla proposent d'aborder la modélisation de scénario pédagogique dans le cadre du projet SceClasse. Leur but dans ce projet est de fournir aux enseignants concepteurs les moyens de produire des scénarios collaboratifs pour leurs classes, sous une forme réutilisable et partageable, mais aussi opérationnalisable au sein d'un ENT. Le résultat de ce projet était la proposition d'un langage pour exprimer ces scénarios : LDL (*Learning Design Language*) s'appuyant sur un méta-modèle défini pour le CSCW (*Computer supported cooperative work*), et une infrastructure pour se charger de l'opérationnalisation des scénarios : LDI (*Learning Design Infrastructure*) [82]. Ce langage repose sur un paradigme organisationnel particulier appelé modèle de participation [142].

Présentation

LDL est un EML s'appuyant sur un méta-modèle défini pour le *Computer Supported Cooperative Work* (CSCW) et composé de sept concepts principaux (Figure 3.6) constituant un scénario pédagogique [143] [144] :

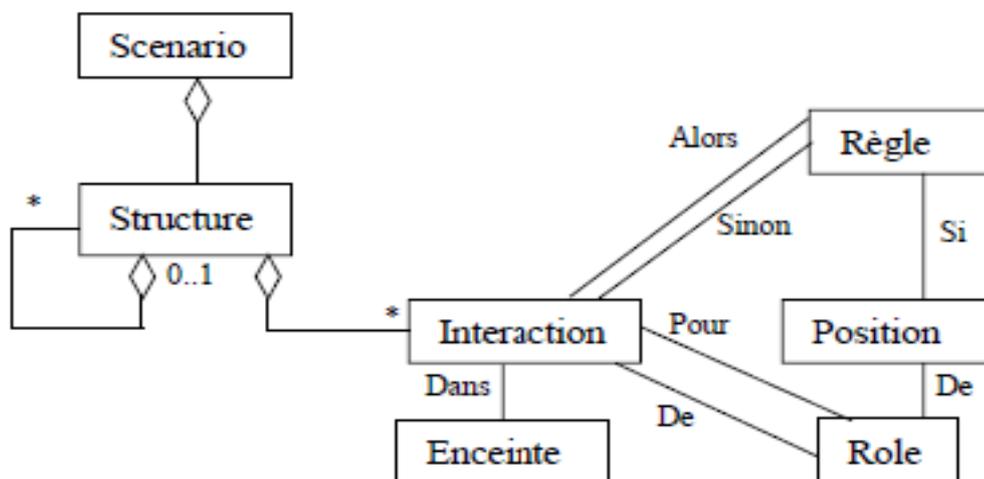


FIG. 3.6: Méta-modèle simplifié LDL

⁴<http://www.syscom.univ-savoie.fr/>

⁵<http://www-clips.imag.fr/arcade/>

- **Structure** d'activité : les structures d'activités décrivent l'enchaînement des interactions entre les acteurs.
- **Interaction** : une interaction est définie par l'action d'échange entre les acteurs durant l'activité pédagogique. Elle est un schéma d'action entre deux ou plusieurs acteurs. Il s'agit de la brique de base principale des scénarios.
- **Enceinte** : l'enceinte est le lieu de déroulement des interactions. Elle peut être de type contenu (ex : un cours, un exercice, etc.), ou de type service (ex : forum, chat, etc.). Il est possible d'inclure une enceinte au sein d'autres enceintes si leur type le permet.
- **Rôle** : les rôles sont définis par les interactions que les acteurs ont entre eux.
- **Règles** : elles sont utilisées pour définir le début et la fin des conditions d'interactions et de structures. Elles permettent lors de leur application de vérifier la complétion des actions, l'état des positions au cours de l'activité, l'adaptation des activités d'apprentissage en fonction des positions des participants, etc.
- **Position** : une position traduit le point de vue d'un acteur sur un autre acteur ou sur une de ses productions. Elle recouvre des notions différentes telles que : point de vue des acteurs, de sa disponibilité, de la difficulté d'une activité, d'une marque, etc.
- **Observables** : ce sont les valeurs sélectionnées par le scénario résultant de l'observation de ses différents composants.

Si nous prenons le même scénario PPC étudié dans la sous-section 3.3.1, nous remarquons que LDL n'est pas facile à utiliser pour scénariser une situation d'apprentissage de la Pédagogie par Projet Collectif. Pour définir le projet, les étapes, et les tâches, nous sommes obligés d'utiliser des concepts comme "Scénario" (pour décrire le projet), "Structure" (pour décrire les étapes), "Interaction" (pour décrire les tâches collectives). Il nous semble que les concepts de LDL sont loin de la spécificité de la Pédagogie par Projet Collectif (PPC) car nous considérons que le projet possède un niveau de décomposition très spécifique et nous pensons que nous ne pouvons pas remplacer ses concepts par des concepts génériques comme ceux de LDL.

Discussion

LDL s'appuie sur un méta-modèle qui permet la représentation de diverses situations et modèles d'activités, notamment collaboratives. Les concepts de ce méta-modèle permettent de représenter des activités très différentes. Le concept "position" permet d'articuler les activités de nature différente mais en relation de dépendance les unes des autres. Cette notion de position est comparable à la notion de propriété du langage IMS-LD [70]. Le concept de position sert à exprimer des résultats de l'apprenant durant l'activité pédagogique. Ces résultats sont pro-

duits par un ou plusieurs apprenants dans une enceinte. Mais, dans une optique de Pédagogie par Projet Collectif (PPC), nous considérons que LDL est générique et ne peut pas répondre à la spécificité de la PPC. Les concepts : Projet, Etape, Tâche Individuelle, Tâche Collective etc. ne peuvent pas être remplacées facilement et simplement par les concepts : Scénario, Structure, Enceinte, Interaction etc. Cette analyse ne remet pas du tout en cause le langage LDL. Par contre, nous pensons que LDL répond bien aux aspects collaboratifs puisque ses concepts : "Interaction" et "Position" traitent les descriptions quantitatives et qualitatives de la collaboration.

3.3.4 CPM

Introduction

Le LIUPPA (Laboratoire Informatique de l'Université de Pau et des Pays de l'Adour) a développé le langage de modélisation, CPM (*Cooperative Problem-based learning Meta-model*) définissant un profil UML. Ce langage est un langage basé sur UML visant à modéliser, en amont, des situations d'apprentissage par résolution de problèmes [126], [128], [152].

CPM est composé de [126] [128] :

- une syntaxe abstraite : représentée par le méta-modèle CPM qui définit les concepts et leurs relations ;
- une syntaxe concrète : représentée par le profil CPM définissant la notation des concepts et 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).

Présentation

CPM est un profil UML permettant à une équipe pluri-disciplinaire de définir de manière itérative une activité d'apprentissage de type PBL (*Problem Based Learning*). CPM s'intéresse aux phases d'analyse et de conception jusqu'à l'implémentation via le langage IMS-LD d'une situation d'apprentissage. Le langage CPM repose sur un méta-modèle dédié à l'apprentissage par résolution de problèmes. Ce méta-modèle conceptuel résume l'idée selon laquelle une personne joue différents rôles en réalisant différentes activités pour résoudre le problème donné dans une PBL coopérative. Le méta-modèle utilisé pour exprimer les situations de PBL doit permettre l'expression :

- De la **structuration hiérarchique** des activités d'une PBL coopérative.

- Des éléments utilisés 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.).
- Des éléments permettant de décrire les **aspects sociaux des PBL** (rôles, activités collaboratives, restrictions, droits, différents modes de coopération, etc.).

Le méta-modèle CPM est défini par un profil spécifique UML comprenant :

La syntaxe abstraite : Le méta-modèle CPM a été conçu de manière préliminaire à l’élaboration du profil afin de séparer la terminologie de la notation [152]. Il repose sur deux principaux paquetages : le paquetage de fondation représentant un sous-ensemble du méta-modèle UML, et le paquetage des extensions regroupant tous les concepts, relations et contraintes du langage (figure 3.7). Le paquetage des extensions est décomposé en quatre sous-paquetages interdépendants : paquetage des éléments de base, paquetage pédagogique, paquetage structurel, et paquetage social. La sémantique des différents concepts et relations proposés est exprimée via des contraintes OCL, lorsque cela est possible, sinon en langage naturel [126]. **La syntaxe concrète** : Le profil CPM est une spécialisation d’UML

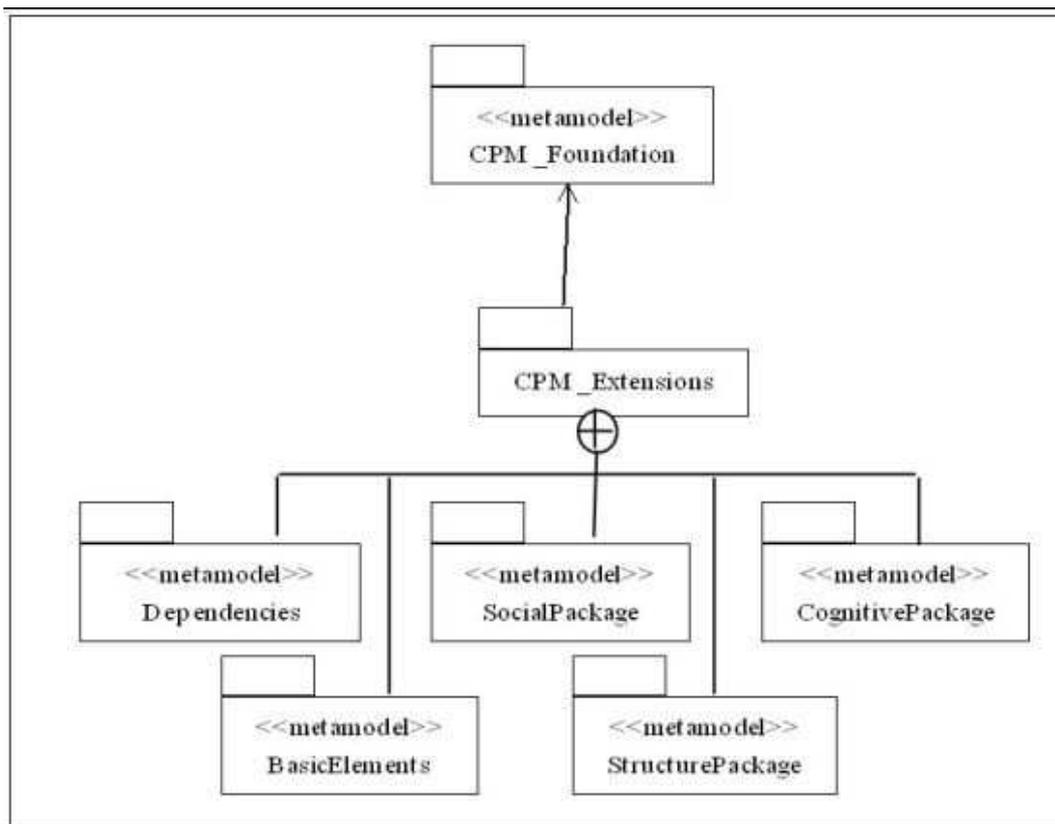


FIG. 3.7: Paquetages du méta-modèle CPM

dédiée à la modélisation de situations d'apprentissage par résolution de problèmes coopératifs. Il est composé entre autres de [126] [128] [152] :

- 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 méta-classe 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 associées aux stéréotypes.

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	

FIG. 3.8: Stéréotypes du profil CPM

Dans la figure 3.8 on retrouve trois des concepts présents dans l'extrait du méta-modèle CPM. A chaque stéréotype correspondent des méta-classes UML de référence qui vont être "étendues" par ce nouveau concept. Des icônes sont associées aux stéréotypes les plus utilisés afin de simplifier leur interprétation sémantique dans les diagrammes où ils apparaissent.

Nous notons qu'un certain nombre d'éléments ou stéréotypes constituant le méta-modèle du langage CPM a un équivalent IMS-LD figure 3.9. D'après ce tableau de comparaison entre la terminologie d'IMS-LD et celle de CPM, nous remarquons que la structuration du dernier ressemble beaucoup à celle d'IMS-LD. Par suite, cette structuration, pour les mêmes raisons citées dans la sous-section 3.3.1 traitant IMS-LD d'un point de vue Pédagogie par Projet Collectif (PPC), reste

Terminologie d'IMS-LD	Terminologie dans CPM
<i>method</i>	LearningProcess
<i>play, act</i>	LearningPhase
<i>activity, learning activity, support activity</i>	Activity
<i>activity-structure</i>	ActivityStructure
<i>role, learner, staff</i>	Role
<i>role-part</i>	\emptyset
<i>environment, learning object, service</i>	Resource
<i>learning objective</i>	Objective, Postcondition
<i>prerequisite</i>	StaticPBLElement, Precondition
<i>condition et property</i>	PBLConstraint, StaticPBLElement
<i>notification</i>	Resource, PBLConstraint

FIG. 3.9: Correspondances entre la terminologie d'IMS-LD et celle de CPM

loin de la structuration de la Pédagogie par Projet Collectif (PPC) même si les concepts sont nommés différemment.

Discussion

Le langage CPM facilite l'élaboration de modèles pour la conception de situations d'apprentissage par résolutions de problèmes (PBL) [126], [128], [152]. Il est un langage de modélisation graphique spécialisant UML pour la conception de PBL en amont des langages de type EML. Après quelques expériences [126] [128] [152], les auteurs ont conclu que l'utilisation des profils UML permet une conceptualisation explicite des différentes connaissances sur la situation d'apprentissage en cours de spécification : "*les stéréotypes et les valeurs marquées décrivent les concepts et les relations sous-jacents. Ceci permet une meilleure capitalisation des connaissances et améliore la communication entre les différents intervenants de l'équipe pluridisciplinaire de conception*" [126].

Vu la structuration de CPM qui est très proche d'IMS-LD (existence d'un concept voisin CPM pour chaque concept IMS-LD), nous considérons que CPM n'est pas satisfaisant pour scénariser une situation d'apprentissage dans un contexte de Pédagogie par Projet Collectif (PPC).

3.3.5 Approche de production de modèles d'expression de scénarios pédagogiques

Dans cette sous-section nous expliquons l'approche proposée par [119], [73] consistant en une conceptualisation s'appuyant sur des modèles spécifiques au

contexte pédagogique. Cela passe par l'élaboration puis l'utilisation de modèles, non pas préexistants, mais extraits de l'activité même de conception des futurs usagers de ces modèles.

La plupart des langages de modélisation pédagogique offrant des concepts pré-définis "*supposés génériques*", ne permettent pas toujours d'exprimer les différents aspects de l'expertise issue du contexte pédagogique ciblé. En effet, les concepteurs doivent adapter ces modèles pédagogiques préétablis en fonction de leurs représentations. Ceci les amène à analyser et à appréhender ces modèles en question. Selon [119], [73], les modèles doivent être produits par les enseignants concepteurs eux-mêmes. Ces derniers sont considérés, selon cette approche, comme des experts du domaine. Tout cela va leur permettre d'appréhender mieux la description des scénarios qui seront produits. L'architecture de cette approche repose sur trois niveaux :

- **le niveau "Scénario pédagogique"** : ce niveau permet aux concepteurs de produire des scénarios qui décrivent leurs situations d'apprentissage.
- **le niveau "Langage de modélisation pédagogique"** : ce niveau permet aux concepteurs de produire des modèles d'activités formalisés sous forme de langages de modélisation pédagogique sur lesquels l'expression des scénarios est fondée.
- **le niveau "Méta-langage"** : ce niveau permet aux concepteurs de représenter les concepts du domaine pédagogique. C'est un méta-modèle de langages que chaque groupe de concepteurs va instancier dans son propre fonctionnement.

Nous utiliserons cette approche et montrerons que proposer un méta-modèle dédié à la Pédagogie par Projet (PPC) (méta-modèle métier), tout en laissant ce méta-modèle ouvert et flexible pour l'enseignant concepteur de la Pédagogie par Projet Collectif (PPC), est une contribution aidant cet enseignant à mieux appréhender la description des scénarios PPC qui seront produits.

3.4 Conclusion

Dans ce chapitre nous avons étudié la définition d'un scénario pédagogique, et son processus de conception (section 3.1). Par suite, nous avons étudié les langages existants pour décrire un scénario pédagogique (3.2). Nous avons commencé (sous-section 3.3.2) par l'étude d'IIMS-LD que nous avons critiqué pour sa difficulté d'utilisation par des enseignants concepteurs non spécialisés et par son manque d'expressivité par rapport à la Pédagogie par Projet Collectif (PPC). Après, nous avons étudié le langage LDL (sous-section 3.3.3) dédié aux situations d'apprentissage collaboratif, puis le langage CPM (sous-section 3.3.4) dédié à l'apprentissage par résolution de problèmes (PBL). Nous avons terminé ce chapitre par l'étude de l'approche consistant à scénariser une situation d'apprentissage en s'appuyant

sur des modèles spécifiques au contexte pédagogique.

L'étude faite dans ce chapitre, notamment l'étude des travaux d'autres chercheurs qui ont proposé des langages dédiés comme LDL ou CPM, malgré leurs grandes qualités, nous permet de conclure qu'un méta-modèle dédié à la Pédagogie par Projet Collectif (PPC) peut être une alternative à l'insuffisance et les difficultés rencontrées dans IMS-LD. L'approche expliquée dans la sous-section 3.3.5 et s'appuyant sur des modèles spécifiques au contexte pédagogique, appuie notre choix de méta-modélisation dédiée à la Pédagogie par Projet Collectif (PPC).

Il y a un consensus général, qu'obliger des enseignants concepteurs à utiliser n'importe quels concepts est difficile [10], parce qu'adopter des spécifications externes au domaine apporte des difficultés supplémentaires et peut décourager les enseignants. Afin de réduire les obstacles vers l'adoption de notre méta-modèle PPC, nous avons choisi de modéliser dans le chapitre suivant le domaine de la PPC.

Chapitre 4

Méta-modélisation de la Pédagogie par Projet Collectif

4.1 Introduction

La modélisation consiste à représenter une chose d'un monde réel ou virtuel dans un but cognitif. Frantz [86] décrit le processus de modélisation sous forme de deux traductions successives : le monde réel est traduit en un modèle conceptuel (par un mécanisme d'abstraction) ; le modèle conceptuel est ensuite traduit en un modèle de simulation (par un mécanisme d'implémentation).

La modélisation d'un domaine vise à définir les différents concepts partagés par tous les acteurs de ce domaine d'activité. Le contexte de la Pédagogie par Projet Collectif (PPC) n'échappe pas à cette règle. Un modèle est une description à un niveau abstrait d'un système. Dans notre contexte de travail de recherche, le système représente une situation d'apprentissage et le modèle le décrivant est le scénario pédagogique. Ce modèle est porteur de caractéristiques du contexte pédagogique de ses concepteurs. Il reflète une démarche en des termes propres à une expertise pédagogique cible et spécifique.

Au lieu d'adopter une approche généraliste (ex : IMS-LD), nous choisissons d'adopter une approche de modélisation plus spécifique dédiée à la Pédagogie par Projet Collectif (PPC). Nous visons une définition d'un modèle dédié à la Pédagogie par Projet Collectif (PPC) parce que nous pensons qu'il pourrait s'avérer plus pertinent et permettrait de pallier à l'inadéquation de la description inhérente à une approche généraliste. Ceci nous amène à réfléchir à un moyen réellement optimisé pour la description de situations en Pédagogie par Projet Collectif (PPC). Ceci passe à notre sens par un langage dédié qui pourrait ainsi s'avérer plus pertinent parce qu'il est conçu à une seule fin de traiter un problème ou domaine précis, et

c'est là une de nos hypothèses de recherche.

Ainsi, nous pensons que la production d'un méta-modèle dédié à la Pédagogie par Projet Collectif (PPC) présente de nombreux avantages : il utilise les idiomes et le niveau d'abstraction du domaine ciblé (PPC), il est plus riche sémantiquement par rapport aux spécialistes dans notre contexte de la PPC car il est facile à appréhender ; il est léger, donc facilement maintenable, portable et réutilisable ; il peut être plus facilement documenté, cohérent et fiable. Le méta-modèle dédié à la PPC fait collaborer les concepteurs experts de la PPC autour d'un même dialecte, en plus il permet de structurer la réflexion sur le domaine.

Cependant, cela peut présenter aussi un inconvénient. En effet, utiliser un langage dédié, donc spécifique a un coût : il nécessite de le concevoir, l'implémenter, le maintenir, et définir précisément son champ d'utilisation.

Dans ce chapitre, nous présentons, et expliquons notre première contribution dans cette thèse vis-à-vis des conclusions de l'étude réalisée dans les chapitre précédents. Pour cela, nous commençons tout d'abord (section 4.1) par expliquer l'approche de modélisation choisie, l'utilisation du méta-modèle proposé, et la méthodologie suivie pour l'élaborer. Dans la section 4.2, nous présentons le méta-modèle dédié à la PPC en expliquant son mode d'élaboration. Dans la section 4.3, nous montrons un scénario PPC exemple qui a été élaboré, à l'IUT de Laval pour la première année en informatique, à partir du méta-modèle PPC proposé. Cet exemple a été conçu en utilisant l'outil ModX du laboratoire LIFL (ex-Trigone) de l'Université de Lille 1.

4.2 Approche, utilisation et méthodologie

Dans cette partie, nous abordons l'approche de modélisation choisie pour proposer le méta-modèle PPC, l'utilisation de ce méta-modèle, et la méthodologie suivie pour l'élaborer.

4.2.1 Approche de modélisation proposée

Nous adoptons la distinction de l'OMG de quatre couches de modélisation pour décrire les niveaux d'expression [94] [27] ¹. En s'appuyant sur l'exemple de la programmation, si M0 désigne le niveau d'exécution d'une instance d'un programme, M1 désigne le niveau d'expression du programme lui-même, M2 le niveau d'expression du langage dans lequel est écrit le programme [82], M3 est le niveau du fonctionnement d'une grammaire formelle. Le niveau M3 définit les notions de base permettant l'expression des méta-modèles M2 (ex : MOF, eCore, etc.). IMS-LD par exemple est situé au niveau M2, le scénario lui-même exprimé en IMS-LD est au niveau M1 et l'implémentation sur la plate-forme est au niveau M0.

¹nous traitons cette approche en détail dans le chapitre 6 pour expliquer les transformations de modèles

Le méta-modèle PPC que nous proposerons dans ce chapitre est situé au niveau M2. En effet, nous voulons permettre aux enseignants concepteurs d'exprimer des modèles de scénario au niveau M1, qui seront exécutés par les apprenants ou les enseignants, au niveau M0 (figure 4.1). Cette approche est adoptée et utilisée par

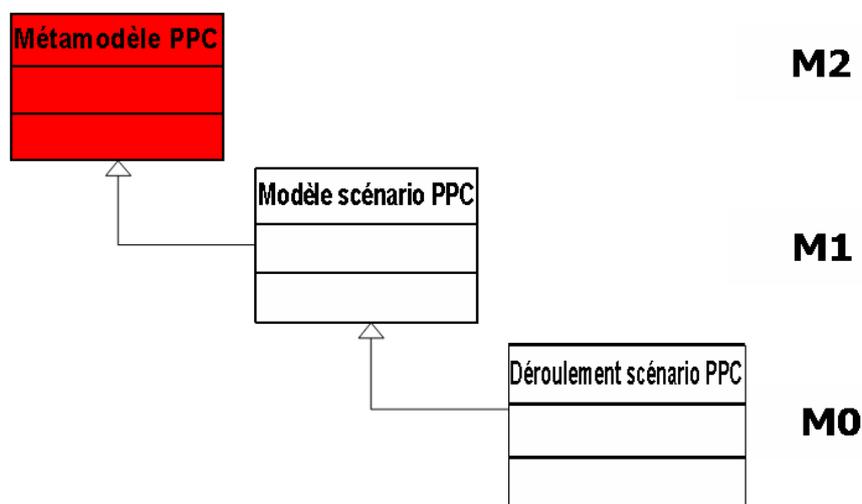


FIG. 4.1: Approche de modélisation proposée

plusieurs travaux en EIAH. Nous citons : la proposition du standard IMS-LD, le langage CPM, et le langage LDL expliqués dans le chapitre précédent.

4.2.2 Utilisation du méta-modèle PPC

L'enseignant concepteur qui souhaite mettre en oeuvre une situation d'apprentissage de PPC, a besoin de décrire un scénario PPC et le déployer dans une plate-forme orientée PPC. Pour cela, il pourra utiliser notre méta-modèle PPC pour modéliser les scénarios PPC qu'il veut mettre en oeuvre (figure 4.2). L'utilisation de notre méta-modèle doit passer par un éditeur existant qui pourrait être ModX ou ECOS, etc.². Ces scénarios sont déployés et exécutés dans la plate-forme cible.

4.2.3 Méthodologie suivie pour élaborer le méta-modèle PPC

Dans cette sous-section, nous expliquons la méthodologie adoptée pour élaborer le méta-modèle PPC. La méta-modélisation de la Pédagogie par Projet Collectif consiste à modéliser le domaine de la PPC. Pour élaborer le méta-modèle PPC, nous adoptons le point de vue de [200] qui se base (figure 4.3), d'un côté, sur

²Nous choisissons dans la section 4.3 ModX comme outil

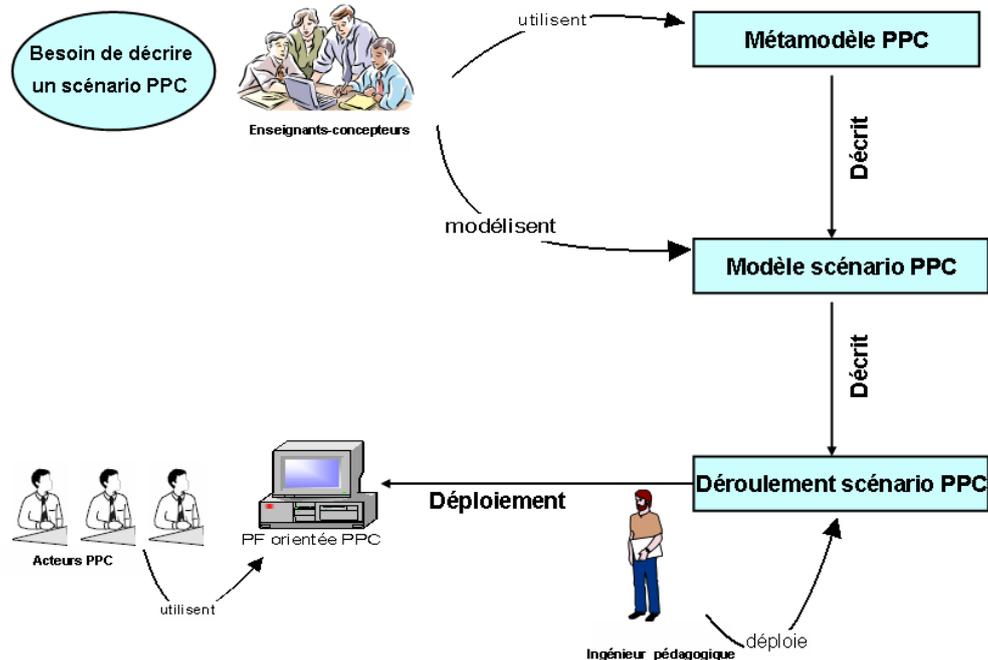


FIG. 4.2: Utilisation du méta-modèle PPC

une étude théorique (bibliographique) pour récolter tous les éléments et concepts d'une Pédagogie par Projet Collectif. D'un autre côté, et pour assurer l'aspect réaliste de notre méta-modèle, ce point de vue se base sur quelques études empiriques (ex : Mepulco-Université). Notre méta-modèle PPC permet à son tour d'interpréter, d'un côté, l'étude théorique achevée et, d'un autre côté, de catégoriser et de structurer les phénomènes (expériences) étudiés.

4.3 Elaboration du méta-modèle Pédagogie par Projet Collectif

Dans la section précédente, nous avons expliqué l'approche et la méthodologie choisies pour élaborer ce méta-modèle de la PPC. Nous sommes conscients qu'une modélisation d'un domaine (PPC) tout entier n'est pas un travail simple. En effet, un même domaine peut être modéliser différemment et par suite peut avoir plusieurs méta-modèles correspondants à des points de vue particuliers. Comme ces points de vue ne sont pas disjoints, il existera des concepts communs aux différents méta-modèles. Nous ne pouvons pas traiter tous les points de vue et proposer **LE** méta-modèle PPC. Pour cela, nous souhaitons proposer un méta-modèle PPC "ouvert" et "évolutif" dans le sens qu'on peut ajouter ou supprimer des concepts et relations entre concepts. Ce méta-modèle peut être personnalisable

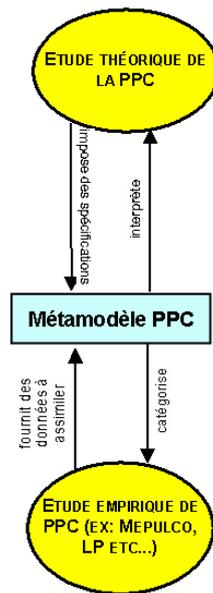


FIG. 4.3: Méthodologie adoptée pour élaborer le méta-modèle PPC

parce qu'il permet l'élaboration de scénarios pédagogiques adaptés aux besoins des différentes équipes de conception l'utilisant. Nous revenons ici à l'approche de [119], [73] considérant que les enseignants deviennent eux-mêmes concepteurs de leurs propres modèles pour mieux appréhender la description des scénarios qui seront produits. En effet, le méta-modèle PPC proposé est un méta-modèle évolutif et ouvert permettant à l'enseignant de le modifier selon ses pratiques de Pédagogie par Projet Collectif (PPC) ³.

Le méta-modèle proposé est présenté selon trois points de vue. Le but de la vue générale (Figure 4.4) est de préciser la **structure du projet** dans le contexte d'une Pédagogie par Projet Collectif (niveau projet). Le but de la deuxième vue (Figure 4.5) est de se concentrer sur les **aspects d'apprentissage**. Cette vue du métamodèle propose un cadre permettant la création d'une session de formation (niveau étape). La troisième vue (Figure 4.6) est liée à la partie de **production du projet** (niveau tâche).

4.3.1 Méta-modèle conceptuel général

Décomposition du projet en étapes composées en tâches

Dans notre étude théorique de la PPC, nous avons distingué différents points de vue d'un projet chez plusieurs chercheurs travaillant sur la Pédagogie par Pro-

³Nous montrerons cet aspect pratiquement lors de l'élaboration de ce méta-modèle dans les sous-sections suivantes

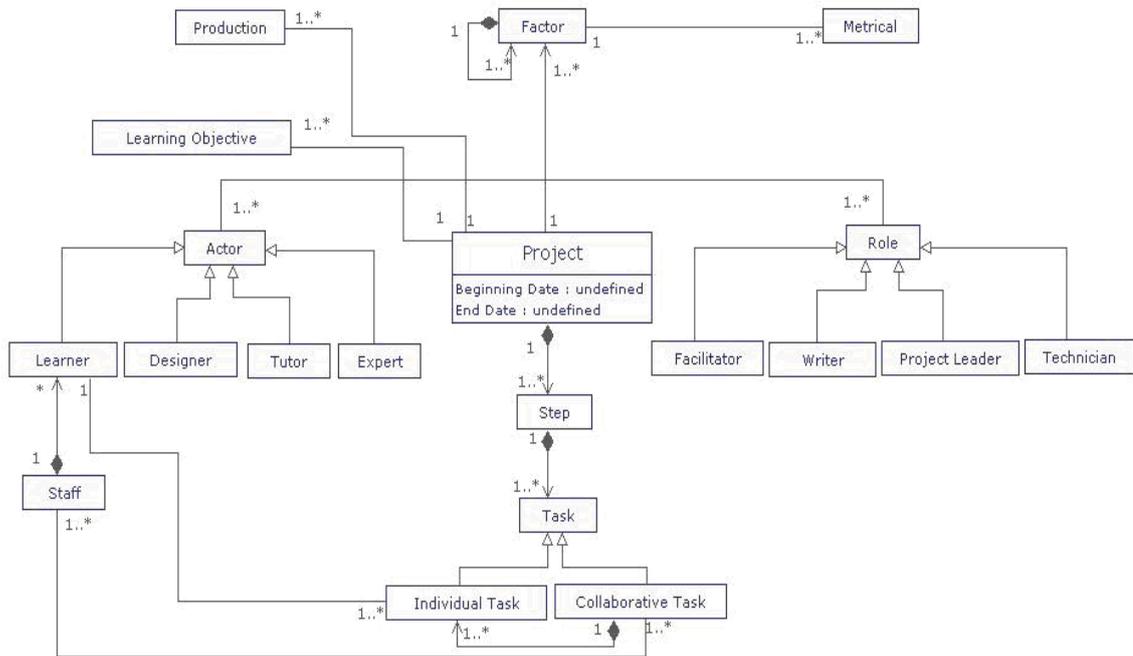


FIG. 4.4: Méta-modèle général de la PPC

jet Collectif (PPC). Des chercheurs ont décomposé un projet en plusieurs étapes [181], [87], [134], [103], [104] "... la construction d'un projet ... procède de certaines étapes" [130], d'autres en phases [7], [84] [83], et d'autres qui ont considéré qu'un projet est "orchestré" en plusieurs scénarios séquentiels [183] ou activités [81], [169].

Dans l'industrie, le projet selon [135], est composé de phases : "phases ayant un objectif, un travail à réaliser et des décisions à prendre."

Donc, un projet est décomposé tout d'abord en plusieurs étapes (phases, unités, activités etc.). Ces chercheurs : [202], [87], [134], [183], [135], [84], [83] décomposent l'étape (la phase, l'unité ou l'activité) en plusieurs tâches. Lebrun [130] décompose l'étape en plusieurs phases, et ces chercheurs [181], [103], [84], [83] la décomposent en plusieurs activités "Each step of the project plan can be defined as a number of actions that learners need to accomplish" [104]. Dans l'industrie, un projet est découpé en phases, une phase en actions : "Ces phases sont des combinaisons d'actions finalisées qui se réalisent selon un cycle identique, qu'il agisse d'actions de management ou d'actions de réalisation du produit du projet." [135]. L'action est considérée comme une tâche aussi ; L'ISO [10006 :2003, article 3] définit l'action comme la "plus petite tâche identifiée dans un processus de projet."

D'un autre côté, nous avons remarqué dans les expériences Licence Professionnelle

menées à l'IUT de Laval et celles menées dans le cadre de la méthode Mepulco-Université (traitée dans le deuxième chapitre) de l'Université de Littoral Côte d'Opale que les apprenants ont tendance à décomposer le projet en étapes et les étapes en tâches.

Selon notre analyse, un projet se décompose (représenté par la classe *Projet*) en plusieurs **étapes de projet** (relation de composition entre les deux classes *Projet* et *Etape de projet*) (figure 4.4). Et nous choisissons de décomposer chaque étape de projet en une ou plusieurs **tâche(s)** (relation de composition entre les deux classes *Etape de projet* et *Tâche*). La tâche peut être une **tâche individuelle** (dans le cas d'une Pédagogie par Projet), ou une **tâche collective** (dans le cas d'une Pédagogie par Projet Collectif) (classes héritées de la classe *Tâche* sont : *Tâche individuelle*, *Tâche collective*). Cette décomposition est modifiable ; c'est-à-dire le concepteur de scénarios PPC pourrait la changer en renommant les concepts et décomposant par exemple le projet en phases ou unités et ces dernières en activités.

Pour apprendre, les apprenants doivent accomplir des tâches individuelles (classe *Tâche individuelle*), ou collectives (classe *Tâche Collective*). Les apprenants apprennent à travailler collectivement en se partageant des tâches, en organisant et planifiant ces tâches [134].

La Pédagogie par Projet Collectif (PPC) comprend l'apprentissage "Collaboratif" et l'apprentissage "Coopératif" et c'est pour cela nous utilisons le terme "Collectif" englobant les deux termes. La démarche coopérative, plus structurée, réserve à l'enseignant un plus grand contrôle sur l'apprentissage. Par contre, la démarche collaborative propose aux apprenants plus de liberté. La tâche collaborative, contrairement à la tâche coopérative, n'est pas morcelée entre les participants. Tandis que dans une tâche coopérative, et au sein d'une équipe, chaque membre individuellement est responsable de réaliser une sous-tâche. La tâche (coopérative) est complète lorsque tous les membres de l'équipe mettent en commun leurs réalisations [105], [106]. Donc en apprentissage coopératif, il faut décomposer la tâche coopérative en sous-tâche et distribuer le travail à chaque membre du groupe. Par contre, en apprentissage collaboratif, chaque apprenant doit pouvoir choisir la stratégie et le cheminement qui lui convient pour apprendre. Les contributions se font selon les habilités de chacun [105], [106]. La tâche collaborative est la même pour tous les membres d'équipe tandis que la tâche coopérative est divisée en sous-tâches individuelles. Ce qui nous permet de conclure qu'il faut distribuer les rôles et les responsabilités dans une tâche coopérative et non pas dans une tâche collaborative.

Si la tâche collective correspond à une tâche coopérative, elle sera divisée en sous-tâches individuelles (relation de composition entre les deux classes *Tâche collective* et *Tâche individuelle*). Si elle correspond à une tâche collaborative, elle pourra ne pas être divisée en sous-tâches individuelles.

Production et Objectifs d'apprentissage

Hubert [110] définit le projet comme étant "*une activité pratique planifiée et signifiante, impliquant des recherches, la résolution de problèmes et menant à la réalisation d'un produit concret qui sera socialisé.*" Selon Leroux [134], la pédagogie par projet permet aux apprenants de travailler à partir d'un cahier des charges visant une **production** effective. Un projet selon Kilpatrick [120] est défini comme "*une activité qui possède un **but précis**.*" Une pédagogie par projet vise un objet qui peut être une **connaissance**, une **production** technique ou artistique [130]. Selon Piccinini *et al.* [169], outre la définition des **objectifs d'apprentissage**, le projet doit inclure des objectifs de **production**. Le standard IMS-LD contient aussi le concept "*Learning objective*" traduit en français en "*Objectifs d'apprentissage*". Pour représenter la production ⁴ concrète et celle de la connaissance, nous proposons et associons au concept "*Projet*" les deux concepts : "*Production*" et "*Objectifs d'apprentissage*" (relation d'association entre d'une part, les deux classes *Projet* et *Production* et d'autre part, entre les deux classes *Projet* et *Objectifs d'Apprentissage*).

Facteur, Métrique

Avec son approche innovatrice sur l'apprentissage, la PPC exige une approche innovatrice concernant l'évaluation. Elle exige une évaluation diverse et fréquente, y compris l'évaluation de l'enseignant concepteur, l'évaluation des apprenants, et l'auto-évaluation [193].

Pour évaluer un projet, [145] évalue "trois **facteurs**" (*Product, Process, People*). Évaluer un projet, c'est évaluer le produit final, les apprenants, et le processus de travail. Pour cela, nous proposons un concept "*Facteur*" (relation d'association entre les deux classes *Projet* et *Facteur*). Ce facteur peut être composé d'un ensemble de facteurs (relation de composition réflexive). Pour évaluer chaque facteur, nous proposons le concept "*Métrique*" (relation d'association entre les deux classes *Facteur* et *Métrique*) qui est composé d'un ensemble d'outils d'évaluation du travail des apprenants "*... evaluation tool for teachers evaluating students work...*" [103], [104], d'outils d'évaluation de processus de travail collectif et individuel [145], ou d'outils de technologie de la programmation tels que les techniques de révision "*Software reading techniques*" [204].

⁴Le cahier des charges est considéré comme une production

Acteurs : Apprenant, Equipe, Concepteur, Tuteur, et Expert

Dans une Pédagogie par Projet Collectif, certains parlent des deux acteurs principaux : **l'enseignant** et **l'apprenant** [181], [87], [130], [134], [81], [77], [169]. Le nombre des acteurs est étendu : l'expert, le coordinateur, le tuteur ou le suiveur, et le client peuvent aussi exister [77], [83]. Talon *et al.* [196], [195] dans leurs expériences sur Mepulco-Université distinguent quatre types d'acteurs : l'apprenant, le tuteur, l'expert et le client (l'enseignant concepteur) ⁵. Dans l'industrie, on utilise les concepts "Acteur" ou "Partie intéressée" définie par l'ISO comme "personne ou groupe de personnes ayant un intérêt dans le fonctionnement ou le succès d'un organisme." IMS-LD utilise le concept "Personne" pour faire référence aux acteurs du scénario [113]. Nous choisissons de modéliser les acteurs par une classe commune (Acteur) qui peut être un **apprenant**, un **tuteur**, un **enseignant concepteur** (client) ou un **expert**. Nous laissons aux utilisateurs du méta-modèle de la PPC la liberté d'ajouter de nouveaux acteurs ou de supprimer des acteurs existants ou de les modifier.

Commençons par l'apprenant (classe *Apprenant* héritée de la classe *Acteur*), qui peut former tout seul ou avec d'autres apprenants une équipe [87], [134] (classe *Equipe*) ou un groupe [84], [83] qui est composée d'un ou de plusieurs apprenant(s) (relation de composition entre les deux classes *Equipe* et *Apprenant*). Nous appelons "**équipe**" un ensemble d'apprenants engagés dans un même projet [87].

Un apprenant (classe *Apprenant*) est responsable d'une ou de plusieurs tâche(s) ou sous-tâches individuelles tandis qu'une tâche ou sous-tâche individuelle est affectée à un seul apprenant (relation d'association entre les deux classes *Apprenant* et *Tâche individuelle*). Une équipe (classe *Equipe*) est responsable d'une ou de plusieurs tâche(s) collective(s) tandis qu'une tâche collective ne peut être affectée qu'à une seule équipe (relation d'association entre les deux classes *Equipe* et *Tâche collective*).

L'enseignant concepteur (représenté par la classe *Concepteur* héritée de la classe *Acteur*) "*intervenant en amont du projet*" [87], propose un projet au minimum et définit un cahier des charges ⁶. Il constitue une ou plusieurs équipe(s). Il nomme ou joue le rôle de tuteur [196], [195] qui sera chargé de suivre l'avancement et la planification du projet et c'est lui seulement qui définit les étapes du projet.

Le tuteur, chef du projet [87], chef d'orchestre [183], (représenté par la classe *Tuteur* héritée de la classe *Acteur*) est chargé de suivre la planification et le déroulement du projet. "*Son rôle est celui d'un facilitateur, d'un consultant et d'un régulateur qui est au contact des apprenants*" [77], les assiste à leur demande ou lorsqu'il en sent la nécessité. Il prend en charge l'environnement social ainsi que tous les processus pédagogiques se déroulant [77]. Un projet peut avoir plus d'un tuteur. Un tuteur dirige une ou plusieurs équipe(s) et une équipe peut être dirigée

⁵L'enseignant-concepteur peut jouer le rôle du client du projet

⁶Le cahier des charges est une instance de la classe *Production*

par un ou plusieurs tuteur(s).

L'expert (représenté par la classe *Expert* héritée de la classe *Acteur*), aidant une ou plusieurs équipe(s) afin de dépasser et résoudre ses problèmes dus à un manque de compétences techniques ou de temps, peut être demandé par une ou plusieurs équipes [196], [195]. De même, une équipe peut demander l'aide d'un ou de plusieurs expert(s). Un expert peut ne pas être affecté à un projet ou affecté à plusieurs projets, et un projet peut ne pas nécessiter l'aide d'un expert ou nécessiter l'aide de plusieurs experts.

Rôle : Facilitateur, Rédacteur, Chef de projet, Technicien

Nous avons modélisé les acteurs dans la Pédagogie par Projet Collectif (PPC) en décrivant leurs rôles brièvement. Mais une question se pose : Est-ce que les rôles joués dans une PPC sont toujours joués par les mêmes acteurs et pendant tout le projet ? Lorsque nous parlons du tuteur, nous disons qu'il est chargé de suivre le déroulement et la planification du projet. Il "*joue un rôle moteur dans la formation. La qualité du suivi permet de garantir la motivation de l'apprenant et d'éviter qu'il abandonne sa formation en cours de route...*"⁷. Son rôle est important dans les tâches d'animation et d'assistance du groupe. Son rôle vis-à-vis du groupe est, selon [49], de :

- Constituer les groupes.
- Assurer le suivi pédagogique de la formation (réponses aux questions des apprenants, analyse de la progression, conseils personnalisés).
- Aider à circonscrire le sujet.
- Prendre part à la recherche de la documentation.
- Suggérer des processus de résolution.
- Gérer les communications synchrones.
- Faciliter la communication entre les membres du groupe.
- Favoriser la mémoire du groupe.

D'après Charlier *et al.* [49], le tuteur joue quatre rôles différents vis-à-vis de ses apprenants :

- Facilitateur : il aide les apprenants à choisir leur projet, facilite leur expression, écoute les autres apprenants, prend en compte les opinions de leurs pairs.
- Modérateur : il synthétise, critique, structure les contenus, gère et renforce les délais de réalisation des activités.
- Expert : il aide les apprenants à trouver les documents et les ressources, leur apporte son expérience personnelle.
- Soutien affectif : il s'engage personnellement, et encourage les apprenants.

⁷<http://www.demos.fr/lexique.asp>

L'enseignant concepteur selon [169] joue un rôle principal de contrôle, de guide et de commande, qui apparaît au moins sur les trois étapes suivantes d'évolution de projet :

- Explication initiale du projet.
- Installation du projet. Avant de commencer chaque travail de projet, les apprenants sont fortement encouragés à demander de conseils auprès de l'enseignant-concepteur.
- Évaluation finale.

Selon Scherz *et al.* [181], le rôle de l'enseignant concepteur dans le projet est de résoudre les problèmes immédiats, d'aider les apprenants plutôt que de faire de leçons particulières.

Les pratiques menées dans Mepulco [196], [195] affectent différents rôles aux apprenants; par exemple elles affectent à l'apprenant le rôle d'un responsable de projet, chef de projet, d'un rédacteur ou d'un technicien, etc.

Nous remarquons que les rôles affectés aux acteurs de la PPC ne sont pas toujours les mêmes. Cela est dû à la différence entre plusieurs manières d'appliquer la Pédagogie par Projet Collectif (PPC). Il y a certains qui affectent la responsabilité de tout le projet à l'enseignant concepteur, d'autres au tuteur, et d'autres laissent cette responsabilité pour un apprenant. Par exemple le rôle du tuteur est de suivre l'avancement et la planification du projet. Mais le fait d'affecter ce rôle à l'enseignant concepteur, à une équipe d'apprenants ou à un seul apprenant est tout à fait possible. Ainsi, nous définissons un concept général qu'on nomme "rôle" (classe *Rôle*) avec des concepts hérités : "facilitateur" (classe *Facilitateur*), "rédacteur" (Classe *Rédacteur*), "chef de projet" (classe *Chef de projet*), "technicien" (Classe *Technicien*). Nous modélisons les rôles les plus joués dans la Pédagogie par Projet (PPC) mais cela ne veut pas dire que nous les restreignons. Nous construisons le concept "rôle" (classe *Rôle*) en héritant plusieurs rôles principaux et après c'est à l'utilisateur d'ajouter de nouveaux rôles ou de ne pas utiliser tous les rôles existants.

Chaque acteur (classe *Acteur*) joue un ou plusieurs rôles (relation d'association entre les deux classes *Acteur* et *Rôle*) et un rôle est joué par un ou plusieurs acteurs. En effet, nous laissons le choix à l'utilisateur de définir le rôle de chaque acteur.

4.3.2 Méta-modèle d'une étape de la PPC

Nous avons modélisé ci-dessus la Pédagogie par Projet Collectif au niveau projet. Dans ce qui suit (figure 4.5), nous modélisons l'étape d'un projet. L'acteur responsable de définir les étapes du projet diffère d'une méthode à l'autre. Selon [87], [130], [134], c'est l'enseignant concepteur qui est l'acteur responsable de structurer le projet en étapes et de les définir. Dans les expériences Licence

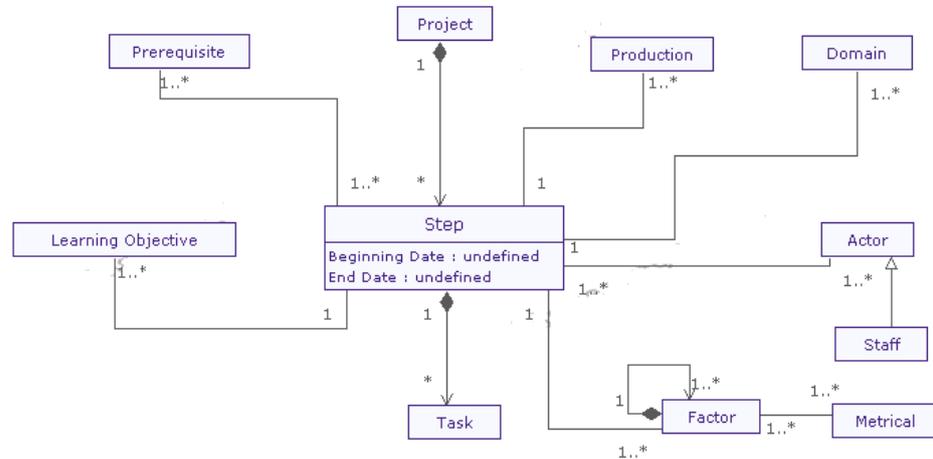


FIG. 4.5: Méta-modèle d'une étape de la PPC

Professionnelle à l'IUT de Laval c'est le tuteur, et dans Mepulco [196], [195] c'est parfois l'apprenant. Pour être mise en oeuvre, une étape nécessite plusieurs acteurs qui peuvent être l'enseignant concepteur, le tuteur, l'apprenant et l'expert. Ainsi, nous lions le concept d'acteur avec celui de l'étape (relation d'association entre les deux classes *Acteur* et *Etape*).

Pour définir une étape, l'acteur responsable doit la décrire en détail. Cette description comporte selon [17], la présentation des **objectifs d'apprentissage** de l'étape (relation d'association entre les deux classes *Etape* et *Objectifs d'Apprentissage*), les acteurs ou/et les équipes concernés, et les tâches à effectuer (relation de composition entre les deux classes *Etape* et *Tâche*), le **domaine** auquel cette étape appartient (relation d'association entre les deux classes *Etape* et *Domaine*). Un projet pluridisciplinaire comporte des étapes appartenant à plusieurs domaines.

Selon [181], chaque étape finit par un **produit intermédiaire** qui sera combiné avec d'autres produits ou **rapports intermédiaires** dans le produit ou le rapport final (relation d'association entre les deux classes *Etape* et *Production*). Pour que les apprenants puissent commencer une étape, il faut qu'ils aient des **pré-requis pédagogiques** [196], [195](relation d'association entre les deux classes *Etape* et *Pré-requis*).

"On percevra, dans chacune des étapes, la place et l'importance de l'évaluation. . . "

[130]. L'évaluation d'une étape du projet permet d'un côté l'évaluation proprement dite et d'un autre côté la correction de travail des apprenants. Nous liions aussi le concept "facteur" avec celui de l'étape (relation d'association entre les deux classes *Etape* et *Facteur*). Ce facteur peut être composé de sous-facteurs (relation de composition réflexive). Comme dans l'évaluation de tout le projet, pour évaluer chaque facteur, nous utilisons des métriques (relation d'association entre les deux classes *Facteur* et *Métrique*).

4.3.3 Méta-modèle d'une tâche de la PPC

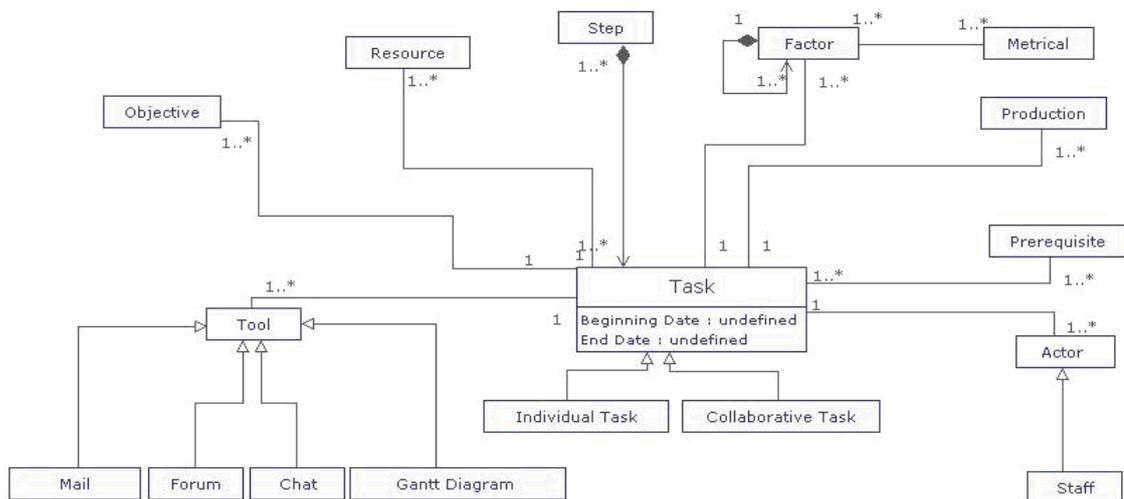


FIG. 4.6: Méta-modèle d'une tâche de la PPC

Dans ce qui suit (Figure 4.6), nous modélisons la tâche d'une étape; cette tâche peut être individuelle, ou collective. Dans chaque étape, l'apprenant (ou l'équipe d'apprenants) doit définir son plan de tâches. Le plan de tâches veut dire la structuration d'une étape en une succession de tâches [202] [183]. Ces tâches sont définies par l'apprenant lui-même, par l'équipe si elle est collective (collaborative ou coopérative), par le tuteur ou par l'enseignant concepteur. Donc, une tâche est affectée à un acteur ou une équipe (relation d'association entre les deux classes *Acteur* et *Tâche*). Lorsque l'acteur souhaite définir une tâche, il doit définir ses **objectifs** (relation d'association entre les deux classes *Tâche* et *Objectifs*), ses responsables [83], les acteurs et/ou les équipes concernés (relation d'association entre les deux classes *Tâche* et *Acteur*), les outils nécessaires pour la réaliser (relations d'association entre les deux classes *Tâche* et *Outil*), les **ressources** demandées "Les tâches sont caractérisées par un début et une fin, consommatrices de ressources et reliées entre elles par une relation d'antériorité" [84], [83] (relation

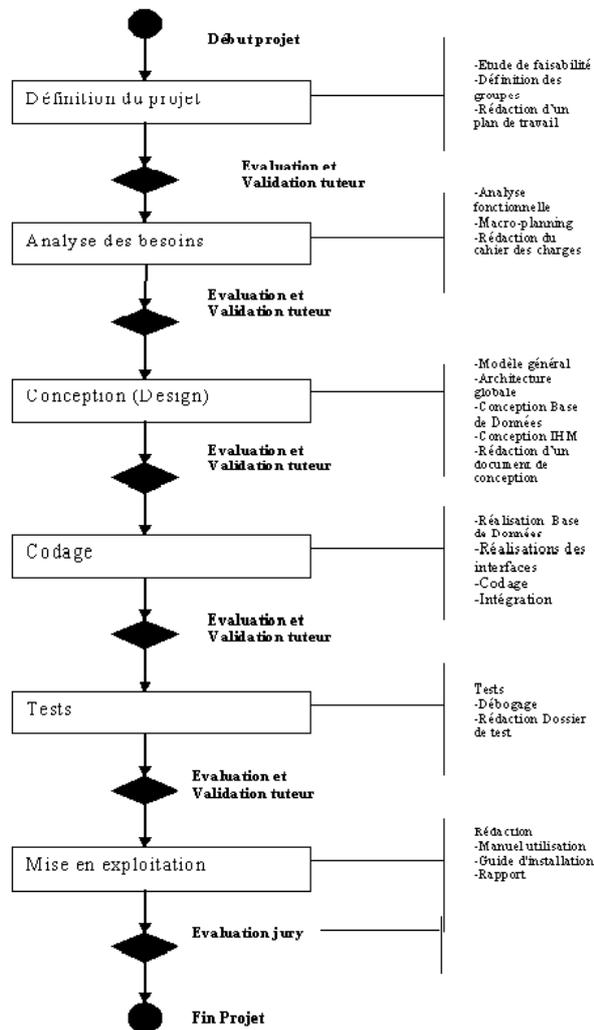


FIG. 4.7: Scénario PPC à l'IUT de Laval

d'association entre les deux classes *Tâche* et *Ressource*)⁸, le résultat attendu ou bien sa **production** (relation d'association entre les deux classes *Tâche* et *Production*), la nature de la tâche (individuelle, ou collective) [17]. Comme dans une étape de projet, les apprenants nécessitent d'avoir des **pré-requis** pédagogiques pour pouvoir commencer une tâche [196], [195], (relation d'association entre les deux classes *Tâche* et *Pré-requis*).

Les outils nécessaires pour la réalisation d'une tâche peuvent être soit un chat, un forum, une messagerie, un diagramme de Gantt, des outils spécifiques au domaine, outils de réunion [87] (classes héritées de la classe *Outil* sont : *Chat*, *Messagerie*, *Forum*, *Diagramme de Gantt*, etc.). L'utilisateur peut ajouter à ces outils n'im-

⁸Nous notons que les ressources peuvent être la production d'une tâche précédente

porte quel outil qu'il trouve nécessaire pour l'accomplissement de sa tâche. L'apprenant, dans la réalisation d'une tâche, peut évaluer ses résultats et les productions [130]. Dans le méta-modèle PPC proposé, nous lions aussi le concept "*facteur*" avec celui de la tâche (relation d'association entre les deux classes *Tâche* et *Facteur*). Pour évaluer chaque facteur, nous avons besoin des **métriques** (relation d'association entre les deux classes *Facteur* et *Métrique*). Nous remarquons qu'il y a des concepts qui sont présents dans chacun des niveaux de conceptualisation (Projet, Etape, et Tâche). Nous avons mis le concept "*Objectifs d'apprentissage*" au niveau projet parce que ce dernier a, comme nous l'avons expliqué, outre des objectifs concrets, un objectif de construction de la connaissance chez les apprenants [130]. Ce concept est repris pour chaque étape afin de détailler une partie précise des objectifs d'apprentissage. Concernant le concept "*Pré-requis*", nous posons que pour chaque étape et de même pour chaque tâche il peut y avoir des pré-requis pédagogiques nécessaires pour pouvoir démarrer l'étape ou la tâche. Enfin, en ce qui concerne le concept "*Production*" répété dans les trois niveaux, chaque tâche aboutit à une production (écriture d'un mail, développement d'un code d'une application, etc.). Chaque étape aboutit à une production intermédiaire (rapport intermédiaire, conception générale, etc.). Et un projet aboutit à une production finale (rapport final, produit final).

4.4 Exemple d'un scénario PPC

Dans cette section, nous présentons un scénario PPC exemple destiné à un public constitué d'apprenants en première année de Diplôme Universitaire de Technologie (DUT) Informatique à l'Institut Universitaire de Technologie (IUT) de Laval. Ce scénario est élaboré à partir du méta-modèle PPC proposé ci-dessus. Le méta-modèle PPC proposé ci-dessus est situé au niveau M2, le scénario que nous allons montrer est au niveau M1 et son implémentation sur une plate-forme cible est au niveau M0.

4.4.1 Cadre général

Dans le cadre de leur formation les apprenants doivent réaliser un projet visant à mettre en oeuvre et approfondir les enseignements de leur première année de formation. Comme il est spécifié dans le Programme Pédagogique National (PPN) des IUT, le but de ces projets est de "*favoriser l'acquisition d'un savoir-faire et d'un savoir être dans une optique professionnelle*" [69].

Le projet de première année se déroule tout au long du deuxième semestre (de début février à fin juin). Il est réalisé en groupe de quatre à cinq apprenants encadrés par l'ensemble des enseignants du département informatique. Des réunions sont organisées tous les quinze jours pour tous les groupes avec un tuteur différent de telle façon que tous les enseignants auront rencontré une fois chaque groupe

projet. Le projet est évalué en fin de période par toute l'équipe pédagogique. Deux rapporteurs sont nommés pour chaque groupe et la soutenance est organisée avec tous les tuteurs.

Les apprenants doivent mener un projet informatique dans leur cursus de formation. Ce projet sera mené dans un cadre collectif. Il se déroule sur une période fixée par l'équipe pédagogique. Le projet concerne l'étude et le développement d'une application web qui met en oeuvre les compétences acquises au cours du premier semestre de la première année de formation en matière de développement informatique, de conduite et gestion de projet mais aussi pour des compétences transversales comme le travail en groupe, l'autonomie, la rédaction de documents. Une attention particulière est portée à l'organisation du développement (conduite et gestion du projet) par les apprenants. Ces derniers doivent également produire des documents techniques et un rapport de fin de projet.

Les apprenants doivent avec l'enseignant concepteur définir leur projet (définir les rôles des membres du projet, définir les étapes et les tâches du projet, définir le calendrier/planning, ...) dans un cadre prescriptif donné (utilisation de la plate-forme de gestion de projet PHPCollab). Ils devront en particulier bien définir la planification de leur projet, organiser les réunions avec les tuteurs et avant chaque réunion avec les tuteurs, bien définir des responsabilités individuelles pour les tâches définies. Au cours des réunions de projet, le tuteur s'assurera de l'avancement du travail à la fois au niveau collectif et individuel du projet (notamment à partir de la planification de la quinzaine passée et de la quinzaine à venir).

A la fin du projet, un jury comportant les tuteurs et l'enseignant-concepteur évalue le projet des apprenants. Cette évaluation se fait par l'examen d'un rapport final, de la documentation technique accompagnant le produit livré, le produit (application web) et une soutenance du groupe afin de mettre une notation finale. L'application et l'ensemble des documents demandés sont à rendre sur un CD.

4.4.2 Présentation du scénario

Le projet possède une durée pour être achevé. Cette durée est fixée dès le début par l'enseignant-concepteur qui divise ce projet en étapes. Chaque étape est composée de plusieurs tâches à accomplir.

Les acteurs du projet sont : l'enseignant-concepteur, le tuteur, et les apprenants. L'enseignant-concepteur constitue les groupes d'apprenants (équipe). Ces groupes sont suivis par les tuteurs.

L'organisation du projet se fait de la manière suivante :

- Les apprenants travaillent par groupe de 3 à 5 personnes.
- Les apprenants utilisent une plate-forme de formation ou de travail collectif (Moodle, PhpCollab, Egroupware, etc.) pour gérer leur projet (définir les tâches, les répartir entre les membres du groupe, les planifier), déposer des documents...

- La planification de leur projet pourra être faite "à grosse maille" au début du projet (définition d'un macro-planning) et de manière plus fine pour chaque quinzaine de manière à permettre au tuteur d'évaluer au cours des réunions le travail accompli à celui planifié. A la fin du projet les apprenants devront faire un bilan et comparer leur planification initiale à la réalisation.

Les étapes du projet sont :

- Définition du projet
- Analyse des besoins
- Conception
- Codage
- Test
- Mise en exploitation
- Evaluation (Soutenance)

Dans la figure ci-dessus (figure 4.7), nous représentons les étapes du projet ainsi que les tâches les composant. Nous mettons en annexe A une description détaillée des étapes, des tâches, des acteurs, des outils utilisés, des équipes etc. . .

4.5 Réalisation technique

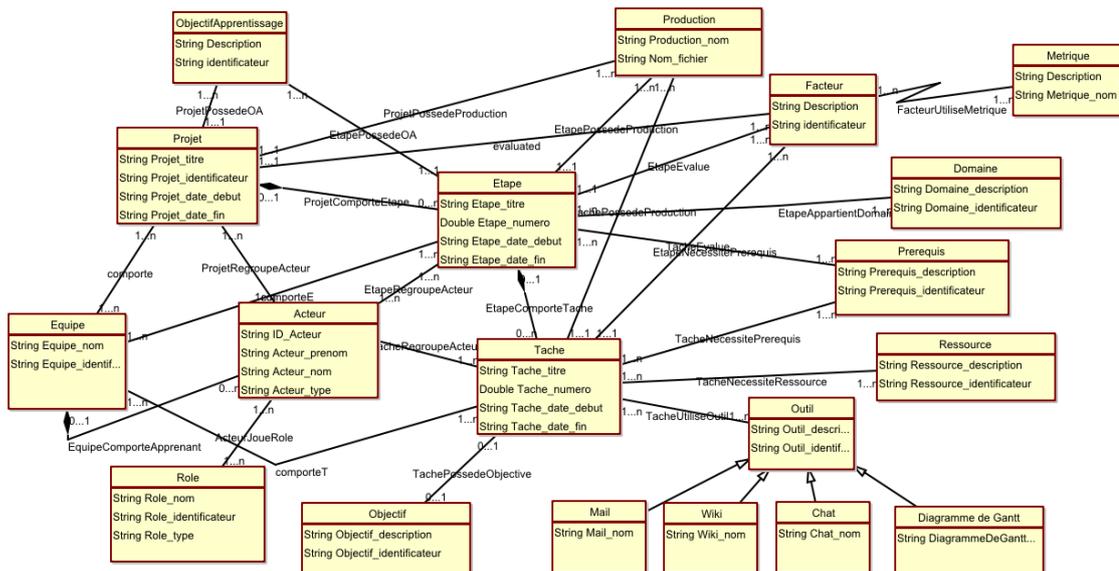


FIG. 4.8: Méta-modèle PPC représenté par ModX

Dans cette section, nous présentons l'outil utilisé pour représenter le méta-modèle PPC proposé qui permet d'élaborer les scénarios PPC qui sont conformes à ce méta-modèle. Parmi les éditeurs existants comme ECoS écrit en java fournis-

sant une interface partagée et collective de modélisation pédagogique, CopperCore [56] spécifique à IMS-LD etc. . . , nous avons choisi l'outil ModX [149] pour des raisons de facilité d'utilisation, richesse de représentation, et de compatibilité relative avec des outils de transformations de modèles utilisés dans le chapitre 6.

ModX est un outil graphique permettant la création de modèles et de méta-modèles basés sur MOF 1.4 [42]. Il utilise une notation graphique proche de l'UML et propose un schéma de persistance selon le format XMI 1.1 pouvant être compatible avec un grand nombre d'outils MDA ⁹. Pour élaborer le scénario PPC, il est nécessaire de créer au préalable le méta-modèle PPC (figure 4.8). Le scénario PPC, étant au niveau modèle est créé de manière graphique à l'aide de ModX sous le contrôle du méta-modèle que nous venons de créer (figure 4.8).

Nous montrons une partie du scénario PPC expliqué dans la section précédente. Dans la figure 4.9, nous montrons la partie correspondante à la première étape du projet (définition du projet) et les tâches qui composent cette étape. Dans la

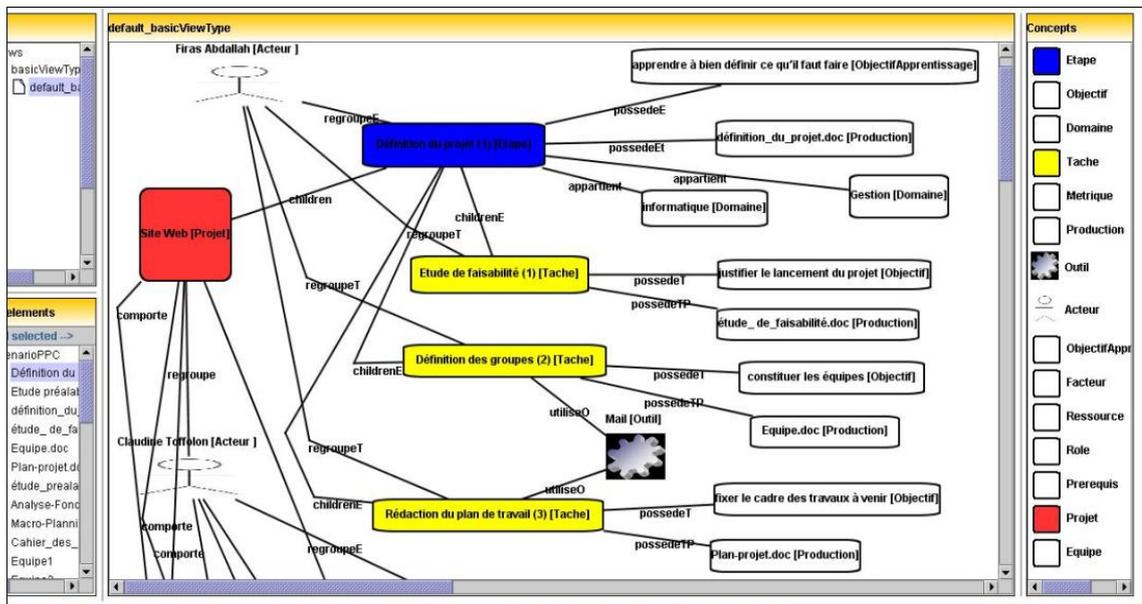


FIG. 4.9: Représentation d'une partie du scénario PPC

figure 4.10, nous montrons la partie du scénario PPC concernant la constitution des équipes composées des apprenants.

⁹Dans notre cas, le fichier XMI du scénario PPC généré par ModX est utilisé dans l'outil de transformation ATL expliqué en détail dans le chapitre 6

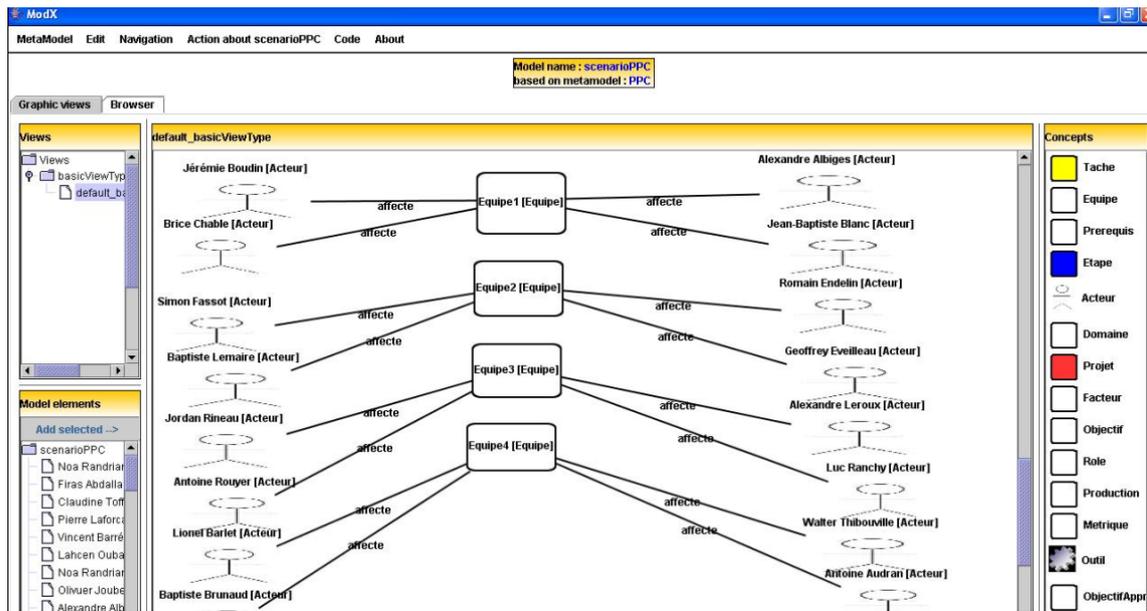


FIG. 4.10: Représentation des équipes du scénario PPC

4.6 Conclusion

Dans ce chapitre, nous avons présenté notre première contribution à cette thèse. Nous nous sommes intéressés dans la section 4.3 principalement à l'élaboration d'un méta-modèle permettant à l'enseignant concepteur de produire un scénario pédagogique d'une situation d'apprentissage dans un contexte d'une Pédagogie par Projet Collectif (PPC). Nous avons introduit ce méta-modèle selon trois points de vue. La vue générale (sous-section 4.3.1) permettant de préciser la **structure du projet** dans le contexte d'une Pédagogie par Projet Collectif (PPC). La deuxième vue présentée dans la sous-section 4.3.2, consiste en l'étude des **aspects d'apprentissage** dans une PPC proposant un cadre permettant la création d'une session de formation. La troisième vue présentée dans la sous-section 4.3.3, interprète l'aspect de **production** dans une PPC.

Afin de montrer que le méta-modèle PPC proposé est utilisable par un enseignant concepteur de la PPC, nous avons présenté (section 4.4) un scénario PPC élaboré à partir de ce méta-modèle. Ce scénario est destiné à un public constitué d'apprenants en première année de DUT Informatique à l'IUT de Laval.

Dans la section 4.5, nous avons présenté ModX, outil choisi qui nous a permis d'élaborer le méta-modèle PPC ainsi que le scénario PPC conforme. Cet outil permet de générer un fichier XMI du scénario élaboré. Les questions qui se posent maintenant sont : quelle plate-forme choisir pour déployer un scénario PPC ? Plus particulièrement lorsque l'on a le choix entre des plates-formes de formation

et des plates-formes de travail collectif. Comment déployer ce scénario dans la plate-forme choisie ? Ces deux importantes questions seront l'objet des chapitres suivants.

Chapitre 5

Aide au choix d'un environnement d'apprentissage dédié à la Pédagogie par Projet Collectif

5.1 Introduction

Un des problèmes majeurs pour les enseignants est de concevoir, adapter, ou simplement choisir une plate-forme ou système d'apprentissage assisté par les Technologies de l'Information et de la Communication (un site Web par exemple) supportant la pédagogie qu'ils désirent mettre en oeuvre. Ainsi, dans le chapitre précédent, nous avons proposé un méta-modèle PPC permettant à l'enseignant concepteur d'élaborer les scénarios PPC qu'il souhaite mettre en oeuvre. Nous avons expliqué les bases de construction de ce méta-modèle. Nous avons également montré l'élaboration d'un exemple de scénario PPC étant conforme à ce méta-modèle.

La première contribution dans cette thèse (chapitre précédent) a consisté à définir un méta-modèle pour la PPC. Ce méta-modèle, associé à un modèleur graphique comme ModX, permet d'encadrer et d'assister l'enseignant-concepteur dans l'élaboration de ses scénarios PPC. Cependant tout cela ne suffit pas pour que la formation basée sur la PPC soit diffusée à travers le Web par les plates-formes spécialisées. Pour que le scénario PPC soit "joué" par les acteurs de la Pédagogie par Projet Collectif (PPC), il doit être déployé et opérationnalisé dans un Environnement Informatique pour l'Apprentissage Humain (EIAH).

Le résultat final, la plate-forme doit concilier le point de vue pédagogique et le

point de vue technologique : un outil qui respecte les intentions pédagogiques de l'enseignant dans le contexte réel dans lequel il se trouve.

La stratégie de la thèse consiste à proposer à l'enseignant de choisir une plate-forme existante la plus adaptée à son scénario de Pédagogie par Projet Collectif (PPC), puis d'opérationnaliser son scénario dans la plate-forme choisie.

Nous avons remarqué dans la section 1.2 du premier chapitre qu'il existe des centaines de plates-formes commerciales et "Open Source" ¹, mais qu'aucune de ces plates-formes ne semble directement adaptée à une situation d'apprentissage dans un contexte de la PPC [87], [17]. Ce phénomène est d'ailleurs vrai pour n'importe quelle pédagogie. Le Collectif AS Plateforme [75] observe que l'extrême complexité et diversité des technologies et compétences à maîtriser rend difficile non seulement le développement et la mise en oeuvre d'un EIAH mais aussi, faute de capitalisation suffisante, sa réutilisation dans un contexte nouveau. Ces problèmes de développement et réutilisation sont d'ailleurs une problématique centrale dans les activités du génie logiciel [192], [96], mais ils pointent sans doute avec plus d'acuité dans le domaine nouveau et fondamentalement transdisciplinaire des EIAH.

Nous nous situons ainsi dans la stratégie de l'aide au choix et de l'adaptation de plate-forme plutôt que celle de la conception-développement. Notre stratégie consiste à assister l'enseignant pour l'implémentation de son scénario dans la plate-forme choisie, plutôt que développer une nouvelle plate-forme. Notre travail cherche à réutiliser ce que les plates-formes existantes proposent. Dans ce chapitre, nous posons la question suivante : comment choisir une plate-forme pour médiatiser ou outiller les scénarios PPC décrits avec le méta-modèle PPC proposé ? Dans cette optique, afin d'avoir une pertinence et une stratégie de réutilisation, nous proposons un modèle de fonctionnalités pour guider le choix des enseignants concepteurs afin qu'ils obtiennent une plate-forme la mieux adaptée à leurs besoins formulés à travers un scénario PPC. L'enseignant peut être amené à choisir une plate-forme parmi plusieurs mises à sa disposition ou bien évaluer dans quelle mesure une certaine plate-forme est bien ou mal adaptée à son besoin (scénario). Le choix d'une plate-forme la plus proche de la pédagogie à mettre en oeuvre permet de réduire les pertes sémantiques lors de l'opérationnalisation des scénarios. Dans la section 5.2, nous étudions des plates-formes de formations parmi les plus connues. Nous essayons de les étudier au regard de la Pédagogie par Projet Collectif (PPC). Dans les sous-sections 5.3.1 et 5.3.2, nous étudions les deux domaines de CSCW (*Computer Support Cooperative Work*) et CSCL (*Computer Support Cooperative Learning*). Dans la sous-section 5.3.3, nous étudions les liens entre ces deux domaines et traitons des exemples d'environnements informatiques de types CSCW et CSCL. Ces environnements sont étudiés aussi au regard de la Pédagogie par Projet Collectif (PPC). Dans la section 5.4, nous proposons un mo-

¹<http://thot.cursus.edu/>

dèle de fonctionnalités PPC dérivé à partir du méta-modèle PPC proposé dans le chapitre précédent. Ce modèle de fonctionnalités PPC va aider l'enseignant concepteur à choisir une plate-forme.

5.2 Plates-formes de formation

5.2.1 E-learning à la base de la plate-forme

Les progrès considérables des Technologies de l'Information et de la Communication appliquées à l'Education (TICE) ont permis l'émergence à grande échelle des EIAH et l'apparition de grands réseaux d'information, à la naissance de la e-formation ("*e-learning*" en anglais). Le "*e-learning*" est un terme anglophone pour e-formation. Il représente l'utilisation des nouvelles technologies multimédias et de l'Internet, pour améliorer la qualité de l'apprentissage. Notamment en facilitant l'accès à des ressources et des services, ainsi que la gestion des échanges et de la collaboration à distance entre apprenants. Le "*e-learning*" définit tout dispositif de formation qui utilise un réseau local, étendu ou l'Internet pour diffuser, interagir ou communiquer, ce qui inclut l'enseignement à distance, en environnement distribué, l'accès à des sources par téléchargement ou en consultation sur le Net. Il peut faire intervenir du synchrone ou de l'asynchrone, des systèmes tutorés, des systèmes à base d'autoformation, ou une combinaison de ces éléments évoqués. Le "*e-Learning*" résulte donc de l'association de contenus interactifs et multimédias, de supports de distribution (PC, Internet, Intranet, extranet), d'un ensemble d'outils logiciels qui permettent la gestion d'une formation en ligne et d'outils de création de formations interactives. L'accès aux ressources est ainsi considérablement élargi de même que les possibilités de collaboration et d'interactivité [72].

La e-formation est un cadre d'utilisation particulier des situations d'apprentissage au centre de nos préoccupations de recherche. Un dispositif privilégié de la e-formation est la **plate-forme**. Comme nous allons voir dans la sous-section 5.2.2, ce type d'environnement d'exécution semble cependant peu disposé à prendre en charge la mise en oeuvre et la réalisation d'apprentissage centré sur l'activité de l'apprenant comme dans les situations d'apprentissage dans un contexte de Pédagogie par Projet Collectif (PPC).

5.2.2 Définition d'une plate-forme de e-formation

Une plate-forme de e-formation, de FOAD (Formation ouverte et à distance), de formation à distance sont tous des termes référant la définition générale de dispositif de e-formation utilisant les réseaux informatiques comme support. Ces dispositifs de formation permettent de réaliser des formations ou des cours

avec des objectifs, des documents, des méthodes, des moyens humains, matériels et financiers. Une plate-forme d'e-formation regroupe les outils nécessaires à ses utilisateurs permettant d'incorporer des ressources pédagogiques multimédias, de participer à des activités et d'effectuer le suivi pédagogique et administratif des apprenants. L'intérêt d'une plate-forme réside dans l'intégration de fonctionnalités en une seule entité accessible de partout. Une plate-forme est utile si elle évite à ses utilisateurs de recourir à plusieurs outils séparés [156]. Dans sa fiche pratique sur les utilisations d'une plate-forme, Ecoutin définit la plate-forme par un logiciel assistant la conduite des formations ouvertes [71]. "*Une plate-forme de formation ouverte et à distance est un logiciel qui fournit aux trois principaux utilisateurs - formateur, apprenant, administrateur - un dispositif qui a pour premières finalités la mise à disposition, la consultation à distance de contenus pédagogiques, l'individualisation de l'apprentissage et le télé-tutorat. Autour de ces premières finalités, peuvent s'ajouter d'autres fonctionnalités et d'autres rôles : des fonctionnalités relatives aux référentiels de formation et à la gestion de compétences, aux catalogues de produits de formation, au commerce électronique, à la gestion administrative, à la gestion des ressources pédagogiques, à la gestion de la qualité de la formation ; des rôles d'administration des matériaux pédagogiques, des rôles d'administration de la scolarité ou de la formation, etc ...*" [71]. La plate-forme utilise plusieurs médias et modes de communication afin d'enrichir les procédures d'échange de données avec d'autres systèmes d'information. Cela est dû à l'évolution des techniques, des infrastructures de réseau et des normes [71], [155].

Le marché des plates-formes de e-formation est très divisé et on compte plus de 300 plates-formes différentes (<http://thot.cursus.edu/>) dont un certain nombre en Open-Source : Ganesha, Moodle, uPortal, ou encore Claroline (une liste plus exhaustive est disponible à l'adresse http://www.fnl.ch/LOBs/LOs_Public/OpenSourcePlatf.html). Nous notons que nous écartons de cette définition certains types d'outils parfois appelés plates-formes comme les portails de formation qui ne sont que de simples sites regroupant des offres de formation.

5.2.3 Types de plates-formes de formation

Nous remarquons dans la littérature qu'il existe plusieurs acronymes désignant les plates-formes de formation. Ces différents acronymes correspondent à plusieurs objectifs d'utilisation de ces dispositifs. Nous citons :

LMS : système de gestion d'apprentissage (*Learning Management System*). Il est considéré dans bien des cas comme le coeur du dispositif de la e-formation [126]. Il permet de gérer et de suivre toutes les activités de formation d'une organisation. Ce terme anglo-saxon désigne le système informatique conçu pour optimiser, sur un réseau Internet ou Intranet, la gestion de l'ensemble des activités de formation,

depuis l'information sur l'offre, l'inscription des participants, la distribution des ressources, l'organisation de parcours individualisés, le suivi par le tuteur et du tutorat (gestion intégrée des interactions apprenants-formateur), l'animation de communautés d'apprentissage. Selon leur conception, elles favorisent l'entrée par les contenus ou les compétences, le travail collaboratif ou individuel, l'acquisition de compétences ou l'organisation de connaissances [72]. Ganesha, WebCT, Claroline, Open USS sont des exemples des LMS existants.

CMS : systèmes de gestion de contenu (*Content Management System*). Ces systèmes sont une famille de logiciels de conception et de mise à jour dynamique de site web ou d'application multimédia ². Ils ont pour but de simplifier la création et la gestion du contenu en ligne. Ceci repose sur deux principes essentiels de fonctionnement des CMS [126] :

- La séparation entre le fond et la forme pour permettre aux auteurs de se concentrer uniquement sur leur contenu. Pour se faire, ils disposent de modèles de présentation prédéfinis spécifiques à chaque élément composant le document (en-tête, format du titre, emplacement d'une image, intégration d'un fichier multimédia, etc.). Par suite, l'auteur intègre son contenu dans cette architecture.
- ces systèmes induisent des procédures de publication des contenus. Avant la publication, il y a deux étapes qui sont : la création, et la validation [126]. Les CMS les organisent selon les règles propres à l'entreprise ou l'entité l'utilisant.

Parmi les CMS existants, nous citons :

- Joomla incluant des fonctionnalités telles que des news, une version imprimable des pages, des blogs, des sondages, des recherches etc. . .
- Drupal possédant un très grand nombre de modules pour gérer les publications, l'organisation et les droits d'accès d'une communauté d'utilisateurs.
- SPIP permettant de créer un site Web, contient une grande liste de fonctionnalités dont la plus importante est que sa conception favorise l'accueil de rédacteurs du site dont il ne sera demandé aucune autre compétence que de connaître son navigateur.
- Lodel, Jaws, MySource, Textpattern, etc. . .

LCMS : système de gestion de contenu d'apprentissage (*Learning Content Management System*). Un LCMS est un système (le plus souvent basé sur les technologies Web) qui permet de créer, valider, publier et gérer des contenus d'apprentissage [72]. Combinant les fonctionnalités des LMS (les *Learning Management Systems*, ou systèmes de gestion de la formation) et des CMS (*Content Management Systems*), ils forment des systèmes complets de création et de gestion de contenus adaptés aux problématiques de la formation ouverte et à distance (FOAD). Les LCMS intègrent toutes les fonctionnalités de base d'un LMS. Les

²Wikipédia 2008

LCMS et les LMS (les plates-formes e-learning) sont cependant deux types d'outils différents. La confusion vient du fait que, souvent, les LCMS intègrent toutes les fonctionnalités de base des plates-formes : individualisation et suivi des parcours d'apprentissage ; outils de communication (messagerie, forums) ; et gestion des groupes d'apprenants. Les LCMS s'appuient sur le modèle des *Learning Objects* (LO) nommés en français "objets d'apprentissage". Un *Learning Object* est composé d'objectifs de formation, d'évaluations et de contenus. Des données appelées "metadata" y sont associées. A partir de ces données, on peut individualiser les contenus selon les profils des apprenants. Les LCMS permettent la création des bibliothèques de LO, une bibliothèque de grains de contenu de formation indépendants, pouvant être réutilisés et associés indifféremment les uns aux autres. La logique et la capacité d'individualisation intégrées au niveau des LO sont répercutées au niveau de la publication par le biais de ces méta-données. Les LCMS peuvent donc, pour un apprenant donné, gérer la distribution et le suivi de l'apprentissage à un niveau très fin : celui de l'objet d'apprentissage [126].

La ressemblance entre les LCMS et les CMS se manifeste dans la gestion des procédures et des flux de publication en assurant la mise en place d'une organisation garantissant le respect des règles de publication. L'auteur crée un LO et le soumet à la procédure pour validation. Il sera publié s'il est approuvé et rejeté dans le cas contraire pour modification.

5.2.4 Exemples de plates-formes de e-formation

Dans cette sous-section, nous étudions des exemples de plates-formes de e-formation au regard de la Pédagogie par Projet Collectif (PPC).

Il existe de nombreux plates-formes informatiques supportant l'enseignement et l'apprentissage. Une grande partie parmi elles réalise des approches considérant que les plates-formes sont "*orientées pédagogie*". Ces approches "*se focalisent sur comment organiser et enseigner des contenus bien identifiés*" [198]. Ces plates-formes permettent aux apprenants d'accéder à des informations en ligne déjà préparées en parcourant ou suivant une structure prédéfinie de l'information. D'un autre côté, il existe des plates-formes "*orientées performance*" se focalisant sur la réalisation d'activités par les apprenants [198].

Comme exemples de plates-formes, nous avons choisi de présenter de manière successive les plates-formes suivantes :

ARIADNE : la plate-forme ARIADNE, née en 1995, résulte d'un projet européen réunissant 24 partenaires et 8 sponsors de sept pays européens. Il est soutenu par le programme de la communauté européenne pour les applications de la télématique et la Suisse. L'originalité d'ARIADNE est de fournir un vivier de documents pédagogiques, réparti sur 15 serveurs en Europe, alimenté et partagé par l'ensemble des adhérents de l'association ARIADNE (*Alliance of Remote Instructional Authoring and Distribution Networks for Europe*). ARIADNE propose

un ensemble d'outils pour la mise en oeuvre de formations à distance. Cela en se basant sur une architecture de système d'information (et de formation) distribuée et collaborative, [33] :

- Le Segmenteur de textes et générateur d'hypertextes pédagogiques : ce segmenteur permet de marquer sémantiquement un texte tout en s'appuyant sur une méthodologie relativement simple.
- Le Segmenteur de séquences vidéo : c'est un outil qui permet de sélectionner et de numériser des vidéo-clips et de leur adjoindre un en-tête pédagogique.
- Le Générateur de simulations pédagogiques.
- Le Générateur d'exercices d'auto-évaluation : utilisant un modèle pédagogique développé à Grenoble. Ce modèle se base sur la capacité de l'apprenant à s'autoévaluer.
- Le Générateur de questionnaires : est un système complet pour inclure des questions de types différents.

LearningSpace : la plate-forme *LearningSpace* est éditée par Lotus, filiale de la société IBM [111]. Cette plate-forme est une application qui s'appuie sur les fonctionnalités fournies par *Notes/Domino* et qui permet la création, la diffusion et le gestion de contenus et d'activités pédagogiques. *LearningSpace* propose cinq principaux outils [87] :

- le *calendrier* qui présente un emploi du temps personnalisé ;
- le *centre des médias* contenant le matériel des cours (page Web) ;
- la *salle de classe* sous forme d'un forum de discussion ;
- le *gestionnaire des profils* contenant les informations sur les apprenants ;
- le *gestionnaire d'évaluations* qui permet aux enseignants de noter et de commenter les travaux des apprenants.

La gestion des utilisateurs s'effectue par personne, par classe correspondant à un cours ou par profil type. Quatre rôles sont prévus : administrateur, concepteur, instructeur, stagiaire [33]. La participation à *LearningSpace* consiste à suivre des cours proposant des activités (items) programmées (sessions) ou à programmer. Ces activités supposent la consultation de ressources et la réalisation de travaux. La participation aux discussions (forum ou chat) avec les autres stagiaires de la classe permet de partager des textes, documents. Les stagiaires peuvent ouvrir et mener des discussions. D'un autre côté, *LearningSpace* ne soutient pas une coopération et collaboration entre les apprenants faute de manque d'outil de communication synchrone [32].

LearningSpace est un produit commercial [111] pour le travail à distance qui a été adapté à l'enseignement. Cette plate-forme a été à l'origine conçue pour être utilisée par les entreprises pour leurs besoins internes de formation de leurs clients et partenaires.

Campus Virtuel : le *Campus Virtuel* propose une approche intéressante pour le support aux activités collectives. Pour son développement un accord a été passé

entre le laboratoire Trigone de l'Université de Lille [109] et la société Archimed ³. Dans cette plate-forme, nous trouvons la notion de groupe supportant tout type d'apprentissage collectif [109] : résolution collective de problème, étude de cas ou tutorage à distance. George [87] reproche à cette plate-forme de ne pas guider les apprenants pour un type d'activité collective particulier et qu'ils doivent prendre en charge la gestion du travail collectif. Il n'y a pas d'outil de communication synchrone interne au *Campus Virtuel* mais malgré cela, il y a la possibilité de lancer *Netmeeting* à partir de cette plate-forme. George [87] pense que cette plate-forme "*demande aux apprenants certaines compétences pour travailler collectivement et un certain niveau en utilisation d'outils informatiques.*"

5.2.5 Discussion

Nous remarquons d'après l'étude de ces plates-formes que la plupart d'entre elles proposent des outils de communication entre apprenants ou bien entre participants tels que les forums de discussions et les messageries électroniques, etc. Cependant, ces plates-formes mettent davantage l'accent sur la diffusion des cours que sur la mise en place d'activités pédagogiques collectives [87]. "*... les plates-formes de téléformation ... sont avant tout conçues comme des outils pour la gestion de cours par correspondance électronique (...), avec une dose plus ou moins forte de travail collaboratif (collectif) (plutôt moins que plus)...*". Par exemple *WebCT*, *FirstClass*, *WebTutor*, ou *Librarian* sont des plates-formes fondant essentiellement leurs modèles pédagogiques sur des ressources pédagogiques (principalement des pages de cours au format html) à fournir aux apprenants tout en permettant une gestion administrative de ces derniers [87].

Les plates-formes *ARIADNE*, *LearningSpace*, *Campus Virtuel* permettent davantage des activités collectives entre apprenants. Ces plates-formes se contentent de fournir des outils de communication et de travail aux apprenants mais ne sont pas vraiment conçues pour soutenir la mise en place et le déroulement d'activités pédagogiques collectives [87].

Ces plates-formes nécessitent des enseignants la mise en place des activités collectives, et des apprenants des compétences dans la gestion de ces activités pour les réaliser. Ces plates-formes ne supportent pas un type particulier d'activité mais elles permettent d'effectuer un large éventail d'activités. Cameron *et al.* [40] estiment, concernant ces plates-formes, que les méthodes pédagogiques sont altérées par la nécessité de s'adapter aux technologies.

Pour répondre aux insuffisances de ces plates-formes pour supporter les activités collectives, Martin Dougiamas, auparavant administrateur de la plate-forme *WebCT* à l'Université de Curtin en Australie a créé la plate-forme *Moodle* ⁴. Cette plate-forme a été proposée comme une plate-forme répondant aux besoins

³<http://www.archimed.fr/>

⁴Wikipédia 2008

de situations d'apprentissage dans un contexte collectif permettant des interactions entre des pédagogues, des apprenants, et des ressources pédagogiques. Elle intègre des outils et des activités d'apprentissage orientés sur : les communications synchrones et asynchrones, et l'apprentissage collectif. Nous présenterons encore plus de détails sur Moodle en fonction du modèle de fonctionnalités PPC élaboré dans la section 5.4.

Afin de soutenir l'apprentissage collectif, un domaine de recherche a émergé : le domaine des CSCL (*Computer-Supported Collaborative Learning*) en relation étroite avec le domaine du CSCW (*Computer-Supported Cooperative Work*).

5.3 Les deux domaines du CSCW et du CSCL

Le terme CSCL est considéré comme une dérivation du terme CSCW qui ne convenait pas pour certaines recherches. Les deux champs de recherche (CSCL et CSCW) se ressemblent et ont beaucoup en commun. Ces deux champs peuvent profiter de leurs apports mutuels en vue de mieux comprendre et mieux soutenir les interactions humaines. Cependant, nous remarquons qu'il y a des particularités et des spécificités propres à chacun d'eux. Tout d'abord, nous commençons par le domaine du CSCW, puis celui du CSCL pour terminer par l'étude des liens entre eux.

5.3.1 CSCW (Computer-Supported Cooperative Work)

Le terme CSCW (*Computer Supported Cooperative Work*) a été défini par Paul Cashman et Irène Grief en 1984 lors d'un workshop organisé dans le but de regrouper des personnes de différentes disciplines s'intéressant à la façon dont les hommes travaillent et comment la technologie pourrait les supporter [101]. Ils ont inventé ce terme pour décrire cet intérêt commun. Depuis la date de cette conférence, de nombreux chercheurs et professionnels ont adopté le terme CSCW et ont suivi des conférences le concernant.

Le terme CSCW est très général puisqu'il s'intéresse à la fois à l'étude de la nature des groupes de travail et à celle des technologies informatiques permettant de soutenir ces groupes. Ce terme est connu en français par le sigle TCAO (Travail Coopératif Assisté par Ordinateur) [76] qui est "*Une application interactive multi-participant grâce à laquelle les hommes peuvent réaliser une tâche "en commun" à partir de leurs postes de travail respectifs*" [62].

L'objectif du CSCW est d'assurer à un ensemble d'acteurs de travailler via une structure informatique [15]. Mais la question consiste maintenant à savoir comment les technologies informatiques peuvent être utilisées pour améliorer les relations de travail dans le cadre collectif (collaboratif et/ou coopératif) et comment l'ordinateur peut servir à minimiser les problèmes logistiques et humain du travail collectif [13]. Le domaine de CSCW englobe plusieurs domaines de recherche

comme l'informatique, la psychologie, la sociologie et l'ethnologie [87]. C'est un domaine interdisciplinaire se concrétisant par la rencontre entre les sciences de l'ingénieur et les sciences humaines [138]. Il représente un fusionnement des personnes, des thèmes, des approches et des langues [101].

Selon Grudin [100], le CSCW est défini très différemment par les chercheurs selon leur domaine d'appartenance (informatique, sciences humaines, etc...). Certains considèrent la technologie permettant d'accéder à des fichiers centraux communs comme des systèmes de CSCW (Crowley in Ensor, 1990) cité dans [47]; d'autres considèrent les systèmes d'e-mail comme des systèmes de CSCW. Bannon et Schmidt (1989) cités dans [47] pensent que les systèmes de CSCW doivent être basés sur un arrangement des aspects coopératifs du travail à soutenir et également le refléter. Dans un autre sens, l'e-mail n'identifie aucun rôle au delà du récepteur et de l'expéditeur et par suite ne doit pas être considéré comme un système de CSCW.

Le domaine de recherche de CSCW (*Computer-Supported Cooperative Work*) peut être défini comme une tentative d'explorer comment la technologie de l'information peut être conçue et employée afin de permettre à des personnes impliquées dans des activités coopératives d'accomplir leur travail plus efficacement [182].

Schmidt et Bannon (1992) cités dans [47] discutent une approche pour le CSCW qui se base sur le besoin de comprendre la nature du travail coopératif dans l'ordre d'établir un fondement pour la conception de système d'information support de ce travail coopératif. "*CSCW should be conceived of as an endeavour to understand the nature and requirements of cooperative work with the objective of designing computer-based technologies for cooperative work arrangements.*"

Les activités du CSCW peuvent être caractérisées selon la matrice espace/temps d'Ellis [74] (un axe temps composé des caractéristiques "asynchrone" et "synchrone" et un axe espace composé des caractéristiques "local" ou "distant") ou selon la situation des systèmes dans le trèfle d'Ellis (Coordination, Communication, Production) (figure 5.1). L'objectif commun des travaux du CSCW est de permettre à un groupe d'individus de communiquer, de produire ensemble et de se coordonner. Cette conceptualisation (les trois ensembles de cette figure) permet d'analyser les activités en fonction des outils utilisés et des objectifs visés.

Collet *et al.* [53] ont identifié quatre objectifs qu'un environnement de développement de CSCW doit respecter :

- Faciliter la modélisation de systèmes interactifs pour l'activité collective spécialisée dans un domaine donné. Les services de base inclus dans un environnement CSCW doivent offrir une malléabilité organisationnelle et fonctionnelle. Un cadre de collaboration doit faciliter le déploiement, la sécurité et l'administration de l'application dans le contexte inter-organisationnel.
- Soutenir le processus de Co-évolution : les utilisateurs peuvent modifier collectivement leur environnement de fonctionnement. Les développeurs et

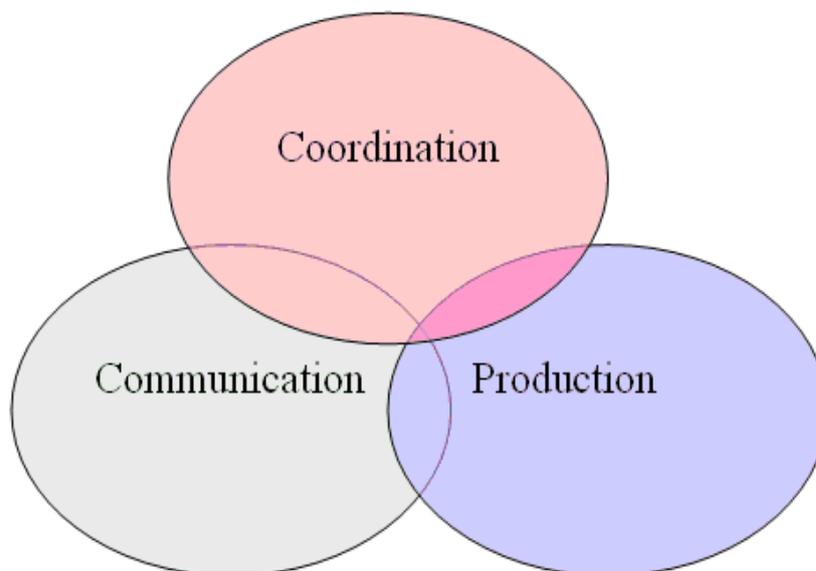


FIG. 5.1: Trèfle fonctionnel des collecticiels

les utilisateurs doivent coopérer à concevoir le système.

- Offrir un service de partage de ressources. Un tel service est utile dans le cas de différentes activités partageant les mêmes ressources. Ce service inclut un "moteur de conscience de coopération" (*cooperation awareness engine*). De cette façon, l'utilisateur peut améliorer le suivi de l'activité.
- Faciliter l'intégration des objets nomades (*nomad object*) et renvoyer les services nomades attendus (*nomad awaited services*).

Une partie des travaux sur les systèmes et outils de CSCW portent sur les collecticiels (*groupware*) et d'autres sur les *workflows*.

Collecticiels (groupware)

Collecticiels ("*groupware*" ou "*workgroup computing*") sont des termes qui décalent la focalisation du travail supporté vers la technologie, et suggèrent de petites unités d'organisation; "*Groupware*" or "*workgroup computing*" are terms that shift the focus from the work being supported to the technology, and suggest small organizational units." [101]. Un collecticiel (ou *groupware*) désigne un système informatique qui assiste un groupe d'utilisateurs à la réalisation d'un projet commun, d'une tâche commune, et qui fournit une interface à un environnement partagé [74]. "*Le groupware est l'ensemble des technologies et des méthodes de travail associées qui, par l'intermédiaire de la communication électronique, permettent le partage de l'information sur un support numérique à un groupe engagé dans un travail collaboratif*" [Courbon & Tajan] cité dans [78]. Cette commu-

nication électronique qui est normalement faite entre des acteurs à distance est assurée soit au même moment (activité synchrone), soit à des moments différents (activité asynchrone).

Benali *et al.* [15] proposent trois types de collecticiels : les outils élémentaires, les collecticiels centrés tâches, et les collecticiels centrés modèles :

- Les outils élémentaires sont les outils de communication classiques fréquemment utilisés : chat, forum, mail, tableau blanc partagé, etc ;
- Les collecticiels centrés tâches correspondent aux collecticiels organisés autour de la nature de la tâche ;
- Les collecticiels centrés modèles correspondant aux collecticiels basés sur des modèles de l'activité conjointe : théorie de l'activité, modèles de participation, etc.

Workflows

Les systèmes de management de *workflow* décrivent les technologies supportant les processus du groupe dans un contexte d'organisation particulier ou spécifique à une méthodologie de travail [101]. Le glossaire du *Workflow Management Coalition* ⁵ définit le *workflow* par : "*The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.*" [52]. Le *workflow* est un système s'occupant de l'automatisation de procédures durant lesquelles des documents, des informations, ou des tâches sont transmis d'un participant à un autre selon un ensemble de règles dans le but d'accomplir et de réaliser un objectif global. Il permet un enchaînement automatique des opérations d'une activité complexe.

Les participants peuvent être à la fois humains et non-humains (traitement automatique d'un document assuré par l'ordinateur) [207]. La procédure automatisée par le système *workflow* est un ensemble coordonné d'activités qui sont entrelacées de façon à atteindre un objectif commun. Chaque activité d'une procédure définit le travail à faire formant une étape logique à l'intérieur d'une procédure. Selon Saikali & David [179], une partie des chercheurs considèrent les *workflows* comme un sous-ensemble des collecticiels et une autre les séparent. Les collecticiels sont plutôt orientés vers le travail collectif dont ils cherchent à couvrir les aspects synchrones ou asynchrones, tandis que les *workflows* concernent des cas asynchrones et aident plus la coordination et l'organisation ; c'est l'exemple du contrôle et de l'automatisation des processus d'entreprise.

Nous remarquons que les collecticiels et les *workflows* se différencient au niveau des mécanismes de synchronisation et de coordination [17]. En ce qui concerne les collecticiels, la synchronisation et la coordination sont faites au niveau de l'acti-

⁵<http://www.wfmc.org/>

tivité commune, mais dans les *workflows* ces mécanismes se situent au niveau du processus.

5.3.2 CSCL (Computer-Supported Collaborative Learning)

Le domaine du CSCL *Computer-Supported Collaborative Learning* n'est pas strictement défini parce qu'il regroupe une multitude d'approches de recherches de différentes origines. Le sigle de CSCL est apparu dans un Workshop d'une conférence en 1989 [12]. Ce sigle est traduit en français par : Apprentissage Collaboratif Assisté par Ordinateur (ACAO) [109] ou par Environnements Informatiques d'Apprentissage Coopérant (EIAC) [11], ces traductions étant peu employées dans le milieu francophone [87]. Nous choisissons le sigle CSCL. Le travail dans les CSCL a pour objet l'étude de l'utilisation des technologies pour la collaboration/coopération dans l'éducation. C'est l'utilisation de l'informatique pour soutenir l'apprentissage collaboratif ⁶.

Le workshop dédié au CSCL en 1991 [17] avait pour objectif de réunir des personnes travaillant dans des domaines divers mais s'intéressant à l'apprentissage collaboratif par le biais d'une pratique d'expériences en classe, des approches pédagogiques, ou du développement du logiciel. La première conférence internationale dédiée à ce domaine s'est déroulée à Bloomington (Indiana) en 1995.

Dans ce domaine, les chercheurs ont plusieurs préoccupations : l'apprentissage (*Learning*), l'apprentissage collaboratif (*Collaborative Learning*), le support soutenant cet apprentissage collaboratif (*Supported Collaborative Learning*), et la machine ou l'ordinateur (informatique) assurant ce support et ce soutien (*Computer*).

Le domaine des CSCL ne se résume pas au développement des outils informatiques mais aussi à la conception de l'activité pédagogique collective. Cette dernière crée une interdépendance entre les apprenants car dans une activité collective la réussite du groupe dépend de celle de chaque individu du groupe surtout si les rôles des membres de groupe sont bien structurés. Cette interdépendance est un facteur de cohésion dans un groupe favorisant l'apprentissage [115]. Salomon [180] affirme la nécessité d'interdépendance entre les apprenants parce qu'il considère que l'interdépendance possède les caractéristiques suivantes : une nécessité de partager de l'information, une division du travail dans laquelle les rôles de chaque membre se complètent, et un besoin d'une activité de réflexion commune qui peut être explicitée, examinée, modifiée et améliorée par ses membres. Le système informatique tout seul n'est pas capable de stimuler et produire cette interdépendance. C'est pour cela qu'il ne faut pas se préoccuper seulement des technologies pour favoriser l'apprentissage collectif mais aussi de l'activité collective à mettre en place avant de définir les spécifications informatiques du système qui doit soute-

⁶Le terme "collaboratif" est en fait utilisé sans prendre en compte vraiment la distinction entre les deux termes "collaboratif" et "coopératif"

nir ces activités. Le domaine de CSCL s'intéresse à étudier comment l'ordinateur peut être utilisé pour soutenir l'apprentissage collaboratif en présentiel ainsi qu'à distance.

La base pédagogique sur laquelle repose le CSCL est l'approche constructiviste et socio-constructiviste [17] qui encourage les interactions entre les apprenants et l'apprentissage par la pratique. Cette approche induit une pédagogie basée sur l'interaction entre l'apprenant et son environnement, plutôt que la transmission d'un savoir pré-construit d'un enseignant à un apprenant comme dans les plates-formes de formation classiques étudiées dans le chapitre 2.

5.3.3 Liens entre CSCW et CSCL

De nombreux chercheurs considèrent le CSCL comme un sous domaine du CSCW. Ils considèrent que CSCL met en oeuvre les technologies de CSCW dans un contexte d'apprentissage et par suite l'éducation est considérée comme un domaine de recherche appliqué au CSCW. Ces chercheurs voient une relation d'inclusion entre les deux domaines CSCW et CSCL. Croisy [57] considère que l'apprentissage coopératif est un exemple de travail coopératif. Mais George [87] n'adopte pas ce point de vue puisqu'il considère qu'il y a un risque pour les chercheurs du domaine CSCL de se concentrer sur les moyens technologiques et non pas sur les activités d'apprentissage. Bannon [12] soutient également ce point de vue : "*there is a tendency to focus too much on features of the technology per se, and not on the learning activity.*"

De notre point de vue, il nous semble effectivement que ces deux domaines (CSCW et CSCL) ont de nombreux points communs. Mais, nous ne souhaitons pas simplifier le domaine des CSCL à un sous-domaine du CSCW. Chacun de ces deux domaines est appliqué dans un contexte différent de l'autre : le CSCL est appliqué dans un contexte d'apprentissage tandis que le CSCW est appliqué dans un contexte industriel. Faerber [76] a expérimenté l'adaptation des collecticiels à la formation à distance en utilisant le collecticiel BSCW ⁷. Cette expérimentation n'a pas été entièrement satisfaisante pour plusieurs raisons [76] :

- l'obstacle technique à vaincre par la communauté éducative pour s'adapter à un tel collecticiel est trop important ;
- l'adjonction d'un collecticiel aux autres outils nécessaires dans les formations conduit à un dispositif hétéroclite dont les éléments ne sont pas intégrés et interoperables ;
- l'interface qui en résulte manque d'unité. Chaque outil possède sa présentation des fonctionnalités et requiert une adaptation constante des utilisateurs qui ne voient pas la continuité qui garantit une logique générale au dispositif.

En CSCW, le but est centré sur l'efficacité de l'accomplissement de la tâche, tandis qu'en CSCL, le but porte sur l'efficacité de l'apprentissage. L'activité collective

⁷basic support for cooperative work

est un "levier" pour réaliser cet apprentissage [87]. Dans sa thèse, Heeren [102] résume ainsi cette différence :

- L'intérêt principal du CSCW porte sur l'efficacité dans l'obtention des produits externes.
- L'intérêt principal du CSCL porte sur l'efficacité du processus d'apprentissage individuel et collaboratif.

Cette différence est due au fait que les recherches du domaine du CSCW visent la réduction de l'effort mental, et les recherches du domaine du CSCL se concentrent sur l'augmentation de l'effort mental [102].

Les environnements CSCL se différencient de ceux de CSCW par le fait qu'ils doivent faciliter la création et la mise en place de l'activité collective. Ces environnements doivent favoriser la sensation d'appartenance à un collectif chez les apprenants. Les environnements CSCL possèdent une structuration plus forte des activités par rapport aux environnements CSCW. L'environnement de CSCL est plus actif que celui du CSCW dans le sens qu'il cherche à guider l'activité collective des apprenants [124]. D'un autre côté, les environnements de CSCL doivent être faciles à utiliser et à prendre en main pour ne pas prendre une longue période de formation ce qui implique que les outils informatiques utilisés dans un contexte éducatif doivent être simples et intuitifs [87].

Dans le cadre de notre travail, nous nous intéressons au travail collectif entre apprenants, de ce fait nous considérons que notre travail de recherche est plus proche du domaine du CSCL. Cependant, le domaine des CSCW nous intéresse parce qu'il permet de mieux mettre en évidence les aspects de coordination et d'organisation, aspects fondamentaux dans la Pédagogie par Projet Collectif (PPC). En ce qui nous concerne, nous pensons que la conception d'une activité collective dans un contexte d'apprentissage ne se réduit pas simplement à une adaptation des principes du travail collectif. Le processus de conception et les spécifications de l'environnement informatique support sont conditionnés par la prise en compte du contexte d'apprentissage.

Afin de donner des exemples d'environnements CSCL et CSCW existants, nous proposons d'étudier quelques environnements informatiques dédiés au travail collectif de type CSCW et d'autres utilisés dans un contexte d'apprentissage de type CSCL ⁸ :

Belvedere : Belvedere est un logiciel *open source* conçu pour supporter les scénarios d'apprentissage collectif en situation de résolution de problèmes [55] par l'observation et la construction de cartes conceptuelles. L'objectif de cet environnement est le développement de compétences de recherche ("*critical inquiry skills*").

Belvedere est un environnement conçu non pas en fonction du domaine d'application mais en fonction des compétences visées. Il repose plutôt sur le concept

⁸Ces environnements sont cités dans la thèse de M. L. Betbeder [17]

de collaboration, les différents membres du groupe participent ensemble à l'évolution d'une tâche. Sa particularité est d'organiser un discours afin de favoriser l'argumentation, c'est-à-dire écrire / schématiser un discours dans une interface informatique. Le système ressemble à un logiciel de dessin. Les arguments sont représentés par des boîtes et sont reliés les uns aux autres par des flèches. Il est possible de typer ces flèches afin de spécifier diverses relations entre les propositions.

SPLACH : SPLACH [87] est un environnement support de projets collectifs intégrant un système d'analyse et d'assistance aux activités collectives. Il est conçu pour permettre la mise en oeuvre d'une pédagogie par projet dans un contexte de distance. Le groupe d'apprenants est réparti en équipe de trois apprenants et le tuteur de l'activité joue le rôle de chef de projet. Un projet est structuré en phases synchrones et asynchrones. Il s'agit d'un environnement dédié à l'apprentissage.

Memo-Net : Memo-Net [136] [137], développé en Lotus Notes Domino, est un collecticiel permettant à un groupe de résoudre d'une façon coopérative des problèmes rencontrés au cours d'un problème de conception, en alternant des phases de synthèse (élaboration d'une solution) et d'analyse (diagnostic d'une situation). Par la structuration des échanges, l'objectif de ce collecticiel est de guider le processus de résolution de problèmes et d'aider les utilisateurs à converger vers la meilleure solution possible à un problème donné. Cet environnement permet :

- de structurer les interactions en proposant une typologie : avant d'intervenir, les participants doivent choisir un type d'argument correspondant au "rôle" de leur interaction ;
- une meilleure représentation des relations entre les catégories d'intervention (visualisation de la caractéristique de "pré-requis" d'une intervention sur une autre).

Bien que ce collecticiel puisse, selon ses auteurs, être utilisé dans un contexte d'apprentissage, il n'est cependant pas conçu pour ce type de contexte.

BSCW - Synergia : BSCW (*Basic Support for Cooperative Work*) est un serveur Web qui permet le travail collaboratif [206]. La métaphore centrale de ce système est centrée sur l'espace de travail partagé. Les espaces de travail partagés peuvent contenir des informations de divers types tels que des documents, des images, des liens vers des pages Web, des discussions ou bien des informations pour contacter un membre du groupe de travail. Le système propose des fonctionnalités d'authentification (pour accéder aux espaces de travail), et de gestion des droits d'accès sur les objets (contrôle complet ou partiel), des forums de discussion, des facilités de recherche de documents, des conversions de format de documents, le contrôle des versions de documents, le support multi langage et des informations de conscience mutuelle.

Ce type de logiciel propose un cadre pour mettre en place des activités collectives. L'utilisation de ce type d'environnement pour supporter une ACCA (Activité Col-

lective en Contexte d'Apprentissage) n'est cependant pas particulièrement simple [76].

A partir de ce système, une version destinée à l'apprentissage a été développée, ce logiciel s'appelle BSCL (*Basic Support for Cooperative Learning*) et a été intégré au système Synergia [191]. Synergia a été conçu pour supporter la construction de connaissances collaboratives. Il comprend, en plus du composant asynchrone BSCL, un autre composant synchrone appelé MapTool. L'environnement Synergia est composé d'un espace de travail structuré et partagé au sein duquel l'apprentissage collectif est favorisé par le partage de documents, la sauvegarde de discussions et le développement de représentations de connaissances.

Fle3 : Fle3 [132] est un environnement d'apprentissage accessible par le Web. Il est développé par l'Université de Helsinki et conçu pour supporter des activités centrées sur le groupe dont l'objectif est de créer et développer des représentations de connaissances. Fle3 contient trois outils d'apprentissage et plusieurs outils d'administration. L'outil "*WebTops*" peut être utilisé par les enseignants et les apprenants pour stocker différents documents (fichiers, liens, notes) les organiser et les partager avec d'autres. L'outil "*Knowledge Building*" supporte les dialogues de construction de connaissances et les débats en les sauvegardant. Le troisième outil "*Jamming*" est un espace partagé permettant la construction collaborative d'images, de texte ou de vidéo. Pour les enseignants et les administrateurs, Fle3 offre des outils de gestion d'utilisateurs et de projets.

5.4 Aider au choix d'un système d'apprentissage : Le modèle de fonctionnalités

Un modèle pour l'assurance de la qualité est un ensemble normalisé ou sélectionné d'éléments de système qualité associés pour satisfaire les besoins d'assurance de la qualité dans une situation donnée (ISO 8402). Pour assurer la meilleure qualité d'une situation d'apprentissage de Pédagogie par Projet Collectif (PPC), nous proposons un modèle qualité appelé modèle de fonctionnalités qui a pour objet de représenter les fonctionnalités PPC associées aux concepts (et leurs relations) définis par le méta-modèle proposé. Ce modèle de fonctionnalités permettra ensuite de caractériser une plate-forme pédagogique par rapport à des besoins spécifiques identifiés par l'enseignant-concepteur ou son équipe à travers le scénario pédagogique élaboré à partir de ce meta-modèle.

Des travaux similaires ont été menés dans le cadre des systèmes d'apprentissage [64] et dans le cadre des outils de collaboration [213]. DeVries propose une typologie des systèmes d'apprentissage selon trois axes : la théorie pédagogique, les fonctions pédagogiques et le statut des connaissances présentées. Miller and Miller [148] ont définis des stratégies de mise en oeuvre d'outils de communication en fonction des objectifs d'une expérience pédagogique spécifique : quels outils pour

quels buts pédagogiques (*Instructional Goals and Communication strategies*). Ces travaux ne suffisent pas cependant pour guider pratiquement les enseignants dans la mise en oeuvre de leur pédagogie.

Oubahssi *et al.* [157] ont proposé une typologie intéressante de fonctionnalités pour les environnements d'apprentissage. Ils ont basé leur typologie sur le cycle de vie d'une formation ouverte à distance (Elaborer, Orienter, Apprendre, Evaluer, Gérer). Ces auteurs proposent l'implémentation d'un noyau de fonctionnalités à partir de cette classification. D'autres travaux ont étudié plus précisément les fonctionnalités proposées par certaines plates-formes dans le contexte plus particulier du travail coopératif (Campus virtuel) ou de l'apprentissage coopératif (outils de groupware). Caron [42] a identifié les fonctionnalités nécessaires pour mettre en oeuvre une approche socioconstructiviste :

- Outils d'administration permettant de créer entités et liens composant le dispositif pédagogique
- Outils de rétroaction de groupe synchrone et asynchrone (outils de rating, wiki peer reviewing, ...)
- Outils de production de groupe (Wiki, Traitement de texte/Tableurs "avec option coopérative", glossaire, ...)
- Outils de résolution de conflit (outil de sondage, vote, ...)
- Outils de workflow permettant de favoriser le conflit, de le mettre en scène

Caron a poursuivi ce travail en mettant en relation différents modèles pédagogiques avec les fonctionnalités offertes par les plates-formes dédiées. Il s'est basé sur une typologie de six fonctionnalités proposée par [129] à partir de laquelle le site de comparaison Edutools a été conçu : *Communiquer, Augmenter la productivité, Mener une évaluation, Administrer, Dispenser des cours et Respecter la charte de la plateforme.*

A partir de cette typologie, dans le cadre de nos travaux sur la PPC, nous regroupons ces fonctionnalités en plusieurs axes. De ce fait selon la PPC "dispensation de cours" n'a plus lieu d'être, car en PPC la notion de cours transmissif est occasionnel, mais est intégré dans un axe "Pédagogie" avec la fonctionnalité "mener une évaluation". Il apparaît les axes "Coopération/collaboration" et "Coordination", de même une recomposition entre action individuelle et collective nous a paru nécessaire. La dernière fonctionnalité proposée "Respecter la charte de la plate-forme" faisant référence aux respects des normes et des standards n'a pas d'objet dans notre contexte d'étude. Ainsi, nous obtenons les six axes suivants :

- **Pédagogie** : regroupant les fonctionnalités répondant aux exigences pédagogiques de la Pédagogie par Projet Collectif (PPC).
- **Administration** : regroupant les fonctionnalités d'administration d'une plateforme hébergeant une situation d'apprentissage dans un contexte PPC.
- **Production** : regroupant les fonctionnalités servant à produire individuellement et à mettre en commun les résultats de ces productions individuelles

dans un contexte PPC.

- **Communication** : regroupant les fonctionnalités permettant aux participants d'échanger des informations par l'écrit, la parole ou le geste. Dans le contexte de la PPC, la communication peut se faire entre enseignant et apprenant et entre les apprenants. Cette communication peut-être synchrone ou asynchrone.
- **Coordination** : regroupant tous les outils de coordination nécessaires à une synergie technique des différents membres.
- **Coopération/Collaboration** : regroupant toutes les fonctionnalités permettant aux acteurs de travailler collectivement.

On remarque que l'on retrouve dans cette typologie les groupes de fonctions traditionnelles des systèmes coopératifs : communication, coordination, collaboration, production [140].

5.4.1 Le Modèle proposé

Le modèle de fonctionnalités que nous proposons (figure 5.2) est construit à partir des six axes de fonctionnalités décrits précédemment. Nous avons tenté de caractériser les fonctionnalités de chaque axe à partir du méta-modèle de la PPC. Ce modèle nous offrira un moyen de catégoriser les plates-formes et d'identifier celles qui sont dédiées plus spécifiquement à la PPC.

Axe de fonctionnalité "Pédagogie". Nous avons observé sur le méta-modèle PPC proposé dans la chapitre 4 que le concept principal est le Projet et que ce concept peut être décomposé en Etape puis en Tâche. Une plate-forme mettant en oeuvre une PPC devra permettre de définir un projet et permettre son organisation et donc la définition d'un ou plusieurs niveaux de décomposition selon ce qui est souhaité par le concepteur, enseignant ou apprenant. Elle devra permettre en outre la navigation entre ces différents niveaux d'organisation du projet.

Aux concepts de Projet, et Etape nous avons associé dans le méta-modèle le concept "Objectif d'Apprentissage" pour donner à l'enseignant-concepteur la possibilité de décrire les objectifs pédagogiques et/ou d'apprentissage du projet et de ses étapes. Afin de décrire les pré-requis et domaines de connaissances mis en oeuvre au cours d'une étape, le concept "Etape" est associé aux concepts "Pré-requis", et "Domaine". Une tâche aura également des pré-requis et des objectifs de réalisations spécifiques. Au niveau plate-forme cela se traduira par la possibilité de définir des champs de description associés aux concepts Projet, Etape et Tâche.

Nous avons associé aux concepts Projet, Etape, et Tâches le concept "Facteur", pour définir les facteurs d'évaluation d'un projet, d'une étape, d'une tâche. Le concept "Métrique" servant à décrire les métriques d'évaluation de ces Facteurs. Pour mettre en oeuvre ce besoin, la plate-forme PPC devra posséder des fonctionnalités d'évaluation comme par exemple des outils de rating, de wiki peer

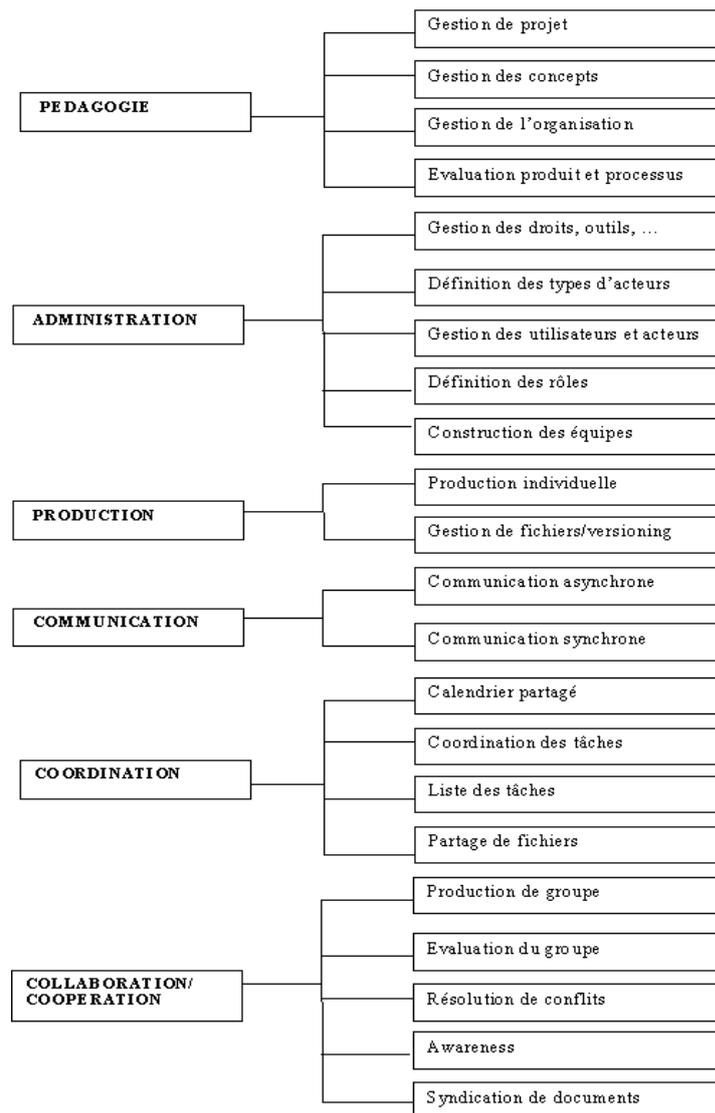


FIG. 5.2: Les six axes du modèle de fonctionnalités PPC

reviewing, de quiz etc. . . et permettre de répondre ainsi à cet aspect d'évaluation du projet, des apprenants et du processus de travail.

Axe de fonctionnalité "Administration". Dans le méta-modèle PPC, on rencontre de nombreux acteurs pouvant jouer des rôles variés en équipe ou non. La plateforme devra permettre de définir ces différents acteurs, leurs rôles, les équipes ou groupes grâce à des outils de gestion des utilisateurs, équipes, droits, outils, privilèges associés à l'apprenant dans son groupe et au groupe dans la formation.

Axe de fonctionnalité "Production". Dans le méta-modèle PPC, les apprenants doivent produire un travail, une oeuvre commune, l'activité de production, no-

tamment de document est donc importante. Ainsi dans le méta-modèle PPC nous avons identifié le concept "Production", concernant un projet, une étape et/ou une tâche. Nous avons mis en évidence l'articulation travail individuel/travail collectif de la PPC (chapitre 4). Les fonctionnalités suivantes pourront être associées à cet axe production : production individuelle et leurs communications grâce à la gestion de fichiers et le versioning pour permettre aux personnes d'apporter leur effort individuel à l'effort collectif.

Axe de fonctionnalité "Communication". La communication joue un rôle central dans un environnement d'apprentissage à distance. En particulier dans le contexte de la PPC, pour mener à bien leur projet les apprenants ont un réel besoin de communiquer avec tous les acteurs : enseignant/tuteur, expert d'un domaine de connaissance ou entre les membres apprenants du groupe de projet. La communication a pour but ultime d'organiser les activités, collaborer, se coordonner dans la production des différentes tâches à mener. Cette communication peut-être synchrone ou asynchrone.

Axe de fonctionnalité "Coordination". En PPC la notion de coordination ou d'organisation est également importante. Les apprenants sont placés dans une situation relativement autonome pour mener à bien un ensemble de tâches qui concourent à la réalisation d'une oeuvre commune. L'objectif pédagogique du projet étant plus de réussir ensemble que le but à atteindre. Pour ce faire, ils ont besoin de coordonner/organiser leur travail en mettant en oeuvre des techniques traditionnelles de gestion de projet : découpage du projet en étapes/tâches, planification de ces étapes/tâches, répartition du travail entre les membres du groupe,.... Cette préoccupation recouvre ainsi plusieurs fonctionnalités correspondantes aux fonctionnalités basées sur la coordination synchrone ou asynchrone (diagramme de Gantt, calendrier partagé par exemple).

Axe de fonctionnalité "Coopération/Collaboration". Dans le méta-modèle PPC, nous avons défini le concept de "Tâche" qui correspond à un niveau d'organisation du projet. En PPC une tâche peut-être menée par un apprenant individuellement ou de manière coopérative ou collaborative. Dans le contexte de la PPC, cet axe de fonctionnalité contribue à la mise en oeuvre d'"une négociation permettant l'explicitation et la socialisation de l'action". Ce type de tâche est supporté dans les plates-formes collaboratives par les fonctions suivantes : Production de groupe, Résolution et mise en scène de conflits, Evaluation de groupe, *Awareness* (conscience de groupe).

Dans le méta-modèle PPC le concept d'"Outil" fait référence aux divers outils mis en place pour réaliser les tâches d'un projet. Ce sont les outils qui vont aider à mettre en oeuvre les fonctionnalités des divers axes dégagés (communication, production, coopération/collaboration, coordination).

Nous regroupons tous les axes de fonctionnalités avec leurs fonctionnalités correspondantes dans un schéma récapitulatif (Figure 5.2 intitulé "Les six axes du

modèle de fonctionnalités PPC").

5.4.2 Mise en oeuvre du modèle de fonctionnalités PPC

Parmi le grand nombre de plates-formes disponibles ⁹, nous avons choisi d'étudier dans un premier temps Moodle. Cette plate-forme figure actuellement parmi les plus utilisées et a été conçue pour mettre en oeuvre une pédagogie socioconstructiviste. Après avoir brièvement présenté cette plate-forme, nous nous proposons d'étudier à partir de notre modèle de fonctionnalités PPC si elle conviendrait pour mettre en oeuvre un scénario pédagogique PPC.

Présentation de Moodle

Moodle est une plate-forme permettant la mise en place de cours en ligne et de sites web. C'est un projet bénéficiant d'un développement actif et conçu pour favoriser un cadre de formation socioconstructiviste ¹⁰. C'est une plate-forme sous licence open source servant à créer des communautés d'apprenants autour de contenus et d'activités pédagogiques. A un système de gestion de contenu (SGC), Moodle ajoute des fonctions pédagogiques ou communicatives pour créer un environnement d'apprentissage en ligne : c'est une application permettant de créer, par l'intermédiaire du réseau, des interactions entre des pédagogues, des apprenants et des ressources pédagogiques (Wikipédia 2008). Nous faisons ici une étude rapide pour Moodle en utilisant le modèle de fonctionnalités PPC proposé. Cette étude est résumée dans le tableau 5.1. Moodle intègre ainsi des outils ou activités d'apprentissage orientés sur : les communications synchrones (personnes en ligne visibles, chat), les communications asynchrones (forum), l'apprentissage collaboratif (groupes, glossaire, wiki, atelier, base de données), la réflexion critique (sondage : vote, consultation : les attentes), la personnalisation (page personnelle : profil, blog personnel). Il est à noter que l'élaboration évoluée de certains outils permet d'assurer plusieurs fonctionnalités. Le glossaire permet aussi bien la communication que l'apprentissage collectif. Dans ce paragraphe nous associons les fonctionnalités de notre grille à des activités Moodle (sous Moodle une activité correspond à un outil). Il peut y avoir plusieurs stratégies d'utilisation de la plate-forme et la proposition qui suit n'est donnée qu'à titre d'illustration et correspond à une version 1.8 de Moodle. De plus, dans un cadre réel, cette association doit être assistée à travers un outil informatique, et non un simple tableau (tableau 5.1), afin de permettre de conseiller et préciser à l'utilisateur la portée et les limites de chaque outil proposé au regard de l'intention pédagogique souhaitée.

Cette rapide étude nous montre que Moodle peut servir de support à une Pédagogie par Projet Collectif (PPC) malgré quelques insuffisances. Nous remarquons

⁹<http://thot.cursus.edu>

¹⁰<http://docs.moodle.org/>

<i>Axes de fonctionnalités</i>	<i>Proposition de fonctionnalités Moodle</i>
<p><i>Pédagogie</i></p> <ul style="list-style-type: none"> ✓ Création d'entités et liens composant l'organisation pédagogique <ul style="list-style-type: none"> ○ Gérer projet ○ Gérer les différents niveaux d'organisation d'un projet ○ Gérer les différents concepts associés (objectifs, prérequis, ..) ✓ Evaluation des apprenants <ul style="list-style-type: none"> ○ Par les enseignants/tuteurs ○ Par les pairs (wiki peer reviewing par exemple) 	<p>Organisation en module « <i>course</i> » (unité d'enseignement) L'enseignant définit un module, les ressources et activités associées, les personnes autorisées à le consulter, la durée du module. Un module est décrit par différents champs : résumé, introduction, ... permettant de supporter les concepts associés. Les wikis permettent de « <i>simuler</i> » l'organisation (la future version de Moodle possèdera une activité Gestion de projets ce qui résoudra plus complètement le problème). Définition d'activités d'évaluation (test, quizz, devoirs, ...)</p>
<p><i>Administration</i></p> <ul style="list-style-type: none"> ✓ Prise en compte de l'équipe <ul style="list-style-type: none"> ○ Gérer utilisateurs et acteurs ○ Gérer les groupes ✓ Assurer les responsabilités <ul style="list-style-type: none"> ○ Gérer les droits ○ Gérer les rôles ○ Assurer les rôles aux intervenants acteurs et groupes 	<p>Gestion de comptes utilisateurs Définition de rôles : rôles prédéfinis apprenant, tuteur, enseignant et possibilités de définir des rôles sur mesure (avec la notion Moodle de « Capacité ») Définition de groupe. Application de cette notion aux activités utilisées (avec la notion Moodle de Contexte). La notion de groupe séparé ou non permet de déclarer un Wiki unique et d'affecter à chaque membre du groupe un exemplaire du Wiki qui lui est propre</p>
<p><i>Production</i></p> <ul style="list-style-type: none"> ✓ Produire individuellement ✓ Gestion de fichier -Versioning 	<p>Blog Glossaire (activité individuelle et collective) Composition de pages texte, pages web</p>
<p><i>Communication</i></p> <ul style="list-style-type: none"> ✓ Communication synchrone ✓ Communication asynchrone 	<p>Zone de dépôt de fichiers (téléchargement de fichiers dans des dossiers) Outils de communication: Chat, Forum, Messagerie, ...</p>
<p><i>Coordination</i></p> <ul style="list-style-type: none"> ✓ Coordonner les tâches (outil de workflow) ✓ Liste des tâches (Diagramme de Gantt) ✓ Partage de fichiers 	<p>Calendrier partagé Devoir</p>
<p><i>Coopération/collaboration</i></p> <ul style="list-style-type: none"> ✓ Production de groupe (Wiki, éditeur/tableur collaboratif, ...) ✓ Résolution de conflit (sondage, vote...) ✓ Evaluation de groupe synchrone/asynchrone ✓ Syndication de documents ✓ Awareness (boîte à crier, membres en ligne, ...) 	<p>Wiki Glossaire Sondage, Vote Liste des utilisateurs connectés</p>

TAB. 5.1: Grille d'étude des fonctionnalités de Moodle

dans le tableau 5.1 que cette plate-forme ne répond pas à tous les critères de l'axe de fonctionnalité "Pédagogie". En fait, elle ne permet pas l'organisation du projet en étapes et tâches parce qu'elle ne possède pas la possibilité de définir un ou plusieurs niveaux de décomposition, ni la navigation entre ces différents niveaux d'organisation du projet. Plus de détails dans le remplissage du tableau, nous montreraient cependant certaines limites au niveau du travail de paramétrage et de respect de la notion de rôle de la PPC. Les travaux de [46] et les évolutions

constantes des plates-formes tentent d'y répondre.

Ce modèle de fonctionnalité permet de situer la plate-forme (ex :Moodle) par rapport à la Pédagogie par Projet Collectif. L'enseignant, à travers ses besoins formulés dans son scénario PPC, peut comparer entre plusieurs plates-formes étudiées à partir de ce modèle de fonctionnalités pour choisir la plate-forme qui correspondent le plus à son scénario PPC. Autrement dit, des enseignants considérant que l'axe de fonctionnalité "Pédagogie" est primordial dans leur scénario PPC ne choisissent pas la même plate-forme choisie par des enseignants considérant que l'axe de fonctionnalité "Communication" est primordial.

5.5 Conclusion

Dans ce chapitre nous avons étudié les plates-formes de formations existantes et leurs différents types. Nous avons présenté un ensemble représentatif de plates-formes parmi les plus utilisées (section 5.2). Dans la section 5.3, nous avons fait une étude de quelques environnements informatiques de travail collectif existants. Nous avons expliqué que certains de ces environnements sont utilisés dans un contexte d'apprentissage. Nous avons expliqué avant de les présenter que ces environnements appartiennent aux deux domaines de CSCW (*Computer Support Cooperative Work*) et CSCL (*Computer Support Cooperative Learning*). Après l'étude de quelques plates-formes de formations existantes et environnements informatiques de travail collectif, nous avons proposé dans la section 5.4 un modèle de fonctionnalités PPC correspondant au méta-modèle PPC proposé dans le chapitre 4. Ce modèle de fonctionnalités est construit à partir de six axes de fonctionnalités. Nous avons tenté de caractériser les fonctionnalités de chaque axe à partir du méta-modèle de la PPC. Ce modèle de fonctionnalités dans son état actuel nous offre un moyen de catégoriser les plates-formes et d'identifier celles qui sont dédiées plus spécifiquement à la PPC. Il favorise la communication entre l'ingénieur pédagogique et l'enseignant parce qu'il permet à ce dernier, à partir de son scénario PPC, de bien exprimer ses besoins en termes de fonctionnalités, forme d'expression nécessaire pour l'ingénieur pédagogique. Ce modèle doit permettre, de rationaliser le choix d'une plate-forme au regard de la PPC. Son utilisation incite non seulement à faire un choix compatible PPC mais elle fournit aussi de facto les principales lignes de la paramétrisation de la plate-forme sélectionnée. Dans la sous-section 5.4.2, nous avons choisi la plate-forme Moodle comme cas d'étude pour la classer suivant ce modèle de fonctionnalités de la Pédagogie par Projet Collectif (PPC).

Nous avons proposé dans le chapitre 4 un méta-modèle de la Pédagogie par Projet Collectif (PPC) permettant à l'enseignant concepteur d'élaborer des scénarios pédagogiques d'une situation d'apprentissage dans un contexte de Pédagogie par Projet Collectif (PPC). Dans ce chapitre, nous avons proposé un modèle de fonc-

tionnalités PPC correspondant au méta-modèle PPC permettant d'aider au choix d'une plate-forme. Le choix de la plate-forme la plus satisfaisante aux besoins formulés dans le scénario PPC doit permettre de minimiser les pertes sémantiques lors de la transformation du scénario PPC en un scénario PPC opérationnalisable par cette plate-forme cible. Il nous reste à répondre à l'objectif déclaré au premier chapitre consistant à opérationnaliser un scénario PPC. Dans le chapitre suivant, nous allons voir comment transformer un scénario PPC élaboré à partir du méta-modèle PPC en un scénario PPC opérationnalisable sur la plate-forme cible.

Chapitre 6

Opérationnalisation d'un scénario PPC

6.1 Introduction

Dans le chapitre précédent, nous avons proposé un modèle de fonctionnalités PPC qui correspond au méta-modèle PPC proposé dans le chapitre 4. Nous avons expliqué comment ce modèle offre un moyen de catégoriser des plateformes de formations et d'identifier celles qui sont dédiées plus spécifiquement à la PPC. Nous avons expliqué que ce modèle de fonctionnalités PPC aide l'enseignant concepteur à choisir la plate-forme la plus adaptée aux besoins qu'il a exprimés à travers son scénario PPC. Nous avons illustré notre proposition sur la plate-forme Moodle.

Dans ce chapitre nous proposons une démarche par transformation de modèles [42], qui permet d'intégrer un scénario PPC dans une plate-forme de formation. En particulier, nous adoptons dans le domaine des EIAHs, l'approche MDA (*Model Driven Architecture*) du génie logiciel. Cette approche [94], basée sur les modèles, met au centre la technique de transformation de modèles. Cette technique permet la traçabilité entre les différents modèles produits à différents niveaux d'abstraction. Cela doit faciliter le dialogue entre l'enseignant, détenteur des connaissances de la situation d'apprentissage et l'ingénieur pédagogique, expert de la plate-forme. Dans cette approche, l'enseignant conceptualise son scénario d'apprentissage PPC basé sur le méta-modèle PPC. Puis, son scénario PPC est adapté automatiquement à la plate-forme choisie en utilisant nos règles de transformations. Pour illustrer cette proposition, la plate-forme Moodle, utilisée à l'IUT de Laval, est choisie comme cas d'étude [2], [4].

Dans ce chapitre, nous étudions tout d'abord, l'approche IDM (Ingénierie Dirigée

par les Modèles) et ses quatre niveaux (section 6.2). Dans la section 6.3, nous étudions une démarche particulière de l'approche IDM, le MDA (*Model Driven Architecture*) défini par l'OMG [94]. Dans la section 6.4, nous expliquons comment introduire MDA dans l'activité de scénarisation et d'opérationnalisation pédagogique. Dans la section 6.5, nous expliquons les principes des transformations des modèles servant à produire un modèle appelé modèle cible (ex : un scénario PPC opérationnalisable par Moodle) à partir d'un modèle source (ex : un scénario PPC). Nous expliquons dans cette section le langage de transformation ATL (*Atlas Transformation Language*) et puis nous le décrivons à travers le standard QVT (*Query View Transformation*). Nous expliquons aussi les étapes techniques nécessaires pour implémenter les règles de transformations ATL depuis le méta-modèle PPC vers le méta-modèle de Moodle. Dans la section 6.6, nous définissons, expliquons et argumentons les règles de transformations élaborées depuis le méta-modèle PPC vers celui de la plate-forme Moodle correspondant à une vision simplifiée de la version 8.0 de Moodle. Dans la section 6.7, nous montrons comment le scénario PPC élaboré par ModX est converti en un scénario PPC opérationnalisable ¹ par Moodle.

Les règles de transformations développées nécessitent une installation de la plate-forme Eclipse et de quelques plug-ins, plus le chargement de scénario PPC créé sous ModX ². Ce processus n'est pas facile à faire pour l'enseignant. C'est pourquoi nous proposons dans la section 6.8 un outil permettant à l'enseignant de charger son scénario PPC généré par ModX et d'utiliser les règles de transformations développées.

6.2 IDM (Ingénierie Dirigée par les Modèles)

Face à l'évolution rapide du logiciel, les méthodes de modélisation sont considérées lourdes et non souples. Ces méthodes ont amené à la notion de **modèle contemplatif** ; un modèle contemplatif est loin d'être un modèle productif mais par contre il sert à la communication et la compréhension. Les méthodes de modélisation à base de modèles contemplatifs sont utilisées dans les phases amont d'analyse et de conception, par contre elles prennent insuffisamment en compte les liaisons avec le code source, le déploiement et la maintenance.

L'ingénierie dirigée par les modèles (IDM), en anglais *Model Driven Engineering* (MDE), est une approche spécifique du génie logiciel ayant pour objectif de définir un cadre théorique pour générer du code en utilisant des transformations successives de modèles, les rendant ainsi productifs. Un **modèle productif**, concept central dans l'approche IDM, est interprétable et manipulable par une machine [25]. Pour cela, il est nécessaire de formaliser non seulement les modèles, mais aussi

¹Nous voulons dire par "opérationnalisable" que le scénario peut être opérationnalisé et non pas implémenté automatiquement

²Comme nous l'avons expliqué dans le chapitre 4

bien les langages les décrivant ainsi que les méta-modèles dans lesquels sont décrits ces langages. L'originalité de l'approche IDM vient de ce qu'elle se distingue des méthodes de modélisations traditionnelles (ex : Merise) par le fait qu'elle se préoccupe de rendre les modèles productifs plutôt que contemplatifs.

L'**Ingénierie dirigée par les modèles** est le domaine de l'informatique mettant à disposition des outils, concepts et langages pour créer et transformer des modèles. Ce que propose l'approche de l'ingénierie des modèles est simplement de mécaniser le processus que les ingénieurs expérimentés suivent à la main. L'intérêt pour l'IDM a été fortement amplifié à la fin du XXème siècle lorsque l'organisme de standardisation OMG (*Object Modeling Group*) a rendu publique son initiative MDA (*Model Driven Architecture*) [94], qui peut être vue comme une restriction de l'IDM à la gestion de l'aspect particulier de dépendance d'un logiciel à une plate-forme d'exécution. De manière plus générale, et comme de nombreuses autres approches de génie logiciel, l'IDM est une forme d'ingénierie générative, qui se singularise par une démarche par laquelle tout ou partie d'une application informatique est générée à partir de modèles.

L'approche *Model Driven Architecture* [94] de l'OMG (étudiée dans la section 6.3), désire proposer autour de la notion de modèle et de méta-modèle une vision unificatrice [42] permettant de rassembler :

- Différents courants, méthodes et langages de modélisation spécifiques à des domaines favorisant l'application de techniques de génération de code (via un assistant, un fichier de configuration, ...) [93].
- Différentes communautés informatiques : génie logiciel, Web Sémantique, Ingénierie des systèmes interactifs, Ingénierie des systèmes d'information etc. [79]
- Différentes disciplines plus anciennes, Sémiotique, Linguistique, etc... pouvant apporter à l'ingénierie dirigée par les modèles des recherches abouties mettant en oeuvre des solutions différentes de celles que l'IDM propose.

6.2.1 Les quatre niveaux de l'IDM

L'IDM est fondé sur une architecture à quatre niveaux, initialement proposée dans le cadre du MDA. Cette architecture repose sur la définition d'un méta-méta-modèle [190], [27]. Concernant le domaine des EIAHs, nous trouvons des références à cette architecture dans de nombreux travaux désirant se situer par rapport à l'IDM, par exemple [42], [43], [44], [45], [82], [126] etc... Les quatre niveaux de cette architecture sont M3, M2, M1, et M0.

Le niveau M3

Le niveau M3 (MétaMétaModèle) est le niveau le plus abstrait de cette architecture [201]. Il définit les notions de base permettant la représentation des niveaux inférieurs (Méta-Modèles M2, et Modèles M1) ainsi que lui-même. Il permet l'expression des règles de conformités liant les entités du niveau M1 à celle du niveau M2. Des méta-méta-modèles comme le MOF (*Meta Object facility*) standardisé par l'OMG [190], eCore défini dans le cadre du "*Eclipse Modeling Framework* [38], différents formalismes comme des profils UML (*Unified Modeling Language*), des graphes conceptuels, sNets, CDIF, OWL [68] sont des exemples de méta-méta-modèles pouvant appartenir au niveau M3. Parmi ces méta-méta-modèles, le standard le plus utilisé maintenant est le MOF, défini par l'OMG dans le cadre du MDA [190]. MOF permet de décrire la structure des méta-modèles, d'étendre ou de modifier les méta-modèles existants. Le MOF est réflexif.

L'outil ModX que nous avons utilisé dans le chapitre 4 pour éditer le méta-modèle PPC afin d'élaborer des exemples de scénarios PPC, est une plate-forme de méta-modélisation graphique reposant sur l'implémentation du MOF 1.4 [149], [42]. Cette plate-forme développée par l'équipe TRIGONE de Lille définit un environnement au sein duquel il est possible d'éditer graphiquement et dynamiquement des méta-modèles et modèles conformes au MOF.

Le niveau M2

Le niveau M2 (ou Méta-Modèle) définit le langage de modélisation et la grammaire de représentation des modèles M1 tout en utilisant le vocabulaire et la grammaire spécifiée au niveau M3 [201]. Le méta-modèle UML qui est décrit dans le standard UML, et qui définit la structure interne des modèles UML, fait partie de ce niveau. Les profils UML, qui tendent le méta-modèle UML (exemple profils CPM [126]), appartiennent aussi à ce niveau. Les méta-modèles sont des instances du MOF.

Le méta-modèle PPC que nous avons élaboré dans le chapitre 4 est situé au niveau M2. Un méta-modèle (ex méta-modèle PPC), peut être considéré comme un langage spécialisé pour un aspect d'un système.

Le niveau M1

Le niveau M1 est le niveau modèle. Il définit l'ontologie utilisée pour la description de l'application du monde réel [201]. Tout en respectant le vocabulaire et la grammaire spécifiés par son méta-modèle au niveau M2, un modèle au niveau M1 décrit les informations de M0. Les modèles M1 sont des instances de méta-modèle M2.

Un scénario PPC, situé au niveau M1, est édité à partir du méta-modèle PPC situé au niveau M2. Pour favoriser leur manipulation par des outils informatiques

et leur opérationnalisation, l'approche IDM a pour objectif la production des modèles pouvant être validés formellement [28]. L'IDM se distingue alors par cette validation des méthodes de modélisations antérieures. Un modèle est valide s'il est conforme à son méta-modèle, les règles de conformités sont définies au niveau du méta-méta-modèle. Par exemple, le respect de contraintes de cardinalité exprimé par le méta-modèle sont considérées comme des règles de conformités. Il existe plusieurs types de conformités d'un modèle avec son méta-modèle : empirique, littérale, théorique, et outillée [28] :

- Vérification de conformité empirique : elle intervient dans le cas où l'on veut produire des modèles d'une application et l'on ne possède pas son méta-modèle.
- Vérification de conformité littérale : elle intervient dans le cas où il existe une représentation du méta-modèle qui n'est ni formelle ni exécutable. C'est le cas d'une comparaison à la main entre un document fourni et le code rédigé (ex une comparaison entre les diagrammes UML et le code qui les réalise).
- Vérification de conformité théorique : elle intervient dans le cas où il existe une représentation du méta-modèle qui est formelle mais non exécutable. Le méta-modèle est décrit par une théorie et c'est l'utilisateur lui-même qui vérifie que son modèle est conforme à la théorie : "démonstration à la main".
- Vérification de conformité outillée : elle intervient dans le cas où il existe une représentation du méta-modèle qui est exécutable mais non formelle. Un outil de vérification de conformité est associé à un méta-modèle et c'est à l'utilisateur de valider son modèle par cet outil.
- Vérification de conformité par "spécifications exécutables" : elle intervient dans le cas où il existe une représentation du méta-modèle qui est formelle et exécutable.

Il est souhaitable que cette conformité soit basée sur une représentation de méta-modèle qui soit formelle et exécutable pour qu'elle soit productive [28]. Une plateforme de modélisation qui implémente un méta-méta-modèle permet d'établir de telles règles de conformité bien fondées entre méta-modèles et modèles.

Le niveau M0

Le niveau M0, appartient au monde réel et non pas au monde de la modélisation. Ce sont les informations réelles de l'utilisateur, instance du modèle de M1. C'est l'exemple du déroulement d'un scénario d'apprentissage comme le déroulement du scénario PPC élaboré dans le chapitre 4.

Nous avons donc expliqué les quatre niveaux (figure 6.1) faisant l'objet d'un consensus au sein de la communauté IDM. Cette communauté met en scène deux relations [25] : la première est celle de la "conformité" liant un modèle M1 à son

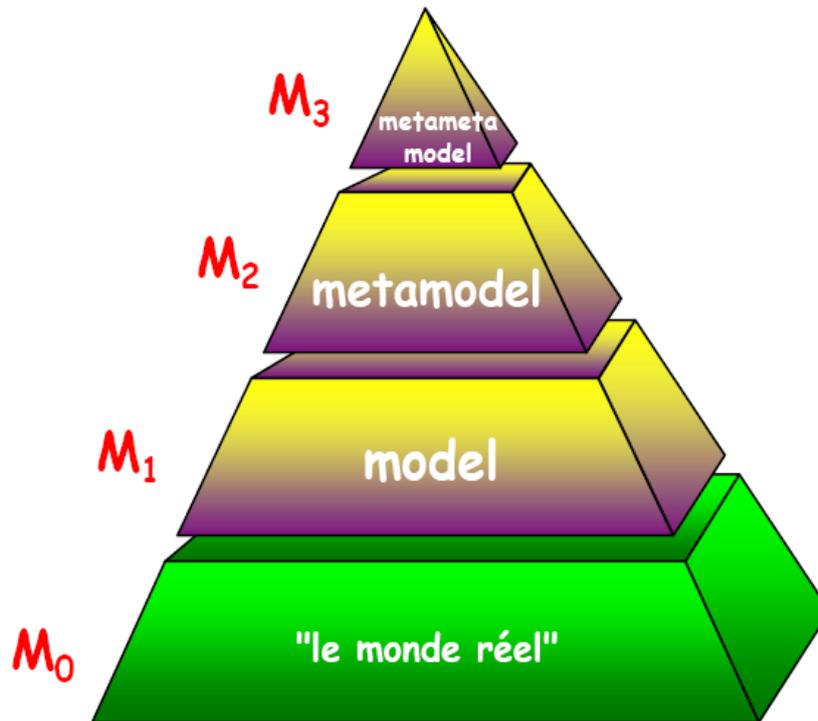


FIG. 6.1: Les quatre niveaux IDM. Source [22]

méta-modèle M_2 , et la deuxième est celle de la "représentation de" liant un modèle à l'élément M_0 qu'il représente. C'est l'existence de cette dernière relation qui rend le modèle opérationnel. Donc, les modèles manipulés en IDM possèdent deux caractéristiques les distinguant des modèles classiquement utilisés par des méthodes de modélisation. Ils sont définis d'une façon formelle par la relation de "conformité" les liant à leur méta-modèle, et ils sont opérationnalisables informatiquement.

6.2.2 Concepts fondamentaux de l'IDM

L'approche IDM se base sur deux relations fondamentales qui sont : la relation appelée *ReprésentationDe*, liée à la notion de modèle, et la relation appelée *EstConformeA*, permet de définir la notion de modèle par rapport à celle de méta-modèle.

La relation *ReprésentationDe* et les modèles

Il n'existe pas une définition unique pour la notion de modèle. Par contre les définitions existantes ne sont pas très différentes. Mais en général, le modèle

représente la simplification subjective d'un système existant [27], [23]. Seidwitz [184] définit le modèle par : "*A model is a set of statements about some system under study (SUS)*". Bézivin et Gérbé dans [27] définissent le modèle par : "*A model is a simplification of a system built with an intended goal in mind. The model should be able to answer questions in place of the actual system.*"

Nous remarquons d'après ces définitions que les deux concepts "modèle" et "système" sont dissociables dans une relation de modélisation liant un modèle au système qu'il modélise [25]. Cette relation s'appelle *ReprésentationDe* [184], [20], [9].

Nous trouvons dans ces définitions toutes les caractéristiques d'un modèle descriptif, modèle manipulé par les méthodes de modélisation traditionnelles. Or dans le cadre de l'approche IDM, il est très intéressant de manipuler des modèles formalisés pour les rendre productifs. Un modèle productif signifie que son traitement est automatisable, et la construction du système qu'il représente est dirigée par les modèles et les méta-modèles. Des chercheurs ont intégré ce caractère primordial dans la définition de modèle. Kleppe *et al.* [121], dans le contexte de l'approche IDM, ont défini le modèle par : "*A model is a description of (part of) a system written in a well-defined language*".

Nous concluons de cette définition de modèle que ce dernier fait référence à la notion de langage et par suite à la notion de méta-modèle qui permet de définir ce qu'est un langage.

La relation *EstConformeA* et les méta-modèles

Nous avons évoqué ci-dessus que l'objectif premier de l'approche IDM est que le modèle soit productif. Pour le rendre productif, il faut qu'il puisse être manipulé par une machine. La manipulation du modèle nécessite que le langage dans lequel il est exprimé soit clairement défini [25]. Pour définir un langage, l'IDM utilise la notion de méta-modèle. Un méta-modèle est un modèle d'un langage de modélisation. Nous trouvons dans la littérature plusieurs définitions de la notion de méta-modèle. Ces définitions sont assez convergentes.

Dans la spécification MOF (*Meta Object Facility*) [190] définit le méta-modèle par : "*A meta-model is a model that defines the language for expressing a model*". Kleppe *et al.* [121] définissent le méta-modèle par : "*A metamodel is a specification model for a class of SUS³ where each SUS in the class is itself a valid model expressed in a certain modelling language*".

La notion de méta-modèle mène à l'identification d'une deuxième relation nommée *ConformeA* [80], [20], [18]. La reconnaissance de l'importance de la relation *ReprésentationDe* en informatique doit être attribuée aux méthodes de modélisation traditionnelles [25]. L'originalité de l'approche IDM est de mettre l'accent sur la relation *ConformeA* liant un modèle au méta-modèle auquel il est conforme.

³System Under Study

Si un modèle est bien conforme à son méta-modèle ça veut dire qu'il est bien et correctement construit d'un point de vue théorique ainsi qu'opérationnel et par suite il est prêt à être transformé automatiquement en lui appliquant de règles de transformations.

6.3 Le MDA (*Model Driven Architecture*)

En novembre 2000, l'OMG du domaine du génie logiciel, consortium de plus 1000 entreprises, a initié la démarche MDA [94], [188] dont les concepts sont orientés-modèles plutôt que orientés-objets. Le *Model Driven Architecture* MDA [94] offre le pouvoir d'abstraction, de raffinement et de vues différentes sur les modèles. Ce standard a pour but d'apporter une nouvelle façon de concevoir des applications en séparant le logique métier de l'entreprise, de toute plate-forme technique afin d'augmenter la réutilisation de code déjà développé, réduire le temps de développement, et faciliter l'intégration des nouvelles technologies [31]. Il donne la possibilité de concevoir des modèles indépendants des plates-formes et de l'environnement d'implémentation [201]. MDA permet de séparer deux visions extrêmes du même système [51] :

- ses spécifications fonctionnelles d'une part,
- son implémentation physique d'autre part,

incluant plusieurs aspects de la vie du logiciel, à savoir ses tests, ses exigences qualité, la définition des itérations successives, etc.

L'architecture du MDA se décompose en quatre couches (figure 6.2). Au centre, il existe le standard UML (*Unified Modeling Language*), MOF (*Meta-Object Facility*) et CWM (*Common Warehouse Metamodel*). La deuxième couche contient le standard XMI (*XML Metadata Interchange*) permettant le dialogue entre les *middlewares* (Java, CORBA, .NET, et web services). Dans la couche suivante, se trouve les services permettant de gérer les événements, la sécurité des transactions, et les répertoires. La dernière couche propose des *frameworks* spécifiques au domaine d'application (Télécommunication, médecine, commerce électronique, finance, etc.) Un concepteur pour créer sa propre application peut utiliser UML comme il peut utiliser d'autres langages. En partant du centre de la figure 6.2, le concepteur dirigera son application en évoluant de couche en couche pour aller vers le domaine d'application qui l'intéresse.

Donc selon cette architecture indépendante du contexte technique, MDA propose de structurer les besoins avant de se livrer à une transformation de cette modélisation fonctionnelle en modélisation technique, tout en testant chaque modèle produit [51]. Il s'agit de modéliser l'application que l'on veut créer de manière indépendante de l'implémentation cible (niveau matériel ou logiciel). Ceci permet une grande réutilisation des modèles [163]. MDA est considéré une démarche

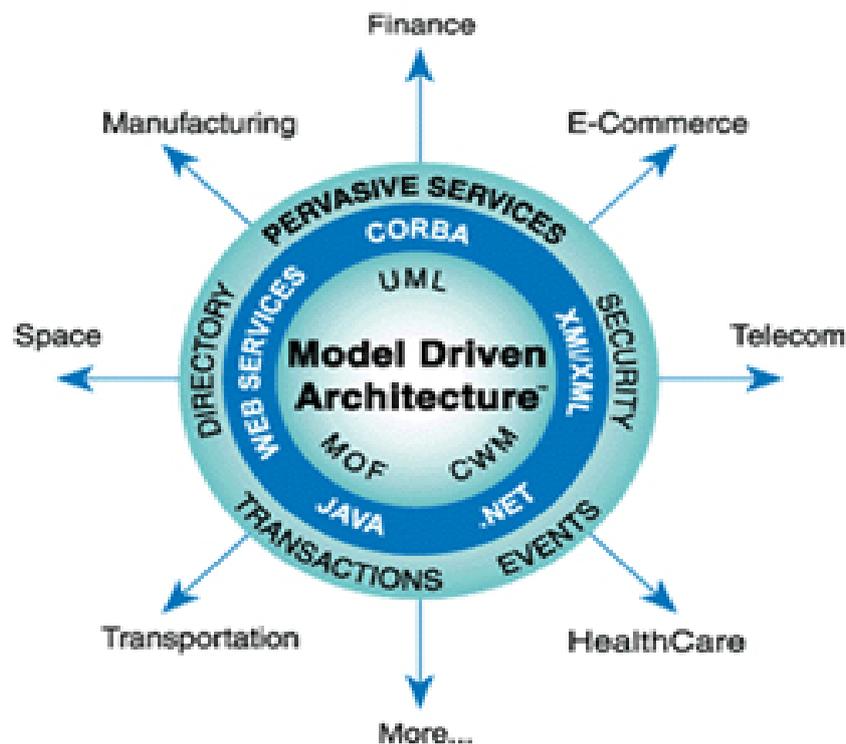


FIG. 6.2: Architecture du MDA

ayant l'ambition de proposer une vision la plus large possible du cycle de vie du logiciel, ne se contentant pas uniquement de sa production. De plus, cette vision globale se veut décrite dans une syntaxe unifiée.

Un des postulats sous-jacents du MDA est que l'opérationnalisation d'un modèle abstrait n'est pas un problème trivial. Un des bénéfices du MDA est de résoudre ce problème [46]. Le MDA propose de concevoir une application au travers d'une chaîne logicielle se déclinant en quatre phases dans l'objectif d'implémentation flexible, d'intégration, de maintenance et de test :

- L'élaboration d'un modèle sans préoccupation informatique (CIM : *Computer Independent Model*).
- Sa transformation manuelle en un modèle dans un contexte technologique particulier (PIM : *Platform Independent Model*) ;
- Sa transformation automatique en un modèle associé à la plate-forme de réalisation cible (PSM : *Platform Specific Model*), modèle qui doit être raffiné ;
- Sa réalisation dans la plate-forme cible.

CIM : *Computer Independent Model*

Le modèle CIM est indépendant de toute implémentation informatique. C'est le modèle métier ou le modèle du domaine (*domain/business model*), recensant les besoins des utilisateurs en utilisant le vocabulaire du praticien. Le modèle CIM ne présente donc, aucune trace informatique [42]. L'indépendance technique de ce modèle lui permet de garder tout son intérêt au cours du temps et il est modifié uniquement si les connaissances ou les besoins métier changent. Le savoir faire est recentré sur la spécification CIM au lieu de la technologie. Il joue un rôle important dans l'établissement d'un lien de rapprochement entre les experts du domaine et leurs besoins d'un côté, et les concepteurs des artefacts d'un autre côté [94]. Il sert de base à la définition d'un modèle PIM (*Platform Independent Model*).

PIM : *Platform Independent Model*

Ce dernier est indépendant de toute plate-forme technique et ne contient pas d'informations sur les technologies qui seront utilisées pour déployer l'application [94], [24]. C'est un modèle informatique représentant une vue partielle d'un CIM. Le modèle PIM correspond à la spécification de la partie "métier" d'une application ou d'un modèle de conception. Il décrit les services rendus par le système, conformément à une analyse informatique en cherchant à répondre aux besoins métiers indépendamment de la technologie de mise en oeuvre. Le modèle PIM doit être pérenne et durer au cours du temps. Il contient des informations sur la persistance, les transactions, et la sécurité du système. Ces concepts permettent de transformer plus précisément le modèle PIM vers le modèle PSM [94].

PSM : *Platform Specific Model*

Le modèle PSM (*Platform Specific Model*) correspond à la spécification d'une application après projection sur une plate-forme technologique donnée [94]. Il est dépendant de la plate-forme technique spécifiée par l'architecte. Il sert essentiellement de base à la génération de code exécutable dans les plates-formes techniques. Ce modèle est chargé de décrire comment le système va utiliser telle ou telle plate-forme. Le modèle PSM est obtenu par transformations successives du modèle PIM jusqu'à l'obtention du code dans un langage spécifique (Java, C++, etc.) [24]. Un PSM d'implémentation contient par exemple des informations comme le code du programme, les types pour l'implémentation, les programmes liés, les descripteurs de déploiement [94].

6.4 Introduire MDA dans l'activité de scénarisation et opérationnalisation pédagogique

Pour concilier les deux approches : pédagogique et technologique, nous adoptons une approche dirigée par les modèles MDA (*Model Driven Architecture*). Nous avons expliqué ci-dessus que cette approche, proposée par l'OMG (*Object Management Group*), divise la vision d'un système en trois points de vue [94] :

- CIM : (*Computer Independent Model*) correspondant au modèle du domaine indépendant de toute implémentation informatique,
- PIM : (*Platform Independent Model*) correspondant à une vision du système indépendante de toute technologie,
- PSM : (*Platform Specific Model*) correspondant à une vision dépendante d'une plate-forme spécifique.

Bien que les trois modèles CIM/PIM/PSM du MDA semblent convenir pour la scénarisation pédagogique, nous adoptons le point de vue de [127] consistant à proposer un nouveau cadre, inspiré de celui du MDA et en l'inscrivant dans une optique de scénario d'apprentissage. Ce nouveau cadre est représenté par la catégorisation des scénarios pédagogiques en trois types inspirés par les trois points de vue de MDA : **Scénario à niveau connaissance (SC)** qui capture le vocabulaire commun à une communauté d'enseignants-concepteurs. Ce vocabulaire peut concerner des approches pédagogiques particulières, des plates-formes de formation connues, etc. L'objectif de ce type de scénario est de guider et faciliter (pour la communauté d'enseignants) l'activité de conception consistant à définir le scénario d'apprentissage. **Scénario abstrait (SA)** qui, afin d'être réutilisé, a son vocabulaire indépendant de celui de la plate-forme de formation. Ce niveau permet à un vocabulaire d'un EML (*Education Modelling Language*) abstrait de s'abstraire aussi d'approches pédagogiques trop spécifiques. Ce niveau a pour objectif l'interopérabilité des dispositifs et l'abstraction des EML spécifiques à un domaine. **Scénario spécifique (SS)** à une plate-forme de formation qui a un vocabulaire conforme à celui d'une plate-forme précise.

Le processus de génie logiciel supporte les trois modèles de l'approche MDA. Par contre, "*un processus de scénarisation pédagogique ne nécessite pas forcément de prendre en compte ces trois aspects*" [127].

Cela veut dire que la transformation depuis le niveau SC vers celui de SS, sans passer par le niveau SA, est tout à fait appropriée. Dans l'approche MDA, le modèle CIM est transformé en PIM et puis ce dernier est transformé en PSM. Puisque le méta-modèle PPC capture le vocabulaire commun de la Pédagogie par Projet Collectif, le scénario PPC proposé dans le chapitre 4 est alors situé au niveau SC. D'autre part, le scénario PPC opérationnalisable par Moodle est situé au niveau SS, possédant un vocabulaire conforme à la plate-forme Moodle.

Cette transformation directe du niveau SC vers celui du SS va nous permettre de transformer le scénario PPC élaboré à partir du méta-modèle PPC en un scénario PPC opérationnalisable par la plate-forme Moodle.

6.5 Transformation du scénario PPC en un scénario PPC opérationnalisable par Moodle

La question de la transformation de modèles se situe au centre de l'approche MDA [21]. Elle consiste à manipuler des modèles pour en préciser le sens et les rendre opérationnalisables [94]. Cette manipulation peut se définir au niveau M2, entre les méta-modèles, sous forme de règles de transformations.

Plus précisément, la transformation de modèles est une opération qui sert à produire un modèle, appelé modèle cible (ex : scénario PPC opérationnalisable par Moodle), à partir d'un modèle, appelé modèle source (ex : scénario PPC) [94]. Elle est constituée, selon le MDA de deux étapes [42]. La première est une spécification des règles de transformations au niveau méta-modèle. Ces règles décrivent la correspondance entre les concepts des méta-modèles sources et cibles. La deuxième étape est une application de ces règles définies au niveau méta-modèle pour générer la transformation du modèle source en modèle cible [19], [59]. Donc, ces règles permettent de mener des transformations entre modèles, c'est-à-dire la génération automatique d'un ensemble de modèles cibles à partir d'un ensemble de modèles sources [59], [60]. Elles doivent être écrites dans un langage bien défini qu'un outil pourra compiler et exécuter. La génération automatique des modèles cibles sous-entendent que les modèles doivent être écrits dans un langage traitable par les machines [1].

Les activités de transformations sont réalisées en tant que processus automatisés qui prennent un ou plusieurs modèles sources en entrée et produisent un ou plusieurs modèles de production cible en suivant un ensemble de règles de transformations. Ce processus étant référé comme Modèle de Transformation [186]. La transformation de modèle se fait par "*mapping*" entre un modèle initial et un modèle cible [94]. Chaque modèle est décrit par un méta-modèle, qui recense les caractéristiques de ce modèle [163]. Le "*mapping*" est alors défini comme une traduction entre le méta-modèle initial et le méta-modèle cible [162], [163].

6.5.1 Le langage de transformation ATL (*Atlas Language Transformation*)

Dans leur utilisation opérationnelle d'ATL, Canals *et al.* [41] affirment que pour traiter de la transformation de modèles, il est difficile et lourd d'utiliser des langages objet puisqu'il faut consacrer autant d'efforts au développement de la transformation qu'à la définition du *FRAMEWORK* permettant de la mettre en

oeuvre. L'utilisation d'un langage comme XSLT si elle est plus directe et plus adaptée reste par contre difficile à maintenir [41]. Nous suivons leur choix en nous intéressant à la mise en oeuvre d'outils centrés autour des approches, MDA (*Model Driven Architecture*), MDE (*Model Driven Engineering*) et QVT (*Queries View Transformation*).

Nous avons expliqué dans la section 6.3 que l'idée principale véhiculée par l'approche MDA est de permettre de transformer ces modèles métier (ou des modèles existants), vers la représentation cible souhaitée. Ainsi quelque soit la technologie utilisée, *nous pourrions facilement passer de cette technologie vers une autre, à condition, bien sûr, d'avoir les outils de transformation nécessaires* [41]. Nous utiliserons des langages de transformations respectant la norme QVT [154] proposée par l'OMG.

Query/View/Transformation (QVT) [154] est un standard défini par l'OMG. Il s'agit d'un langage standardisé pour exprimer les transformations de modèles. QVT n'est pas aujourd'hui suffisamment avancé dans sa définition pour les aspects *Queries* et *View*. Par contre l'aspect transformation exprimé par l'approche MDA a donné lieu à diverses expérimentations (ex : Triskell, ATL ...) tant au niveau académique que commercial.

Pour établir la transformation, il est nécessaire d'avoir des outils de transformations. Ces derniers sont basés sur des langages de transformations devant respecter la norme QVT [154] proposée par l'OMG [26]. Il existe une offre d'outils libres (ATL, MTF, MTL, QVTP, etc.) et commerciaux (ex : MIA). Nous avons choisi ATL (*Atlas Transformation Language*) parmi l'offre d'outils libres, dans la mesure où *seul ATL propose un esprit conforme à OMG/MDA/MOF/QVT* [41].

Description du langage ATL

Atlas Transformation Language (ATL) a été conçu pour réaliser des transformations dans le cadre du *Framework* MDA proposé par l'OMG [98], [54]. Le langage ATL se base principalement sur le fait que les modèles sont des entités de première classe. En effet, les transformations sont considérées comme des Modèles de Transformation [186]. Puisque les transformations sont considérées elles-mêmes comme des modèles, on peut donc leur appliquer des transformations. Cette possibilité d'ATL est considéré comme un point important. En effet, il fournit les moyens de réaliser des transformations d'ordre supérieur (*Higher- Order Transformation HOT*) [54]. Une transformation d'ordre supérieur est une transformation dont les modèles sources et cibles sont eux-mêmes des transformations. Comme ATL est parmi les langages de transformation de modèles respectant la norme QVT [154] proposée par l'OMG [26], nous décrivons sa structure par rapport à cette norme (QVT).

Sa syntaxe abstraite a été décrite comme un méta-modèle MOF [26]. Quant à sa syntaxe concrète textuelle, elle a été définie en correspondance avec ce méta-

modèle. ATL fournit aux développeurs un moyen pour spécifier comment un (ou plusieurs) modèle cible peut être élaboré à partir d'un (ou plusieurs) modèle source [98]. ATL est mis à disposition dans le cadre des outils ADT (*ATL Development Tools*). La distribution actuelle d'ADT tourne sur la plate-forme Eclipse, et est basée sur EMF (*Eclipse Modelling Framework*) [38].

Relation QVT / ATL

- *Queries* (Requêtes) : une requête ATL est écrite en OCL (*Object Constraint Language*) [95], [54]. L'expression OCL peut retourner des valeurs primitives (booléens, String, Integer etc.), des éléments de modèle, des tuples, etc. [26].
- *Views* (Vues) : une vue est un cas particulier de transformation [26]. Cependant et grâce à quelques propriétés du langage de transformations, les vues peuvent être rendues plus pratiques. Notamment les transformations incrémentales rendent possible de mettre à jour une vue depuis sa source sans exécuter à nouveau l'intégralité des transformations. La bidirectionnalité des transformations permet de définir des vues changeables et dont les modifications se propagent au modèle source. Il est à noter que la version actuelle d'ATL [98] ne supporte pas ces fonctionnalités [54], [26].
- *Transformations* (Transformations) : A condition que les méta-modèles sources et cibles soient définis, un ensemble de modèles sources peut être transformé en un ensemble de modèles cibles grâce au modèle de transformation ATL. Tous les modèles conformes à un méta-modèle MOF sont actuellement manipulables par ATL [26].

Syntaxe abstraite

L'étude de la syntaxe abstraite du langage ATL consiste à étudier deux fonctionnalités fournies par ce langage plus les règles de transformations. La première fonctionnalité, **Navigation**, permet d'étudier la possibilité de navigation entre les méta-modèles sources et cibles. La deuxième fonctionnalité, **Opérations**, permet de décrire la possibilité de définir des opérations sur des éléments des modèles. Enfin, l'étude des **règles de transformations** permet de décrire les types de ces règles, la manière dont elles sont appelées et le type des résultats qu'elles retournent.

- Navigation [98], [26] : cette fonctionnalité d'ATL est offerte par le langage (*Object Constraint Language*) [95]. La navigation n'est autorisée que si les éléments de modèles sont complètement initialisés. Les éléments du modèle cible ne peuvent être définitivement initialisés qu'à la fin de l'exécution de la transformation. Par conséquent, la navigation dans ATL ne peut être faite qu'entre des éléments du modèle (ou méta-modèle) source et du modèle (ou méta-modèle) cible.

- Opérations [98] : cette fonctionnalité d’ATL est aussi fournie par le langage OCL (*Object Constraint Language*) [95]. Dans OCL, les opérations peuvent être définies sur les éléments du modèle. ATL reprend cette possibilité d’OCL pour permettre la définition des opérations sur des éléments du modèle source et du modèle de transformation [26].
- Les règles de transformations : il existe plusieurs types de règles de transformations basées sur la manière dont elles sont appelées et le type de résultat qu’elles retournent (figure 6.3).

CalledRule [98] : règle appelée explicitement en utilisant son nom et en

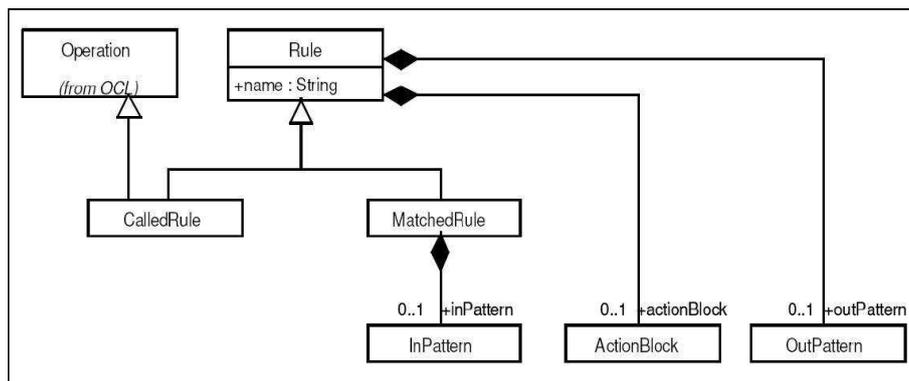


FIG. 6.3: Règles de transformation ATL

initialisant ses paramètres.

MatchedRule [98] : règle exécutée lorsque qu’un schéma type (*InPattern*) est reconnu dans le modèle source.

Le résultat d’une règle peut être un ensemble d’éléments de modèles prédéfinis (*OutPattern*), soit un bloc d’instructions impératives (*ActionBlock*). Si la règle est de type *MatchedRule* et si son résultat est un ensemble d’éléments du modèle cible (*OutPattern*), elle est nommée **déclarative**. Si elle est de type *CalledRule* dont le résultat est un bloc d’instructions, elle est alors appelée **Procédure**. Les combinaisons de règles (**déclarative et impérative**) sont appelées règles **hybrides** [26].

- *Pattern* : dans les règles de transformations ATL, il existe deux types de *pattern* utilisés par les règles de transformations (figure 6.4). Une règle de type *MatchedRule* spécifie les *InPatterns* utilisés pour détecter l’ensemble d’éléments particuliers dans le modèle source d’une part, et d’autre part spécifie les *OutPatterns* étant définis comme des ensembles de types issus du méta-modèle cible. Les *OutPatterns* représentent les éléments de modèles résultant de l’application d’une règle de transformations.

Les filtres sont définis dans le but d’obtenir la correspondance entre éléments du modèle source et *InPattern* d’une règle. Les filtres, décrits à l’aide des expressions booléennes OCL, permettent de guider le choix des éléments à

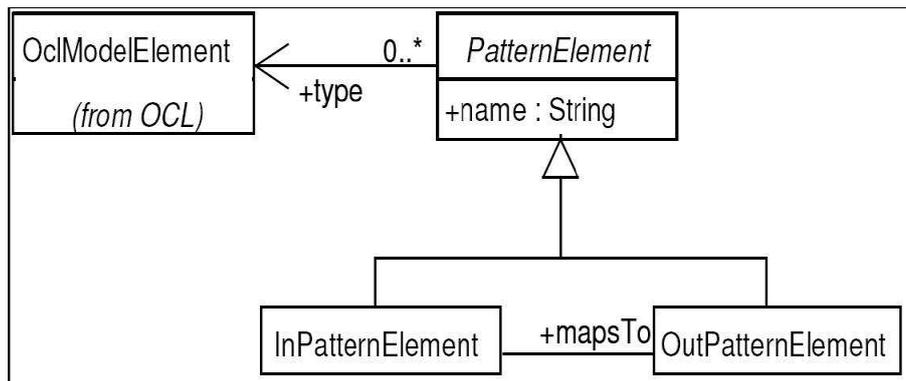


FIG. 6.4: Les Patterns cibles et sources des règles ATL

mettre en correspondance avec le *Pattern* en entrée de la règle (figure 6.5). Les éléments du pattern cible (*OutPattern*) peuvent être mis en relation par

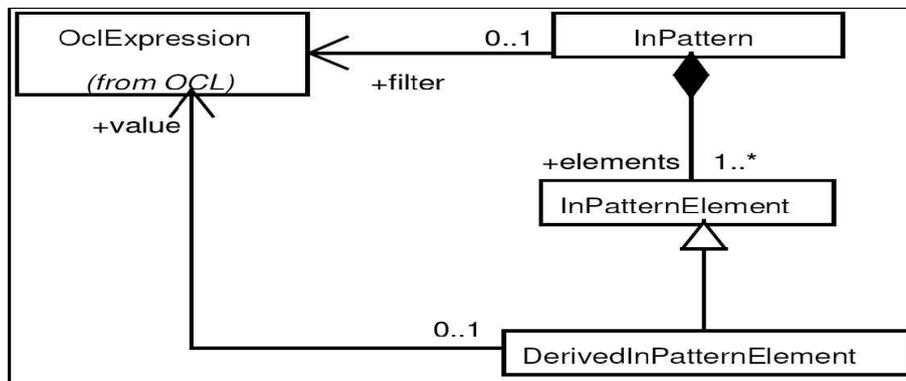


FIG. 6.5: Les Patterns sources des règles ATL

le biais de "*bindings*" (attaches, liens) qui spécifient la valeur d'initialisation d'une propriété d'une instance d'un élément (figure 6.6).

6.5.2 Etapes à faire pour implémenter les règles de transformations ATL

Pour implémenter les règles de transformations ATL, la première étape consiste à installer l'ADT (*ATL Development Tooling*), atelier dédié au développement des règles de transformations ATL.

ATL Development Tooling (ADT)

ADT [6] est l'atelier de développement ATL créé par l'équipe ATLAS à l'université de Nantes. Il s'agit d'un IDE ⁴ basé sur la plate-forme Eclipse [38]. Cette

⁴Integrated Development Environment

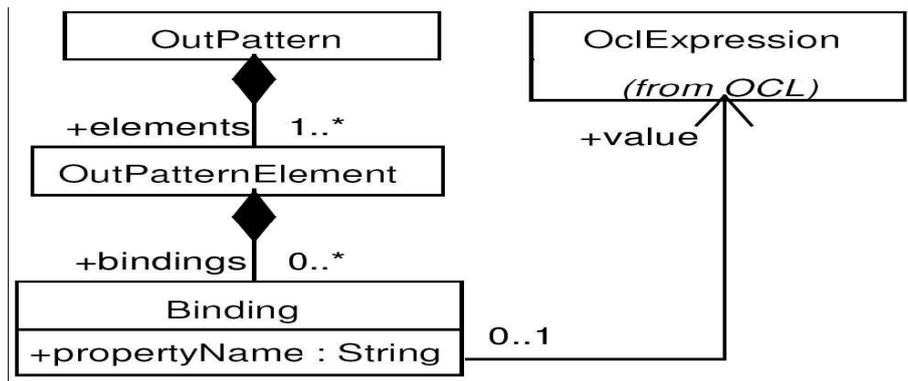


FIG. 6.6: Les Patterns cibles des règles ATL

dernière propose une architecture ouverte à base de plugins. ADT utilise des fonctionnalités du plugin EMF [95] développé par IBM.

EMF est un *framework* de modélisation facilitant la manipulation de modèles et la génération de code pour le développement d'outils basés sur un modèle structuré. EMF peut ainsi être utilisé dans le cadre du développement logiciel, aussi bien par des éditeurs que par des industriels. A partir d'un modèle de spécifications, EMF fournit des outils et un socle d'exécution pour produire un ensemble de classes Java pour le modèle, un ensemble de classes qui permettent la visualisation et l'édition du modèle.

ADT fournit un environnement complet de développement de transformations de modèles en ATL. Cela permet d'aider les développeurs dans leurs travaux de transformations. ADT dans sa version actuelle fournit déjà des outils précieux pour le développeur [6] :

- Un éditeur textuel fournissant une coloration syntaxique adaptable, et possédant une technique de reconnaissance d'accolades (indentation automatique). Cet éditeur est associé à une fenêtre affichant les propriétés de la transformation en cours de développement (*content outline view*),
- Une perspective complète eclipse (environnement graphique sous eclipse),
- Des icônes intelligentes pour les éléments spécifiques d'ATL,
- Un compilateur, et un debugger,
- Une aide à la création de projet ATL (*Project creation Wizard*).

En revanche l'installation d'ADT en tant que plug-in eclipse n'est pas réellement intuitive. Les détails d'installation d'ADT sont décrits dans [97].

Implémentation des règles de transformations ATL

Dans le domaine de l'ingénierie des modèles, les modèles sont des entités de première classe. Un modèle doit être défini en fonction des spécifications imposées par son méta-modèle : un modèle est dit alors conforme à son méta-modèle. De

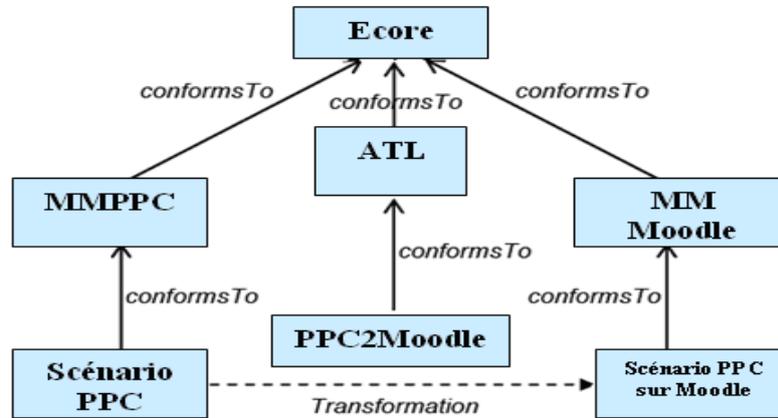


FIG. 6.7: Transformation ATL de PPC vers Moodle

la même manière, un méta-modèle doit être conforme à un méta-méta-modèle. Dans cette architecture de trois couches (modèles, méta-modèles, et méta-méta-modèles), le méta-métamodèle est souvent conforme à sa propre sémantique (il peut être défini par ses propres concepts). MOF, défini par l'OMG, et Ecore, introduit par l'EMF (*Eclipse Modelling Framework*), sont des exemples des méta-méta-modèles existants auto-définis.

Une des caractéristiques principales en ingénierie des modèles est de considérer tous les objets manipulés comme des modèles. La transformation de modèle doit être définie comme un modèle. Ce modèle de transformation doit être conforme à un méta-modèle de transformation qui définit les sémantiques de transformation de modèle. Comme les autres méta-modèles, le méta-modèle de transformation doit, à son tour, être conforme à son méta-méta-modèle.

Formellement, un modèle simple de transformation doit définir la manière de générer un modèle M_b , qui est conforme au métamodèle MM_b , à partir d'un modèle M_a conforme au méta-modèle MM_a . Le modèle de transformation est défini et vu en tant que modèle. Ce modèle de transformation doit être conforme au méta-modèle définissant les sémantiques de transformation de modèle. Comme tous les métas-modèles, le méta-modèle de transformation est conforme à un méta-méta-modèle.

La figure 6.7 fournit, un aperçu des éléments nécessaires à la transformation ATL permettant de générer un modèle de Scénario PPC opérationnalisable sur la plateforme Moodle et qui est conforme au méta-modèle Moodle, à partir du modèle Scénario PPC qui est conforme au méta-modèle PPC. Les trois méta-modèles (PPC, ATL, et Moodle) sont exprimés par les sémantiques du méta-méta-modèle Ecore.

Pratiquement et dans l'ordre d'achever la transformation, nous avons besoin :

- d'un méta-modèle source en KM3 (MMPPC) [99].
- d'un modèle source (en XMI transformé automatiquement en ecore) qui est conforme au méta-modèle PPC (Scénario PPC).
- d'un méta-modèle cible en KM3 (MMMoodle) [99].
- d'un modèle de transformation en ATL (PPC2Moodle).

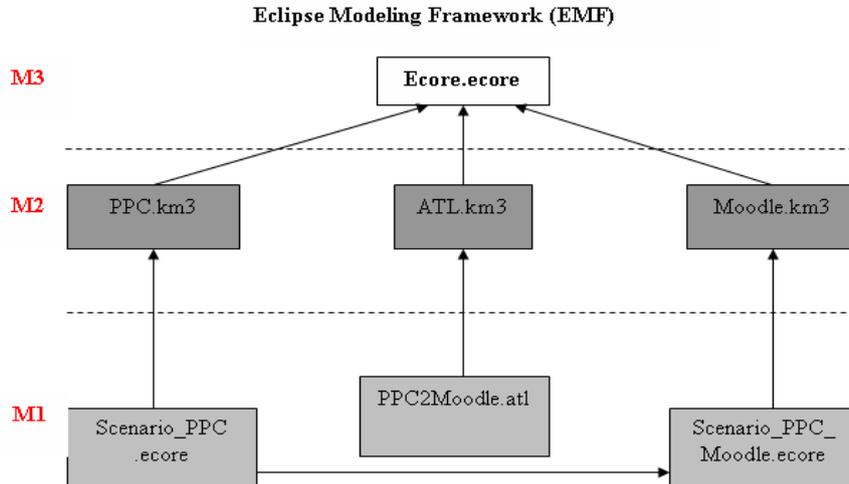


FIG. 6.8: Extensions des fichiers

Le coeur de travail de transformation est fait en écrivant les règles de transformation ATL dans le fichier "PPC2Moodle.atl" (figure 6.8). Avant l'exécution de cette transformation, le fichier résultant "Scénario_PPC_Moodle.ecore" n'existe pas. Il est créé par la transformation elle-même. Mais avant de définir la transformation, nous avons besoin de définir les méta-modèles source et cible (PPC.km3 et Moodle.km3) dans des fichiers km3 (*Kernel MetaMetaModel*) [99]. Nous notons que la définition du langage ATL existe dans le fichier ATL.km3 et que la définition récursive du méta-méta-modèle existe dans le fichier "Ecore.ecore". Donc, avant de définir le fichier des règles de transformations, nous créons les fichiers correspondant aux deux méta-modèles PPC et Moodle (PPC.km3 et Moodle.km3) (figure 6.9), nous les écrivons en tant que méta-modèle KM3. Ensuite il s'agit de créer les règles de transformations dans le fichier "PPC2Moodle.atl" (figure 6.10). Pour cela, nous utilisons le langage ATL décrit ci-dessus pour coder les règles de transformations ATL depuis le méta-modèle PPC vers celui de la plate-forme Moodle. La logique de la création de ces règles est décrite dans le reste de ce chapitre. Après la création du fichier des règles de transformations "PPC2Moodle.atl", vient l'insertion de n'importe quel scénario PPC sous forme d'un fichier XMI (converti en ecore automatiquement) (figure 6.11). Nous présentons dans ce qui suit l'exemple de scénario PPC élaboré à l'IUT de Laval

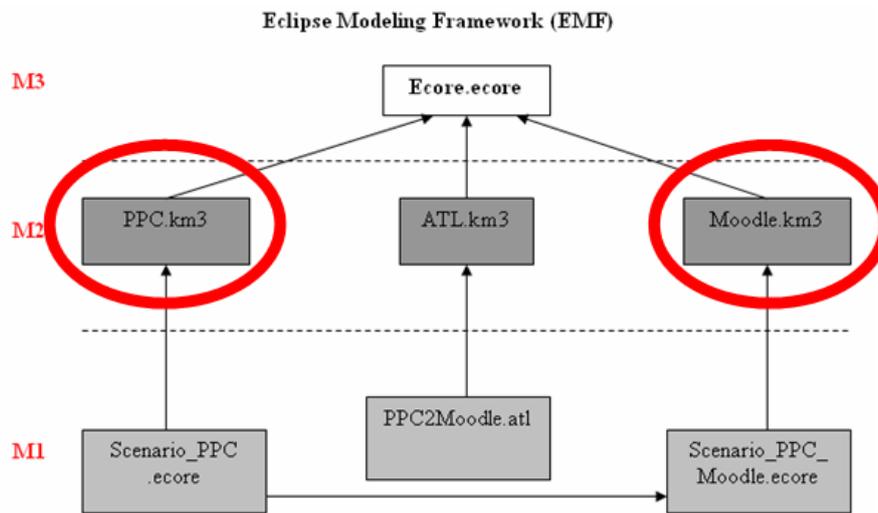


FIG. 6.9: Création des Méta-modèles PPC et Moodle

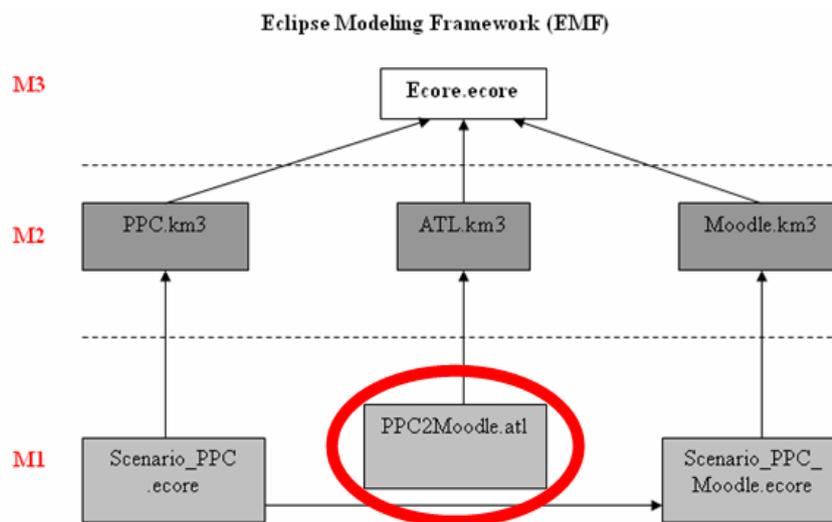


FIG. 6.10: Création des règles de transformations dans le fichier "PPC2Moodle.atl"

et expliqué dans le chapitre 4. Le format XMI du scénario PPC, à transformer, est fourni automatiquement par l'outil de scénarisation pédagogique ModX [149] comme nous l'avons expliqué dans le chapitre 4.

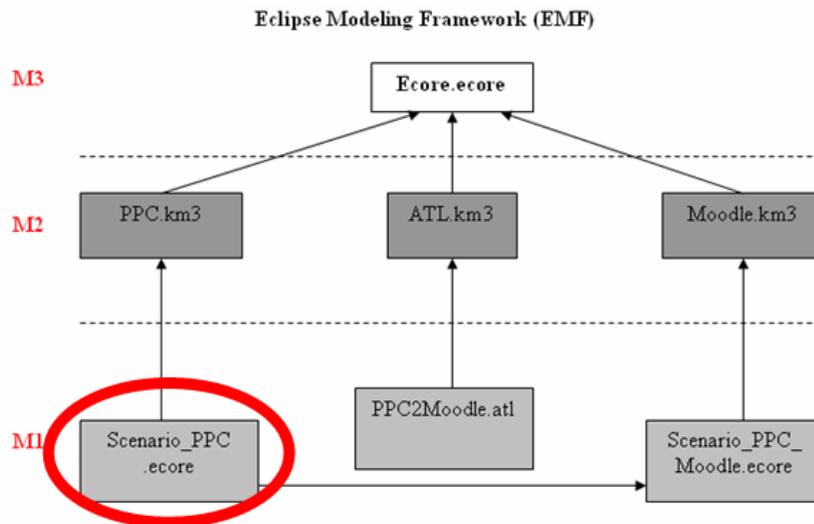


FIG. 6.11: Création du modèle "scénario_PPC.ecore"

6.6 Règles de transformations depuis le méta-modèle PPC vers le méta-modèle Moodle

Dans cette section, nous définissons, expliquons et argumentons les règles de transformations élaborées depuis le méta-modèle PPC vers celui de la plate-forme Moodle. Une règle définit le "*mapping*" entre les classes des méta-modèles ainsi que les règles de manipulation des attributs et des relations des classes. L'exécution d'un modèle de transformation nécessite une bonne compréhension de la syntaxe et la sémantique de la source et la cible. Le méta-modèle PPC source a été présenté dans le chapitre 4. Dans la figure 6.12, nous présentons une représentation du méta-modèle de la plate-forme cible Moodle. Les transformations entre deux domaines différents, sont difficiles à mettre en oeuvre en pratique [127] parce que certains concepts/reliions d'un méta-modèle n'ont pas forcément de concepts/reliions "voisins" dans l'autre (et vice versa). Cela peut mener à des pertes sémantiques (disparition de certaines informations spécifiées dans un scénario), comme à des manques sémantiques (certains informations nécessaires au langage "métier" cible ne peuvent parfois être précisées : le scénario obtenu est sémantiquement incomplet). Douze règles de transformations ont été élaborées depuis le méta-modèle PPC vers le méta-modèle Moodle [2], [4]. Ces règles de transformations ont deux buts :

- le premier est de transformer le scénario PPC vers l'univers technocentré de la plate-forme Moodle afin d'assurer son opérationnalisation.
- Le deuxième est de qualifier, par rapport à la PPC, la perte sémantique de

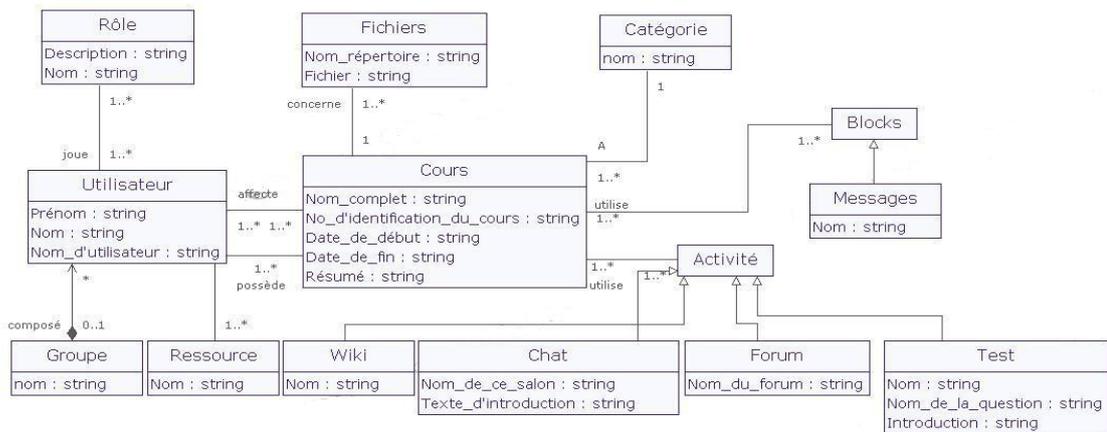


FIG. 6.12: Méta-modèle de la plate-forme Moodle

l'application de chaque règle.

Nous notons que dans cette section nous essayons de valider nos hypothèses de recherche et non pas de produire des plates-formes totalement réelles. C'est-à-dire que les règles de transformations proposées correspondent à notre choix qui n'est pas le seul. La proposition d'autres règles de transformations, correspondant à une autre vision, entre les deux méta-modèles (PPC et Moodle) est tout à fait possible. Nous donnons quelques exemples de code de développement des règles de transformations en ATL. Pour plus de détails, le code entier est décrit dans la partie "Annexe" à la fin de ce document.

6.6.1 Règles entre Projet, Etape, et Tache du côté méta-modèle PPC et Course du côté méta-modèle Moodle

Dans le méta-modèle PPC, il existe trois niveaux de conceptualisation à respecter : Projet, Etape et Tâche. Ces trois concepts sont reliés par des relations de composition (un projet est composé d'étapes qui sont composées de tâches). Tandis que dans le méta-modèle de la plate-forme Moodle, nous n'avons trouvé que le concept (Course) pouvant décrire ces trois concepts de PPC avec l'absence de relation de composition ; dans Moodle un cours ne peut pas être composé d'autres cours. Pour cela, nous créons trois règles entre le concept "Projet" du méta-modèle PPC et "Course" de celui de Moodle, entre "Etape" du méta-modèle PPC et "Course" de celui de Moodle, et entre "Tache du méta-modèle PPC et "Course" de celui de Moodle.

Concernant la **première règle (rule Projet2Course)** (figure 6.13), nous avons besoin de créer une règle de type (*MatchedRule*) qui va avoir comme classe source (Projet) et comme classe cible (Course). Cette règle fait le "mapping" entre la

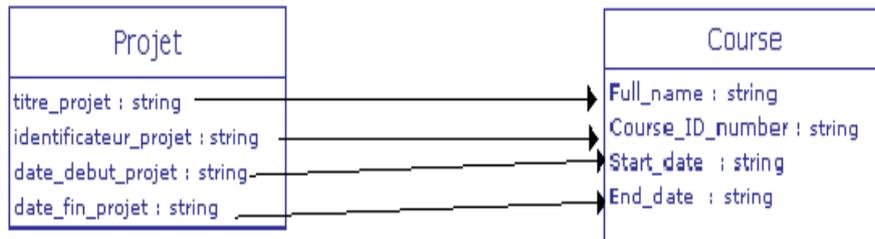


FIG. 6.13: rule Projet2Course

classe `Projet` du méta-modèle PPC et `Course` du méta-modèle Moodle ainsi que la manipulation des attributs de ces classes de la façon suivante : elle affecte la valeur de l'attribut "titre_projet" (`Projet`) à "Full_name" de `Course`, la valeur de l'"identificateur_projet" (`Projet`) à "Course_ID_number", la valeur de "date_debut_projet" (`Projet`) à "Start_date", et la valeur de "date_fin_projet" (`Projet`) à "End_date" de `Course`. Concernant la manipulation de la relation entre les classes `Projet` et `Etape` du méta-modèle PPC, nous sommes dans l'impossibilité d'avoir une relation voisine dans le méta-modèle Moodle parce qu'il n'existe pas dans le méta-modèle cible (Moodle) de relation équivalente. Dans Moodle, nous n'avons pas la possibilité de construire un `Course` qui est composé de plusieurs `Courses`. Pour cela, il y a à ce niveau là **une perte sémantique** (certaines informations spécifiées dans un scénario disparaissent) (figure 6.18). Et c'est de la responsabilité de l'utilisateur de bien nommer les étapes de projet pour savoir quelles étapes correspondent à quels projets. Nous verrons dans la règle suivante comment nous concaténons le titre de l'étape avec celui de son projet et le numéro de cette étape avec l'identificateur de son projet pour répondre à ce problème.

Concernant la **deuxième règle (rule `Etape2Course`)** (figure 6.14), nous avons

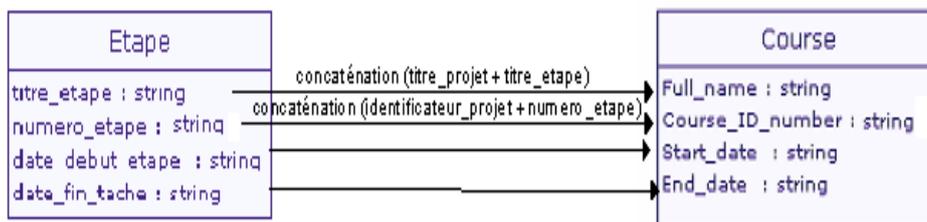


FIG. 6.14: rule Etape2Course

choisi aussi une règle de type (*MatchedRule*) qui a comme classe source (`Etape`) et

comme classe cible (Course). Cette règle fait le "*mapping*" entre la classe Etape du méta-modèle PPC et Course du méta-modèle Moodle ainsi que la manipulation des attributs de ces classes de la façon suivante : elle affecte la valeur de l'attribut "date_debut_etape" (Etape) à "Start_date" (Course), la valeur de "date_fin_etape" (Etape) à "End_date" (Course), la valeur de "titre_projet" (Projet) concaténée avec la valeur "titre_etape" (Etape) à l'attribut "Full_name" (Course), et la valeur de l'"identificateur_projet" (Projet) concaténée avec la valeur "numero_etape" (Etape) à la valeur de "Course_ID_number" (Course) (figure 6.15).

La concaténation n'est pas assurée à l'intérieur de la règle directement. Pour

```
rule Etape2Cours {
  from
    e : PPC!Etape
  to
    c : Moodle!Cours {
      Nom_complet <- e.x + '_' + e.Etape_titre,
      No_identification_du_cours <- e.w + '_' + e.Etape_numero,
      Date_de_debut <- e.Etape_date_debut,
      Date_de_fin <- e.Etape_date_fin,
      Resume <- e.getEtapeObjectifApprentissage(),
      UserAffectedCourse <- e.EtapeRegroupeActeur
    }
}
```

FIG. 6.15: Code rule Etape2Course

cela, nous spécifions que nous utilisons un "*Helper*" [98]. Nous définissons deux "*Helper*" (figure 6.16) ⁵ dans le contexte du concept "Etape". Le premier permet de retourner la valeur de l'attribut "titre_projet" du concept "Projet" en l'affectant au "*Helper*". Le deuxième permet de retourner la valeur de l'attribut "identificateur_projet" du concept "Projet" en l'affectant au "*Helper*".

Après dans la règle (*MatchedRule*), nous utilisons les "*Helper*" définis pour concaténer la valeur qu'ils rendent avec les valeurs des "titre_etape" et "numero_etape" de l'Etape et les affecter aux attributs "Full_name" et "Course_ID_number" de Course.

Comme pour la manipulation des relations entre les classes Etape et Projet, nous n'avons pas trouvé de relations voisines dans le méta-modèle Moodle pour traiter les relations entre les classes Etape et Tache du méta-modèle PPC. Dans Moodle, nous n'avons pas la possibilité de décomposer un Course en plusieurs Courses. Pour cela, il y a à ce niveau là **une perte sémantique** à noter (figure 6.18).

Concernant la **troisième règle (rule Tache2Course)** (figure 6.17), elle suit

⁵Un "*Helper*" est l'équivalent d'une opération additionnelle d'UML

```

helper context PPC!Etape def: x : String =
    self.PereEtape.Projet_titre;
helper context PPC!Etape def: w : String =
    self.PereEtape.Projet_identificateur;
helper context PPC!Tache def: y : String =
    self.PereTache.PereEtape.Projet_titre + '_' + self.PereTache.Etape_titre;
helper context PPC!Tache def: z : String =
    self.PereTache.PereEtape.Projet_identificateur + '_' + self.PereTache.Etape_numero;
    
```

FIG. 6.16: Code Helper

la même logique de la règle "Etape2Course" expliquée ci-dessus. Seulement nous utilisons deux autres "Helper" (figure 6.16) définis dans le contexte du concept Tache. Le premier "Helper" retourne la valeur du "titre_projet" du Projet concaténée à la valeur du "titre_etape" de l'Etape pour la concaténer à la valeur du "titre_tache" de la Tache et après l'affecter à l'attribut "Full_name" de Course. Le deuxième "Helper" retourne la valeur de "identificateur_projet" du Projet concaténée à la valeur du "numero_etape" de l'Etape pour la concaténer à la valeur du "numero_tache" de la Tache et après l'affecter à l'attribut "Course_ID_number" de Course. Dans les classes TacheIndividuelle et Tache-

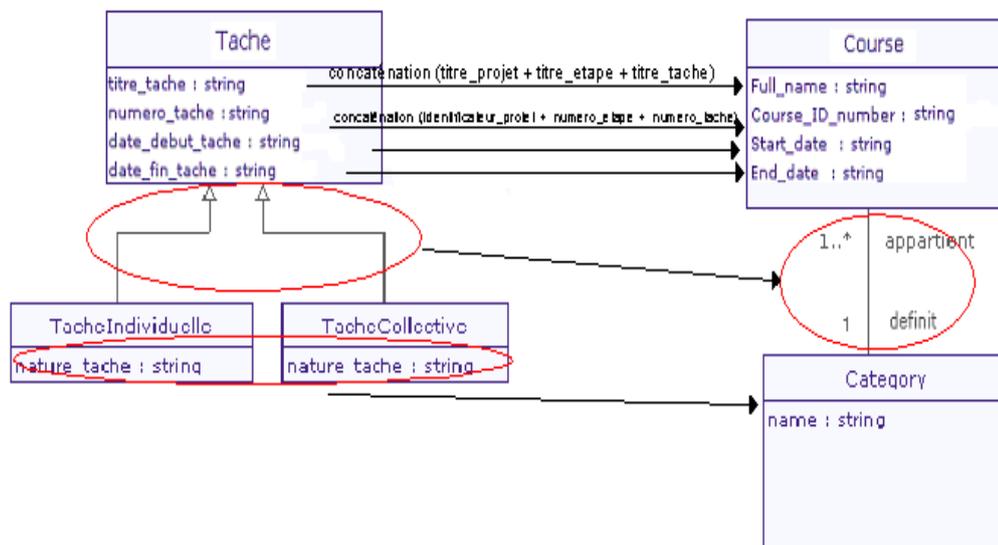


FIG. 6.17: rule Tache2Course

Collective du méta-modèle PPC, il y a l'attribut "nature_tache" qui spécifie si la tâche est individuelle ou collective. Le problème qui se pose c'est que dans le concept "Course" de Moodle il n'y a pas un attribut similaire pour lui affecter

la valeur de "nature_tache". Pour cela dans la règle "rule Tache2Course", nous faisons le "mapping" entre les classes TacheIndividuelle et TacheCollective du méta-modèle PPC et Category du méta-modèle Moodle ainsi que la manipulation des attributs de ces classes de la façon suivante : nous affectons la valeur de l'attribut "nature_tache" (TacheIndividuelle et TacheCollective) à "name" de Category.

En pratique, l'enseignant crée une tâche et c'est la valeur de l'attribut "nature_tache" qui spécifie si cette tâche est individuelle ou collective. Donc, la création du concept Tache va engendrer automatiquement l'utilisation de la classe Category qui est en relation avec la classe Course dans le méta-modèle Moodle (relation "appartient - définit") (figure 6.17).

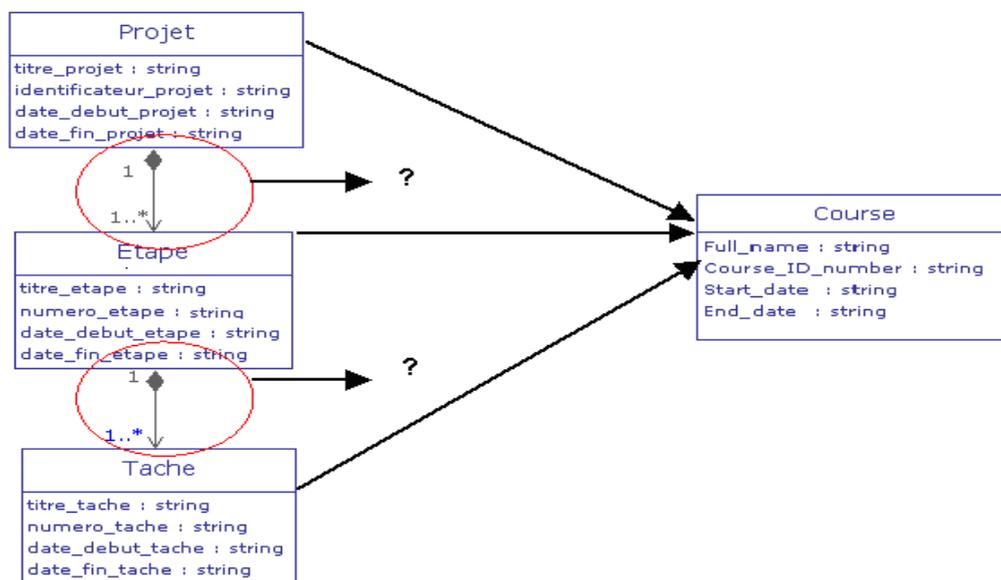


FIG. 6.18: Perte Sémantique

6.6.2 Règles entre Acteur, Role du côté méta-modèle PPC et User, Role du côté méta-modèle Moodle

Dans le méta-modèle PPC, il existe le concept Acteur pour décrire les acteurs de la PPC, et il existe le concept Role permettant d'affecter les différents rôles aux différents acteurs. Ces deux concepts ont deux concepts équivalents dans le méta-modèle Moodle : User et Role.

Concernant la **quatrième règle (rule Acteur2User)** (figure 6.19), nous créons une règle de type (*MatchedRule*) qui a comme classes sources "Acteur", "Apprenant", "Enseignant/Concepteur", "Tuteur", et "Expert" et comme classe cible

"User". Cette règle fait le "mapping" entre les classes sources du méta-modèle PPC et User du méta-modèle Moodle ainsi que la manipulation des attributs de ces classes de la façon suivante : elle affecte la valeur de l'attribut "type" des classes "Apprenant", "Enseignant/Concepteur", "Tuteur" et "Expert" à l'attribut "Username" de la classe "User" de Moodle, la valeur du "nom" de la classe "Acteur" à "Surname", la valeur du "prenom" à "First_name".

Concernant la cinquième règle (rule **Role2Role**) (figure 6.19), nous créons

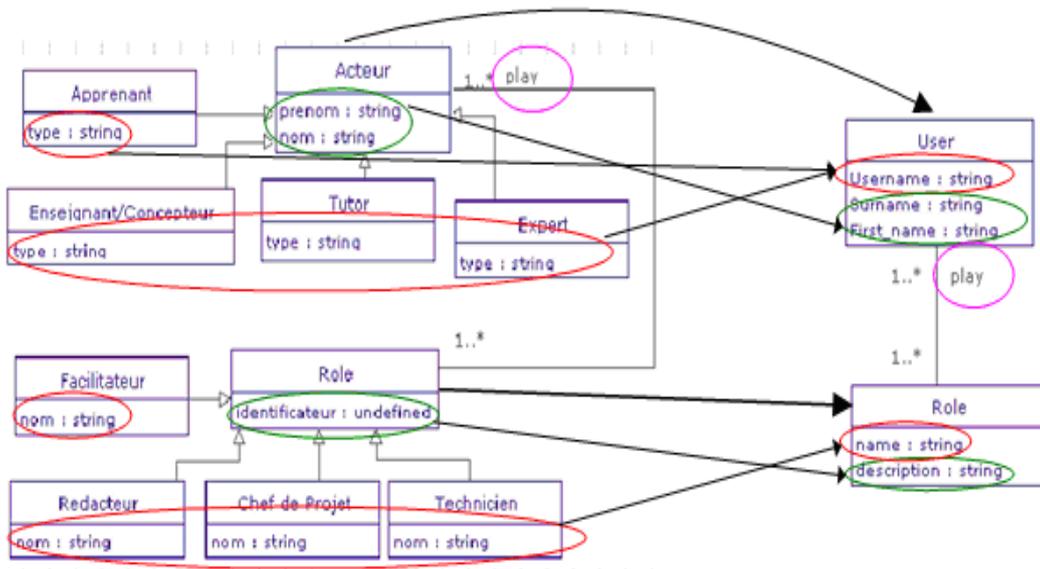


FIG. 6.19: rule Acteur2User et Role2Role

une règle de type (*MatchedRule*) qui a comme classes sources "Role", "Facilitateur", "Redacteur", "Chef de Projet", et "Technicien" et comme classe cible "Role". Nous faisons le "mapping" entre les classes "Role", "Facilitateur", "Redacteur", "Chef de Projet", et "Technicien" du méta-modèle PPC et "Role" du méta-modèle Moodle ainsi que la manipulation des attributs de ces classes de la façon suivante : nous affectons la valeur de l'attribut "nom" des classes "Facilitateur", "Redacteur", "Chef de Projet", et "Technicien" à l'attribut "name" de la classe "Role" de Moodle, et la valeur de "identificateur" de la classe "Role" à "description" de Moodle. Concernant la manipulation des relations, nous remplaçons la relation entre les classes "Acteur" et "Role" ("play") du méta-modèle PPC par la relation entre "User" et "Role" ("play") du méta-modèle Moodle. Les relations ("participe") entre d'un côté "Acteur" et "Projet", "Etape", et "Tache" du méta-modèle PPC sont remplacées par la relation entre "User" et "Course" ("participe") du méta-modèle Moodle (figure 6.20).

Revenons à la première règle (rule **Projet2Course**) (figure 6.22), nous

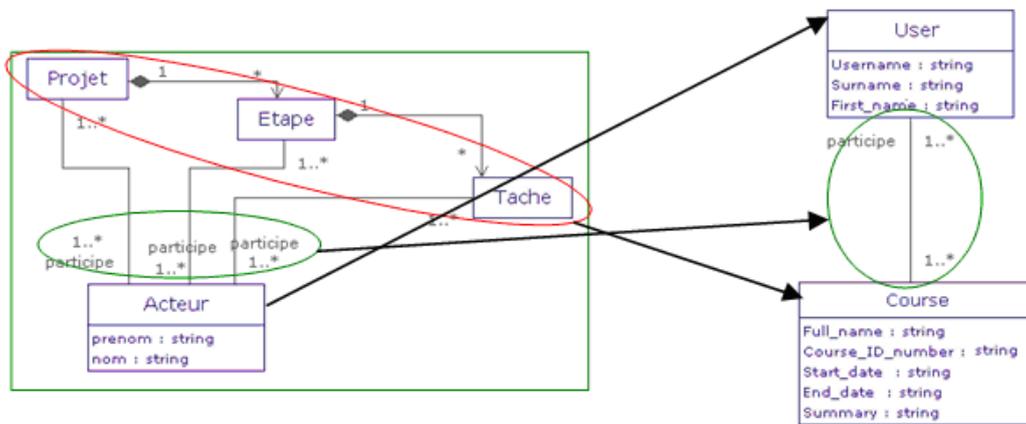


FIG. 6.20: rule Acteur2User et Role2Role

avons la classe "ObjectifApprentissage" dans le méta-modèle PPC qui n'a pas un concept voisin ou similaire dans le méta-modèle Moodle. Pour cela, nous décrivons les objectifs d'apprentissage (ObjectifApprentissage) concernant le projet (Projet) en se servant de l'attribut "Summary" du concept "Course" de Moodle. Mais, cet attribut prend une seule valeur en instanciant le concept "Course". Dans un autre sens, il faut affecter plusieurs objectifs d'apprentissage (relation "possede" de la classe "Projet" (cardinalité 1) avec la classe "ObjectifApprentissage" (cardinalité 1..*)) concernant le projet à **un seul attribut ("Summary")** de la classe "Course". Pour cela, nous créons un "*Helper*" dans le contexte de la classe "Projet" cherchant à récupérer un ensemble des objectifs d'apprentissage (attribut "description" de la classe "ObjectifApprentissage") pour les concaténer dans une chaîne de caractères (figure 6.21).

Dans la règle (rule Projet2Course), nous affectons la valeur du "*Helper*" à l'attribut "Summary" de la classe "Course". Nous notons que techniquement le problème est résolu, mais au niveau logique il y a une perte sémantique parce que nous utilisons un attribut propre à la classe Course ("Summary"), pour décrire un autre concept du méta-modèle PPC (ObjectifApprentissage).

Revenons à la **deuxième règle (rule Etape2Course)** (figure 6.23), nous avons la classe "ObjectifApprentissage" dans le méta-modèle PPC qui n'a pas de concept voisin dans le méta-modèle Moodle. Pour cela, nous décrivons les objectifs d'apprentissage (ObjectifApprentissage) concernant l'étape (Etape) en se servant de l'attribut "Summary" du concept "Course" de Moodle. Mais comme nous avons

```

helper context PPC!Projet def: getProjetObjectifApprentissage() : String =
    self.ProjetPossedeOA->collect(e | e.Description) ->
        asSet()->
        iterate (ObjecApp; acc : String = '' |
            acc +
                if acc = ''
                then ObjecApp
                else ' and ' + ObjecApp
                endif);
    
```

FIG. 6.21: Code Helper

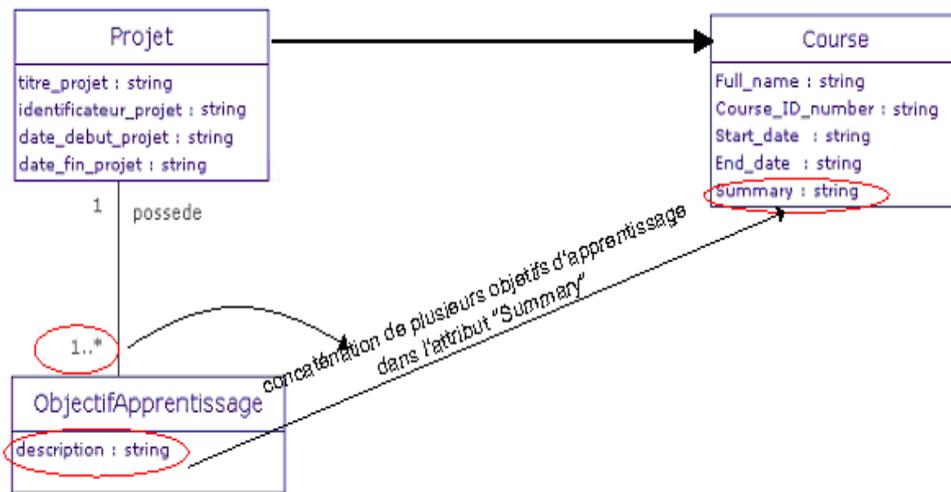


FIG. 6.22: Affecter un ensemble de valeurs d'une classe à un seul attribut d'une autre classe

dit, cet attribut prend une seule valeur en instanciant le concept "Course". Dans un autre sens, il faut affecter plusieurs objectifs d'apprentissage (relation "possede" de la classe "Etape" (cardinalité 1) avec la classe "ObjectifApprentissage" (cardinalité 1..*)) concernant l'étape à **un seul attribut ("Summary")** de la classe "Course". Pour cela, nous créons un "Helper" dans le contexte de la classe "Etape" cherchant à récupérer un ensemble des objectifs d'apprentissage (attribut "description" de la classe "ObjectifApprentissage") pour les concaténer dans une chaîne de caractères.

Revenons à la **troisième règle (rule Tache2Course)** (figure 6.24), nous avons la classe "Objectif" dans le méta-modèle PPC qui n'a pas un concept voisin dans le méta-modèle Moodle. Pour cela, nous décrivons les objectifs (Objectif) concernant la tâche (Tache) en se servant de l'attribut "Summary" du concept "Course"

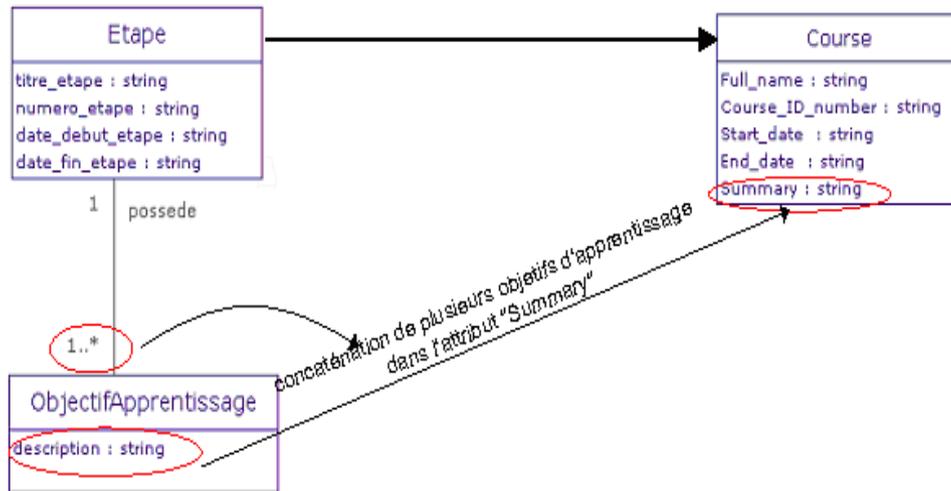


FIG. 6.23: Affecter un ensemble de valeurs d'une classe à un seul attribut d'une autre classe

de Moodle. De la même manière que ci-dessus (création d'un "Helper" dans le contexte de la classe "Tache" concaténant les objectifs dans une seule chaîne de caractères), nous résolvons le problème d'affectation de plusieurs objectifs (relation "possede" de la classe "Tache" (cardinalité 1) avec la classe "Objectif" (cardinalité 1..*)) concernant la tâche à un seul attribut ("Summary") de la classe "Course".

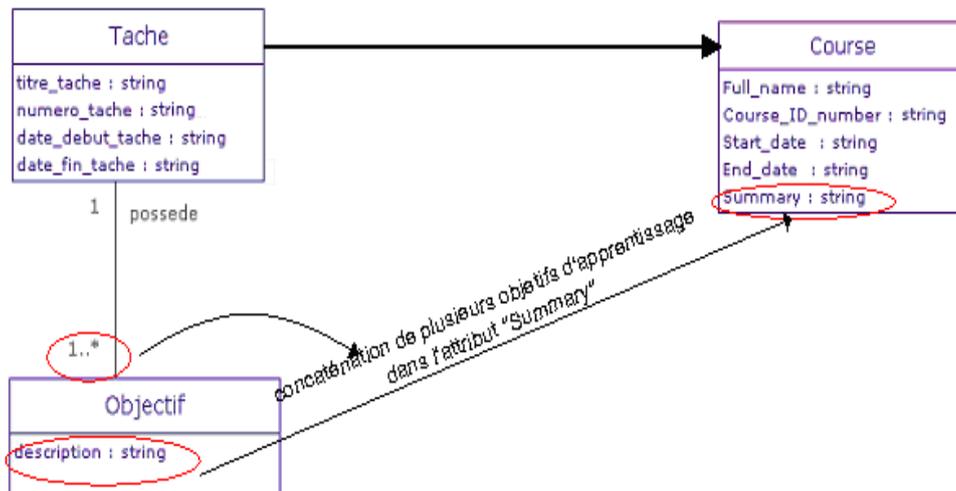


FIG. 6.24: Affecter un ensemble de valeurs d'une classe à un seul attribut d'une autre classe

6.6.3 Règle entre Production du côté méta-modèle PPC et Files du côté méta-modèle Moodle

Dans le méta-modèle PPC, il existe le concept "Production" permettant de décrire la production d'un projet, étape, et tâche. Ce concept a un concept voisin dans la plate-forme Moodle pouvant le remplacer : "Files". Dans la plate-forme Moodle, nous pouvons mettre les productions des cours en utilisant le concept "Files" et en y uploadant des fichiers.

Concernant la **sixième règle (rule Production2Files)** (figure 6.25), nous établissons une règle entre la classe "Production" du méta-modèle PPC et la classe "Files" du méta-modèle Moodle. Nous affectons la valeur de l'attribut "nom_production" de la classe "Production" du méta-modèle PPC à l'attribut "Folder_name" de la classe "Files" de Moodle, et la valeur de l'attribut "nom_fichier" du méta-modèle PPC à l'attribut "File" de celui de Moodle. Concernant la manipulation des relations, nous remplaçons les relations ("possede") entre "Production" d'un côté et "Projet", "Etape", et "Tache" de l'autre côté du méta-modèle PPC par la relation ("possede") entre "Files" et "Course" dans Moodle.

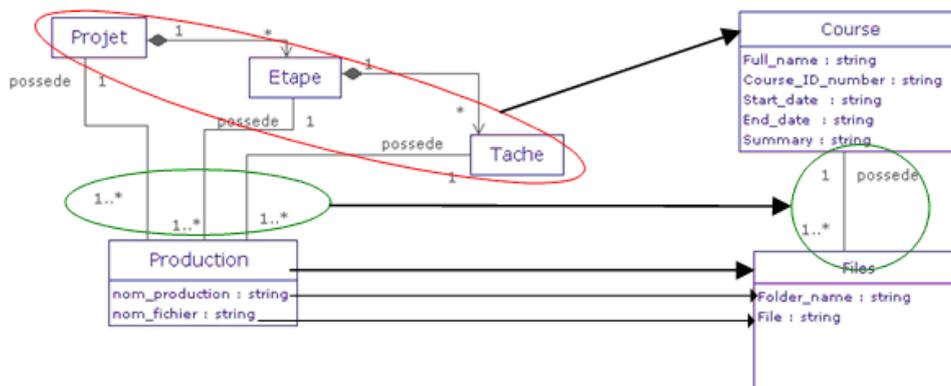


FIG. 6.25: rule Production2Files

6.6.4 Règle entre Ressource du côté méta-modèle PPC et Resource du côté méta-modèle Moodle

Dans le méta-modèle PPC, il existe la classe "Ressource" permettant de décrire les ressources d'une tâche. Cette classe a un équivalent dans le méta-modèle de la plate-forme Moodle ("Resource").

Concernant la **septième règle (rule Ressource2Resource)** (figure 6.26), nous établissons une règle de type (*MatchedRule*) qui a comme classe source "Ressource" du méta-modèle PPC et classe cible "Resource" du méta-modèle Moodle. Cette règle fait le "*mapping*" entre la classe "Ressource" du méta-modèle PPC et "Resource" du méta-modèle Moodle ainsi que la manipulation des attributs de ces classes de la façon suivante : elle affecte la valeur de l'attribut "description" de la classe "Ressource" à l'attribut "Name" de la classe "Resource" de Moodle. Concernant la manipulation des relations, nous remplaçons la relation entre les classes "Ressource" et "Tache" ("necessite") du méta-modèle PPC par la relation entre "Resource" et "Course" ("necessite") du méta-modèle Moodle.

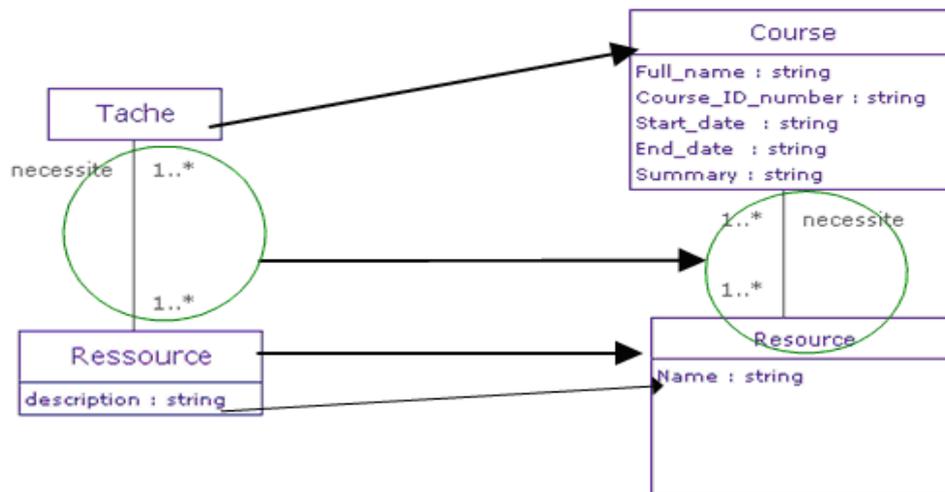


FIG. 6.26: rule Ressource2Resource

6.6.5 Règle entre Domaine, et Prerequis du côté méta-modèle PPC et Resource du côté méta-modèle Moodle

Dans le méta-modèle PPC, nous spécifions les domaines de chaque étape de projet par le concept "Domaine" et les prerequis d'une étape par le concept "Pre-requis". Dans la plate-forme Moodle, nous n'avons pas trouvé des concepts équivalents. Pour cela, nous n'avons trouvé pour définir les domaines et les prerequis

que le concept "Resource" du méta-modèle de la plate-forme Moodle pour lui affecter leurs valeurs.

Concernant la **huitième règle (rule Domaine2Resource)** (figure 6.27), nous avons, dans le méta-modèle PPC, le concept "Domaine" qui sert à décrire les domaines auxquels appartient l'étape. Dans le méta-modèle Moodle, nous avons choisi la classe "Resource" pour décrire le domaine d'une étape. Nous établissons une règle de type (*MatchedRule*) qui a comme classe source "Domaine" du méta-modèle PPC et classe cible "Resource" du méta-modèle Moodle. Cette règle fait le "*mapping*" entre la classe "Domaine" du méta-modèle PPC et "Resource" du méta-modèle Moodle ainsi que la manipulation des attributs de ces classes de la façon suivante : elle affecte la valeur de l'attribut "description" de la classe "Domaine" à l'attribut "Name" de la classe "Resource" de Moodle. Concernant la manipulation des relations, nous remplaçons la relation entre les classes "Domaine" et "Etape" ("appartient") du méta-modèle PPC par la relation entre "Resource" et "Course" ("necessite") du méta-modèle Moodle.

Concernant la **neuvième règle (rule Prerequis2Resource)** (figure 6.28),

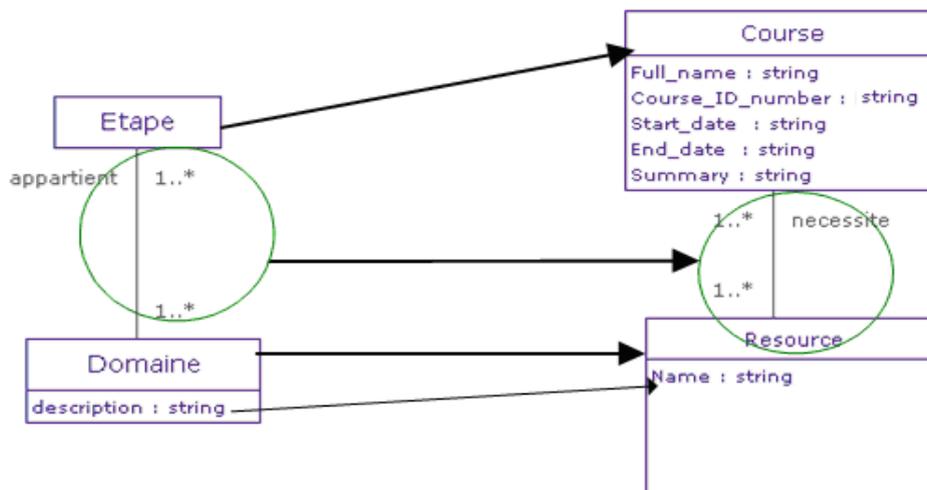


FIG. 6.27: rule Domaine2Resource

nous avons, dans le méta-modèle PPC, le concept "Prerequis" qui sert à décrire les prérequis nécessaires pour une étape et/ou une tâche. Aussi pour remplacer la classe "Prerequis" du méta-modèle PPC, nous avons choisi dans le méta-modèle Moodle la classe "Resource" pour décrire les prérequis d'une étape et/ou d'une tâche.

Nous établissons une règle de type (*MatchedRule*) qui a comme classe source "Prerequis" du méta-modèle PPC et classe cible "Resource" du méta-modèle Moodle. Cette règle fait le "*mapping*" entre la classe "Prerequis" du méta-modèle PPC

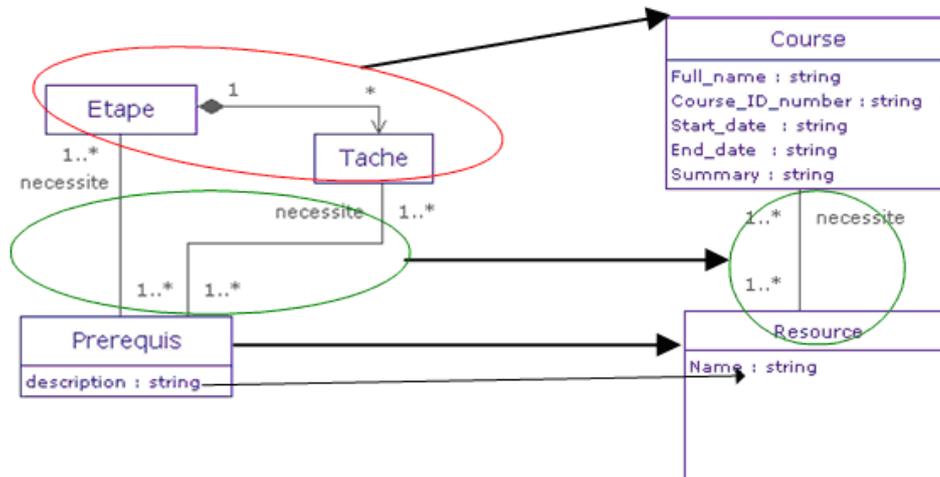


FIG. 6.28: rule Prerequis2Resource

et "Resource" du méta-modèle Moodle ainsi que la manipulation des attributs de ces classes de la façon suivante : elle affecte la valeur de l'attribut "description" de la classe "Prerequis" à l'attribut "Name" de la classe "Resource" de Moodle. Concernant la manipulation des relations, nous remplaçons les relations entre la classe "Prerequis" et ("Etape" et "Tache") d'un côté, et ("necessite") de l'autre côté du méta-modèle PPC par la relation entre "Resource" et "Course" ("necessite") du méta-modèle Moodle.

6.6.6 Règle entre Outil du côté méta-modèle PPC et Activity du côté méta-modèle Moodle

Dans le méta-modèle PPC, nous avons le concept Outil permettant de spécifier les outils nécessaires pour achever une tâche. Tandis que dans Moodle, le concept équivalent est "Activity". Dans le méta-modèle PPC, nous avons spécifié l'outil "Diagramme de Gantt" qui n'existe pas dans Moodle et à ce niveau là nous signalons une perte sémantique (figure 6.29).

Concernant la dixième règle (rule **Outil2Activity**) (figure 6.29), nous établissons une règle de type (*MatchedRule*) qui a comme classes sources "Outil", "Forum", "Chat" du méta-modèle PPC et classes cibles "Activity", "Forum", "Chat" du méta-modèle Moodle. Cette règle fait le "mapping" entre les classes "Outil", "Forum", "Chat" du méta-modèle PPC et "Activity", "Forum", "Chat" du méta-modèle Moodle ainsi que la manipulation des attributs de ces classes de la façon suivante : elle affecte la valeur de l'attribut "description" de la classe "Outil" à l'attribut "Name" de la classe "Activity" de Moodle. Concernant la

manipulation des relations, nous remplaçons la relation entre la classe "Outil" et "Tache" ("utilise") du méta-modèle PPC par la relation entre "Activity" et "Course" ("utilise") du méta-modèle Moodle.

Concernant la **onzième règle (rule Outil2Messages)** (figure 6.29), dans le mé-

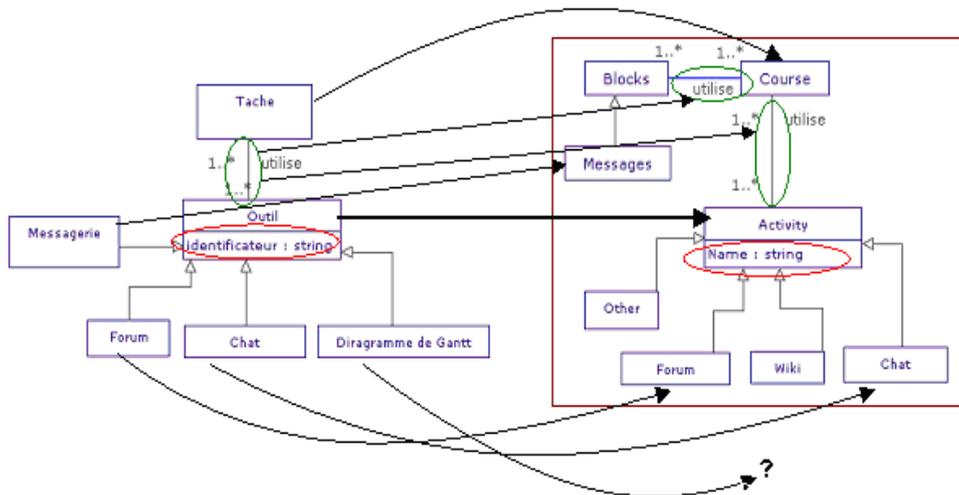


FIG. 6.29: rules : Outil2Activity et Outil2Messages

ta-modèle PPC la classe "Messagerie" hérite de la classe "Outil" tandis que dans le méta-modèle Moodle la classe "Messages" hérite de la classe "Blocks". Pour cela, nous établissons une règle de type (*MatchedRule*) entre les classes sources "Outil", "Messagerie" du méta-modèle PPC et les classes cibles "Blocks", "Messages" du méta-modèle Moodle. Cette règle fait le "mapping" entre les classes "Outil", "Messagerie" du méta-modèle PPC et "Blocks", "Messages" du méta-modèle Moodle ainsi que la manipulation des attributs de ces classes de la façon suivante : elle affecte la valeur de l'attribut "description" de la classe "Outil" à l'attribut "Name" de la classe "Blocks" de Moodle. Concernant la manipulation des relations, nous remplaçons la relation entre la classe "Outil" et "Tache" ("utilise") du méta-modèle PPC par la relation entre "Blocks" et "Course" ("utilise") du méta-modèle Moodle.

6.6.7 Règle entre Metrique du côté méta-modèle PPC et Quiz du côté méta-modèle Moodle

Dans le métamodèle PPC, nous avons le concept "Facteur" servant à décrire les facteurs d'évaluation d'un projet, une étape, et une tâche. Cette classe (Facteur) n'a pas une classe voisine dans le méta-modèle Moodle. Pour cela, nous considérons qu'à ce niveau là il y a une perte sémantique. Dans la plate-forme

Moodle, il n'est pas facile d'affecter des facteurs d'évaluation à un cours. Par contre, cette plate-forme contient des outils d'évaluation des étudiants affectés à un cours. Nous représentons ces outils par la classe Quiz.

Concernant la **douzième règle (rule Metrique2Quiz)** (figure 6.30), nous avons établi, malgré l'absence (dans Moodle) du concept voisin du concept "Facteur" (dans le méta-modèle PPC) une règle de type (*MatchedRule*) entre la classe source "Metrrique" du méta-modèle PPC et la classe cible "Quiz" du méta-modèle Moodle. Cette règle fait le "mapping" entre la classe "Metrrique" du méta-modèle PPC et "Quiz" du méta-modèle Moodle ainsi que la manipulation des attributs de ces classes de la façon suivante : elle affecte la valeur de l'attribut "nom" (Metrrique) à l'attribut "Name" (Quiz), la valeur de l'attribut "type" à l'attribut "Question_name", et la valeur de l'attribut "description" à l'attribut "Introduction". Concernant la manipulation des relations, nous rencontrons aussi une **perte sémantique** parce que la classe "Metrrique" dans le méta-modèle PPC est liée directement à la classe "Facteur" qui n'a pas d'équivalent dans Moodle. Cette règle de transformation va mener à des pertes sémantiques parce que certaines informations spécifiées dans le scénario disparaissent.

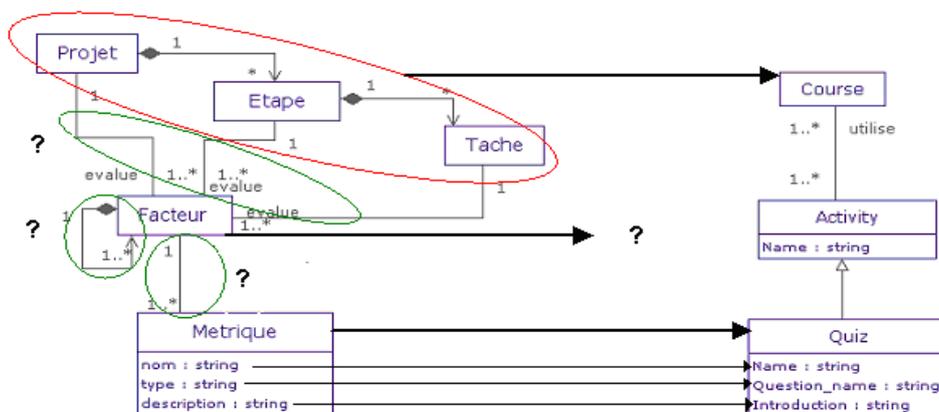


FIG. 6.30: rule Metrique2Quiz

6.6.8 Résultats

Le scénario PPC élaboré par ModX (représenté en format Ecore dans la partie gauche de la figure 6.31) est converti avec succès en un scénario PPC opérationnalisable par Moodle (représenté en format Ecore dans la partie droite de la figure

6.31). L'ingénieur pédagogique, expert de la plate-forme Moodle, peut maintenant déployer dans Moodle le scénario transformé.

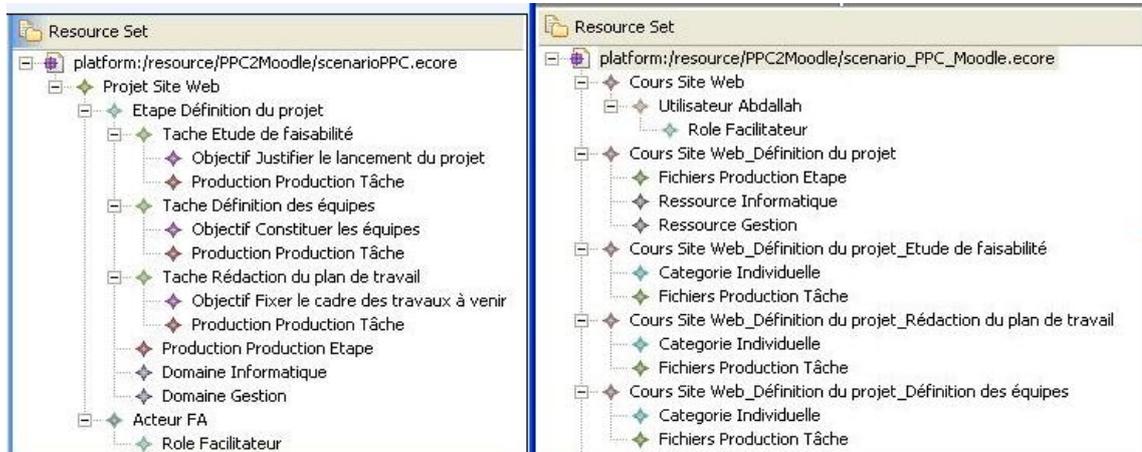


FIG. 6.31: Scénario PPC et scenario PPC transformé

6.7 Outil de Transformation d'un scénario PPC (OTCPPC)

Les règles de transformations développées nécessitent une installation de la plate-forme Eclipse et de quelques plug-ins, plus le chargement de scénario PPC créé sous ModX ⁶. Ce processus n'est pas évident pour l'enseignant. Pour cela, nous proposons dans cette section un **Outil de Transformation d'un scénario PPC (OTCPPC)** permettant à l'enseignant de charger son scénario PPC généré dans un fichier XMI par ModX et de lancer l'exécution des règles de transformations développées. Dans le cadre de cette thèse, nous n'avons développé que les règles de transformations depuis le méta-modèle PPC vers celui de la plate-forme Moodle permettant d'opérationnaliser le scénario PPC dans la plate-forme Moodle. Pour permettre à l'outil OTCPPC de transformer un scénario PPC en un scénario PPC opérationnalisable par n'importe quelle plate-forme, nous proposons une architecture permettant de capitaliser dans une base de règles, des règles de transformations depuis le méta-modèle PPC vers des méta-modèles de plusieurs plates-formes existantes.

L'architecture informatique (figure 6.33) que nous proposons permet de définir une base de règles de transformations depuis le méta-modèle PPC vers plusieurs méta-modèles correspondant à plusieurs plates-formes. L'outil de transformation OTCPPC est lié à une base de règles de transformations. Lorsque l'enseignant choisit la plate-forme avec laquelle il souhaite travailler, il la spécifie dans l'outil

⁶Comme nous avons expliqué dans le chapitre 4

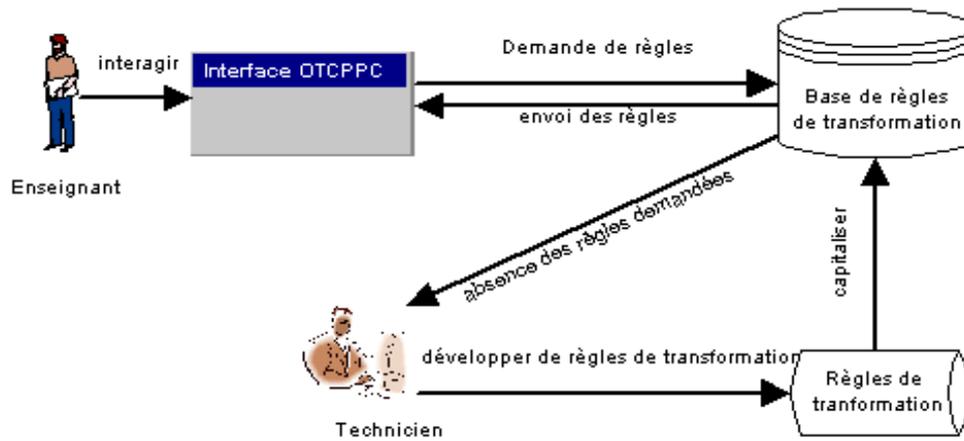


FIG. 6.32: Architecture proposé

qui va chercher dans la base de règles de transformations les règles correspondantes à la plate-forme choisie. Si l'outil de transformation trouve les règles correspondantes à la plate-forme choisie, il les exécute et transforme le scénario PPC en un scénario PPC opérationnalisable par la plate-forme choisie. Au cas où les règles de transformations ne se trouvent pas dans la base de règles, il devient nécessaire que le technicien intervienne pour définir et développer ces règles afin de les indexer et les sauvegarder dans la base de règles pour être capitalisées.

Pour implémenter cette architecture, nous avons besoin de définir les règles de transformations ATL à part et puis les appeler depuis le code d'un langage de programmation permettant de définir des interfaces graphiques pour l'outil. Nous avons trouvé sur le site d'ATL des programmeurs qui ont travaillé sur comment lancer des règles de transformations ATL depuis du code Java ⁷

6.7.1 Lancer des transformations ATL depuis Java

Pour lancer des transformations ATL depuis du code Java, nous avons utilisé des types prédéfinis par l'équipe ATLAS permettant de spécifier les méta-modèles (PPC et Moodle) et les modèles (scénario PPC et scénario PPC opérationnalisable par Moodle) dans EMF (*Eclipse Modeling Framework*).

Donc, l'idée principale est de coder les méta-modèles en `.ecore` (ex `PPC.ecore` et `Moodle.ecore`) et développer les règles de transformations ATL à part, dans des fichiers possédant une extension `.atn` (ex : `PPC2Moodle.atn`), comme nous avons expliqué dans la sous section 6.5.2. A l'exécution des transformations, l'environnement EMF avec les plug-ins ATL génèrent automatiquement un fichier `.asm` (ex : `PPC2Moodle.asm`). Ce fichier avec les deux fichiers des deux méta-modèles

⁷http://wiki.eclipse.org/ATL_Howtos#How_do_I_launch_transformations_programmatically.3F

doivent être appelés depuis des fonctions Java.

Les fonctions nécessaires pour lancer la transformation ATL sont :

- une fonction de type `ASMMModel` (type prédéfini) permettant d'initialiser, en spécifiant le chemin de chaque méta-modèle et celui du fichier où les règles de transformations ATL se trouvent (ex : `PPC2Moodle.asm`),
- une fonction de type `ASMMModel` (type prédéfini) permettant d'injecter des fichiers XMI d'entrée de modèle XMI (scénario PPC),
- une fonction de type `ASMMModel` (type prédéfini) permettant de retourner le modèle de sortie (ex : scénario_PPC_Moodle.ecore correspondant au scénario PPC opérationnalisable par Moodle),
- une fonction de type `ASMMModel` (type prédéfini) permettant de lancer la transformation ATL. Cette fonction a les paramètres suivants : le modèle de sortie (ex : scénario_PPC_Moodle.ecore correspondant au scénario PPC opérationnalisable par Moodle), le modèle d'entrée pour la transformation (ex : scénario_PPC.ecore correspondant au scénario PPC), le méta-modèle d'entrée pour la transformation (ex : PPC.ecore correspondant au méta-modèle PPC), le méta-modèle de sortie pour la transformation (ex : Moodle.ecore correspondant au méta-modèle Moodle), des paramètres supplémentaires pour la transformation, et les bibliothèques supplémentaires spécifiques à la transformation ATL.

Toutes ces fonctions ont été définies dans une classe Java que nous avons nommée "ATLTransformations.java". Cette classe est la classe "noyau" de l'Outil de Transformation de scénario PPC (OTCPPC). Donc, pour chaque plate-forme ⁸, il faut définir dans un fichier Java les transformations développées en ATL (comme nous avons montré). Chaque fichier Java (qui correspond au lancement des règles de transformation ATL depuis le méta-modèle PPC vers une plate-forme) est appelé depuis les boutons des interfaces de l'outil.

L'outil OTCPPC est composé donc : des classes Java contenant toutes les fonctions nécessaires pour lancer les règles de transformations ATL (un fichier Java par plate-forme), et des fichiers correspondant aux interfaces développées aussi avec le langage Java.

Dans la sous-section ci-dessous, nous présentons l'Outil de Transformation de scénario PPC (OTCPPC).

6.7.2 Présentation de l'outil OTCPPC

Dans cette sous-section, nous présentons l'Outil de Transformation de scénario PPC (OTCPPC) que nous avons développé dans le Laboratoire d'Informatique de l'Université du Maine (LIUM). Au lancement de cet outil, nous avons l'interface principale encore appelée interface d'accueil (figure 6.33). Cette interface est

⁸Nous n'avons défini que les règles transformations depuis le méta-modèle PPC vers celui de la plate-forme Moodle

composée de deux blocs de boutons principaux plus le bouton "Quitter". Le premier bloc contient les boutons permettant de choisir une plate-forme de formation, et le deuxième bloc contient les boutons permettant de choisir une plate-forme de gestion de projet. En cliquant sur un de ces boutons, l'enseignant spécifie vers quelle plate-forme il souhaite transformer son scénario PPC.

Toutes ces fonctions présentées ci-dessus sont accessibles aussi depuis le menu principal. L'enseignant peut ouvrir le menu principal "Fichier" pour choisir "Transformer" et après il choisit entre "Transformer vers une plate-forme de formation" ou "Transformer vers une plate-forme de gestion de projet" pour trouver les plates-formes de formation et de gestion de projet. Lorsque l'enseignant choisit

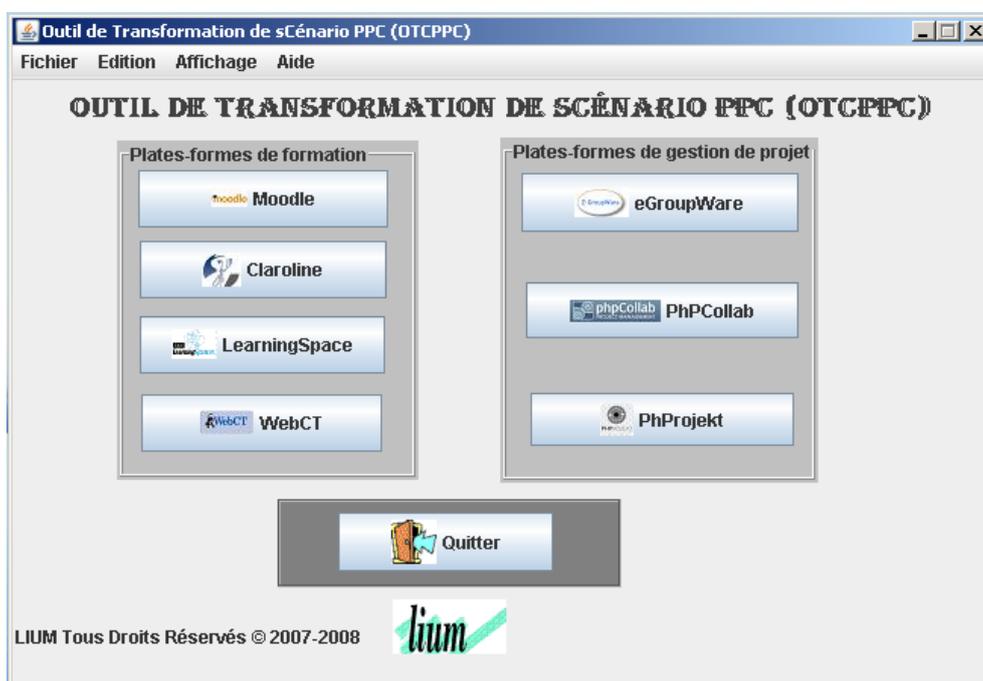


FIG. 6.33: Accueil OTCPPC

une plate-forme (de formation ou de gestion de projet), un autre menu s'ouvrant, est distingué par le nom de la plate-forme choisie au dessus des deux boutons "Parcourir" et "Transformer". Sur le bouton "Transformer", il y a aussi un logo de la plate-forme choisie. En bas, nous avons mis deux boutons : "Accueil" permettant l'enseignant de revenir à l'interface d'accueil si jamais il trouve qu'il veut changer la plate-forme, et "Quitter" lui permettant de quitter l'environnement de travail (figure 6.34). Cette interface permet à l'enseignant de charger son scénario PPC. Pour le charger, l'enseignant doit cliquer sur le bouton "Parcourir" lui permettant de spécifier l'emplacement de son scénario PPC (qui est un fichier xmi généré par exemple par ModX) sur l'ordinateur ou sur un espace de mémoire



FIG. 6.34: Choix de la plate-forme cible

périphérique. Après le chargement du scénario PPC, l'enseignant doit cliquer sur "Ouvrir" afin de l'ouvrir et qu'il soit prêt à la transformation (figure 6.35). Donc, l'outil OTCPPC permet à l'enseignant de visualiser le scénario PPC choisi dans un "conteneur" avec un "Scrollbar". Maintenant le scénario PPC est prêt à être transformé en un scénario PPC opérationnalisable par la plate-forme déjà choisie tout simplement en cliquant sur le bouton "Transformer" désigné aussi par le logo de la plate-forme cible (figure 6.36). Lorsque l'enseignant clique sur le bouton "Transformer", une nouvelle fenêtre s'affiche (figure 6.37) contenant le scénario PPC transformé en un scénario PPC opérationnalisable par la plate-forme choisie. Dans cette fenêtre, l'enseignant peut : cliquer sur le bouton "Transformer un autre scénario PPC" pour revenir à la fenêtre de chargement d'un scénario PPC, sauvegarder le scénario PPC transformé en cliquant sur le bouton "Sauvegarder" en bas à droite, revenir à la fenêtre d'accueil en cliquant sur le bouton "Accueil" afin de choisir une autre plate-forme, ou quitter l'outil en cliquant sur "Quitter".

6.8 Conclusion

Dans ce chapitre nous avons proposé une démarche par transformation de modèles afin d'intégrer un scénario PPC dans une plate-forme de formation. Nous avons commencé ce chapitre par l'étude de l'approche IDM (Ingénierie Dirigée

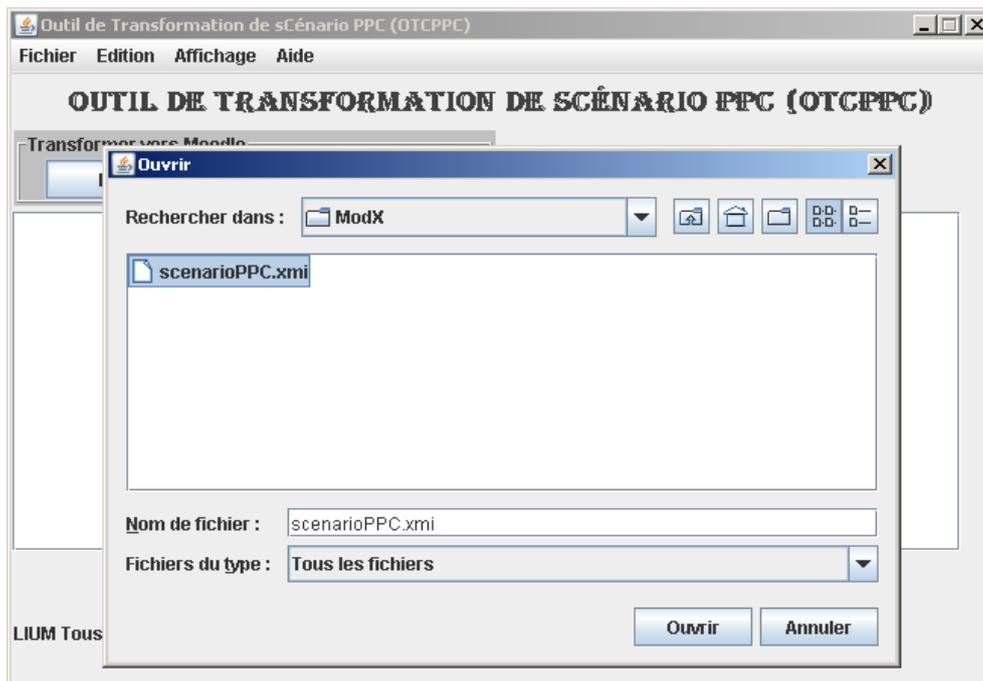


FIG. 6.35: Charger le scénario PPC

par les Modèles)(section 6.2). Dans la section 6.3, nous avons fait une étude d'une démarche particulière de l'approche IDM, le MDA (*Model Driven Architecture*) défini par l'OMG. Nous avons vu aussi comment introduire MDA dans l'activité de scénarisation et d'opérationnalisation pédagogique. Après cette utilisation de la démarche MDA dans l'activité de scénarisation, nous avons expliqué, dans la section 6.5, les principes des transformations de modèles en situant le scénario PPC en modèle source et le scénario PPC opérationnalisable par Moodle en modèle cible. Dans cette section, nous avons choisi le langage de transformation ATL (*Atls Transformation Language*) en l'expliquant et le décrivant et étudiant toutes les étapes techniques permettant de l'utiliser et l'implémenter. Nous avons proposé, dans la section 6.6, une vision de la plate-forme Moodle en proposant un méta-modèle de Moodle. Cette vision nous a permis de faire un travail de "*mapping*" entre le méta-modèle PPC proposé et celui de la plate-forme Moodle afin de définir, expliquer, et argumenter les règles de transformations ATL depuis le méta-modèle PPC vers celui de Moodle. Dans la section 6.7, nous avons utilisé le scénario PPC développé dans le chapitre 4 pour montrer comment le transformer en un scénario PPC opérationnalisable par Moodle. Notre processus de transformation ayant en méta-modèle de sortie celui de la plate-forme Moodle, le modèle produit est de facto opérationnalisable sous Moodle. Le résultat de la transformation que nous avons présentée dans cette section montre que le scénario

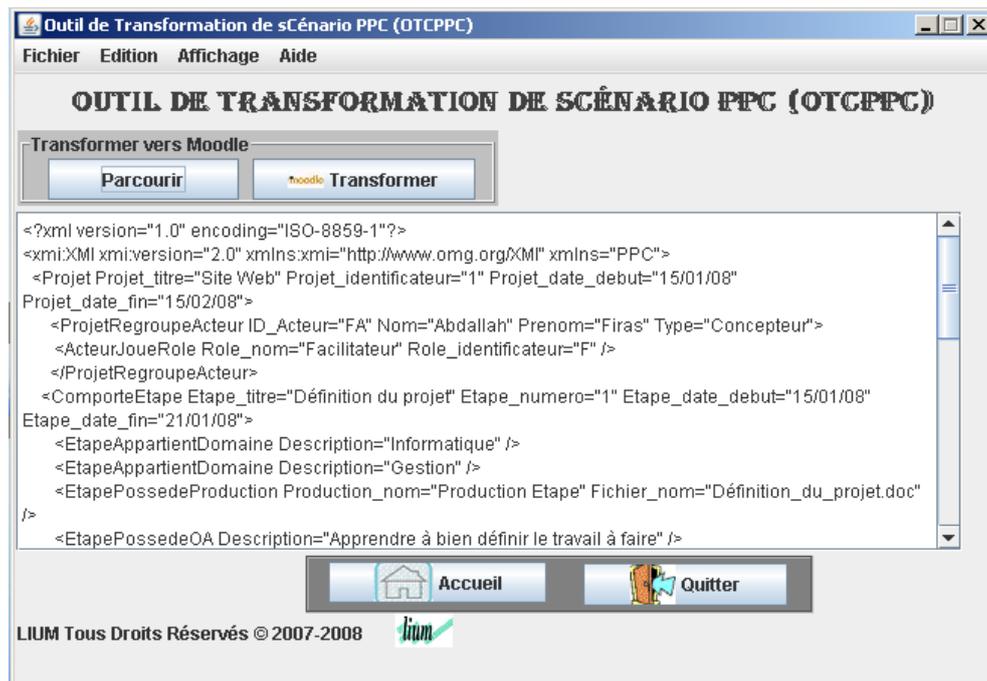


FIG. 6.36: Visualiser et transformer le scénario PPC

PPC opérationnalisable par Moodle est effectivement décrit par des concepts du méta-modèle de Moodle ; la chose qui prouve que l'ingénieur pédagogique, expert de la plate-forme Moodle, peut techniquement déployer dans Moodle le scénario transformé.

Afin de faciliter l'utilisation de ces règles de transformations par un enseignant non informaticien, nous avons proposé dans le section 6.8 un outil lui permettant de charger son scénario PPC généré par ModX (a priori) et d'utiliser les règles de transformations développées. L'architecture proposée de cet outil permet une capitalisation, dans une base de règles, des règles de transformations depuis le méta-modèle PPC vers les méta-modèles des plates-formes existantes.

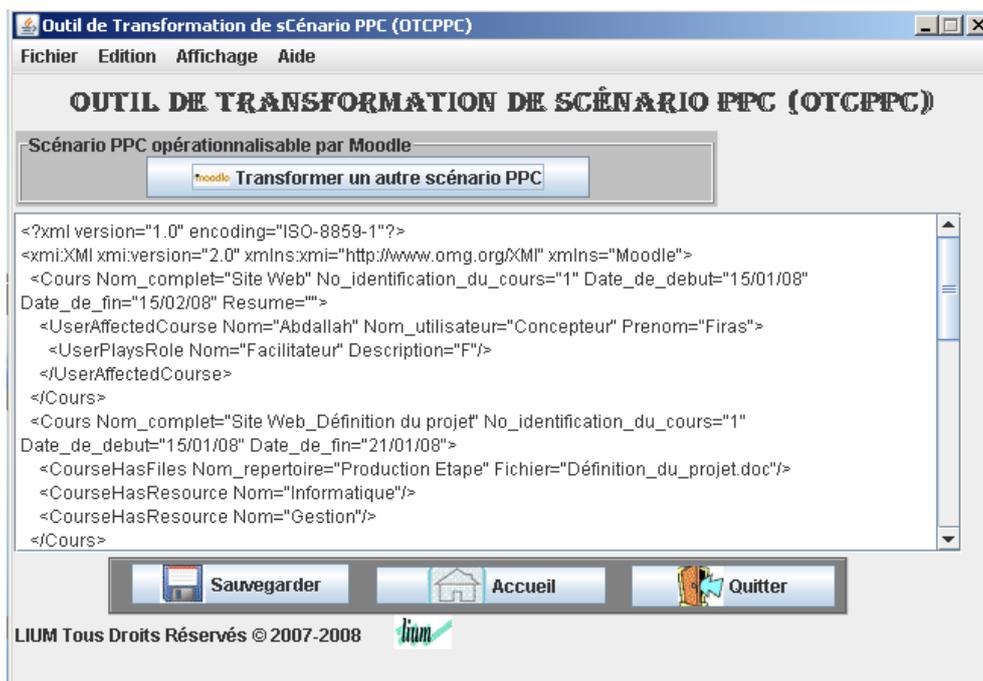


FIG. 6.37: Scénario PPC transformé

Chapitre 7

Conclusion générale

Dans cette thèse, nous nous sommes intéressés principalement à l'élaboration d'un méta-modèle dédié à la Pédagogie par Projet Collectif (PPC) permettant la mise en oeuvre de modèles de scénarios PPC pour conceptualiser une situation d'apprentissage dans un contexte de Pédagogie par Projet Collectif (PPC). Nous concluons, maintenant, en présentant un bilan des travaux réalisés. Enfin, nous terminons en exprimant et en discutant des perspectives pour ce travail.

7.1 Bilan des travaux réalisés

Notre contribution se présente sous la forme de trois propositions complémentaires :

- un méta-modèle dédié à la Pédagogie par Projet Collectif (PPC) permettant à l'enseignant concepteur d'une situation d'apprentissage de PPC d'élaborer son scénario PPC,
- un modèle de fonctionnalités PPC permettant à l'enseignant concepteur d'une situation d'apprentissage de la PPC de choisir une plate-forme opérationnelle existante la plus adaptable à ses besoins exprimés dans son scénario PPC,
- un Outil de Transformation de sCénario PPC (OTCPPC) permettant à l'enseignant concepteur d'une situation d'apprentissage de Pédagogie par Projet Collectif (PPC) de transformer son scénario PPC en un scénario PPC opérationnalisable par la plate-forme cible. Cet outil s'appuie sur un système formel de transformations de méta-modèle que nous avons développé dans le langage ATL.

7.1.1 Méta-modèle dédié à la Pédagogie par Projet Collectif (PPC)

Nous avons proposé un méta-modèle dédié à la Pédagogie par Projet Collectif (PPC) facilitant l'élaboration de modèles pour la conception de situations d'apprentissage dans un contexte de la PPC sur des plates-formes de formations et/ou de gestion de projet. L'étude des différents langages correspondant en partie à cet objectif a permis de mettre en évidence i) les limites de l'approche généraliste IMS-LD, les difficultés rencontrées par les enseignants pour la maîtriser et exprimer leurs besoins de scénarisation ii) l'existence des langages dédiés à un domaine précis : le langage de modélisation CPM (*Cooperative Problem-based learning Meta-model*) est dédié à l'apprentissage par problème (*Problem Based Learning*) [126], le langage LDL (*Learning Design Language*) s'appuyant sur un méta-modèle défini pour le CSCW [143]. Ce constat nous a alors amené à positionner notre proposition comme un méta-modèle graphique pour la conception des situations d'apprentissage de PPC.

Le méta-modèle PPC proposé permet de décrire les différents scénarios de la PPC. Ces scénarios aident l'équipe pluridisciplinaire de conception à décrire, spécifier et documenter une situation d'apprentissage de la PPC qu'ils conçoivent. De plus, le caractère visuel du méta-modèle et des scénarios (les modèles) permet d'abstraire la complexité de la PPC mais constitue également une représentation adéquate pour favoriser le dialogue, la compréhension et l'implication des différents intervenants.

Cette première proposition a fait l'objet d'une publication dans une conférence internationale [3]. Le méta-modèle PPC, permet la modélisation de scénarios d'apprentissage indépendamment des fonctionnalités des plates-formes. La seconde proposition étend notre contribution en prenant en compte ces plates-formes.

7.1.2 Modèle de fonctionnalités PPC

Nous avons proposé un modèle de fonctionnalités pour guider le choix des enseignants concepteurs pour une plate-forme la mieux adaptée à leurs besoins formulés à travers un scénario PPC. L'enseignant peut être amené à choisir une plate-forme parmi plusieurs mises à sa disposition ou bien évaluer dans quelle mesure une certaine plate-forme est bien adaptée à son besoin (scénario). Ce modèle favorise la communication entre le pédagogue et l'ingénieur et permet de rationaliser le choix d'une plate-forme au regard de la PPC. Son utilisation incite non seulement à faire un choix compatible PPC mais elle fournit aussi de facto les principales lignes de la paramétrisation de la plate-forme sélectionnée.

Cette proposition répond à un des problèmes majeurs pour les enseignants qui est le choix d'une plate-forme ou système d'apprentissage assisté par les Technologies de l'Information et de la Communication (un site Web par exemple) supportant la pédagogie qu'ils désirent mettre en oeuvre. La meilleure solution est celle qui

concilie le point de vue pédagogique et le point de vue technologique de la façon la plus satisfaisante : un outil qui respecte les intentions pédagogiques de l'enseignant dans le contexte réel dans lequel il se trouve.

Cette proposition nous a permis d'adopter une stratégie qui consiste à assister l'enseignant pour l'implémentation de son scénario dans la plate-forme choisie, plutôt que développer une nouvelle plate-forme. Notre travail cherche à réutiliser ce que les plates-formes existantes proposent.

7.1.3 Outil de Transformation de sCénario PPC (OTCPPC)

Nous avons proposé un Outil de Transformation de sCénario PPC (OTCPPC). Cet outil prend en entrée des règles de transformation du méta-modèle PPC vers le méta-modèle de la plate-forme et les applique au scénario PPC exprimé dans un fichier XMI pour le transformer en un scénario opérationnalisable par la plate-forme cible. Nous avons adopté pour définir les règles de transformation une démarche par transformation de modèles depuis le méta-modèle de la situation d'apprentissage de la PPC vers celui de la plate-forme cible.

Cette proposition, faisant l'objet de deux publications [2], [4] nous a permis d'adopter une stratégie qui consiste à assister l'enseignant à élaborer son scénario d'apprentissage de la PPC indépendamment de toute technologie particulière (plate-forme cible). Après l'élaboration de son scénario PPC, il utilise l'outil de transformation OTCPPC pour transformer son scénario PPC en un scénario PPC opérationnalisable sur la plate-forme cible.

7.2 Apports de la thèse

Nos travaux répondent en partie à de nombreuses problématiques actuellement étudiées dans la communauté française des EIAH. Concernant le cadre de recherche général de l'ingénierie des EIAH, nous nous sommes focalisés dans ce travail de thèse sur la phase de conception d'une situation d'apprentissage de la Pédagogie par Projet Collectif (PPC) en vue d'une formation médiatisée. Il s'agit d'un contexte dans lequel les travaux "orientés pédagogie" [198] à forte connotation d'ingénierie sont actuellement les plus nombreux. Plus précisément, nous avons défini dans cette thèse un processus de conception et de diffusion des situations d'apprentissage dans un contexte de Pédagogie par Projet Collectif (PPC). Ce processus est un processus générique d'ingénierie qui intègre la nécessaire compréhension partagée entre l'enseignant et l'expert de la plate-forme (ingénieur pédagogique) lors du choix et de l'opérationnalisation d'un scénario. Cette thèse a montré qu'un tel processus peut s'articuler sur la notion de méthode comme moyen de scénarisation pédagogique et sur une approche d'Architecture Dirigée par les Modèles (MDA) comme moyen privilégié de compréhension partagée.

Le méta-modèle dédié à la Pédagogie par Projet Collectif (PPC) que nous avons défini permet d'encadrer, et par là d'assister, l'expression du scénario pédagogique PPC par les enseignants. Notre travail met en évidence et décrit l'apport de la méta-modélisation, pour spécifier des scénarios d'apprentissage de la PPC. Ceci permet une meilleure capitalisation des connaissances et améliore la communication entre les différents intervenants de l'équipe pluridisciplinaire de conception. La proposition d'un modèle de fonctionnalités permet de guider le choix des enseignants concepteurs pour une plate-forme la mieux adaptée à leurs besoins formulés à travers leurs scénarios PPC. Enfin, notre proposition d'une démarche par transformation de modèles depuis le méta-modèle de la situation d'apprentissage de la PPC vers celui de la plate-forme cible outillée informatiquement assiste l'enseignant et l'ingénieur-pédagogique à médiatiser la situation d'apprentissage.

Notre proposition se résume par le processus suivant : L'enseignant-concepteur utilise le méta-modèle dédié à la Pédagogie par Projet Collectif (PPC) pour élaborer le scénario PPC qu'il souhaite mettre en oeuvre. Dans une seconde étape, l'enseignant-concepteur utilise le modèle de fonctionnalités PPC afin de choisir la plate-forme qui correspond le plus aux besoins formulés dans son scénario d'apprentissage PPC. A la fin, il utilise l'outil de transformation proposé permettant d'appliquer les règles de transformation de modèles pour transformer son scénario PPC en un scénario PPC opérationnalisable par la plate-forme choisie. Ce dernier est fourni à l'ingénieur pédagogique, chargé de le déployer dans la plate-forme cible, afin d'être "joué" par les acteurs de la Pédagogie par Projet Collectif (PPC).

7.3 Perspectives

Les différents travaux de cette thèse participent à un processus générique d'ingénierie intégrant la nécessaire compréhension partagée entre l'enseignant et l'ingénieur pédagogique lors du choix et de l'opérationnalisation d'un scénario d'apprentissage de PPC sur une plate-forme existante. Ce processus ouvre de nombreuses perspectives à moyen et long termes. Toutefois, notre contribution doit être suivie dans l'immédiat de plusieurs travaux de validation à grande échelle : méta-modèle PPC, modèle de fonctionnalités PPC et outil de transformation. Par validation à grande échelle nous voulons dire le retour sur usage par les apprenants et les enseignants ou bien plus précisément par les acteurs de la situation d'apprentissage de la PPC.

C'est vrai que nous avons utilisé le méta-modèle PPC proposé dans cette thèse pour élaborer un scénario d'apprentissage de la PPC (chapitre 4). Mais, cette proposition nécessite pour la généraliser de mener plusieurs expérimentations par des enseignants "non chercheurs" en mettant le méta-modèle PPC à leur disposition. Nous avons donné un exemple d'utilisation du modèle de fonctionnalités PPC

lorsque nous l'avons utilisé comme un moyen de catégoriser la plate-forme Moodle (chapitre 5) et d'identifier si elle est, ou non, dédiée à la PPC. Mais, ce modèle de fonctionnalités PPC n'était pas (faute du temps) validé par une expérimentation avec les enseignants. De même l'outil OTCPPC proposé est validé dans le sens qu'il transforme un scénario PPC en un scénario PPC opérationnalisable par la plate-forme cible. Mais il n'est pas validé dans le sens d'utilisation par l'enseignant. Cette expérimentation nécessite une installation de cet outil sur les machines des enseignants pour évaluer la facilité d'utilisation.

A long terme et concernant la proposition du modèle de fonctionnalités PPC, il est clair qu'une utilisation manuelle telle que nous l'avons illustrée dans le chapitre 5, même si elle a déjà été utilisée, n'est pas optimale et est réservée à des spécialistes. Nous envisageons de la doter d'un outil d'assistance et de l'expérimenter sur des cas réels.

Aussi à long terme, nous comptons nous atteler à l'intégration totale de nos propositions afin d'obtenir un processus automatisé qui permet à l'enseignant de déployer son scénario PPC directement dans la plate-forme cible sans devoir recourir systématiquement à l'expertise de l'ingénieur pédagogique. L'outil de transformation OTCPPC proposé transforme un scénario PPC en un scénario PPC opérationnalisable par la plate-forme cible mais à condition de la présence de l'ingénieur pédagogique. Un des défis majeurs sera de minimiser, voire de faire disparaître, l'intervention de l'ingénieur pédagogique lors du déploiement du scénario pédagogique.

Nous souhaitons aussi intégrer un outil d'élaboration des scénarios PPC dans l'outil de transformation. Nous avons bien utilisé l'outil ModX pour élaborer un scénario PPC et puis nous avons utilisé l'outil OTCPPC pour le transformer en un scénario PPC opérationnalisable par la plate-forme cible. Pour faciliter cette tâche à l'enseignant, l'idéale est proposer un **seul** outil intégrant toutes les fonctionnalités depuis l'élaboration d'un scénario d'apprentissage de la PPC, en passant par les fonctionnalités d'assistance pour le choix d'une plate-forme la plus adaptée aux besoins exprimés dans le scénario PPC, jusqu'à la transformation de ce scénario en un scénario déployé.

7.4 Conclusion

Cette thèse s'inscrit dans le cadre des recherches pluridisciplinaires des EIAH et plus particulièrement de l'ingénierie des EIAH ; dès lors, nos recherche nous ont permis de nous intéresser à plusieurs domaines allant des sciences humaines et sociales jusqu'au génie logiciel. Cette thèse, par l'utilisation et l'application des méthodes et techniques de métamodélisation et de transformations de modèles, est très orientée Génie Logiciel. Cependant elle est, par son domaine d'application, aussi, au coeur des EIAHs.

Bibliographie

- [1] J. Abd-Ali and K. El Guemhioui. Transformation horizontale de psms : de ejb à .net. Technical report, Université du Québec en Outaouais (UQO) Département d'informatique et d'ingénierie, 2005. [cited at p. 124]
- [2] F. Abdallah. Transformation de modèles pour opérationnaliser un scénario pédagogie par projet collective (ppc) : Etude de cas moodle. In *Secondes Rencontres Jeunes Chercheurs en EIAH*, Lille, France, 15-16 mai 2008. [cited at p. 113, 133, 159]
- [3] F. Abdallah, C. Toffolon, and B. Warin. Assistance to project-based learning support : from learning models to platforms. pages 244–251. IADIS, 6-8 Juillet 2007. [cited at p. 15, 158]
- [4] F. Abdallah, C. Toffolon, and B. Warin. Models transformation to implement a project-based collaborative learning (pbcl) scenario : Moodle case study. In *The 8th IEEE International Conference on Advanced Learning Technologies (ICALT'2008)*, Santander, Cantabria, Spain, 01-05 July 2008. [cited at p. 113, 133, 159]
- [5] ADL-SCORM. Adl sharable content object reference model 2004 overview. Technical report, 2004. [cited at p. 49]
- [6] F. Allilaire and T. Idrissi. Adt : Eclipse development tools for atl. Technical report, IRIN, University of Nantes, http://www.sciences.univ-nantes.fr/lina/atl/www/papers/ADT_AllilaireIdrissi.pdf, 2004. [cited at p. 128, 129]
- [7] M. Altet. *Les pédagogies de l'apprentissage*. Collection Pédagogues et Pédagogies, Presses Universitaires de France, 2003. [cited at p. 20, 23, 25, 26, 72]
- [8] L. Arenilla, B. Gossot, M. C. Rolland, and M. P. Roussel. *Dictionnaire de pédagogie*. Bordas, 1996. [cited at p. 8, 20, 21, 22, 25]
- [9] C. Atkinson and T. Kuhne. Model-driven development : A metamodeling foundation. *IEEE Software*, September 2003. [cited at p. 119]

- [10] C. Bailey, M. T. Zalfan, H. C. Davis, K. Fill, and G. Conole. Panning for gold : Designing pedagogically-inspired learning nuggets. *Educational Technology Society*, 9(1) :P. 113–122, 2006. [cited at p. 55, 56, 66]
- [11] M. Baker, E. de Vries, K. Lund, and M. Quignard. Interactions épistémiques médiatisés par ordinateur pour l'apprentissage des sciences : bilan de recherches. In 2001 Hermès, editor, *C. Desmoulin, M. Grandbastien, J-M. Labat*, pages p. 21–32, Paris, 2001. Sixièmes journées francophones Environnements Interactifs d'Apprentissage avec Ordinateur (EIAO'2001). [cited at p. 99]
- [12] L. J. Bannon. Issues in computer-supported collaborative learning. In *C. O'Malley, (Ed.), NATO Advanced Workshop on Computer-Supported Collaborative Learning*, Maratea, Italy, 1989. [cited at p. 99, 100]
- [13] L. J. Bannon and K. Schmidt. *CSCW : Four Character in Search of a Context*, chapter Studies in Computer Supported Cooperative Work : Theory, Practice ans Design, pages p. 3–16. 1991. [cited at p. 95]
- [14] B. Bayard. *Problem-based learning in dietetic education : A descriptive and evaluative case study and an analytical comparison with a lecture based method*. PhD thesis, University of Wisconsin, 1994. [cited at p. 37, 39]
- [15] K. Benali, G. Bourguin, B. David, A. Derycke, and C. Ferraris. Collaboration / coopération. In *Information, Interaction, Intelligence, Actes des 2 èmes assises nationales du GDR I3*, Nancy : Cépaduès, 2002. [cited at p. 95, 98]
- [16] M. Berton, B. Hudon, B. Juteau, M. Synnott, M. Lauzon, M. Bournival, and L. Ferland-Lalancette. La collaboration à distance, un outil stratégique en alphabétisation - rapport de recherche-action. Technical report, Document rédigé pour le Centre des lettres et des mots (CLEM) dans le cadre du projet communautaire EST@CAD, Montréal, Octobre 2005. [cited at p. 8, 20, 21, 29, 31]
- [17] M.L. Betbeder. *Symba : un environnement malléable support d'activités collectives en contexte d'apprentissage*. PhD thesis, Université du Maine, 2003. [cited at p. 12, 29, 78, 80, 88, 98, 99, 100, 101]
- [18] S. Le Beux, P. Marquet, and J.-L. Dekeyser. Model driven engineering benefits for high level synthesis. Technical report, Institu national de recherche en informatique et en automatique, August 2008. [cited at p. 119]
- [19] J. Bézivin. From object composition to model transformation with the mda. In *TOOLS'USA*, Santa Barbara, August 2001. [cited at p. 124]
- [20] J. Bézivin. In search of a basic principle for model driven engineering. *V(2)* :21–24, April 2004. [cited at p. 119]

- [21] J. Bézivin. Sur les principes de base de l'ingénierie des modèles. In *RSTI - L'objet - Où en sont les objets ?*, pages pp. 145–156, Octobre 2004. [cited at p. 124]
- [22] J. Bézivin and Groupe ATLAS. Collaboration université-industrie à partir de la plate-forme eclipse. Technical report, Université de Nantes, <http://www.sciences.univ-nantes.fr/lina/atl/>, 2006. [cited at p. 118, 206]
- [23] J. Bézivin, M. Belaunde, and R. Marvie. Transformations et modèles plates-formes. In *Ingénierie dirigée par les modèles : OFTA*. A. 30 (Ed.), 2004. [cited at p. 119]
- [24] J. Bézivin and X. Blanc. Mda : Vers un important changement de paradigme en génie logiciel. Technical report, <http://www.devreference.net/>, Juillet 2002. [cited at p. 122]
- [25] J. Bézivin, M. Blay-Fornarino, M. Bouzeghoub, J. Estublier, and J. M. Favre. Rapport de synthèse idm. Technical report, Synthèse de l'AS CNRS sur l'Ingénierie Dirigée par les Modèles, décembre 2005. [cited at p. 114, 117, 119]
- [26] J. Bézivin, G. Dupé, F. Jouault, G. Pitette, and J. Eddine Rougui. First experiments with the atl model transformation language : Transforming xslt into xquery. In *OOPSLA 2003 Workshop*, Anaheim, California, 2003. [cited at p. 125, 126, 127]
- [27] J. Bezivin and O. Gerbé. Towards a precise definition of the omg/mda framework. San Diego, USA, November 2001. ASE'01, Automated Software Engineering. [cited at p. 68, 115, 119]
- [28] M. Blay-Fornarino and P. Franchi. Espace technologique langages et idm. Technical report, Synthèse de l'AS CNRS sur l'Ingénierie Dirigée par les Modèles, décembre 2005. [cited at p. 117]
- [29] J. Boaler. Alternative approaches to teaching, learning, and assessing mathematics. Athens, Greece, 1998. Paper presented at the European Conference for Research on Learning and Instruction. [cited at p. 37]
- [30] J. Boaler. Mathematics for the moment, or the millennium ? what a british study has to say about teaching methods. *Education Week*, March 31 1999. [cited at p. 37]
- [31] P. Boulet, J.L. Dekeyser, C. Dumoulin, and P. Marquet. Mda for soc embedded systems design, intensive signal processing experiment. In *SIVOES-MDA workshop at UML 2003*, San Francisco, October 2003. [cited at p. 120]
- [32] D. Boulliber. *La loi du support : leçons de trois ans d'enseignement numérique à distance*, volume 1 (2), pages 145 – 172. Les cahiers du numérique, l'université virtuelle, 2000. [cited at p. 93]

- [33] A. Bouthry, P. Chevalier, S. Ravet, and J-L. Schaff. Choisir une solution de téléformation - étude 2000. Technical report, Etude réalisée par Aska, Le Préau, et Klr.fr, http://www.preau.ccip.fr/etudes/etude_teleformation/teleformation.php, 2000. [cited at p. 93]
- [34] J.P. Boutinet. *Psychologie des conduites à projet*. Presses Universitaires de France, 1993. [cited at p. 26]
- [35] C.M. Brody. Collaboration or cooperative learning? complimentary practices for instructional reform. *The Journal of Staff, Program Organizational Development*, vol.12(no.3) :133–143, 1995. [cited at p. 29, 30]
- [36] M. Bru. *Les méthodes en pédagogie*. Que sais-je ?, 2006. [cited at p. 24, 25, 41]
- [37] J. Bruner. *The process of education*. 1960. [cited at p. 21, 22]
- [38] F. Budinsky, D. Steinberg, R. Ellersick, and T. Grose. *Eclipse Modeling Framework*. Addison-Wesley Professional, 2004. [cited at p. 116, 126, 128]
- [39] D. Burgos, M. Arnaud, M. Neuhauser, and K. Koper. Ims learning design : la flexibilité pédagogique au service des besoins de l’e-formation. <http://www.epi.asso.fr/revue/articles/a0512c.htm>, 2005. [cited at p. 11, 46, 51, 58]
- [40] T. Cameron, H. S. Barrows, and S. M. Crooks. Distributed problem-based learning at southern illinois university school of medicine. pages p. 86–93, Stanford University, Palo Alto, California, USA. Computer Supported Collaborative Learning (CSCL’99). [cited at p. 94]
- [41] A. Canals, C. Le-Camus, M. Feau, G. Jolly, V. Bonnafous, and P. Bazavan. Une utilisation opérationnelle d’atl : l’intégration de la transformation de modèles dans le projet topcased. In *Génie Logiciel (73)*, pages 21–26, 2005. [cited at p. 124, 125]
- [42] P. A. Caron. *Ingénierie dirigée par les modèles pour la construction de dispositifs pédagogiques sur des plateformes de formation*. PhD thesis, Université des sciences et technologies de Lille, Lille, France, 18 Juin 2007. [cited at p. 57, 58, 84, 104, 113, 115, 116, 122, 124]
- [43] P.A. Caron. Contextualisation de dispositifs pédagogiques sur des applications web 2.0. Strasbourg, France, 2006. Actualité de la Recherche en Education et en Formation. [cited at p. 11, 12, 115]
- [44] P.A. Caron. Bricoles : une approche dispositive des applications web 2.0 utilisables pour enseigner. Lausanne, Suisse, 2007. EIAH. [cited at p. 12, 115]
- [45] P.A. Caron, A. Derycke, F. Hoogstoel, X. Le Pallec, and B. Warin. Scénarios et dispositifs de formations spécialisés : application de la démarche

- d'ingénierie bricole pour une instanciation sur moodle. Montréal, Canada, 14-15 mai 2007. Colloque international Scénario. [cited at p. 12, 115]
- [46] P.A. Caron, F. Hoogstoel, X. Le Pallec, and B. Warin. Construire des dispositifs sur la plateforme moodle - application de l'ingénierie bricoles. In *MoodleMoot-2007*, Castres, France, 14 - 15 Juin 2007. [cited at p. 109, 121]
- [47] P. H. Carstensen and K. Schmidt. *Computer Supported Cooperative Work : New Challenges to Systems Design*. Handbook of Human Factors, Tokyo, 1999. [cited at p. 96]
- [48] O. Catteau and A. Mayere. Gestion des projets tuteurés à l'aide d'une plateforme pédagogique. In *Colloque sur les Technologies de l'Information et de la Communication dans l'Enseignement Supérieur et l'Entreprise TICE*, Toulouse, France, 2006. [cited at p. 26]
- [49] B. Charlier, A. Daele, F. Docq, M. Lebrun, S. Lusalusa, R. Peeters, and N. Deschryver. Tuteurs en ligne : quels rôles, quelle formation ? In *CNED*, Décembre 1999. [cited at p. 45, 76]
- [50] C. Choquet and S. Iksal. Modélisation et construction de traces d'observation d'une activité d'apprentissage : une approche pour la réingénierie d'un eiah. Présentation pour la journée Traces réflexives et EIAH, 18 Avril 2007. [cited at p. 12]
- [51] A. Clave. D'uml à mda en passant par les méta modèles. Technical report, La Lettre d'ADELI n 56, 2004. [cited at p. 120]
- [52] Workflow Management Coalition. Terminology glossary. Technical report, Technical Report WfMC-TC-1011, Version 3.0, February 1999. [cited at p. 98]
- [53] L. Collet, A. Derycke, F. Hoogstol, and X. Le Pallec. A new user-centered design of distributed collaboration environments : the réciprocité project. In *ICEIS 2003, 5th International Conference on Enterprise Informations Systems*, Ecole Supérieure d'Electronique de l'Ouest, Angers - France, 23-26 Avril 2003. [cited at p. 96]
- [54] B. Combemale and S. Rougemaille. *ATL - Atlas Transformation Language*. Master 2 Recherche SLCP, module rtm edition, 2005. [cited at p. 125, 126]
- [55] M. A. Constantino-Gonzales and D. Suthers. Coaching collaboration in a computer-mediated learning environment. In *In Proceedings of Computer Support for Collaborative Learning*, pages pp. 583-584, Boulder, January 7-11 2002. Hillsdale : Lawrence Erlbaum Associates. [cited at p. 101]
- [56] Coppercore. *Coppercore, The IMS Learning Design Engine*. <http://coppercore.sourceforge.net>, 2007. [cited at p. 11, 47, 51, 84]

- [57] P. Croisy. *Collecticiels temps réel et apprentissage coopératif : des aspects sociaux et pédagogiques jusqu'au modèle multi-agents de l'interface de groupe*. PhD thesis, Université des Sciences et Technologies de Lille, Lille, France, 1995. [cited at p. 100]
- [58] S. Crozat. *Éléments pour la conception industrialisée des supports pédagogiques numériques*. PhD thesis, Université de technologie de Compiègne, Compiègne, France, 22 mars 2002. [cited at p. 21, 22]
- [59] K. Czarnecki and S. Helsen. Classification of model transformation approaches. In *OOPSLA'03 Workshop on the Generative Techniques in the Context Of Model-Driven Architecture*, Anaheim, California, USA, 2003. [cited at p. 124]
- [60] K. Czarnecki and S. Helsen. Classification des approches de transformation de modèles. Technical report, centre universitaire d'informatique de Genève, Modélisation et vérification de logiciel, 2006. [cited at p. 124]
- [61] N. Dabbagh, D. Jonassen, and H. Yueh. Assessing a problem-based learning approach to an introductory instructional design course : A case study. *Performance Improvement Quarterly*, 3(3) :60–83, 2000. [cited at p. 36]
- [62] B. David, F. Tarpin-Bernard, and C. Vial. Ergonomie du travail coopératif en conception. In *Actes de la conférence ERGO-IA*, 1996. [cited at p. 95]
- [63] C. Desmoulins and M. Grandbastien. Des ontologies pour indexer des documents techniques pour la formation professionnelle. In *Ingénierie des Connaissances IC'2000*, pages pp. 49–57, Toulouse, France, 10-12 mai 2000. [cited at p. 45]
- [64] E. DeVries. Les logiciels d'apprentissage : panoplie ou éventail? *Revue Française de pédagogie*, 137 :105–116, 2001. [cited at p. 103]
- [65] J. Dewey. *L'école et l'enfant*. Delachaux et Niestlé, 2ème édition, 1922. [cited at p. 30]
- [66] J. Dewey. Democracy and education. an introduction to the philosophy of education. *New York : Free Press*, 1966. [cited at p. 27]
- [67] P. Dillenbourg. What do you mean by "collaborative learning". In *Collaborative Learning : Cognitive and Computational Approaches*, pages 1–19, Amsterdam, 1999. [cited at p. 29]
- [68] T. L. A. Dinh, O. Gerbé, , and H. Sahraoui. Un méta-métamodèle pour la gestion de modèles. In *IDM 06 Actes des 2èmes Journées sur l'Ingénierie Dirigée par les Modèles*, Lille, France, 2006. [cited at p. 116]
- [69] BOEN du ministère de l'éducation nationale. *Programmes Pédagogiques Nationaux des I.U.T.* <http://www.education.gouv.fr/cid178/programmes-pedagogiquesnationaux.html>, 15 janvier 2007. [cited at p. 8, 81]

- [70] G. Durand. *La scénarisation de l'évaluation des activités pédagogiques utilisant les Environnements Informatiques d'Apprentissage Humain*. PhD thesis, Université de Savoie, Savoie, France, 24 octobre 2006. [cited at p. 45, 46, 60]
- [71] E. Ecoutin. Fiche pratique no. 1 : les utilisations d'une plate-forme. Technical report, 2001. [cited at p. 90]
- [72] Educnet. Glossaire e-formation. Technical report, <http://www.educnet.education.fr/superieur/glossaire.htm>, 2008. [cited at p. 89, 91]
- [73] H. El-Kechai. Approche dirigée par les modèles pour accompagner la production de modèles d'expression de scénarios pédagogiques. pages P. 515–526, Lausanne, 2007. *Environnements Informatiques pour l'Apprentissage Humain*. [cited at p. 44, 47, 48, 50, 64, 65, 71]
- [74] C. Ellis, S.J. Gibbs, and G. Rein. Groupware : some issues and experiences. *Communication of ACM*, 34(1), 1991. [cited at p. 96, 97]
- [75] J.M. Adam et al. Contributions de l'action spécifique "conception d'une plate-forme pour la recherche en eia". *Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation, sticf*, 12, 2005. [cited at p. 12, 88]
- [76] R. Faerber. In Larose F Karsenti T. (Ed.), *La place des TICE en formation initiale et continue à l'enseignement : bilan et perspectives*, chapter Le groupe d'apprentissage en formation à distance : ses caractéristiques dans un environnement virtuel, pages 99–128. Sherbrooke : Editions du crp, université de sherbrooke edition, 2002. [cited at p. 95, 100, 103]
- [77] R. Faerber. Caractérisation des situations d'apprentissage en groupe. *Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation STICF*, vol. 11, 2004. [cited at p. 26, 32, 34, 75]
- [78] M. Favier. *Le travail en groupe à l'âge des réseaux*. 1998. [cited at p. 97]
- [79] J. M. Favre. Concept fondamentaux de l'idm, tutoriel. In *IDM 06 Actes des 2èmes Journées sur l'Ingénierie Dirigée par les Modèles*, Lille, France, 2006. [cited at p. 115]
- [80] J.M. Favre. Towards a basic theory to model driven engineering. In *3rd Workshop in Software Model Engineering, WiSME*, 2004. [cited at p. 119]
- [81] E. Fernandez and D. M. Williamson. Using project-based learning to teach object oriented application development. In *CITC4'03*, Lafayette, Indiana, USA., October 16-18 2003. [cited at p. 72, 75]

- [82] C. Ferraris, A. Lejeune, L. Vignollet, and J.P. David. Modélisation de scénarios d'apprentissage collaboratifs pour la classe. pages P. 285–296, Montpellier, France, 25-27 mai 2005. EIAH 2005. [cited at p. 59, 68, 115]
- [83] A. J. Fougères, P. Canalda, and V. Ospina. Assistance à la gestion et au suivi de projets d'étudiants dans un environnement d'apprentissage coopératif. *e-TI - la revue électronique des technologies d'information*, (Numéro 3, 5), 2007. [cited at p. 72, 75, 79]
- [84] A. J. Fougères and V. Ospina. Gestion et suivi de projets d'étudiants. vers un système de médiation. In *Actes du colloque TICE 04*, Compiègne, France, 2004. [cited at p. 72, 75, 79]
- [85] Communauté française de Belgique. Enseignement.be. Technical report, <http://www.enseignement.be>, 2008. [cited at p. 23]
- [86] F. K. Frantz. A taxonomy of model abstraction techniques. In *Winter Simulation Conference*, pages 1413–1420, 1995. [cited at p. 67]
- [87] S. George. *Apprentissage collectif à distance. SPLACH : un environnement informatique support d'une pédagogie de projet*. PhD thesis, Thèse de doctorat, Université du Maine, 2001. [cited at p. 8, 12, 26, 27, 28, 29, 31, 32, 33, 35, 38, 55, 72, 75, 77, 80, 88, 93, 94, 96, 99, 100, 101, 102]
- [88] A. A. Gokhale. Collaborative learning enhances critical thinking. *Journal of Technology Education*, vol. 7(1), 1995. [cited at p. 29]
- [89] P. Gordon, A. Rogers, and M. Comfort. A taste of problem-based learning increases achievement of urban minority middle-school students. *Educational Horizons*, 79(4) :171–175, 2001. [cited at p. 36, 38]
- [90] P. Gounon and P. Leroux. Ediscène, un editeur de scénarios d'apprentissage et d'encadrement. TICE'2006, 2006. [cited at p. 11]
- [91] P. Gounon, P. Leroux, and X. Dubourg. Décrire l'accompagnement des apprenants. proposition d'une extension du langage de modélisation. pages P. 261–272, Montpellier, France, 2005. Environnements Informatiques pour l'Apprentissage Humain. [cited at p. 58]
- [92] P. Gounon, P. Leroux, and X. Dubourg. Describing learner support : An adaptation of ims-ld educational modelling language. pages 807–809, Amsterdam (Pays-Bas), 18-22 juillet 2005. AIED'05, Supporting Learning through Intelligent and Socially Informed Technology. [cited at p. 11]
- [93] J. Greenfield and K. Short. Software factories : assembling applications with patterns, models, frameworks and tools. In *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, 2003. [cited at p. 115]

- [94] Object Management Group. Technical guide to model driven architecture : The mda guide v1.0.1. Technical report, <http://www.omg.org/docs/omg/03-06-01.pdf>, 2003. [cited at p. 13, 68, 113, 114, 115, 120, 122, 123, 124]
- [95] Object Management Group. *UML 2.0 OCL Specification*. Object Management Group, Inc., final adopted specification. [online] - available : <http://www.omg.org/docs/ptc/03-10-14.pdf> edition, October 2003. [cited at p. 126, 127, 129]
- [96] Standish Group. Chaos report. Technical report, <http://www.standishgroup.com>, 2004. [cited at p. 12, 88]
- [97] ATLAS group LINA and INRIA Nantes. Atl : Atlas transformation language - atl installation guide - version 0.1. Technical report, INRIA University of Nantes, November 2005. [cited at p. 129]
- [98] ATLAS group LINA and INRIA Nantes. Atlas transformation language - atl user manual - version 0.7. Technical report, INRIA University of Nantes, February 2006. [cited at p. 125, 126, 127, 136]
- [99] ATLAS group LINA and NRIA Nantes. *KM3 : Kernel MetaMetaModel Manual - version 0.3*. INRIA University of Nantes, August 2005. [cited at p. 131]
- [100] J. Grudin. Cscw : the convergence of two development contexts. In *Proceedings of the SIGCHI conference on Human factors in computing systems : Reaching through technology*, pages pp. 91–97, New Orleans, Louisiana, United States, 1991. ACM. [cited at p. 96]
- [101] J. Grudin and S.E. Poltrock. Computer-supported cooperative work and groupware. *advanced in Computers, Orlando : Academic Press*, Vol. 45 :pp. 269–320, 1997. [cited at p. 95, 96, 97, 98]
- [102] E. Heeren. *Technology support for collaborative distance learning*. PhD thesis, University of Twente, Netherlands, 1996. [cited at p. 101]
- [103] D. Helic, H. Krottmaier, H. Maurer, and N. Scerbakov. Implementing project-based learning in wbt systems. Technical report, coronet.iicm.edu/denis/pubs/elearn2003.pdf, 2003. [cited at p. 37, 72, 74]
- [104] D. Helic, H. Krottmaier, H. Maurer, and N. Scerbakov. Enabling project-based learning in wbt systems. Technical report, coronet.iicm.edu/denis/pubs/ijel2005.pdf, 2005. [cited at p. 37, 72, 74]
- [105] F. Henri and K. Lundgren-Cayrol. Apprentissage collaboratif et nouvelles technologies. Technical report, LICEF, bureau des technologies d'apprentissage, décembre 1998. [cited at p. 29, 73]

- [106] F. Henri and K. Lundgren-Cayrol. Apprentissage collaboratif à distance. pour comprendre et concevoir les environnements d'apprentissage virtuels. *Sainte-Foy, Québec : Presses Universitaires du Québec*, page 184, 2001. [cited at p. 29, 30, 31, 73]
- [107] D. H. Hernandez-Leo, J.I. Asensio-Pérez, Y.A. Dimitriadis, M. L. Bote-Lorenzo, I. M. Jorrin-Abellan, and E. D. Villasclaras-Fernandez. Reusing ims-ld formalized best practices. *Collaborative Learning Structuring Advanced Technology for Learning*, 2(3) :P. 223–232, 2005. [cited at p. 58]
- [108] A. Mbala Hikolo. *Analyse, conception, spécification et développement d'un système multi-agents pour le soutien des activités en formation à distance*. PhD thesis, Université de Franche-Comté, 2003. [cited at p. 32]
- [109] F. Hoogstoel. *Une approche organisationnelle du travail coopératif assisté par ordinateur. Application au projet Co-Learn*. PhD thesis, Université des sciences et technologies de Lille, Lille, France, 1995. [cited at p. 94, 99]
- [110] M. Hubert. Apprendre en projets. la pédagogie du projet-élèves. Technical report, Chronique sociale, Lyon, France, 1999. [cited at p. 27, 74]
- [111] IBM. *LearningSpace homepage*. [http ://www.lotus.com/learningspace](http://www.lotus.com/learningspace). [cited at p. 93]
- [112] IMS-LD. Ims learning design best practice and implementation. Technical report, IMS Global Learning Consortium - [http ://www.imsglobal.org/learningdesign/index.html](http://www.imsglobal.org/learningdesign/index.html), Février 2003. [cited at p. 11, 51]
- [113] IMS-LD. Ims learning design information model, ims. Technical report, IMS Global Learning Consortium - [http ://www.imsglobal.org/learningdesign/index.html](http://www.imsglobal.org/learningdesign/index.html), Février 2003. [cited at p. 11, 47, 50, 51, 52, 53, 75, 205]
- [114] Norme IMS-LD. Ims learning design xml binding. Technical report, [http ://www.imsglobal.org/learningdesign/index.html](http://www.imsglobal.org/learningdesign/index.html), February 2003. [cited at p. 11]
- [115] D. Johnson, R. Johnson, and K. Smith. *Active learning : Cooperation in the college classroom*. Edina, Mn : Interaction Book Company, 1991. [cited at p. 31, 99]
- [116] R.T. Johnson and D.W. Johnson. *Cooperative Learning : Two heads learn better than one*, volume 18. Context : a Quarterly of Humane Sustainable Culture, 1988. [cited at p. 30]
- [117] C. Joiron. *Une contribution aux systèmes supports de Formation Médicale Continue à distance et d'apprentissage entre pairs : conception et expérimentation du forum DIACOM (Discussion Interactives à bAse de Cas pour*

- le fOrmation Médicale*. PhD thesis, Université de Picardie Jules Verne, 19 décembre 2002. [cited at p. 8]
- [118] B. F. Jones, C. M. Rasmussen, and M. C. Moffitt. Real-life problem solving. : A collaborative approach to interdisciplinary learning. Washington, 1997. DC : American Psychological Association. [cited at p. 36]
- [119] H. El Kechai and C. Choquet. Reusing pedagogical scenarios at a knowledge level : a model driven approach. pages 670–674, Nigata (Japan), 18-20 Juillet 2007. The 7th IEEE International Conference on Advanced Learning Technologies (ICALT'2007). [cited at p. 50, 64, 65, 71]
- [120] W. H. Kilpatrick. *The Project Method*, volume 19 (2), pages 319–335. Teachers College Record, 1918. [cited at p. 27, 74]
- [121] A. Kleppe, S. Warmer, and W. Bast. Mda explained. the model driven architecture : Practice and promise. Addison-Wesley, April 2003. [cited at p. 119]
- [122] R. Koper. Current research in learning design. *Educational Technology Society*, 9(1) :13–22, 2006. [cited at p. 11, 49]
- [123] R. Koper and C. Tattersall. Preface to learning design : A handbook on modelling and delivering networked education and training. *Journal of Interactive Media in Education*, 1(18), 2005. [cited at p. 45]
- [124] V-S. Kumar. computer-supported collaborative learning : Issues for research. In *Eight Annual Graduate Symposium on Computer Science*, University of Saskatchewan, 1996. [cited at p. 101]
- [125] La Sorbonne. *L'approche psycho-culturelle de l'éducation*, Paris, France, 1998. 4ème biennale de l'éducation et de la formation. [cited at p. 33]
- [126] P. Laforcade. *Méta-modélisation UML pour la conception et la mise en oeuvre de situations-problèmes coopératives*. PhD thesis, Université de Pau et des Pays de l'Adour, 15 décembre 2004. [cited at p. 7, 13, 47, 49, 50, 52, 58, 61, 62, 63, 64, 90, 91, 92, 115, 116, 158]
- [127] P. Laforcade, V. Barré, and B. Zeindagui. Scénarisation pédagogique et ingénierie dirigé par les modèles. cadre d'étude pour la définition de langages et environnements-outils de scénarisation pédagogique spécifiques à des domaines. In *EIAH'07*, Lausanne, 2007. [cited at p. 123, 133]
- [128] P. Laforcade, T. Nodenot, and C. Sallaberry. Un langage de modélisation pédagogique basé sur uml. *Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation*, Volume 12, 2005. [cited at p. 61, 63, 64]
- [129] B. Landon. Hard choices for individual situations : Selecting a course management system. In *Acte de la conférence ED-MEDIA*, Denver, Colorado, 2002. [cited at p. 104]

- [130] M. Lebrun. Des méthodes actives pour une utilisation effective des technologies. *Techno et Pédagogie : de la méthode!* - www.ipm.ucl.ac.be/marcel/METHODES.PDF, 19 septembre 2001. [cited at p. 8, 24, 25, 32, 34, 35, 72, 74, 75, 77, 79, 81]
- [131] L. Legrand. Pédagogie de projet, projet technique. *Bulletin de l'Association des Enseignants Activités Technologiques (AEAT)*, vol. 65 (3ème trimestre), 1986. [cited at p. 33, 35, 38]
- [132] T. Leinonen, G. Kligyte, T. Toikkanen, J. Pietarila, and P. Dean. *Learning with Collaborative Software - A guide to Fle3*. Helsinki : Taideteollinen korkeakoulu, (no. isbn : 951-558-127-3.) edition, 2003. [cited at p. 103]
- [133] D.H. Hernandez Leo, J.I. Asensio Pérez, and Y.A. Dimitriadis. Imslearning design support for the formalization of collaborative learning patterns. pages 350 – 354, Joensuu (Finlande), 2004. the IEEE International Conference on Advanced Learning Technologies (ICALT'04). [cited at p. 11, 57, 58]
- [134] P. Leroux. *Machines partenaires des apprenants et des enseignants. Etude dans le cadre d'environnements supports de projets pédagogiques*. PhD thesis, Mémoire de HDR, Université du Maine, 20 décembre 2002. [cited at p. 26, 32, 33, 35, 38, 55, 72, 73, 74, 75, 77]
- [135] S. K. Levan. *Travail collaboratif sur Internet. Concepts, méthodes et pratiques des plateaux projet*. Vuibert, 2004. [cited at p. 72]
- [136] M. Lewkowicz. *Conception de collecticiels pour la gestion coopérative des connaissances*. PhD thesis, Université Paris VI, 14 décembre 2000. [cited at p. 102]
- [137] M. Lewkowicz and M. Zacklad. Analyse cognitive des usages de memo-net, collecticiel pour une gestion coopérative des connaissances. In *In Teulier, R., Charlet, J., Tchounikine, P., Ingénierie des Connaissances*, pages pp. 345–366. L'Harmattan, 2005. [cited at p. 102]
- [138] M. Linard. *Des machines et des hommes*. 1996. [cited at p. 96]
- [139] LOM. Final draft standard for learning object metadata v0.1. Technical Report P 1484.12.1, Document IEEE, 2002. [cited at p. 48]
- [140] J. Longchamp. *Le travail coopératif et ses technologies*. Hermes-Lavoisier publications, 2003. [cited at p. 105]
- [141] C. Marquesuzaà, T. Nodenot, and C. Labanino. La mise en oeuvre d'une pédagogie par projets efficace dans les formations professionnalisantes : quels enjeux pour les tice ? La Havane, Cuba, 2002. 3ème Rencontre Europe Amérique Latine sur la Formation Technologique et Professionnelle. [cited at p. 26]

- [142] C. Martel. *La modélisation des activités conjointes. Rôles, places et positions des participants*. PhD thesis, Université de Savoie, Savoie, France, 1998. [cited at p. 59]
- [143] C. Martel, L. Vignollet, and C. Ferraris. Modelling the case study with ldl and implementing it with ldi. pages 1158–1159. Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06), 2006. [cited at p. 11, 12, 50, 57, 58, 59, 158]
- [144] C. Martel, L. Vignollet, C. Ferraris, and G. Durand. Ldl : a language to model collaborative learning activities. In *In EDMEDIA 2006*. World Conference on Educational Multimedia, Hypermedia and Telecommunications, June 2006. [cited at p. 46, 59]
- [145] S. Matsuura. An evaluation method of project based learning on software development experiment. Houston, Texas, USA, March 1-5 2006. SIGCSE'06. [cited at p. 35, 74]
- [146] P. McAndrew. Can generic models be reused and shared. Technical report, LTNS Generic Center Self Session, 2003. [cited at p. 43]
- [147] S. Meier, R. Hovde, and R. Meier. Problem solving : Teachers' perceptions, content area models, and interdisciplinary connections. *School Science and Mathematics*, pages 230–237, 1996. [cited at p. 36]
- [148] S. M. Miller and K. L. Miller. Using instructional theory to facilitate communication in web-based courses. *Journal of Educational Technology and Society*, ISSN 1436-4522, 2(3), 1999. [cited at p. 103]
- [149] ModX. *Modelling Everything!* <http://noce.univ-lille1.fr/projets/ModX/>, 2007. [cited at p. 12, 84, 116, 132]
- [150] F. Morandi. *Modèles et méthodes en pédagogie*. Nathan, Paris, 2005. [cited at p. 19, 23, 24]
- [151] NetUniversité. *NetUniversité*. <http://www.cepia-hds.utc.fr:8080/CEPIAHV3/web/index-1.jsp>. [cited at p. 11]
- [152] T. Nodenot. *Contribution à l'ingénierie dirigée par les modèles en EIAH : le cas des situations-problèmes coopératives*. PhD thesis, Laboratoire d'Informatique de l'Université de Pau et des Pays de l'Adour (LIUPPA), Pau, France, 30 novembre 2005. [cited at p. 61, 62, 63, 64]
- [153] G. Norman and H. Schmidt. The psychological basis of problem-based learning : A review of the evidence. *Academic Medicine*, 67 :557–565, 1992. [cited at p. 36]
- [154] OMG. *OMG/RFP/QVT MOF 2.0 Query/Views/Transformations RFP*. OMG, October 2002. [cited at p. 125]

- [155] ORAVEP. Etude comparative technique et pédagogique des plates-formes pour la formation ouverte et à distance. Technical report, Novembre 2000. [cited at p. 90]
- [156] L. Oubahssi. *Conception de plates-formes logicielles pour la formation à distance, présentant des propriétés d'adaptabilité à différentes catégories d'utilisateurs et d'interopérabilité avec d'autres environnements logiciels*. PhD thesis, Université René Descartes - Paris V, 01 décembre 2005. [cited at p. 90]
- [157] L. Oubahssi, M. Granbastien, M. Ngomo, and G. Claes. The activity at the center of the global open and distance learning process. In *The 12th International Conference on Artificial Intelligence in Education (AIED 2005)*, pages pp. 386–394, Amsterdam, 18-22 July 2005. [cited at p. 104]
- [158] R. Palloff and K. Pratt. *Building Learning Communities in Cyberspace*. Josey-Bass Publishers, San Francisco, 1999. [cited at p. 22]
- [159] T. Panitz. *A Definition of Collaborative vs Cooperative Learning*. PhD thesis, <http://www.kdassem.dk/didaktik/l3c-1.htm>, 1997. [cited at p. 29, 30]
- [160] G. Paquette. *L'ingénierie pédagogique pour construire l'apprentissage en réseau*. Presses de l'Université du Québec, 2002. [cited at p. 44]
- [161] G. Paquette, F. Grevier, and G. Aubin. Méthode d'ingénierie d'un système d'apprentissage (misa). initiation à la formation/conseil en milieu de travail. Technical report, Téléuniversité, Sainte-Foy, 1998. [cited at p. 9, 10, 45]
- [162] P. Parrend. *MDA : Un Tutoriel*, Mars 2005. [cited at p. 124]
- [163] P. Parrend. *Introduction à MDA : Principe*. <http://pparrend.developpez.com/tutoriel/mda-intro/>, 2006. [cited at p. 120, 124]
- [164] J. P. Pernin. A propos d'objet d'apprentissage, d'activité, de ressource. Lyon, France, Mars 2004. Séminaire DOCSI, ENSSIB. [cited at p. 10, 46]
- [165] J-P. Pernin, T. Nodenot, and M. Grandbastien. Modèles, architectures logicielles et normes pour le développement et l'intégration des eiah. Technical report, 3ème Ecole Thématique EIAH, Juillet 2005. [cited at p. 10, 49]
- [166] J.P. Pernin and A. Lejeune. Dispositifs d'apprentissage instrumentés par les technologies : vers une ingénierie centrée sur les scénarios. pages 42–45, France, 2004. TICE'04. [cited at p. 9, 10, 43, 45]
- [167] P. Perrenoud. Apprendre à l'école à travers des projets : pourquoi ? comment ? <http://p.biebandt.free.fr/PERRENOUD/Perrenoudapprendreavec.p.htm>, 1999. [cited at p. 8, 26, 32, 34, 35]

- [168] J. Piaget. *La psychologie de l'intelligence*. 1967. [cited at p. 22, 23]
- [169] N. Piccinini and G. Scollo. Cooperative project-based learning in a web-based software engineering course. *Educational Technology Society*, 9 (4) :P. 54–62, 2006. [cited at p. 72, 74, 75, 77]
- [170] J. Polman. Bootstrapping a community of practice : Learning science by doing projects in a high school classroom. In Eds., editor, *D. C. Edelson, E. A. Domeshek*, pages 474–479, Charlottesville, VA, 1996. International Conference on the Learning Sciences. [cited at p. 33]
- [171] C. Prigent and al. Le projet pluridisciplinaire à caractère professionnel. Technical report, Ministère de l'Éducation nationale - Direction de l'enseignement scolaire, avril 2002. [cited at p. 39]
- [172] I. Rak, C. Teixido, J. Favier, and M. Cazenaud. *La démarche de projet industriel*. Paris, 1990. [cited at p. 33]
- [173] S. Ratté and J. Caron. Le web pour enseigner par projets et favoriser la collaboration. *International Journal of Technologies in Higher Education*, 1(2), 2004. [cited at p. 26]
- [174] J. Ravitz, J. Mergendoller, T. Markham, C. Thorsen, K. Rice, C. Snelson, and S. Reberry. Online professional development for project based learning : Pathways to systematic improvement. In *Association for Educational Communications and Technology*, Chicago, IL, 21 october 2004. [cited at p. 27, 28]
- [175] A. Rawlings, P. van Rosmalen, R. Koper, M. Rodriguez-Artacho, and P. Lefrere. Survey of educational modelling languages (emls). Technical report, September 19st 2002. [cited at p. 49]
- [176] Reload. *Reusable eLearning Object Authoring Delivery*. <http://www.reload.ac.uk/>, 2006. [cited at p. 11]
- [177] M.D. Rossetti and H.B. Nembhard. Using cooperative learning to activate your simulation classroom. Proceedings of the 1998 Winter Simulation Conference, 1998. [cited at p. 23]
- [178] S. Sage. A natural fit : Problem-based learning and technology standards. *Learning and Leading with Technology*, 28(1) :6–12, 2000. [cited at p. 36]
- [179] K. Saikali and B. David. Using workflow for coordination in groupware applications. In *Actes de la conférence IHM-HCI*, pages pp. 399–411, Lille, France, 10-14 septembre 2001. [cited at p. 98]
- [180] G. Salomon. What does the design of effective cscl require and how do we study its effects? *SIGCUE outlook, special issue on CSCL*, vol. 21(3) :pp. 62–68, 1992. [cited at p. 99]

- [181] Z. Scherz and S. Polak. An organizer for project-based learning and instruction in computer science. In *Proceedings of the 4th annual SIGCSE/SIGCUE ITiCSE conference on Innovation in computer science education ITiCSE '99*, 1999. [cited at p. 72, 75, 77, 78]
- [182] K. Schmidt. Distributed collective practices : A csw perspective. In *Conference on Distributed Collective Practices*, Paris, 19-22 September 2000. [cited at p. 96]
- [183] D. Schneider, B. Class, C. Frété, F. Giradin, F. Lombard, S. Morand, and P. Synteta. Conception et implémentation de scénarios pédagogiques riches avec des portails communautaires. second colloque de Guéret, 4-6 juin 2003. [cited at p. 28, 45, 72, 75, 79]
- [184] E. Seidewitz. What models mean. *IEEE Software*, 20(5) :26–32, September 2003. [cited at p. 119]
- [185] S. Sendall and W. Kozaczynski. Model transformation : The heart and soul of model-driven software development. *IEEE Software, IEEE Society*, pages 42–45, 2003. [cited at p. 16]
- [186] S. Sendall and W. Kozaczynski. Model transformation : The heart and soul of model-driven software development. *IEEE Computer Society*, 2003. [cited at p. 124, 125]
- [187] B. Skinner. *Pour une science du comportement : Le behaviorsime*. 1979. [cited at p. 21]
- [188] R. Soley and the OMG staff. Model-driven architecture. Technical report, [ftp ://ftp.omg.org/pub/docs/omg/00-11-05.pdf](ftp://ftp.omg.org/pub/docs/omg/00-11-05.pdf), November 2000. [cited at p. 120]
- [189] A.L. Soller. Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education*, 1(12) :40–62, 2001. [cited at p. 23]
- [190] Meta Object Facility (MOF) Specification. [cited at p. 115, 116, 119]
- [191] G. Stahl. Groupware goes to school. In *Proceedings of the 8th International Workshop on Groupware : Design, Implementation and Use*, pages pp. 7–24, 2002. [cited at p. 103]
- [192] Swebok. *Guide to the software Engineering of body of knowledge*. IEEE Computer Society, 2004. [cited at p. 12, 88]
- [193] P. Synteta. Eva_pm : Design and development of a scaffolding environment for students projects. Master's thesis, Université de Genève, 6 juillet 2001. [cited at p. 8, 28, 35, 37, 39, 74]

- [194] P. Synteta. *Project-Based e-Learning : The model and the method, the practice and the portal*. PhD thesis, University of Geneva, Geneva, Switzerland, 2002. [cited at p. 27]
- [195] B. Talon, C. Toffolon, and B. Warin. Accompagner les projets en milieu universitaire. présentation d'une méthodologie d'encadrement de projets collaboratifs assistée par le web. Louvain-la-Neuve, Belgique, 24-26 janvier 2005. 4ème colloque "Questions de pédagogies dans l'enseignement supérieur". [cited at p. 15, 40, 41, 42, 75, 76, 77, 78, 80]
- [196] B. Talon, C. Toffolon, and B. Warin. Projet en milieu universitaire : vers une gestion collaborative assistée par le web. *Revue Internationale de Pédagogie Universitaire*, vol. 2(2) :pp. 28–33, 2005. [cited at p. 15, 26, 40, 41, 42, 75, 76, 77, 78, 80]
- [197] P. Tchounikine. Conception des environnements informatiques d'apprentissage : mieux articuler informatique et sciences humaines et sociales. pages 203–210. *Les technologies en éducation : Perspectives de recherche et questions vives*, 2002. [cited at p. 7, 8]
- [198] P. Tchounikine. Pour une ingénierie des environnements informatiques pour l'apprentissage humain. *Revue Information Interaction Intelligence* (www.revue-i3.org), 2002. [cited at p. 7, 8, 9, 92, 159]
- [199] S. Tchudi and S. Lafer. *The interdisciplinary teacher's handbook : Integrated teaching across the curriculum*. Portsmouth, NH : Boynton/Cook, 1996. [cited at p. 36]
- [200] M.N. Terrasse, M. Savonnet, E. Leclercq, T. Grison, and G. Becker. Points de vue croisés sur les notions de modèle et métamodèle. Paris, 2005. *Ingénierie Dirigée par les Modèles IDM*. [cited at p. 14, 69]
- [201] D. Thi-Lan-anh, G. Olivier, and S. Houari. Gestion de modèles : définitions, besoins et revue de littérature. In *Premières Journées sur l'Ingénierie Dirigée par les Modèles*, pages 1–15, Paris, France, 30 Juin- 1 Juillet 2005. [cited at p. 116, 120]
- [202] J. Thomas. A review of research on project-based learning. Technical report, [http ://www.bobpearlman.org/BestPractices/PBL_Research.pdf](http://www.bobpearlman.org/BestPractices/PBL_Research.pdf), 2000. [cited at p. 27, 32, 35, 37, 72, 79]
- [203] E. Thorndike. *The Psychology of Learning*. Educational Psychology, 1913. [cited at p. 21]
- [204] G. H. Travassos, F. Shull, M. Fredericks, and V. R. Basili. Detecting defects in object oriented designs : Using reading techniques to increase software quality. In *Proc. of OOPSLA*, pages P. 47–56, 1999. [cited at p. 74]

- [205] R. Tretten and P. Zachariou. Learning about project-based learning : Assessment of project-based learning in tinkertech schools. *San Rafael, CA : The Autodesk Foundation*, 1997. [cited at p. 37]
- [206] M. van der Klink. Active learning for adaptive internet alfanet ist-2001-33288. Technical report, Project Deliverable Report, 2002. [cited at p. 102]
- [207] T. Vantroys. *Du langage métier au langage technique, une plate-forme flexible d'exécution de scénarios pédagogiques*. PhD thesis, Université des Sciences et Technologies de Lille, Lille, France, 2003. [cited at p. 47, 98]
- [208] M. Viveta. La réception des productions des apprenants : une phase essentielle dans la conduite de projet. In *INRP, I. N. d. R. Pédagogique, (Eds.)*, pages 117–134, Liège, 1993. Quatrième colloque international sur la robotique pédagogique. [cited at p. 35]
- [209] L.S. Vygotsky. *Mind in society : the development of higher psychological processes*. Cambridge, 1978. [cited at p. 22, 23]
- [210] J.D. Ward and C.L. Lee. A review of problem-based learning. *Journal of Family and Consumer Sciences Education*, Vol. 20(No. 1), Spring/Summer 2002. [cited at p. 32, 36, 37, 38]
- [211] J. Watson. *Psychology as the behaviorist views it*. 1913. [cited at p. 20]
- [212] G. Weiss and P. Dillenbourg. What is 'multi' in multi-agent learning? In *Collaborative-learning Cognitive*. In P. Dillenbourg (Ed), 1999. [cited at p. 32]
- [213] E. Wenger. *Communities of practice : Learning, meaning, and identity*. NewYork : Cambridge University Press, 1998. [cited at p. 103]

Appendices

Annexe A

Annexe A : Description du scénario PPC élaboré par ModX

Cette annexe présente des captures d'écran du scénario PPC élaboré en utilisant l'outil ModX. Cet outil permet d'élaborer à partir d'un méta-modèle (méta-modèle PPC) un modèle de scénario (scénario PPC) étant conforme. Le scénario PPC élaboré dans le chapitre 4 est décrit dans les figures ci-dessous.

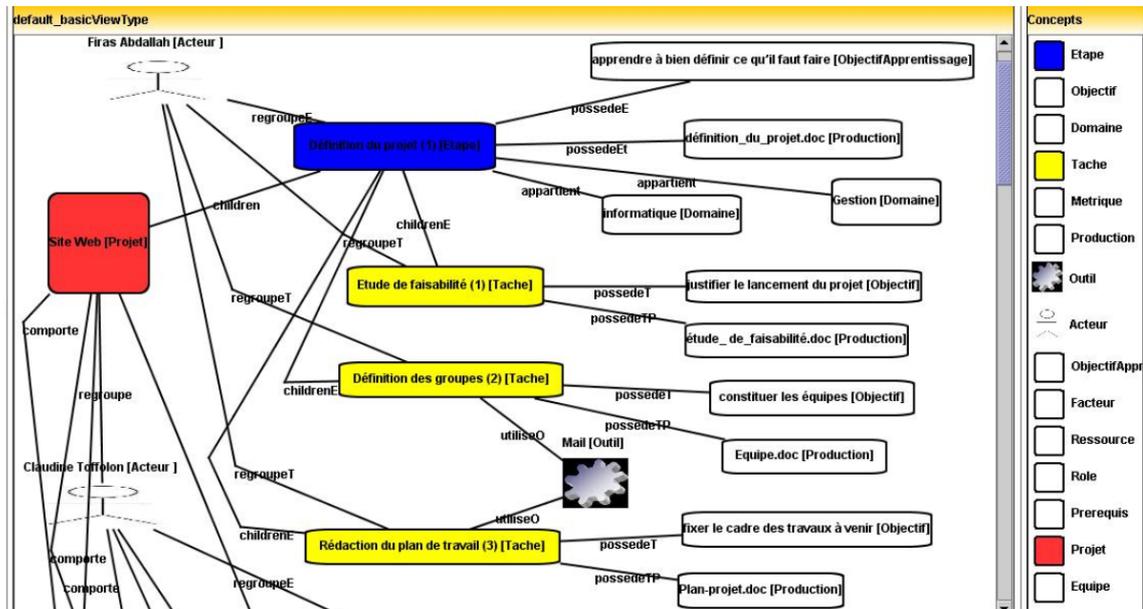


FIG. A.1: Définition du projet, des étapes et des tâches

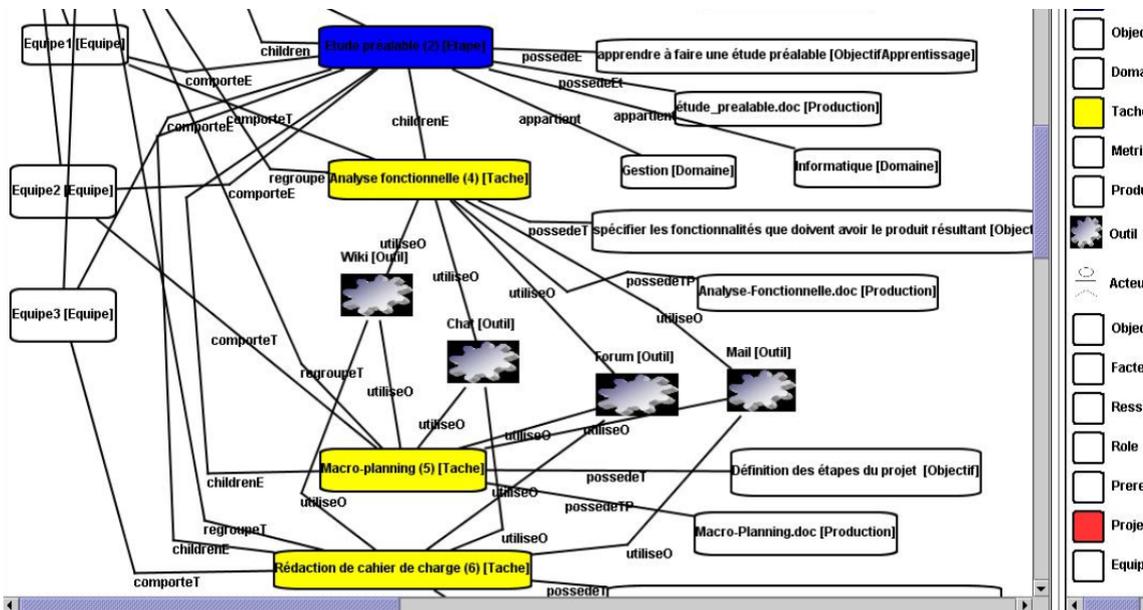


FIG. A.2: Définition du projet, des étapes et des tâches

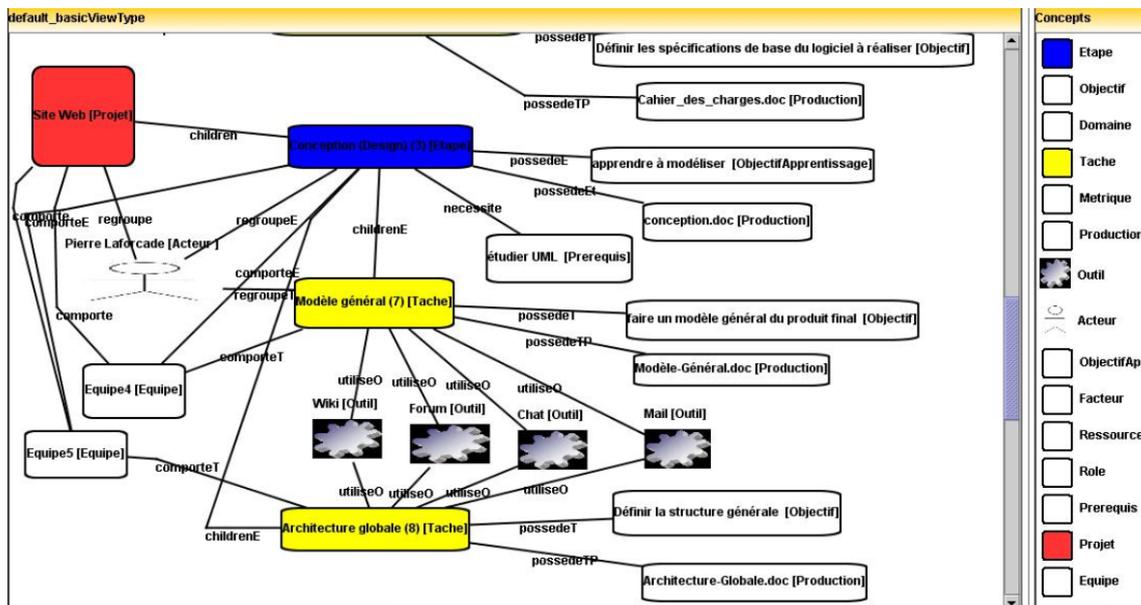


FIG. A.3: Définition du projet, des étapes et des tâches

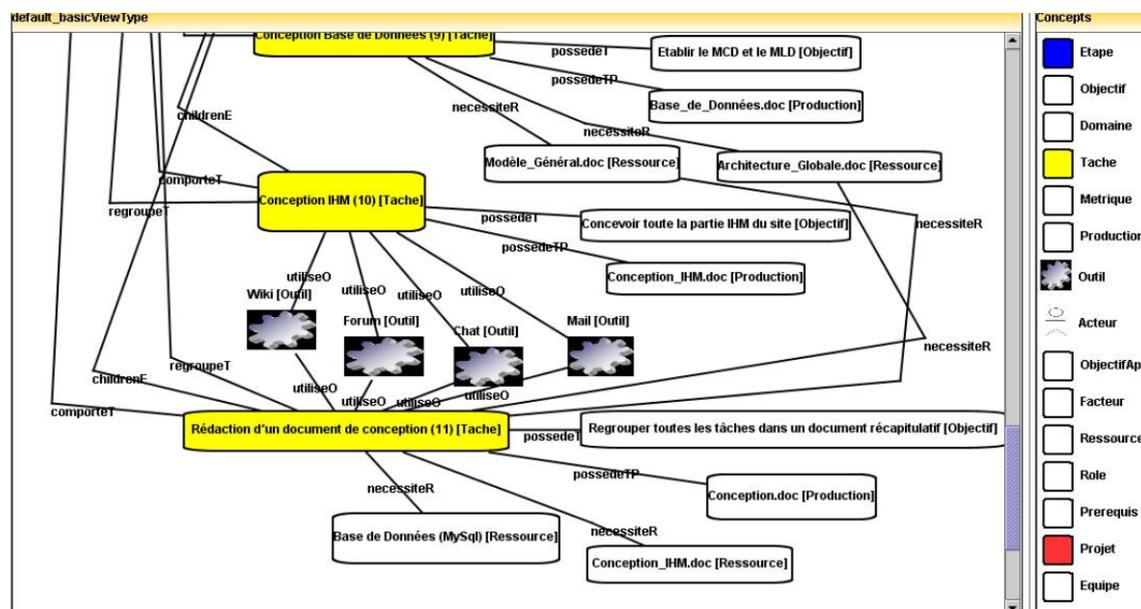


FIG. A.4: Définition du projet, des étapes et des tâches

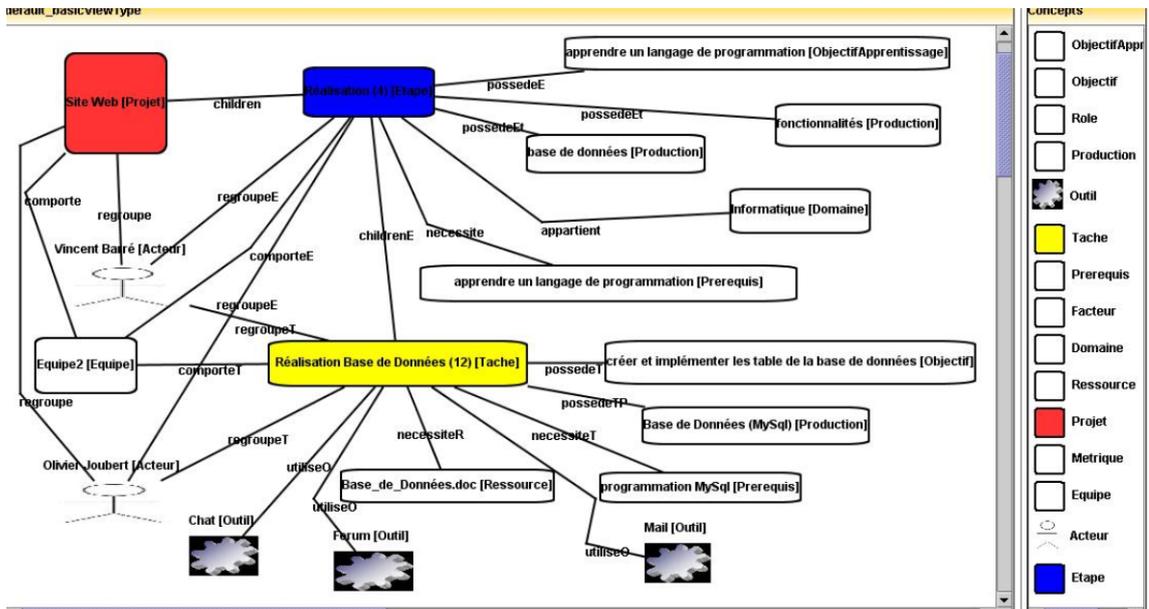


FIG. A.5: Définition du projet, des étapes et des tâches

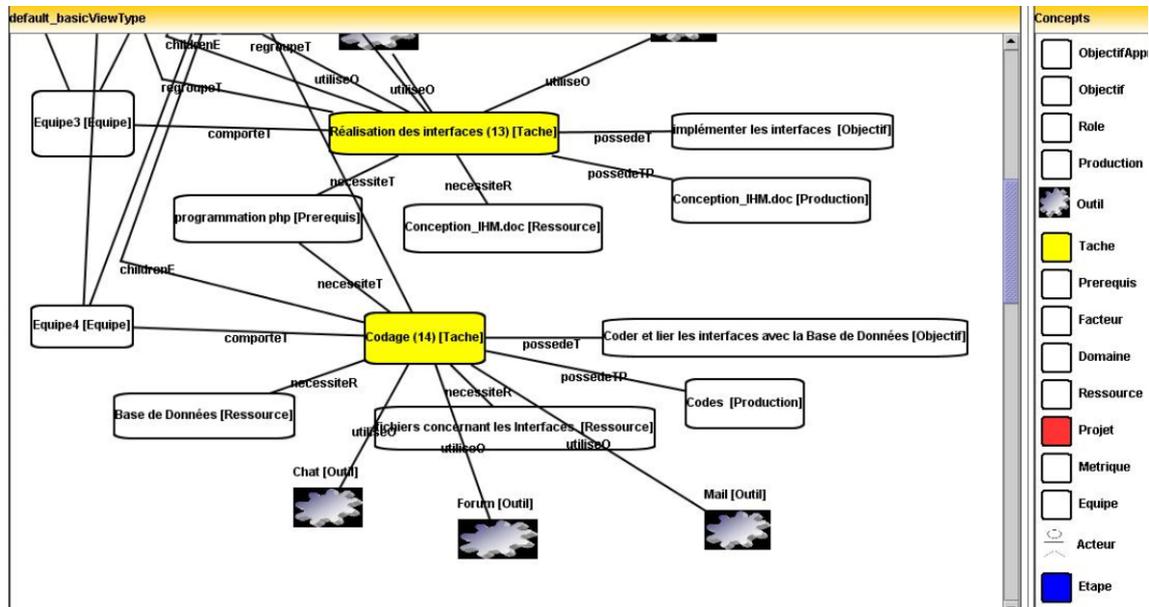


FIG. A.6: Définition du projet, des étapes et des tâches

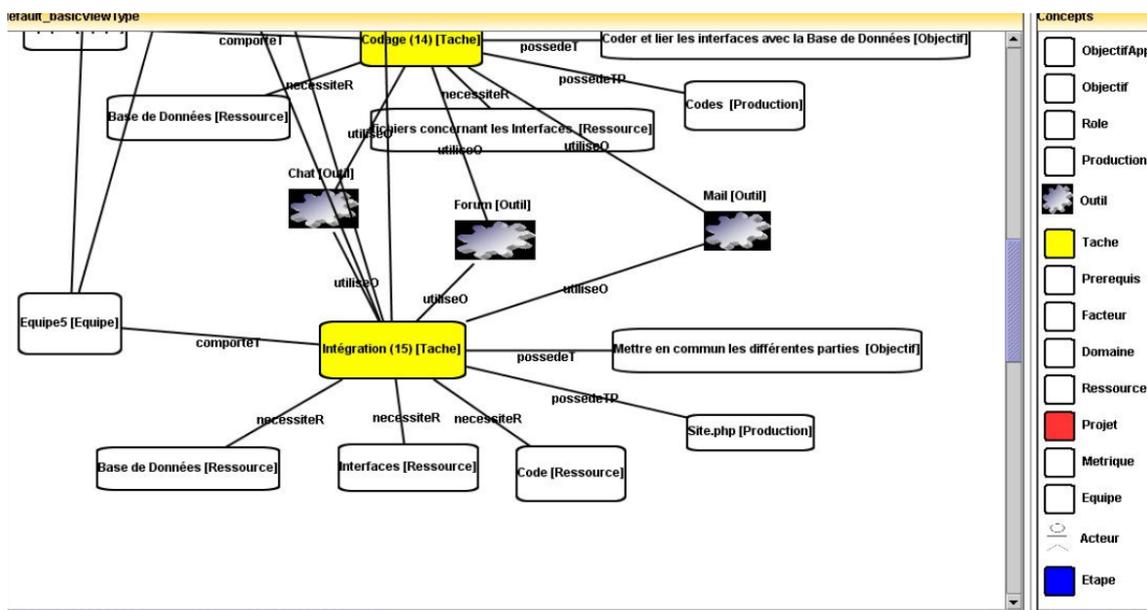


FIG. A.7: Définition du projet, des étapes et des tâches

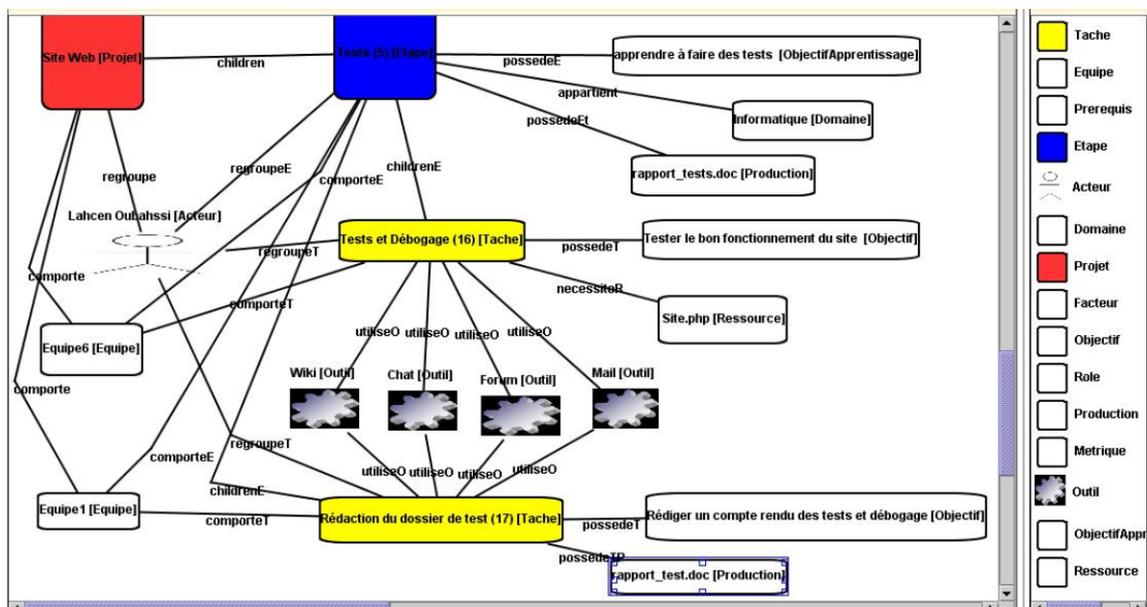


FIG. A.8: Définition du projet, des étapes et des tâches

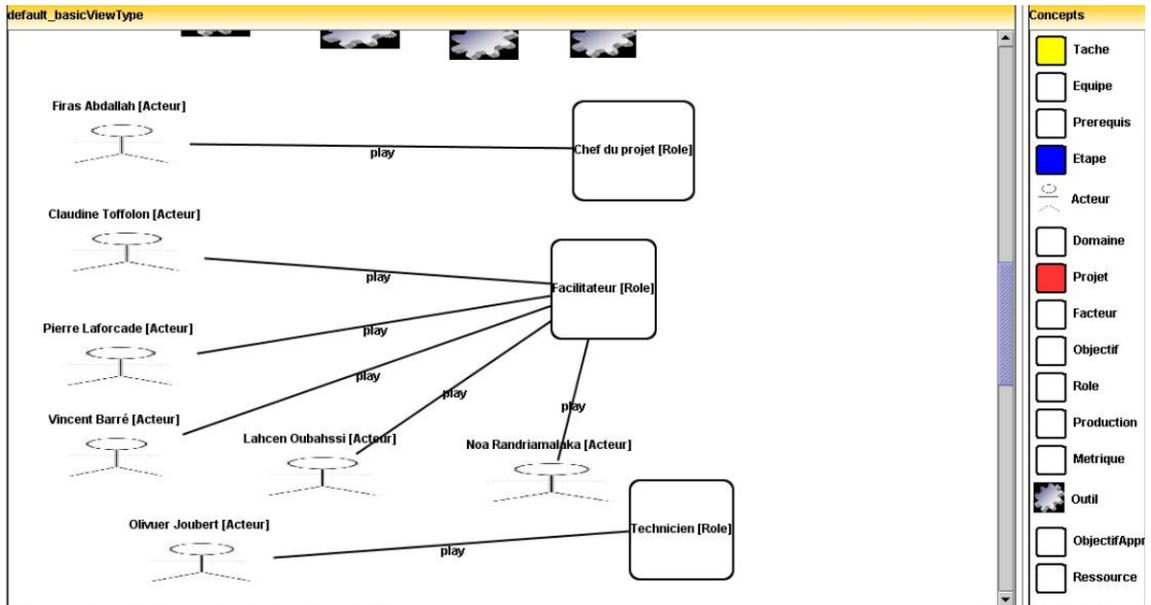


FIG. A.11: Définition des acteurs et de leurs rôles

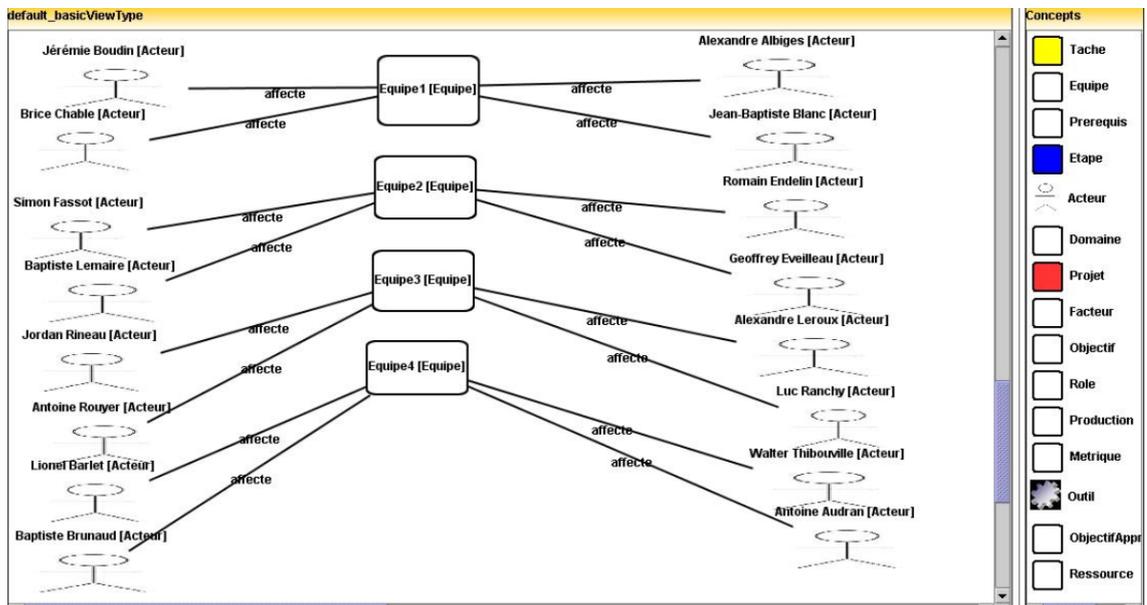


FIG. A.12: Définition des équipes

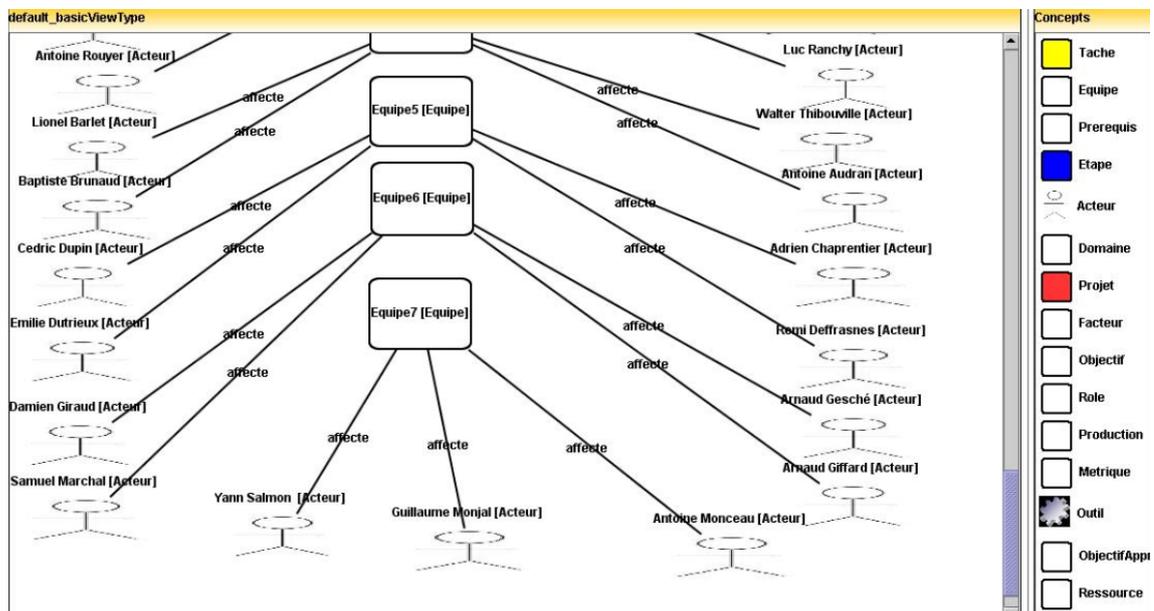


FIG. A.13: Définition des équipes

Annexe B

Annexe B : Implémentation des règles de transformation ATL

Cette annexe présente des extraits de code écrits dans le langage km3 et ATL permettant de créer les règles de transformation depuis le méta-modèle PPC vers le méta-modèle de la plate-forme Moodle.

B.1 Code KM3 de la création du méta-modèle PPC

Le code suivant correspond à la spécification du méta-modèle PPC écrit en langage km3 dans le fichier PPC.km3. Ce code est décrit dans les figures B1, B2, B3, B4, et B5.

B.2 Code KM3 de la création du méta-modèle Moodle

Le code suivant correspond à la spécification du méta-modèle Moodle écrit en langage km3 dans le fichier Moodle.km3. Ce code est décrit dans les figures B6, et B7.

B.3 Code ATL de la création des règles de transformation entre le méta-modèle PPC et le méta-modèle Moodle

Le code suivant correspond à la création des règles de transformation entre le méta-modèle PPC et le méta-modèle Moodle dans le fichier PPC2Moodle.atl. Ce code est décrit dans les figures B8, B9, B10, B11, et B12.

```
package PPC {  
  class Projet {  
    attribute Projet_titre : String;  
    attribute Projet_identificateur : String;  
    attribute Projet_date_debut : String;  
    attribute Projet_date_fin : String;  
    reference ComporteEtape [*] ordered container : Etape oppositeOf PereEtape;  
    reference ProjetRegroupeActeur [1-]* ordered container : Acteur oppositeOf ActeurAffecte;  
    reference ProjetPossedeOA [1-]* container : ObjectifApprentissage oppositeOf concerne;  
    reference ProjetPossedeProduction [1-]* container : Production oppositeOf concerneP;  
    reference ProjetEvaluer [1-]* : Facteur oppositeOf evaluer;  
  }  
  
  class Etape {  
    attribute Etape_titre : String;  
    attribute Etape_identificateur : String;  
    attribute Etape_numero : String;  
    attribute Etape_date_debut : String;  
    attribute Etape_date_fin : String;  
    reference PereEtape [0-1] : Projet oppositeOf ComporteEtape;  
    reference ComporteTache [*] ordered container : Tache oppositeOf PereTache;  
    reference EtapePossedeOA [1-]* : ObjectifApprentissage oppositeOf concerneE;  
    reference EtapeNecessitePrerequis [1-]* container : Prerequis oppositeOf necessaire;  
    reference EtapePossedeProduction [1-]* container : Production oppositeOf concerneEt;  
    reference EtapeAppartientDomaine [1-]* container : Domaine oppositeOf definit;  
    reference EtapeRegroupeActeur [1-]* container : Acteur oppositeOf participeE;  
    reference EtapeEvaluer [1-]* : Facteur oppositeOf evaluerE;  
  }  
}
```

FIG. B.1: Code KM3 de la création du méta-modèle PPC

```

class Tache {
    attribute Tache_titre : String;
    attribute Tache_identificateur : String;
    attribute Tache_numero : String;
    attribute Tache_date_debut : String;
    attribute Tache_date_fin : String;
    attribute Tache_nature : String;
    reference PereTache [1-1] : Etape oppositeOf ComporteTache;
    reference TachePossedeObjectif [1-1] container : Objectif oppositeOf concerneT;
    reference TacheNecessiteRessource [1-1] container : Ressource oppositeOf necessaireR;
    reference TacheEvalue [1-1] : Facteur oppositeOf evaluer;
    reference TachePossedeProduction [1-1] container : Production oppositeOf concerneTP;
    reference TacheNecessitePrerequis [1-1] container : Prerequis oppositeOf necessaireT;
    reference TacheRegroupeActeur [1-1] container : Acteur oppositeOf participeT;
    reference TacheUtiliseOutil [1-1] : Outil oppositeOf utiliseO;
}
class TacheCollective extends Tache {
    reference fils [1-1] : TacheIndividuelle oppositeOf pere;
}
class TacheIndividuelle extends Tache {
    reference pere [1-1] : TacheCollective oppositeOf fils;
}
class Acteur {
    attribute ID_Acteur : String;
    attribute Prenom : String;
    attribute Nom : String;
    attribute Type : String;
    reference ActeurJoueRole [1-1] container : Role oppositeOf played;
    reference ActeurAffecte [1-1] : Projet oppositeOf ProjetRegroupeActeur;
    reference participeE [1-1] : Etape oppositeOf EtapeRegroupeActeur;
    reference participeT [1-1] : Tache oppositeOf TacheRegroupeActeur;
}
class Apprenant extends Acteur {
    reference affecte [1-1] : Equipe oppositeOf EquipeComporteApprenant;
}
class Concepteur extends Acteur {
}
    
```

FIG. B.2: Code KM3 de la création du méta-modèle PPC

```

class Tuteur extends Acteur {
}
class Expert extends Acteur {
}
class Equipe {
attribute Equipe_nom : String;
attribute Equipe_identificateur : String;
reference EquipeComporteApprenant [*] : Apprenant oppositeOf affecte;
}
class Role {
attribute Role_nom : String;
attribute Role_identificateur : String;
reference played [1-]* : Acteur oppositeOf ActeurJoueRole;
}
class Facilitateur extends Role {
}
class Redacteur extends Role {
}
class ChefdeProjet extends Role {
}
class Technicien extends Role {
}
class ObjectifApprentissage {
attribute Description : String;
attribute ObjectifApprentissage_identificateur : String;
reference concerne [1-1] : Projet oppositeOf ProjetPossedeOA;
reference concerneE [1-1] : Etape oppositeOf EtapePossedeOA;
}
class Production {
attribute Production_nom : String;
attribute Fichier_nom : String;
reference concerneP [1-1] : Projet oppositeOf ProjetPossedeProduction;
reference concerneEt [1-1] : Etape oppositeOf EtapePossedeProduction;
reference concerneTP [1-1] : Tache oppositeOf TachePossedeProduction;
}

```

FIG. B.3: Code KM3 de la création du méta-modèle PPC

```
class Facteur {
attribute Description : String;
attribute Facteur_identificateur : String;
reference evalue [1-1] : Projet oppositeOf ProjetEvalue;
reference sedivise [1-*] : Facteur oppositeOf composed;
reference composed [1-1] : Facteur oppositeOf sedivise;
reference FacteurUtiliseMetrique [1-*] : Metrique oppositeOf servir;
reference evalueE [1-1] : Etape oppositeOf EtapeEvalue;
reference evalueT [1-1] : Tache oppositeOf TacheEvalue;
}
class Metrique {
attribute Description : String;
attribute Metrique_identificteur : String;
reference servir [1-1] : Facteur oppositeOf FacteurUtiliseMetrique;
}
class Prerequis {
attribute Description : String;
attribute Prerequis_identificateur : String;
reference necessaire [1-*] : Etape oppositeOf EtapeNecessitePrerequis;
reference necessaireT [1-*] : Tache oppositeOf TacheNecessitePrerequis;
}
class Domaine {
attribute Description : String;
attribute Domaine_identificateur : String;
reference definit [1-1] : Etape oppositeOf EtapeAppartientDomaine;
}
class Objectif {
attribute Description : String;
attribute Objectif_identificateur : String;
reference concerneT [1-1] : Tache oppositeOf TachePossedeObjectif;
}
class Ressource {
attribute Description : String;
attribute Ressource_identificateur : String;
reference necessaireR [1-*] : Tache oppositeOf TacheNecessiteRessource;
}
```

FIG. B.4: Code KM3 de la création du méta-modèle PPC

```
class Outil {
  attribute Description : String;
  attribute Mail_nom : String;
  attribute Forum_nom : String;
  attribute Chat_nom : String;
  attribute Wiki_nom : String;
  reference utilisedO [1-*] : Tache oppositeOf TacheUtiliseOutil;
  attribute Outil_identificateur : String;
  attribute GanttChart_nom : String;
}

package PrimitiveTypes {
  datatype String;
  datatype Integer;
}
```

FIG. B.5: Code KM3 de la création du méta-modèle PPC

```
package Moodle {
  class Cours {
    attribute Nom_complet : String;
    attribute No_identification_du_cours : String;
    attribute Date_de_debut : String;
    attribute Date_de_fin : String;
    attribute Resume : String;
    reference CourseHasCategory [1-*] container : Categorie oppositeOf definit;
    reference UserAffectedCourse [1-*] container : Utilisateur oppositeOf participe;
    reference CourseHasFiles [1-*] container : Fichiers oppositeOf FilesConcerningCourse;
    reference CourseHasResource [1-*] container : Ressource oppositeOf necessaire;
    reference CourseUsesActivity [1-*] : Activite oppositeOf evaluate;
  }
  class Categorie {
    attribute nom : String;
    reference definit [1-1] : Cours oppositeOf CourseHasCategory;
    reference sedivise [1-*] : Categorie oppositeOf composed;
    reference composed [1-1] : Categorie oppositeOf sedivise;
  }
  class Utilisateur {
    attribute Nom : String;
    attribute Nom_utilisateur : String;
    attribute Prenom : String;
    reference affecte [1-1] : Groupe oppositeOf compose;
    reference UserPlaysRole [1-*] container : Role oppositeOf played;
    reference participe [1-*] : Cours oppositeOf UserAffectedCourse;
  }
  class Groupe {
    attribute Groupe_nom : String;
    attribute Groupe_description : String;
    reference compose [*] : Utilisateur oppositeOf affecte;
  }
  class Role {
    attribute Nom : String;
    attribute Description : String;
    reference played [1-*] : Utilisateur oppositeOf UserPlaysRole;
  }
}
```

FIG. B.6: Code KM3 de la création du méta-modèle Moodle

```
class Fichiers {
  attribute Nom_repertoire : String;
  attribute Fichier : String;
  reference FilesConcerningCourse [1-1] : Cours oppositeOf CourseHasFiles;
}
class Ressource {
  attribute Nom : String;
  reference necessaire [1-1] : Cours oppositeOf CourseHasResource;
}
class Activite {
  attribute Description : String;
  attribute Categorie : String;
  attribute Nom_de_la_question : String;
  attribute Introduction : String;
  attribute Nom : String;
  attribute Nom_de_ce_salon : String;
  attribute Texte_introduction : String;
  attribute Nom_du_forum : String;
  attribute Message_nom : String;
  reference evaluer [1-1] : Cours oppositeOf CourseUsesActivity;
}
}
package PrimitiveTypes {
  datatype String;
  datatype Integer;
}
```

FIG. B.7: Code KM3 de la création du méta-modèle Moodle

```

module PPC2Moodle; -- Module Template
create OUT : Moodle from IN : PPC;
helper context PPC!Etape def: x : String =
    self.PereEtape.Projet_titre;
helper context PPC!Etape def: w : String =
    self.PereEtape.Projet_identificateur;
helper context PPC!Tache def: y : String =
    self.PereTache.PereEtape.Projet_titre + '_' + self.PereTache.Etape_titre;
helper context PPC!Tache def: z : String =
    self.PereTache.PereEtape.Projet_identificateur + '_' + self.PereTache.Etape_numero;
helper context PPC!Tache def: naturetache : String =
    self.Tache_nature;
helper context PPC!Projet def: getProjetObjectifApprentissage() : String =
    self.ProjetPossedeOA->collect(e | e.Description) ->
        asSet()->
            iterate (ObjecApp; acc : String = '' |
                acc +
                    if acc = ''
                    then ObjecApp
                    else ' and ' + ObjecApp
                    endif);
helper context PPCL!Etape def: getEtapeObjectifApprentissage() : String =
    self.EtapePossedeOA->collect(e | e.Description) ->
        asSet()->
            iterate (ObjecApp; acc : String = '' |
                acc +
                    if acc = ''
                    then ObjecApp
                    else ' and ' + ObjecApp
                    endif);
helper context PPC!Tache def: getObjectif() : String =
    self.TachePossedeObjectif->collect(e | e.Description) ->
        asSet()->
            iterate (Obj; acc : String = '' |
                acc +
                    if acc = ''
                    then Obj else ' and ' + Obj endif);
    
```

FIG. B.8: Code ATL de la création des règles de transformation entre le méta-modèle PPC et le méta-modèle Moodle

```

rule Projet2Cours {
  from
    p : PPC!Projet
  to
    c : Moodle!Cours (
      Nom_complet <- p.Projet_titre,
      No_identification_du_cours <- p.Projet_identificateur,
      Date_de_debut <- p.Projet_date_debut,
      Date_de_fin <- p.Projet_date_fin,
      Resume <- p.getProjetObjectifApprentissage(),
      UserAffectedCourse <- p.ProjetRegroupeActeur
    )
}
rule Etape2Cours {
  from
    e : PPC!Etape
  to
    c : Moodle!Cours (
      Nom_complet <- e.x + '_' + e.Etape_titre,
      No_identification_du_cours <- e.w + '_' + e.Etape_numero,
      Date_de_debut <- e.Etape_date_debut,
      Date_de_fin <- e.Etape_date_fin,
      Resume <- e.getEtapeObjectifApprentissage(),
      UserAffectedCourse <- e.EtapeRegroupeActeur
    )
}

```

FIG. B.9: Code ATL de la création des règles de transformation entre le méta-modèle PPC et le méta-modèle Moodle

```
rule Tache2Cours {
  from
    t : PPC!Tache
  to
    c : Moodle!Cours (
      Nom_complet <- t.y + '_' + t.Tache_titre,
      No_identification_du_cours <- t.z + '_' + t.Tache_numero,
      Date_de_debut <- t.Tache_date_debut,
      Date_de_fin <- t.Tache_date_fin,
      Resume <- t.getObjectif()
    ),

    d : Moodle!Categorie(
      nom <- t.Tache_nature,
      definit <- c
    )
}

rule Acteur2Utilisateur {
  from
    a : PPC!Acteur
  to
    u : Moodle!Utilisateur (
      Nom <- a.Nom,
      Prenom <- a.Prenom,
      Nom_utilisateur <- a.Type,
      participe <- a.ActorAffected,
      participe <- a.participeE,
      participe <- a.participeT,
      UserPlaysRole <- a.ActeurJoueRole
    )
}
```

FIG. B.10: Code ATL de la création des règles de transformation entre le méta-modèle PPC et le méta-modèle Moodle

```
rule Role2Role {
  from
    l : PPC!Role
  to
    r : Moodle!Role (
      Nom <- l.Role_nom,
      Description <- l.Role_identificateur,
      played <- l.played
    )
}
rule Equipe2Groupe {
  from
    e : PPC!Equipe
  to
    g : Moodle!Groupe (
      Groupe_nom <- e.Equipe_nom,
      Groupe_description <- e.Equipe_identificateur,
      compose <- e.EquipeComporteApprenant
    )
}
rule Production2Fichiers {
  from
    p : PPC!Production
  to
    f : Moodle!Fichiers (
      Nom_repertoire <- p.Production_nom,
      Fichier <- p.Fichier_nom,
      FilesConcerningCourse <- p.concerneP,
      FilesConcerningCourse <- p.concerneEt,
      FilesConcerningCourse <- p.concerneTP
    )
}
```

FIG. B.11: Code ATL de la création des règles de transformation entre le méta-modèle PPC et le méta-modèle Moodle

```
rule Domaine2Ressource {
  from
    d : PPC!Domaine
  to
    r : Moodle!Ressource (
      Nom <- d.Description,
      necessaire <- d.definit)
}
rule Prerequis2Ressource {
  from
    p : PPC!Prerequis
  to
    r : Moodle!Ressource (
      Nom <- p.Description,
      necessaire <- p.necessaire,
      necessaire <- p.necessaireT)
}
rule Ressource2Ressource {
  from
    p : PPC!Ressource
  to
    r : Moodle!Ressource (
      Nom <- p.Description,
      necessaire <- p.necessaireR
    )
}
rule Outil2Activitie {
  from
    o : PBCL!Outil
  to
    a : Moodle!Activitie(
      Description <- o.Description,
      Message_nom <- o.Mail_nom,
      Nom_du_forum <- o.Forum_nom,
      Nom_de_ce_salon <- o.Chat_nom,
      Nom <- o.Wiki_nom,
      evaluate <- o.utilisedO
    )
}
```

FIG. B.12: Code ATL de la création des règles de transformation entre le méta-modèle PPC et le méta-modèle Moodle

Table des figures

1.1	Illustration de processus d'ingénierie d'EIAH	15
1.2	Approche pour proposer le méta-modèle PPC	16
2.1	Situation de la Pédagogie par Projet Collectif	26
2.2	Rôle des apprenants et des enseignants dans les méthodes de la pédagogie traditionnelle	38
2.3	Rôle des apprenants et des enseignants dans la Pédagogie par Projet Collectif	39
3.1	Architecture de la spécification IMS-LD, Niveaux A, B, et C. Source [113] annotée	52
3.2	Définition des rôles dans la PPC en IMS-LD	54
3.3	Définition des rôles dans la PPC en IMS-LD	55
3.4	Définition des tâches de la PPC en IMS-LD	56
3.5	Aspect dynamique	57
3.6	Méta-modèle simplifié LDL	59
3.7	Paquetages du méta-modèle CPM	62
3.8	Séréotypes du profil CPM	63
3.9	Correspondances entre la terminologie d'IMS-LD et celle de CPM	64
4.1	Approche de modélisation proposée	69
4.2	Utilisation du méta-modèle PPC	70
4.3	Méthodologie adoptée pour élaborer le méta-modèle PPC	71
4.4	Méta-modèle général de la PPC	72
4.5	Méta-modèle d'une étape de la PPC	78
4.6	Méta-modèle d'une tâche de la PPC	79
4.7	Scénario PPC à l'IUT de Laval	80
4.8	Méta-modèle PPC représenté par ModX	83
4.9	Représentation d'une partie du scénario PPC	84
4.10	Représentation des équipes du scénario PPC	85

5.1	Trèfle fonctionnel des collecticiels	97
5.2	Les six axes du modèle de fonctionnalités PPC	106
6.1	Les quatre niveaux IDM. Source [22]	118
6.2	Architecture du MDA	121
6.3	Règles de transformation ATL	127
6.4	Les Patterns cibles et sources des règles ATL	128
6.5	Les Patterns sources des règles ATL	128
6.6	Les Patterns cibles des règles ATL	129
6.7	Transformation ATL de PPC vers Moodle	130
6.8	Extensions des fichiers	131
6.9	Création des Méta-modèles PPC et Moodle	132
6.10	Création des règles de transformations dans le fichier "PPC2Moodle.atl"	132
6.11	Création du modèle "scénario_PPC.ecore"	133
6.12	Méta-modèle de la plate-forme Moodle	134
6.13	rule Projet2Course	135
6.14	rule Etape2Course	135
6.15	Code rule Etape2Course	136
6.16	Code Helper	137
6.17	rule Tache2Course	137
6.18	Perte Sémantique	138
6.19	rule Acteur2User et Role2Role	139
6.20	rule Acteur2User et Role2Role	140
6.21	Code Helper	141
6.22	Affecter un ensemble de valeurs d'une classe à un seul attribut d'une autre classe	141
6.23	Affecter un ensemble de valeurs d'une classe à un seul attribut d'une autre classe	142
6.24	Affecter un ensemble de valeurs d'une classe à un seul attribut d'une autre classe	142
6.25	rule Production2Files	143
6.26	rule Ressource2Resource	144
6.27	rule Domaine2Resource	145
6.28	rule Prerequis2Resource	146
6.29	rules : Outil2Activity et Outil2Messages	147
6.30	rule Metrique2Quiz	148
6.31	Scénario PPC et scénario PPC transformé	149
6.32	Architecture proposé	150
6.33	Accueil OTCPPC	152
6.34	Choix de la plate-forme cible	153
6.35	Charger le scénario PPC	154
6.36	Visualiser et transformer le scénario PPC	155

6.37 Scénario PPC transformé	156
A.1 Définition du projet, des étapes et des tâches	184
A.2 Définition du projet, des étapes et des tâches	184
A.3 Définition du projet, des étapes et des tâches	185
A.4 Définition du projet, des étapes et des tâches	185
A.5 Définition du projet, des étapes et des tâches	186
A.6 Définition du projet, des étapes et des tâches	186
A.7 Définition du projet, des étapes et des tâches	187
A.8 Définition du projet, des étapes et des tâches	187
A.9 Définition du projet, des étapes et des tâches	188
A.10 Définition du projet, des étapes et des tâches	188
A.11 Définition des acteurs et de leurs rôles	189
A.12 Définition des équipes	189
A.13 Définition des équipes	190
B.1 Code KM3 de la création du méta-modèle PPC	192
B.2 Code KM3 de la création du méta-modèle PPC	193
B.3 Code KM3 de la création du méta-modèle PPC	194
B.4 Code KM3 de la création du méta-modèle PPC	195
B.5 Code KM3 de la création du méta-modèle PPC	196
B.6 Code KM3 de la création du méta-modèle Moodle	197
B.7 Code KM3 de la création du méta-modèle Moodle	198
B.8 Code ATL de la création des règles de transformation entre le méta- modèle PPC et le méta-modèle Moodle	199
B.9 Code ATL de la création des règles de transformation entre le méta- modèle PPC et le méta-modèle Moodle	200
B.10 Code ATL de la création des règles de transformation entre le méta- modèle PPC et le méta-modèle Moodle	201
B.11 Code ATL de la création des règles de transformation entre le méta- modèle PPC et le méta-modèle Moodle	202
B.12 Code ATL de la création des règles de transformation entre le méta- modèle PPC et le méta-modèle Moodle	203

Liste des tableaux

5.1 Grille d'étude des fonctionnalités de Moodle 109