



HAL
open science

Scénarisation pédagogique pour des EIAH ouverts : une approche dirigée par les modèles et spécifique au domaine métier

El Amine Ouraiba

► To cite this version:

El Amine Ouraiba. Scénarisation pédagogique pour des EIAH ouverts : une approche dirigée par les modèles et spécifique au domaine métier. Autre [cs.OH]. Université du Maine, 2012. Français. NNT : 2012LEMA1020 . tel-01128256

HAL Id: tel-01128256

<https://theses.hal.science/tel-01128256v1>

Submitted on 9 Mar 2015

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

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

Thèse pour obtenir le grade de Docteur de l'Université du Maine

Spécialité : Informatique

Scénarisation pédagogique pour des EIAH ouverts

Une approche dirigée par les modèles et spécifique au domaine métier

par

El Amine OURAIBA

Soutenue le 19 septembre 2012 devant le jury composé de :

Rapporteurs :

M Jean-Charles MARTY

M. Philippe VIDAL

Maître de Conférences HDR, Université de Savoie

Professeur des Universités, Université Paul
Sabatier (Toulouse III)

Examineurs :

M. Mourad Chabane OUSSALAH

Mme Anne LEJEUNE

Professeur des Universités, Université de Nantes

Maître de Conférences, Université de Grenoble

Directeur :

M. Christophe CHOQUET

Professeur des Universités, Université du Maine

Co-Encadrant :

M. Philippe COTTIER

Maître de Conférences, Université du Maine

Année 2012

Numéro d'Ordre :

Remerciements

Je tiens à remercier M. Jean-Charles MARTY, Maître de conférences Habilité à Diriger des Recherches à l'université de Savoie, et M. Philippe VIDAL, Professeur des universités à l'université Paul Sabatier (Toulouse III), de m'avoir fait l'honneur d'être rapporteurs de la thèse.

Je tiens à remercier Mme. Anne LEJEUNE, Maître de conférences à l'université de Grenoble, de m'avoir fait l'honneur d'être examinatrice de la thèse, et M. Mourad Chabane OUSSALAH, Professeur des universités à l'université de Nantes, de m'avoir fait l'honneur d'être examinateur et président de mon jury de thèse.

Je remercie sincèrement M. Christophe CHOQUET, Professeur des universités à l'université du Maine d'avoir dirigé cette thèse. Je le remercie pour sa confiance constante et pour sa relecture, ses retours et ses conseils toujours très constructifs.

Je tiens à remercier également M. Philippe COTTIER, Maître de conférences à l'université du Maine qui a assuré l'encadrement scientifique de cette thèse, pour sa relecture, ses retours et ses conseils toujours constructifs. Je le remercie pour toutes les discussions que nous avons eues, et de m'avoir apporté sa vision SHS de l'ingénierie des EIAH.

J'adresse également tous mes remerciements à tous les membres de l'équipe REDiM sur le site de l'IUT de Laval, ainsi que ceux qui sont sur le site du Mans.

Merci à toutes les personnes qui travaillent à l'IUT de Laval : enseignants du département SRC (Services et Réseaux de Communication) et enseignants du département Informatique pour leurs aides et conseils au niveau des enseignements, personnels IATOSS en particulier les personnels du service informatique pour leurs soutiens techniques, merci également à Nathalie d'avoir toujours pris en charge tous les tracas de mes déplacements.

Merci à Nassim, Faouzi, Karim, Amine, Noureddine et Boubekour pour leur amitié et soutien constants.

Je remercie particulièrement toute ma famille pour leurs encouragements et leur soutien tout au long de ma thèse.

Enfin, je voudrai remercier toute personne qui m'a aidé de près ou de loin.

*Je dédie avec respect ce travail à mes chers parents, à ma famille
et à tous mes amis.*

Amine.

Résumé

Notre travail de recherche s'inscrit dans le cadre du projet REDiM (Réingénierie des EIAH Dirigée par les Modèles), dont un des objectifs est d'intégrer les enseignants dans le processus de conception des scénarios pédagogiques d'un EIAH (Environnement Informatique pour l'Apprentissage Humain).

Nous proposons une approche d'ingénierie et de réingénierie pour rendre un EIAH ouvert à la conception et à l'adaptation de ses scénarios pédagogiques par les enseignants utilisateurs. Nous avons défini un processus de conception basé sur la modélisation de scénarios pédagogiques ouverts (SPO), qui permet l'instrumentation des enseignants pour les aider dans la conception continue (i.e. qui se poursuit dans l'usage) d'une activité d'apprentissage. Nous faisons trois propositions scientifiques :

- un modèle de représentation des SPO, qui permet de les structurer en variantes en fonction des contextes d'exécution. Nous qualifions ce modèle de rationnel puisqu'il s'appuie principalement sur l'approche du Design Rationale que nous avons adaptée à notre problématique.
- un processus itératif et incrémental d'ingénierie et de réingénierie qui guide les enseignants pour concevoir et adapter des SPO conformes au modèle que nous avons défini.
- une méthode dirigée par les modèles et spécifique au domaine métier pour instrumenter le processus d'ingénierie et de réingénierie des SPO d'un EIAH. Cette méthode d'instrumentation, reposant sur l'IDM (Ingénierie Dirigée par les Modèles) et le DSM (*Domain-Specific Modeling*), implique les enseignants utilisateurs de l'EIAH, considérés ici comme des experts du domaine. Elle est structurée en plusieurs phases qui amènent progressivement à définir, de façon spécifique à l'EIAH considéré, un langage d'expression des SPO (ADSGEML - *Adaptive Domain-Specific Graphical Educational Modeling Language*) et un éditeur associé permettant la conception et l'adaptation des SPO dans l'univers métier de l'EIAH.

Afin d'évaluer et de raffiner nos propositions, nous les avons appliquées sur l'EIAH «Hop3x», préalablement conçu au LIUM dans le cadre d'un autre projet pour pratiquer la programmation orientée objet.

Mots clés : Ingénierie des EIAH, conception pédagogique, adaptation, scénario pédagogique ouvert, Design Rationale, IDM, DSM.

Sommaire

| | |
|---|-----------|
| 1 INTRODUCTION GENERALE..... | 13 |
| 1.1 Contexte général et motivation de la recherche..... | 14 |
| 1.2 Questions soulevées par les sciences humaines et sociales autour de la notion d'EIAH ouvert | 17 |
| 1.3 La position de notre travail dans le projet REDiM..... | 20 |
| 1.4 Éléments de la problématique..... | 23 |
| 1.5 Définition des principaux termes utilisés..... | 25 |
| 1.6 Aperçu de nos contributions..... | 26 |
| 1.7 Organisation du document..... | 28 |

I. PREMIERE PARTIE :

POSITIONNEMENT SCIENTIFIQUE ET PROBLEMATIQUE DE RECHERCHE

| | |
|--|-----------|
| 2 POSITIONNEMENT SCIENTIFIQUE..... | 33 |
| 2.1 Adaptation et EIAH..... | 34 |
| 2.1.1 Types d'adaptation possibles dans un EIAH..... | 36 |
| 2.1.2 Limites des STI et SHA | 38 |
| 2.1.3 Conception et adaptation des scénarios pédagogiques..... | 40 |
| 2.1.3.1 Deux approches de conception des scénarios pédagogiques | 42 |
| 2.1.3.1.1 Approche interprétative | 43 |
| 2.1.3.1.2 Approche constructive..... | 44 |
| 2.1.3.2 Nécessité d'adaptation des scénarios pédagogiques | 45 |
| 2.1.3.3 Quand et comment adapter un scénario pédagogique ? | 47 |
| 2.2 Design Rationale..... | 52 |
| 2.2.1 Qu'est-ce que le Design Rationale ? | 52 |
| 2.2.2 Apports du Design Rationale pour la conception..... | 54 |
| 2.2.3 Représentation du Design Rationale | 56 |
| 2.2.3.1 Le modèle QOC (<i>Questions, Options, and Criteria</i>) | 57 |
| 2.3 Travaux existants en lien avec notre objet d'étude | 59 |
| 2.3.1 Approches par langages génériques de modélisation pédagogique..... | 59 |
| 2.3.1.1 Travaux reposants sur la spécification IMS-LD | 59 |
| 2.3.1.1.1 Approche à base d'exceptions en IMS-LD | 60 |
| 2.3.1.1.2 Conception de simples unités d'apprentissage adaptables avec IMS-LD | 61 |
| 2.3.1.1.3 Le projet aLFanet | 62 |
| 2.3.1.1.4 Approche mixant IMS-LD et les systèmes hypermédias adaptatifs..... | 64 |
| 2.3.1.1.5 Extension de la spécification IMS-LD pour permettre l'adaptation et l'intégration des unités d'apprentissage | 66 |

| | |
|--|------------|
| 2.3.1.1.6 Adaptation des unités d'apprentissage en temps réel par l'introduction des petites actions adaptatives | 67 |
| 2.3.1.2 Le langage de modélisation pédagogique LDL..... | 69 |
| 2.3.1.3 Langage de modélisation pédagogique orienté perspectives (POEML) | 71 |
| 2.3.2 Approches par l'ingénierie dirigée par les modèles (IDM) | 72 |
| 1.3.2.1 Présentation de l'IDM..... | 72 |
| 2.3.2.1.1 Principes d'IDM | 72 |
| 2.3.2.1.2 Les niveaux d'abstraction dans IDM | 74 |
| 2.3.2.1.3 Les types de modèles..... | 75 |
| 2.3.2.2 Modélisation spécifique au domaine : DSM | 76 |
| 2.3.2.2.1 Qu'est-ce que le DSM ?..... | 76 |
| 2.3.2.2.1 Comment définir un DSML ?..... | 79 |
| 2.3.2.2.2 L'outillage d'Eclipse Modeling Project | 81 |
| 2.3.2.3 CPM (Cooperative Problem-based learning Metamodel) | 86 |
| 2.3.2.4 Approche à base de Workflow pour permettre la flexibilité des processus d'apprentissage | 87 |
| 2.3.2.5 Le projet Bricoles : une approche dispositif..... | 89 |
| 2.3.2.6 Instrumentation des situations d'apprentissage de Pédagogie par Projet Collectif (PPC)..... | 90 |
| 2.3.2.7. Conception pédagogique dirigée par les modèles (MDLD) | 91 |
| 2.3.2.8 Implication des praticiens dans la définition des langages spécifiques de conception pédagogique | 94 |
| 2.3.2.8.1 Le projet MDEduc | 95 |
| 2.3.2.8.2 Conception collective des scénarios pédagogiques : approche par méta-modélisation située | 96 |
| 2.3.3 Approches liées au Web sémantique..... | 97 |
| 2.3.3.1 Approche à base des documents virtuels adaptatifs..... | 97 |
| 2.3.3.1 Représentation des connaissances par des technologies de Web sémantique | 98 |
| 2.3.4. Analyse et discussion des travaux existants..... | 99 |
| 3 PROBLEMATIQUE ET APPROCHE METHODOLOGIQUE..... | 105 |
| 3.1 Définition de notre problématique de recherche | 106 |
| 3.2 Hypothèses et objectifs de recherche..... | 110 |
| 3.3 Méthodologie de notre recherche..... | 112 |
| II. DEUXIEME PARTIE : | |
| PROPOSITIONS SCIENTIFIQUES | |
| 4 MODELE RATIONNEL DES SPO | 117 |
| 4.1 Introduction | 117 |
| 4.2 Notion d'un scénario pédagogique ouvert (SPO)..... | 118 |
| 4.3 Modèle rationnel des SPO | 123 |
| 4.4 Métamodèle implémentant le modèle rationnel des SPO | 130 |
| 4.5 Conclusion | 133 |

| | | |
|----------|---|------------|
| 5 | PROCESSUS ITERATIF ET INCREMENTAL D'INGENIERIE DES SPO..... | 135 |
| 5.1 | Introduction | 136 |
| 5.2 | Processus itératif et incrémental d'ingénierie des SPO | 137 |
| 5.2.1 | Conception initiale | 138 |
| 5.2.2 | Déploiement..... | 138 |
| 5.2.3 | Exécution | 139 |
| 5.2.4 | Réingénierie | 140 |
| 5.3 | Conclusion..... | 143 |
| 6 | METHODE D'INSTRUMENTATION POUR OUVRIR LES SCENARIOS PEDAGOGIQUES D'UN EIAH..... | 145 |
| 6.1 | Introduction | 146 |
| 6.2 | Vision IDM/DSM sur l'ingénierie des SPO..... | 147 |
| 6.3 | Méthode d'instrumentation dirigée par les modèles et spécifique au domaine métier pour ouvrir des scénarios pédagogiques d'un EIAH..... | 150 |
| 6.3.1. | Collecte d'informations sur le domaine métier de l'EIAH à ouvrir..... | 154 |
| 6.3.2 | Définition de DSEML d'expression des scénarios pédagogiques existants et développement d'un premier éditeur dédié | 155 |
| 6.3.3 | Mise à l'essai du premier éditeur dédié..... | 157 |
| 6.3.4 | Evolution de DSEML existant vers ADSGEMML d'expression des SPO..... | 158 |
| 6.3.5 | Développement dirigé par les modèles d'un éditeur graphique des SPO..... | 159 |
| 6.3.6 | Réingénierie de système d'apprentissage..... | 161 |
| 6.4 | Conclusion..... | 163 |

III. TROISIEME PARTIE :

ETUDE DE CAS

| | | |
|-----------|---|------------|
| 7 | APPLICATION SUR HOP3X | 167 |
| 7.1 | L'environnement Hop3x | 168 |
| 7.1.1 | Architecture et fonctionnement de Hop3x | 168 |
| 7.1.2 | Pourquoi le choix de Hop3x ?..... | 170 |
| 7.2 | Application de notre méthode d'instrumentation pour ouvrir des sessions d'apprentissage de Hop3x..... | 171 |
| 7.2.1 | Collecte d'informations sur le domaine de Hop3x | 171 |
| 7.2.1.1 | Fouille des fichiers XML existants dans le serveur de Hop3x | 172 |
| 7.2.1.2 | Expérimentation en vue d'observer l'activité des usagers de Hop3x..... | 176 |
| 7.2.1.2.1 | Contexte de l'expérimentation | 176 |
| 7.2.1.2.2 | Objectifs de l'expérimentation | 177 |
| 7.2.1.2.3 | La session d'apprentissage mise en place | 177 |
| 7.2.1.2.4 | Observations recueillies..... | 178 |
| 7.2.1.3 | Entretiens | 179 |
| 7.2.1.3.1 | Entretiens avec les concepteurs initiaux de Hop3x | 180 |
| 7.2.1.3.2 | Entretien avec les tuteurs impliqués dans l'expérimentation | 181 |

| | |
|---|------------|
| 7.2.2 Définition du métamodèle des sessions non-ouvertes et prototypage d'un premier éditeur spécifique | 182 |
| 7.2.3 Évolution du métamodèle initial vers un autre des sessions ouvertes | 184 |
| 7.2.4 Développement dirigé par les modèles d'un environnement d'édition graphique dédié à Hop3x..... | 190 |
| 7.2.4.1 Développement de l'éditeur des sessions..... | 193 |
| 7.2.4.1.1 Création de modèle de définition graphique : « <i>sessions.gmfgraph</i> » | 193 |
| 7.2.4.1.2 Création de modèle de définition de l'outillage : « <i>sessions.gmftool</i> »..... | 196 |
| 7.2.4.1.3 Création de modèle de mise en association (mapping) : « <i>sessions.gmfmap</i> » | 198 |
| 7.2.4.1.4 Création de modèle générateur de code : « <i>sessions.gmfgen</i> »..... | 199 |
| 7.2.4.1.5 Génération de code du plug-in de l'éditeur graphique des sessions..... | 199 |
| 7.2.5 Réingénierie du système Hop3x | 201 |
| 7.2.6 Exemple d'utilisation de Hop3X ouvert..... | 205 |
| 7.3 Apports de notre approche pour Hop3x | 210 |
| | |
| 8 CONCLUSIONS ET PERSPECTIVES..... | 213 |
| | |
| 8.1 Synthèse de nos travaux..... | 214 |
| | |
| 8.2 Analyse critique de nos travaux..... | 215 |
| 8.2.1 Apports de nos propositions | 215 |
| 8.2.2 Limites de notre travail | 218 |
| | |
| 8.3 Perspectives | 218 |
| | |
| TABLE DES FIGURES | 221 |
| | |
| REFERENCES BIBLIOGRAPHIQUES | 225 |
| | |
| ANNEXE A LA SESSION HOP3X UTILISEE | 251 |
| | |
| ANNEXE B REGLES DE TRANSFORMATION..... | 259 |

1

Introduction générale

Sommaire

- 1.1. Contexte général et motivation de la recherche
- 1.2. Questions soulevées par les sciences humaines et sociales autour de la notion d'EIAH ouvert
- 1.3. Position de notre travail dans le projet REDiM
- 1.4. Eléments de la problématique
- 1.5. Définition des principaux termes utilisés
- 1.6. Aperçu de nos contributions
- 1.7. Organisation du document

Ce premier chapitre donne un aperçu global du contenu de notre thèse. On y trouvera donc une présentation du contexte général de notre travail, ainsi que les motivations qui nous ont poussées à penser aux solutions efficaces face au faible déploiement des EIAH dans les établissements de formation. Nous pensons que parmi les raisons principales, les EIAH proposés ne sont pas assez accessibles et ouverts aux praticiens. Afin de préciser la notion d'EIAH ouvert, nous nous appuyons sur les théories issues des sciences humaines et sociales, en l'occurrence les réflexions de G. Simondon [Simondon 1958] et de P. Rabardel [Rabardel 1995] sur la genèse instrumentale. Nous présentons ensuite le cadre de notre travail, le projet REDiM [Choquet 2007], où le scénario pédagogique est considéré comme un modèle pertinent et consistant pour étudier l'ingénierie des EIAH. Nous présentons les principaux éléments de notre problématique avant de définir quelques termes spécialisés utilisés dans ce document. Nous présentons enfin un aperçu de nos propositions et les travaux réalisés et terminons par un aperçu de la structure de ce mémoire.

1.1 Contexte général et motivation de la recherche

Ces dernières années ont connu un fort essor de la FOAD (Formation Ouverte et À Distance). Cet essor est dû en grande partie à l'explosion des TIC (Technologies de l'Information et de la Communication) et à leur déploiement dans différents secteurs, notamment la formation et l'éducation. Selon Bérubé, « *l'intégration des TIC dans la réalité du monde de l'enseignement suppose un nécessaire changement de paradigme. D'une conception behavioriste de l'apprentissage où l'étudiant reçoit et reproduit ce que le professeur enseigne, on tend de plus en plus vers une conception constructiviste de l'apprentissage* » [Bérubé 1996]. « *L'apport des nouvelles technologies semble avoir donné au constructivisme un nouvel élan basé sur le principe d'auto-construction du savoir* » [Lalonde et Pinette 1999]. A l'aide des TICE (Technologies de l'Information et de la Communication pour l'Enseignement), L'étudiant exerce un rôle beaucoup plus actif, cognitivement parlant, dans le processus d'apprentissage¹. Selon Martel, « *avec l'arrivée du paradigme du socioconstructivisme et l'intégration des TIC, les pratiques pédagogiques sont amenées à changer et les rôles des acteurs à être redéfinis. De nouvelles perspectives théoriques sur l'apprentissage prenant en compte les dimensions individuelle et sociale de l'acte d'apprendre sont mises en œuvre dans différentes classes* » [Martel 2005].

Notre travail de recherche se situe dans le contexte du développement des TICE et en particulier dans le domaine de l'ingénierie des EIAH (Environnements Informatiques pour l'Apprentissage Humain). Tchounikine et al. définissent un EIAH comme « *un environnement informatique conçu dans le but de favoriser l'apprentissage humain, c'est-à-dire la construction de connaissances chez un apprenant* » [Tchounikine et al. 2004]. Ces environnements relèvent de l'informatique, mais aussi d'autres disciplines comme la pédagogie, la psychologie, etc. Tchounikine définit l'ingénierie des EIAH comme « *les travaux visant à définir des concepts, méthodes et 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* » [Tchounikine 2002].

Depuis quelques années, la recherche en EIAH s'intéresse à l'ingénierie de ces environnements. Selon Cottier et al., « *il s'agit de faire évoluer les processus d'ingénierie classiques de l'informatique en les adaptant ou en les rendant adaptables aux pratiques des enseignants et des formateurs* » [Cottier et al. 2008]. Les auteurs confirment que « *l'un des*

¹ <http://enc-bessieres.org/muc/tic.htm>

enjeux de la recherche en EIAH est donc d'élaborer et d'éprouver des techniques, des méthodes et des outils d'ingénierie capables d'être adoptés et utilisés par des enseignants et des formateurs ».

« *Aucune profession n'est aussi exigeante que la profession d'enseignant, car aucune n'exige que ses membres sachent jouer autant de rôles différents* » [Gueroui 2002]. Au cours du temps, l'enseignant a joué des rôles variés suivant les différentes approches adoptées pour la modélisation d'artefacts d'apprentissage. Dans la première approche, considérée comme artisanale, l'enseignant était auteur ; alors que dans l'approche éditoriale, il était programmeur ; quant à l'approche documentaliste, l'enseignant a joué le rôle d'organisateur de contenu pédagogique, en agençant un ensemble d'objets d'apprentissage à destination des apprenants vus souvent comme des consommateurs de l'information. En revanche, récemment, dans l'approche centrée activité, l'enseignant joue le rôle de scénariste, où des EIAH sont considérés comme une deuxième génération de technologies éducatives laissant beaucoup plus de place au « récepteur » (i.e l'apprenant). Ce courant actuel d'ingénierie des EIAH, centrée sur le concept de « scénario pédagogique », s'est généralisé en liaison avec l'émergence des langages de modélisation pédagogique (EML – *Educational Modeling Language*) [Koper 2001]. Ce courant a amené à la proposition d'une spécification IMS Learning Design [IMS-LD 2003].

Bien que les travaux de recherche en EIAH existent depuis plusieurs années, voire décennies, peu d'EIAH sont utilisés en classe ou à l'université [Cottier et al. 2008]. L'un des verrous identifiés par la communauté de recherche en EIAH concerne la mise à l'épreuve des prototypes d'EIAH produits dans les laboratoires. Depuis maintenant plusieurs dizaines d'années, de nombreux artefacts logiciels ont été développés dans les laboratoires pour explorer de nouvelles modalités d'apprentissage. Une difficulté identifiée par plusieurs auteurs [Tchounikine 2006] [Nodenot 2005] concerne le transfert de propositions expérimentées dans des contextes de laboratoire vers des usages à plus large échelle [Pernin 2007]. Ces technologies éducatives n'ont pas toujours la souplesse requise pour une utilisation dans le cadre éducatif qui demande bien souvent des adaptations rapides à des événements nouveaux et souvent imprévus [Cottier et al. 2008]. Nous entrons en effet dans une nouvelle ère où les technologies progressent assez rapidement, les besoins des acteurs augmentent, les métiers évoluent, les organisations changent et les méthodes de gestion se transforment, etc. Malgré une volonté institutionnelle d'intégrer les EIAH dans les programmes des différentes disciplines, cette intégration reste faible [Previt 2003]. Plusieurs causes peuvent être évoquées : ils sont inaccessibles pour les praticiens ; ils n'ont pas toujours la flexibilité nécessaire pour s'adapter à des contextes réels de formation ; leur coût

de production est souvent élevé ; ils sont conçus dans une perspective technocentrique bien que destinés à l'utilisation humaine. La conception d'un EIAH représente un effort significatif. Coûteuse en temps comme en moyens techniques et humains, elle amène généralement les concepteurs à faire des choix en termes de pédagogie, de modalités d'interaction, de technologies, etc. Les méthodes classiques d'ingénierie des systèmes informatiques sont caractérisées par leur rigidité et leur linéarité. D'une part, dès lors que les systèmes développés ont été implantés sur site, la seule évolution que l'on peut leur faire subir se fait au travers de la maintenance. D'autre part, pour aboutir à ces systèmes, la conception se déroule suivant des étapes qui segmentent le temps, les acteurs, et les tâches à réaliser [Bourguin et Derycke 2005]. Ceci induit différents types de ruptures : d'une part rupture, ou difficulté de jeter des ponts, entre les disciplines scientifiques mobilisées, notamment entre l'informatique et les sciences sociales [Bowker, Star et Turner 1997] [Bourguin et Derycke 2005] ; et d'autre part, rupture entre processus de conception et processus d'appropriation collective (séparation concepteurs/usagers).

Cottier et El-Kechaï affirment que : « *en matière d'ingénierie des EIAH, la conception ne se réduit pas à l'acte de modélisation par un informaticien de l'objet technique, [...] la parole et la pratique du concepteur pédagogique sont fondamentales* » [Cottier et El-Kechaï 2009]. Il est donc nécessaire de concevoir des EIAH qu'on pourrait qualifier d'ouverts, donnant la possibilité aux utilisateurs de continuer eux-mêmes la conception pendant l'usage. En outre, « *l'enseignement est moins utilisateur de logiciels clé en main que de logiciels capables d'être personnalisés par leur utilisateur suivant sa conception propre de ce qu'il doit enseigner et de la façon dont il doit l'enseigner* » [RAP 1999]. Il est donc souhaitable que l'EIAH mis à disposition puisse s'adapter aux évolutions du contexte d'usage, être modifié en fonction de l'évolution des besoins et des pratiques métiers de ses usagers. F. Henri et al. affirment que « *l'environnement d'apprentissage est construit progressivement (work in progress). Il se présente comme un ouvrage malléable, un instrument d'apprentissage autant pour l'enseignant concepteur que pour l'apprenant. Il est modifié, amélioré de session en session. C'est une œuvre qui n'est jamais finie, qui s'élabore au fil de l'évolution de son contenu, des apprenants visés, des besoins et des compétences du concepteur* » [Henri et al. 2007].

1.2 Questions soulevées par les sciences humaines et sociales autour de la notion d'EIAH ouvert

Un EIAH est un système complexe qui mobilise des agents humains (apprenant, enseignant, tuteur) et artificiels dans des interactions conçues dans le but de favoriser l'apprentissage humain [Tchounikine 2002]. On peut considérer alors un EIAH comme un ensemble cohérent d'artefacts, de natures hétérogènes, sur lesquels les usagers projettent et appliquent leurs schèmes d'utilisation pour construire leur propre système d'instruments [Bourmaud 2007]. En réalité, ce système est différent d'un utilisateur à un autre, car chacun se l'approprie à sa manière, il le structure en fonction de son expérience et de ses compétences, tout en respectant les objectifs visés par des actions. Or, l'activité humaine est par nature opportuniste, singulière, située et socialement liée à un contexte particulier. Dans cette section, nous faisons un détour par les sciences humaines et sociales dans le but de clarifier la notion d'EIAH ouvert. Nous nous appuyons pour cela sur le travail de Gilbert Simondon [Simondon 1958] ainsi que sur l'approche instrumentale proposée par Pierre Rabardel [Rabardel 1995].

En matière d'appropriation des EIAH, la question n'est pas celle d'un « bon » ou d'un « mauvais » artefact. Pour G. Simondon : *« la plus forte cause d'aliénation dans le monde contemporain réside dans la méconnaissance de la machine, qui n'est pas une aliénation causée par la machine, mais par la non-connaissance de sa nature et de son essence »*. Selon lui : *« un bon outil n'est pas seulement celui qui est bien façonné et bien taillé. Le sens de l'objet technique est son fonctionnement. Il définit l'objet technique comme le rapport entre le vivant et le milieu naturel »* [Simondon 1958]. L'idée est que la conception d'un objet technique ne se complète que progressivement dans son usage. Cela nécessite une prise en main et une maîtrise de l'objet technique par des utilisateurs considérés comme des sujets et non pas comme des agents. Selon Cottier : *« Cette maîtrise dont on peut sommairement s'accommoder lorsqu'il s'agit de simples objets de notre quotidien, devient une nécessité impérieuse lorsque des objets plus complexes, qui offrent de nouvelles médiations sociales, de nouvelles façons d'appréhender le monde, en l'occurrence, de nouvelles manières d'enseigner, nous sont, au mieux, offertes et au pire, imposées »* [Cottier 2010]. Simondon considère que l'adoption des objets techniques par les sujets s'élabore dans ce qu'il désigne comme une « individualisation ». Il affirme *« la technique individualisée est une technologie concentrée sur la construction de la machine individuelle complète [...] L'invention procède principalement par l'évolution de synergies, par le processus de concrétisation. L'individualisation des êtres techniques est la condition du progrès technique. Cette individualisation est possible par la récurrence de causalité dans un milieu que l'être*

technique crée autour de lui-même et qui le conditionne comme il est conditionné par lui. Ce milieu à la fois technique et naturel peut être nommé milieu associé » [Simondon 1958]. En particulier, dans le domaine de la formation, ce « milieu associé » représente le contexte d'enseignement/apprentissage qui « regroupe l'ensemble des conditions associées à une situation spécifique mise en œuvre. Il correspond à une configuration particulière, organisationnelle et temporelle, des moyens humains et matériels, construite à partir des moyens offerts de façon générale par le dispositif de formation » [Pernin 2007]. La multitude possible et l'évolution continue de ces contextes exigent une grande accessibilité et malléabilité des technologies utilisées. De nombreux réajustements sont nécessaires afin que les usagers puissent développer leur créativité individuellement et/ou collectivement.

Bernard Stiegler va plus loin en parlant de l'appropriation des technologies grâce à la continuation de leur conception dans l'usage, il affirme que : « nous entrons dans une nouvelle époque qui rompt avec celle des producteurs-émetteurs opposés aux consommateurs-récepteurs. La logique du monde numérique veut que nous transmettions ce que nous recevons, après l'avoir enrichi par exemple d'une métadonnée (un nom sur une photo, une modification sur une notice, une ligne de code dans un logiciel libre, etc.). Nous nous trouvons dans une économie de la contribution qui s'appuie sur des technologies collaboratives » [Stiegler 2007]. Cela à l'instar des communautés des applications *open source* où il n'y a pas une différence claire entre le rôle d'utilisateur au sens de consommateur et le rôle de concepteur au sens de producteur. Tout le monde peut consommer et produire en même temps. En fait, suivant leurs besoins, les utilisateurs adaptent des outils existants pour construire leurs propres outils qui, eux aussi, peuvent être utilisés par d'autres utilisateurs. Stiegler continue : « le public veut redevenir actif et profite des possibilités offertes par ces technologies pour prendre des initiatives en ce sens. C'est donc le public qui invente ce nouveau monde industriel, et non l'initiative industrielle elle-même » [Stiegler 2007].

Dans cet ordre d'idées, à notre sens, les EIAH doivent être suffisamment ouverts pour permettre aux usagers d'avoir la main et de pouvoir poursuivre la conception durant l'usage, d'adapter les artefacts en fonction de l'évolution du contexte et des besoins de l'activité menée. Un EIAH ouvert doit donc être conçu dans une perspective anthropocentrique qui s'appuie selon [Rabardel 1995] « sur les compétences existantes des utilisateurs et cherche à les développer. Une technologie anthropocentrique cherche à augmenter les degrés de liberté laissés aux opérateurs pour définir leurs propres objectifs et activités de travail. Le contrôle s'exerce dans le sens homme technologie et non dans le sens inverse ».

P. Rabardel [Rabardel 1995] propose une théorie instrumentale à dominante *anthropocentrique*, véritablement centrée sur un homme acteur (sujet) de son travail et pour cela acteur de la fiabilité, de l'évolution et de la performance des systèmes techniques auxquelles il participe [Rabardel 1995]. Cette théorie s'inscrit dans le courant de la théorie de l'activité [Engestrom 1987], basée sur la notion de *médiation* définie par Lev Vygotsky [Vygotsky 1934]. Elle exprime justement un principe, celui de la continuation de la conception d'un *artefact* pendant son usage, artefact devenant alors *instrument*. Dans la phase de conception initiale, le concepteur institutionnel spécifie les fonctions *constituantes* et le mode de fonctionnement de l'artefact. Puis, pendant l'usage, les usagers appliquent leurs schèmes sur l'artefact en modifiant ses fonctions constituantes ou créant de nouvelles fonctions appelées *constituées*. Ces dernières, dans certains cas, anticipent des fonctions constituantes futures [Rabardel 1995]. L'instrument construit n'est donc pas donné d'emblée, il est conçu par le sujet lui-même dans le cadre d'un double processus de « genèse instrumentale » : un processus « d'instrumentation » dirigé vers le sujet qui s'adapte à l'outil en modifiant ses habitudes à travers la métacognition et la réflexivité, et parallèlement un processus « d'instrumentalisation » dirigé vers l'artefact qui est adapté et personnalisé par l'utilisateur pour atteindre ses objectifs. Rabardel affirme que ces processus « *par lesquels se constituent les composantes schème et artefact de l'instrument impliquent, de la part du sujet, une activité représentative dont le rôle dans la structuration, le contrôle et la régulation des actions est essentiel* » [Rabardel 1995]. L'auteur poursuit ; « *Il s'agit, de la part des utilisateurs, d'une conception pour soi, d'une resingularisation de l'artefact qui lui confère des propriétés nouvelles, extrinsèques, qui n'étaient pas prévues par les concepteurs institutionnels et pourront même, dans certains cas, être structurellement inscrites dans l'artefact* » [Rabardel 1995]. C'est cette resingularisation de l'artefact que [Simondon 1958] a qualifié d'individualisation de l'objet technique.

On ne peut donc observer l'instrument qu'au moment de l'appropriation de l'artefact par le sujet, au cours de son intégration dans l'accomplissement d'une activité [Henri et Maina 2007]. Un instrument est une entité mixte, composée des *schèmes* développés par l'utilisateur, pendant l'usage, et d'*artefacts* eux-mêmes élaborés par d'autres concepteurs.

Nous considérons par conséquent un EIAH ouvert comme un système d'instruments, un ensemble cohérent d'artefacts sur lesquels l'utilisateur projette et applique ses schèmes d'utilisation pour résoudre ses intentions et atteindre ses buts, à travers les différentes reconfigurations de l'EIAH opérées par adaptations dynamiques en fonction des contextes d'enseignement/apprentissage rencontrés. Il nous paraît en conséquence tout à fait nécessaire et indispensable de permettre aux utilisateurs de continuer la conception en

temps réel, via les actions d'adaptation leur permettant de réadapter, en cours d'utilisation, les activités pédagogiques prévues initialement par le concepteur. Bourguin et Derycke affirment que : « *l'activité se révèle difficile à planifier. L'idéal, face à ce problème, serait de permettre aux participants de respécifier une activité au cours de son exécution* » [Bourguin et Derycke 2005]. Autrement dit, pendant le déroulement des situations d'apprentissage, des reconfigurations de l'EIAH peuvent être effectuées à la volée par des utilisateurs impliqués (tuteurs et/ou apprenants), seuls ou en sollicitant l'enseignant concepteur. Pour Rabardel : « *c'est dans la mise en œuvre des systèmes que les instrumentalités potentielles émergent, se révèlent ou s'inventent, sont conçues le plus souvent par les utilisateurs eux-mêmes, seuls ou en collaboration avec les concepteurs initiaux, l'artefact étant alors lui-même remis en question et évoluant également* » [Rabardel 1995]. Pour conclure, F. Henri affirme que l'approche instrumentale « *ouvre en effet une perspective élargie et productive pour penser les pratiques de conception en formation à distance, notamment celles qui s'appuient sur le "design pédagogique". Les plus grandes faiblesses en matière de design pédagogique sont de s'arrêter à l'étape de la " livraison " à l'étudiant du produit de la conception* » [Henri 2007].

1.3 La position de notre travail dans le projet REDiM

Notre travail s'insère dans le cadre du projet REDiM (Réingénierie des EIAH Dirigée par les Modèles) mené au LIUM² [Choquet 2007]. Ce projet a pour objectif principal d'intégrer étroitement les enseignants et les formateurs dans le processus de conception d'un EIAH, afin de faciliter leur prise de décisions de conception, d'adaptation et de réingénierie pédagogique. Nous considérons donc des situations d'ingénierie et de réingénierie d'un EIAH où le concepteur est un enseignant ou un formateur. La position scientifique générale est de considérer la conception comme continue, supportée par un processus itératif alternant des phases d'analyse des usages et de réingénierie de l'EIAH, dans une approche dirigée par les modèles [Choquet 2007]. Pour cela, l'approche IDM (Ingénierie Dirigée par les Modèles) est adoptée dans REDiM, car elle permet une grande productivité et qualité en facilitant la génération (semi) automatique du code (ou une partie) à partir d'un (des) modèle(s) spécifié(s) à un niveau abstrait. Les modèles manipulés dans ce projet sont les scénarios pédagogiques. Mettre ainsi les enseignants usagers de l'EIAH au cœur du processus d'ingénierie et de réingénierie nécessite leur instrumentation par des supports informatiques leur facilitant la tâche à travers le cycle de vie du scénario pédagogique. Les notions de scénarios prédictifs et scénarios descriptifs [Pernin et Lejeune 2004] sont adoptées. Le scénario prédictif décrit formellement la situation d'apprentissage à mettre en œuvre et aide à son analyse qui est menée grâce à l'observation. Le scénario descriptif, en

² LIUM : Laboratoire d'Informatique d'Université du Maine.

décrivant le déroulement effectif de la situation d'apprentissage, permet d'améliorer le scénario prédictif dans un acte de réingénierie.

Un scénario pédagogique doit être conforme à un métamodèle qui décrit le langage de modélisation pédagogique (EML) [Koper 2001]. Étant donné que ce type de langages permet aux enseignants concepteurs de modéliser le déroulement des situations d'apprentissage, la conception des scénarios (scénarisation) est considérée comme une activité de modélisation. Une vue en couches issue de MDA (*Model Driven Architecture*) [OMG 2006] est alors adoptée. Quatre niveaux de description sont considérés, sur lesquels les scénarios pédagogiques et les métamodèles d'expression sont projetés [Choquet 2007] :

- ❖ M3 : niveau de méta-métamodèle auto-descriptif, auquel des métamodèles sont conformes.
- ❖ M2 : niveau de métamodèles d'expression pédagogique. Un métamodèle permet de décrire formellement un langage de modélisation pédagogique (EML).
- ❖ M1 : niveau des modèles, en l'occurrence les scénarios pédagogiques indépendants de l'EIAH.
- ❖ M0 : niveau du monde réel contenant les entités liées à l'EIAH à modéliser.

Ainsi, Chikofsky et Cross définissent la réingénierie d'un système comme « *le processus d'examen et d'altération d'un système afin de le reconstituer sous une nouvelle forme* » [Chikofsky et Cross 1990]. Pour les EIAH, on distingue deux catégories de réingénierie : (a) la réingénierie de la plateforme d'apprentissage en termes de fonctionnalités et ergonomie, (b) la réingénierie du scénario pédagogique qui consiste à améliorer le scénario prédictif en utilisant le scénario descriptif. Le projet REDiM porte précisément sur la deuxième catégorie de réingénierie. C. Choquet a défini la réingénierie pédagogique d'un EIAH comme étant « *l'examen d'une situation pédagogique médiatisée et la modification du dispositif d'apprentissage (l'EIAH proprement dit, mais aussi son contexte, tel que le rôle des acteurs, le contexte d'usage, les possibilités d'observation de l'activité) afin de reconstituer ce dernier sous une nouvelle forme et mettre ainsi en place une nouvelle situation pédagogique, prenant mieux en compte les usages observés et l'évolution des pratiques pédagogiques* » [Choquet 2007, p 6].

Un ensemble de travaux ont été réalisés dans le cadre de ce projet pour aborder les différents axes de recherche. Un premier travail, [Corbière 2006] a défini une organisation du processus global d'ingénierie et de réingénierie des EIAH, où la norme ODP-RM a été adoptée pour la formalisation d'un cadre méthodologique de travail des acteurs de la

conception et du développement d'un EIAH, en termes de processus, d'interactions, de rôles et de tâches.

Un deuxième travail [El-Kechai 2008] a étudié la conception collective des scénarios pédagogiques en se basant sur un métamodèle situé dans le domaine pédagogique des usagers de l'EIAH, qui sont en l'occurrence les enseignants concepteurs.

Un troisième travail [Zendagui 2010] a proposé une approche pour aider les enseignants à spécifier les besoins d'observation des situations d'apprentissage.

Ces deux derniers travaux se situent en amont dans le cycle d'ingénierie des scénarios pédagogiques. En revanche, d'autres travaux se situent en aval, visant la prise en compte des moyens d'observation d'une situation d'apprentissage afin d'adresser aux enseignants concepteurs des indicateurs signifiants [Iksal et al. 2004] [Randriamalaka et Iksal 2006]. Dans cet axe de recherche, le langage UTL (*Usage Tracking Language*) [Choquet et Iksal 2007] a été proposé pour définir et modéliser les données d'observation formalisées comme indicateurs sur l'activité d'apprentissage. Ces indicateurs sont calculés à partir des traces générées par l'EIAH. Ils permettent de découvrir les circonstances réelles de déroulement de scénario pédagogique mis en place, afin de le reconstituer sous une nouvelle forme et mettre ainsi en place une nouvelle situation pédagogique, prenant mieux en compte les usages observés et l'évolution des pratiques pédagogiques [Choquet 2007].

Enfin, le travail de [Pham Thi Ngoc 2011] a proposé le langage DCL4UTL et son outil support comme instruments pour permettre le calcul des indicateurs pédagogiques durant et/ou après le déroulement des sessions d'apprentissage. Cela pour aider à la capitalisation et à la réutilisation de l'expertise de modélisation des indicateurs.

Dans le projet REDiM, notre travail se focalise en particulier sur l'ingénierie et la réingénierie des EIAH que nous qualifions d'ouverts (en anglais : *Adaptive Learning Environments*). Un EIAH ouvert doit pouvoir être modifié et reconfiguré par les usagers eux-mêmes selon les différents contextes d'usages. Notre objectif est donc de rendre un EIAH ouvert aux enseignants leur permettant de poursuivre le processus de conception pendant l'usage. En effet, le scénario en tant que modèle intermédiaire entre les acteurs est suffisamment consistant et pertinent pour approcher la question de l'ouverture des EIAH permettant leur reconfiguration dynamique pendant l'usage. Nous faisons l'hypothèse que cela peut être favorisé par la définition et l'adaptation des scénarios pédagogiques explicites, exprimés dans l'univers métier des enseignants, et en se basant sur les indicateurs qui caractérisent le contexte de l'activité d'apprentissage. Pour cela, nous nous intéressons dans

notre travail à l'instrumentation du processus d'ingénierie et de réingénierie des scénarios pédagogiques ouverts (SPO), qui puissent être adaptés en fonction des évolutions du contexte dans lequel ils sont exécutés, être modifiés en fonction de l'évolution des activités et le changement des besoins des utilisateurs.

1.4 Éléments de la problématique

Dans la pratique, il est parfois difficile, voire impossible, de créer une description complète d'un scénario pédagogique avant le début de son exécution [Zarraounandia 2006b]. D'ailleurs, des études en matière de conception ont montré que le concepteur ne peut compléter et préciser sa représentation mentale que progressivement [Bonnardel et Rech 1998]. En fait, le scénario réel d'apprentissage est unique et situé dans son propre contexte. En raison de la spécificité de ce contexte et de son évolution dynamique, il est difficile de mener le scénario exactement suivant le modèle de conception prescrit, sans dévier [Yongwu 2005]. Cette déviation est gérée par l'enseignant usager qui joue le rôle de superviseur de la session d'apprentissage. Selon Dessus et K.Schneider : « *tout enseignant sait justement opérer des changements importants par rapport à ses intentions initiales, selon les imprévus survenant* » [Dessus et K.Schneider 2006].

De nombreux travaux dans la littérature portent sur l'instrumentation de la conception des scénarios pédagogiques et leur adaptation. Ces travaux peuvent être regroupés en deux catégories principales :

- La première propose de se baser sur des EML génériques, notamment IMS-LD [IMS-LD 2003]. Ces langages visent la généralité et la neutralité des domaines pédagogiques particuliers, ce qui provoque une perte des particularismes. « *Ces langages ne permettent pas en effet de répondre aux besoins très situés des concepteurs pédagogiques* » [Cottier et El-Kechai 2009]. Bien qu'ils soient destinés à l'utilisateur, ils ne sont pas véritablement centrés sur ses pratiques métiers spécifiques, mais bien plutôt sur leur métaphore d'expression pédagogique imposée. De plus, Dessus considère qu'il est nécessaire que les EML « *prennent en compte le caractère dynamique et contextualisé de toute situation d'enseignement, sans jouer sur l'interopérabilité, permettent la saisie en direct des séquences d'enseignement pour un plus grand contrôle de l'enseignant* » [Dessus et K.Schneider 2006]. Quant aux outils de modélisation des scénarios actuellement disponibles, ils demeurent inaccessibles et ne répondent pas effectivement aux besoins réels des enseignants praticiens, notamment les non-experts [Griffiths et al. 2005] [Dodero et al. 2008]. La

majorité de ces outils sont complexes et ne permettent pas d'avoir des modèles productifs et adaptables (des scénarios opérationnels et évolutifs).

- Les travaux de la seconde catégorie ont pour but de permettre aux enseignants de travailler à un niveau plus abstrait, en franchissant les barrières conceptuelles et techniques posées par les EML génériques et leurs outils. Ces travaux adoptent l'IDM comme cadre théorique et pratique en vue d'éviter de proposer aux enseignants des langages ayant une métaphore difficile à comprendre et des outils éloignés de leurs pratiques métiers. Un premier ensemble de ces travaux s'appuie sur la fusion et la transformation des modèles qui impliquent toutefois une perte sémantique. Cela a poussé d'autres travaux, toujours dans le cadre d'IDM, à suivre une démarche DSM en vue d'aider les praticiens dans la définition des langages propres à leurs domaines métiers (DSEML). Ils n'abordent toutefois pas en détail les aspects d'adaptation dynamique des scénarios.

Ces constats nous amènent à poser notre question principale de recherche et un ensemble de questions sous-jacentes :

Comment instrumenter les enseignants par des langages compréhensibles et des outils conviviaux leur permettant de concevoir et d'adapter des scénarios pédagogiques en fonction de l'évolution des contextes, à un niveau élevé d'abstraction ?

- *Par quelles méthodes ?*
- *Selon quel processus ?*
- *Avec quels moyens ? Quels outils ? Quelles techniques ?*

1.5 Définition des principaux termes utilisés

Afin de faciliter la lecture de notre document, nous présentons ici certaines des notions que nous utilisons :

- **IDM** : L'ingénierie Dirigée par les Modèles est une approche qui relève du génie logiciel consistant à mettre les modèles au cœur du processus d'ingénierie des systèmes. La construction des systèmes s'effectue donc à un niveau abstrait grâce à la manipulation des modèles.
- **DSM** : La modélisation spécifique au domaine (en anglais *Domain-Specific Modeling*) s'inscrit dans le courant de l'IDM. Elle consiste à proposer des solutions dédiées pour résoudre un problème à une échelle plus réduite, en se focalisant particulièrement sur le champ de connaissances qui limite les périmètres de ce problème.
- **Domaine métier** : Dans notre contexte de recherche sur les EIAH, le domaine métier correspond au domaine pédagogique traité. Un domaine métier limite le périmètre des connaissances concernées. Il est caractérisé par un ensemble spécifique de concepts, propriétés, relations, règles, contraintes, etc.
- **Concept métier** : il s'agit d'une connaissance du domaine métier. Dans notre contexte, un concept métier est une notion du domaine pédagogique traité, comme par exemple : variable, classe, méthode en programmation orientée objet.
- **EML générique** : Ce type de langages de modélisation pédagogique est faiblement typé quant au domaine pédagogique traité. Ces langages ont leurs propres métaphores d'expression pédagogique. Ils sont décrits généralement par des métamodèles contenant des primitives (concepts, relations, etc.) d'expression de n'importe quel scénario pédagogique indépendamment du domaine métier considéré. On peut citer, à titre d'exemple, IMS-LD [[IMS-LD 2003](#)] qui est proposé pour devenir un standard de conception des scénarios pédagogiques.
- **Scénario pédagogique opérationnel** : Il s'agit d'un scénario qui peut être exécuté sur la plateforme d'apprentissage. Dans le paradigme de l'IDM, on parle de modèle productif permettant de générer le code.
- **DSEML**: *Domain-Specific Educational Modeling Language*. Il s'agit d'un langage dédié à un domaine pédagogique particulier, permettant d'exprimer les scénarios en

utilisant les primitives spécifiques au domaine des usagers de l'EIAH, en l'occurrence des enseignants concepteurs.

- **ADSGEML:** *Adaptive Domain-Specific Graphical Educational Modeling Language*. Au-delà de DSEML, il s'agit d'un langage dédié qui permet de spécifier et d'adapter graphiquement les scénarios pédagogiques ouverts en utilisant les primitives spécifiques au domaine pédagogique des enseignants.
- **EIAH ouvert :** Un EIAH ouvert permet aux usagers de continuer la conception dans l'usage. Il peut être modifié en fonction de l'évolution de l'activité de ses usagers, pour qu'il puisse s'adapter aux différents contextes d'utilisation.
- **Scénario pédagogique ouvert :** Un scénario est ouvert s'il donne la possibilité à l'enseignant de continuer sa conception durant l'exécution, notamment par adaptations dynamiques suivant l'évolution du contexte réel d'exécution.
- **Contexte d'enseignement/apprentissage :** Nous désignons par contexte d'enseignement/apprentissage l'ensemble des circonstances qui influencent le comportement des usagers de l'EIAH et qui caractérisent les spécificités de la situation d'enseignement/apprentissage dans laquelle ces usagers sont impliqués.
- **Design Rationale :** En français « la logique de conception ». Cette approche de conception relève du génie logiciel. Elle consiste à explorer de façon systématique l'ensemble des options de conception possibles (alternatives) tout au long du processus de conception, à justifier les choix de conception et à garder la trace des décisions prises [[Lacaze 2004](#)].

1.6 Aperçu de nos contributions

Afin de rendre un EIAH ouvert à la conception et à l'adaptation de ses scénarios pédagogiques par les enseignants utilisateurs, nous avons proposé une approche de conception basée sur la modélisation de scénarios pédagogiques ouverts (SPO), qui permet l'instrumentation des enseignants par des moyens dédiés afin de les aider dans la conception continue (i.e. qui se poursuit dans l'usage) d'une activité d'apprentissage. Pour cela, nous faisons trois propositions scientifiques :

- ❑ **Un modèle rationnel de représentation des SPO**, structuré en deux couches : couche des éléments statiques et couche des variantes définies suivant les différents contextes d'exécution. Nous qualifions ce modèle de rationnel puisqu'il s'appuie principalement sur l'approche du Design Rationale que nous avons adaptée à notre problématique. Ce modèle permet aux enseignants d'avoir des scénarios qui s'adaptent à la particularité de chaque situation d'apprentissage (des variantes spécifiques aux contextes d'apprentissages) et de conserver le caractère ouvert d'un scénario pour qu'il puisse intégrer d'éventuelles nouvelles variantes, autres que celles déjà répertoriées.

- ❑ **Un processus itératif et incrémental d'ingénierie et de réingénierie** qui guide l'enseignant pour concevoir et adapter des SPO conformes au modèle rationnel que nous avons défini. L'enseignant joue un rôle pivot, il occupe une place au cœur de ce processus qui se constitue de quatre phases définissant le cycle de vie du SPO : la conception initiale ; le déploiement ; l'exécution ; et la réingénierie. Lors de la conception initiale, les variantes prescrites sont définies selon les contextes prévus. Au moment du déploiement, une variante considérée comme la plus appropriée au contexte d'exécution doit être choisie pour l'opérationnalisation. Durant l'exécution, chaque variante est déployée pour un contexte réel d'apprentissage où de nouvelles variantes imprévues peuvent émerger suite aux adaptations dynamiques. Enfin, la réingénierie consiste à capitaliser les variantes émergentes qui peuvent améliorer le SPO.

- ❑ **Une méthode dirigée par les modèles et spécifique au domaine métier pour instrumenter le processus d'ingénierie et de réingénierie des SPO d'un EIAH.** L'objectif est de permettre aux enseignants de surmonter les difficultés sémantiques et techniques posées par des EML génériques et leurs outils. Notre méthode d'instrumentation, reposant sur l'IDM et le DSM, implique les enseignants utilisateurs de l'EIAH, considérés ici comme des experts du domaine. Elle se déroule en plusieurs étapes qui amènent progressivement à définir, de façon spécifique au domaine de l'EIAH considéré, un langage d'expression graphique des SPO (ADSGEML) et un éditeur associé, permettant la conception et l'adaptation des SPO dans l'univers métier de l'EIAH.

Afin d'évaluer et de raffiner nos propositions, nous les avons appliquées sur l'EIAH «Hop3x», préalablement conçu au LIUM dans le cadre d'un autre projet pour pratiquer la programmation orientée objet. Notre objectif a été de rendre Hop3x ouvert pour des enseignants qui veulent l'utiliser. Nous avons donc appliqué notre méthode afin d'instrumenter le processus d'ingénierie des sessions ouvertes. Nous avons fait appel aux

enseignants concepteurs de Hop3x dans le but d'extraire et de cristalliser les connaissances métiers et d'étudier les possibilités d'adaptation des sessions pour les structurer selon le modèle rationnel. Un ADSEML et un environnement d'édition graphique ont été construits pour permettre aux enseignants néophytes de concevoir et d'adapter des sessions ouvertes de Hop3x à un niveau élevé d'abstraction. Pour l'implémentation, nous avons utilisé les frameworks EMF/GMF (*Eclipse Modeling Framework / Graphical Modeling Framework*).

1.7 Organisation du document

Ce document est composé de trois parties.

La première partie est consacrée au positionnement scientifique de nos travaux, ainsi qu'à la précision de notre problématique de recherche. Elle est composée de deux chapitres :

Le chapitre 2 dresse un positionnement scientifique de notre travail. Il s'agit de se focaliser en premier lieu sur l'adaptation en EIAH. Nous montrons la nécessité d'adaptation des scénarios pédagogiques dans la pratique. Ainsi présentons nous les différents travaux existants sur la conception et l'adaptation des scénarios pédagogiques.

Le chapitre 3 détaille notre problématique, présente notre objectif de recherche en s'appuyant sur le positionnement scientifique. Des questions de recherche sont posées et la méthodologie de recherche est présentée.

La deuxième partie de ce document détaille nos propositions scientifiques qui sont présentées dans trois chapitres :

Le chapitre 4 définit la notion d'un SPO et présente le modèle rationnel des SPO et son implémentation sous forme de métamodèle.

Le chapitre 5 présente le processus itératif et incrémental d'ingénierie des SPO structurés selon le modèle rationnel.

Le chapitre 6 est consacré à la présentation de la méthode d'instrumentation que nous avons proposée pour ouvrir les scénarios pédagogiques d'un EIAH. Cette méthode est dirigée par les modèles et spécifique au domaine métier.

Dans la troisième partie de ce document, nous présentons l'application de nos propositions théoriques sur un cas pratique de l'EIAH « Hop3x ».

Le chapitre 7 présente en premier lieu l'environnement Hop3x et les raisons de son choix. Il détaille ensuite les étapes de notre méthode appliquée pour ouvrir des sessions d'apprentissage de Hop3x.

Enfin, **le chapitre 8** dresse un bilan des travaux réalisés et une analyse de nos propositions. Ce dernier chapitre se conclut avec quelques perspectives en termes de travaux que nous envisageons de poursuivre.

I. Première partie :

Positionnement scientifique et problématique de recherche

2

Positionnement scientifique

Sommaire

[2.1 Adaptation et EIAH](#)

[2.2 Design Rationale](#)

[2.3 Travaux existants en lien avec notre objet d'étude](#)

Ce chapitre dresse un positionnement scientifique de notre travail. Il s'agit de se focaliser en premier lieu sur l'adaptation en EIAH. Nous mentionnons les différents types d'adaptation possibles dans un EIAH, ainsi que les limites des STI³ et des SHA⁴ liées notamment à leur accessibilité pour des enseignants. Afin de dépasser ces limites, plusieurs approches récentes sont proposées pour élaborer des EIAH adaptables, notamment autour de la conception et l'adaptation des scénarios pédagogiques et des apports possibles des outils d'édition. Nous distinguons deux approches de conception pédagogique : interprétative, par utilisation d'un EML générique ; et constructive, par utilisation d'un EML spécifique au domaine du concepteur. Ensuite, nous montrons la nécessité d'adaptation des scénarios pédagogiques dans la pratique. Ainsi présentons nous les différents travaux existants sur la conception et l'adaptation des scénarios pédagogiques. Un premier ensemble de travaux adopte les EML génériques, notamment IMS-LD. Puis, les travaux s'appuyant sur IDM et DSM. D'autres approches à base de Web sémantique sont également présentées.

³ STI : Systèmes tutoriels intelligents.

⁴ SHA : Systèmes hypermédias adaptatifs.

2.1 Adaptation et EIAH

L'intégration de la technologie dans le processus pédagogique nécessite d'aborder la problématique de l'amélioration de la qualité de l'enseignement et de l'apprentissage. Elle ne doit pas être considérée simplement comme un énième moyen, même flexible, de délivrer des ressources pédagogiques. [Nichols 2003] [Cristea 2003] [Rumetshofer et Wöß 2003] et [Ayersman et Minden 1995] ont mis en évidence le manque de personnalisation des EIAH. Or, l'activité humaine est par nature opportuniste, singulière et située. L'adaptation constitue alors un défi majeur en matière d'EIAH pour des utilisateurs aux profils très divers, tenant des rôles distincts, exprimant des besoins et centres d'intérêt différents, dans des contextes d'apprentissage variés, etc. L'adaptation des systèmes d'apprentissage est donc indispensable. Elle permet, notamment, de satisfaire les besoins spécifiques des apprenants, d'accomplir leurs objectifs d'apprentissage et de s'adapter aux contextes spécifiques des différentes communautés. L'adaptation permet aussi de ne pas submerger l'apprenant d'informations qui ne sont pas pertinentes pour lui [Ouraiba et al. 2008a] [Ouraiba et al. 2008b] [Ouraiba et al. 2009]. Dolog et al. [Dolog et al. 2007] soulignent l'importance de l'adaptation des environnements en vue d'une individualisation des apprentissages. Selon Boticario et Santos, l'adaptation dans un EIAH consiste à créer une expérience d'apprenant ajustée volontairement aux différentes conditions (par exemple, les caractéristiques personnelles et les centres d'intérêts, les connaissances pédagogiques, les interactions des apprenants, les productions des processus d'apprentissage réels, les contenus disponibles, la similitude avec les pairs, etc.) sur une période de temps avec une intention d'augmenter la réussite selon certains critères prédéfinis (par exemple, l'efficacité de l'apprentissage: le score obtenu, le temps écoulé, le coût économique, la participation et la satisfaction des utilisateurs) [Boticario et Santos 2007].

En outre, selon Burgos [Burgos 2008], il y a de nombreux aspects à prendre en considération pour une adaptation efficace, tels que l'interface utilisateur, les ressources d'apprentissage, les processus d'apprentissage, etc. Il existe une vaste collection d'approches d'adaptation dans des EIAH [Burgos, Tattersall et Koper 2006]. Van Rosmalen, Vogten et al. [Van Rosmalen, Vogten et al. 2006] proposent, par exemple, d'incorporer le tuteur comme un facteur clé dans le processus d'adaptation [Wasson 1997] et de construire un système hybride qui s'appuie fortement sur des agents de l'intelligence artificielle. Toutes ces approches sont basées sur la proposition d'adaptation d'apprentissage personnalisé au contexte de chaque apprenant afin de stimuler son processus d'apprentissage et d'encourager son implication dans ce processus [Fredericksen, Pickett et al. 2000] [He, Kinshuk et al. 2002] [Burgos et Ruiz-Mezcua 2003]. Ces approches retiennent également

que la meilleure performance d'apprentissage provient de la pédagogie personnalisée [Towle et Halm 2005].

On trouve dans la littérature plusieurs recherches récentes définissant deux dimensions importantes, l'adaptivité et l'adaptabilité, qui se focalisent toutes les deux sur l'apprentissage personnalisé [Ahmad et al. 2004] [Chen et Magoulas 2005] [Burgos et al. 2006]. L'adaptivité est la capacité de modifier, automatiquement, des cours déployés dans un EIAH en utilisant différents paramètres et un ensemble de règles prédéfinies. Cette première approche est centrée-machine. En revanche, l'adaptabilité qui est centrée-utilisateur, consiste à donner la possibilité aux utilisateurs de personnaliser un cours par eux-mêmes [Burgos 2008]. Burgos a défini l'adaptation dans les EIAH comme : « une méthode pour créer une expérience d'apprentissage pour l'étudiant, mais aussi pour le tuteur, basée sur la configuration d'un ensemble d'éléments dans une période spécifique visant l'augmentation de la performance des critères prédéfinis [Van Rosmalen, Vogten et al. 2006]. Ces critères peuvent être éducatifs, économiques, relatifs au temps ou à la satisfaction des utilisateurs. Les éléments à modifier/adapter peuvent concerner le contenu, le temps, l'ordre, l'évaluation, l'interface, etc.» [Burgos 2008].

Specht et Burgos [Specht et Burgos 2007] ont relevé une variété importante de recherches effectuées sur les systèmes hypermédias éducatifs adaptatifs pour développer la façon d'adapter les parcours et les contenus d'apprentissage pour les apprenants, individuellement et par groupes [Leutner 1992] [Brusilovsky 1996] [Weber et Specht 1997] [Specht 1998] [De Bra et Calvi 1999].

De point de vue de Specht et Burgos, des méthodes adaptatives dans des applications hypermédias éducatives peuvent être essentiellement structurées selon quatre questions principales [Specht 1998] :

- ❖ *Quels parties ou composants du processus d'apprentissage sont adaptés ?* Cette question met l'accent sur la partie d'une application adaptable qui est adaptée par une méthode adaptative. Il existe des exemples pour personnaliser le rythme de l'enseignement [Tennyson et Christensen 1988] [Leutner 1992], l'adaptation de présentation de contenu, et le séquençage des contenus. Les extensions avec de nouvelles formes de diffusion de l'information permettent la distribution des matériaux d'apprentissage à différents contextes d'apprentissage pertinents pour un utilisateur individuel ou pour des groupes d'utilisateurs.
- ❖ *Pourquoi le système s'adapte ?* Cette question interroge principalement les modèles pédagogiques sous jacents à l'adaptation [Pask 1964] [Frank 1965] [Salomon 1975].

Les systèmes hypermédias classiques s'adaptent pour palier des déficits de connaissances, des problèmes ergonomiques, ou s'adaptent aux styles d'apprentissage pour une introduction plus facile à une matière.

- ❖ *Quelles informations le système utilise pour l'adaptation ?* Dans la plupart des applications éducatives adaptatives, un modèle d'apprenant est utilisé comme base de l'adaptation des paramètres. L'adaptation tient compte, non seulement des connaissances de l'apprenant, de ses préférences, de ses intérêts, de ses capacités cognitives, mais aussi de ses tâches et de ses objectifs.
- ❖ *Comment le système recueille-t-il l'information pour s'adapter ?* Il existe une variété de méthodes pour recueillir des informations : des méthodes explicites, par interrogation des apprenants, et/ou des méthodes implicites, par observation et inférence des données d'interaction. On les trouve décrites généralement dans des travaux sur la modélisation des utilisateurs. Jameson en dresse un aperçu [Jameson 2003].

2.1.1 Types d'adaptation possibles dans un EIAH

L'étude de la littérature permet d'identifier huit types d'adaptation mis en œuvre dans les systèmes d'apprentissage [Specht et Burgos 2006]. Chacun d'entre eux utilise diverses informations et vise à mettre au point les activités et les actions de l'apprenant pour obtenir la meilleure expérience d'apprentissage [Butz et al. 2003]. Les huit types d'adaptation sont présentés ci-dessous [Burgos 2008].

1. adaptation d'interface (appelée aussi navigation adaptative et liée à l'utilisabilité et l'adaptativité) : où les éléments et les options de l'interface graphique d'utilisateur sont positionnés sur l'écran et leurs propriétés sont définies (couleurs, taille, font, etc.) [Ahmad et al. 2004]. Cela est étroitement lié au soutien des personnes ayant des besoins spéciaux, ayant des handicaps tels que le daltonisme ou une mauvaise audition par exemple [Chin 2001].

2. adaptation de flux d'apprentissage : où le processus d'apprentissage est dynamiquement adaptable pour séquencer le contenu du cours de différentes façons. Le chemin d'apprentissage est dynamique et personnalisable pour chaque utilisateur, et, au moment du lancement du cours, un utilisateur peut prendre un autre itinéraire en fonction de ses capacités.

3. adaptation de contenu : où des ressources changent dynamiquement leurs contenus, comme les systèmes éducationnels à base de présentation adaptative [Brusilovsky and Miller 2001] [De Bra et al. 2004].

Ces trois premiers types d'adaptation constituent la base des suivants [Brusilovsky et Paylo 2003].

4. Support interactif de résolution des problèmes : qui guide l'utilisateur sur la prochaine étape à prendre afin d'obtenir la bonne solution à un problème. Ces orientations pourraient provenir d'un tuteur en ligne ou hors ligne ou à partir d'un ensemble de règles prédéfinies.

5. Filtrage d'information adaptatif : consiste à ne fournir que l'information pertinente aux utilisateurs. Un système de filtrage fait parvenir les objets d'apprentissage aux apprenants qui en ont besoin à partir de larges volumes d'objets générés continuellement, voire augmente la quantité d'objets pertinents collectés à partir de différents viviers en ciblant les objets vraiment appropriés [Ouraiba et al. 2008a] [Ouraiba et al. 2008b]. Le filtrage d'informations peut être considéré comme une facilité externe liée à l'activité d'apprentissage et n'est pas une partie réelle de cette activité.

6. Regroupement adaptatif des utilisateurs : permet de créer des groupes spécifiques d'utilisateurs et de soutenir la collaboration en vue de la réalisation de tâches spécifiques. Par exemple, à la suite d'un ensemble de questions, deux groupes se constituent : groupe de débutants et groupe d'experts.

7. Évaluation adaptative : où le modèle d'évaluation, le contenu et le déroulement d'un test peuvent être changés en fonction de la performance de l'élève et l'orientation de l'enseignant [Van Rosmalen et al. 2006].

8. Temps d'adaptation : la possibilité de modifier / adapter un cours par le système ou par le tuteur en temps réel [Van Rosmalen et Boticario 2005], contrairement aux changements définis dans la phase de conception [Merceron et Yacef 2003].

Un ensemble large et cohérent de règles de dépendances entre les utilisateurs, de méthodes et d'objets d'apprentissage est nécessaire pour décrire ces huit types d'adaptation, et d'ailleurs leurs combinaisons possibles [Karampiperis et Sampson 2004].

2.1.2 Limites des STI et SHA

L'adaptation dans les EIAH représente un champ de recherche qui prend ses racines depuis plus de trois décennies à partir notamment : des **Systemes tutoriels intelligents (STI)** [Wenger 1987] et des **Systemes hypermédias adaptatifs (SHA)** [Brusilovsky 2001]. Les STI se basent sur des techniques de l'intelligence artificielle. Ces systèmes adaptent leurs explications et stratégies d'enseignement aux apprenants en regard de leurs niveaux de connaissances et de leur progression d'apprentissage ; quant aux SHA [Brusilovsky 2001] ils permettent de personnaliser la présentation et la navigation pour chaque utilisateur.

Malgré de nombreuses années de recherche et de développement, ces environnements d'apprentissage adaptatifs ont été relativement peu utilisés dans les écoles et les centres de formation [Murray 2004]. Ces systèmes sont complexes à mettre en œuvre, notamment à cause des différents traitements, sélections et combinaisons effectués sur les données, causant des problèmes de désorientation et de surcharge cognitive chez l'apprenant. En outre, l'adaptation dans ces systèmes est prédéfinie par le développeur du système. Malgré une relative souplesse liée à l'utilisation des techniques d'Intelligence Artificielle, les stratégies de raisonnement restent par définition prévues pendant la conception, ce qui limite les possibilités de comportement du système et introduit une certaine rigidité dans l'interaction. En plus, ces systèmes utilisent leurs propres représentations et des mécanismes persistants internes pour l'adaptation pédagogique. Ceci ne permet pas la réutilisabilité et l'échange des solutions adaptatives, et réduit l'intégration des solutions dans la majorité des EIAH utilisés [Specht et Burgos 2007].

Park, Perez et Seidel [Park, Perez et Seidel 1987] affirmaient que les développeurs des STI n'ont pas vraiment réussi à incorporer de nombreux éléments liés à l'apprentissage et aux stratégies pédagogiques mis en évidence par les recherches en éducation et par les éducateurs eux-mêmes. Park et Lee [Park et Lee 2001] ont affirmé que des efforts de coopération entre les experts de différents domaines, notamment ceux de l'apprentissage et de l'enseignement et ceux de l'intelligence artificielle, sont nécessaires à l'élaboration des systèmes mieux adaptés, utilisant des méthodes et des techniques des STI [Park et Seidel 1989] [Seidel, le parc et Perez, 1989]. Néanmoins, les challenges sur la façon d'enseigner avec les systèmes intelligents demeurent difficiles [Park et Lee 2001].

Quant aux SHA, ils visent à améliorer la compréhension par les apprenants du matériel d'apprentissage en leur fournissant des parcours et des contenus adaptés. Mais l'utilisation de ces systèmes dans les environnements réels d'apprentissage est exceptionnelle. Plusieurs facteurs sont évoqués [Murray 2004] [Berlanga 2007] :

- le manque de preuves crédibles de leurs avantages à grande échelle [Murray 2004] ;
- le manque d'intérêt des professionnels de l'éducation et de la formation ;
- le coût élevé de la production et de la maintenance de ces systèmes [Murray 2004] ;
- le temps nécessaire au développement de ces processus complexes ;
- la validité discutable de leurs évaluations, rarement réalisées dans de réels contextes [Weibelzahl 2005] ;
- la prolifération de systèmes autonomes et l'absence d'outils génériques proposés [Stash 2007] ;
- on ne peut exprimer explicitement les méthodes didactiques et les modèles pris en considération. Les méthodes et les contenus sont indissociables [Van Rosmalen 2008] ;
- l'insuffisance des informations significatives sur les scénarios d'apprentissage et les contextes réels d'apprentissage ;
- les enseignants ne trouvent pas que ces systèmes soient pertinents ; ils ne proposent pas aux apprenants d'expériences d'apprentissage convenables ;
- l'absence des mécanismes de partage et de réutilisation des composants, tels que les stratégies d'adaptation ou des ressources d'apprentissage, entre les différents cours et systèmes.

Selon Berlanga [Berlanga 2006], les SHA sont pensés et conçus d'un point de vue technocentrique, sans implication des praticiens. Cette tendance, qui existe encore aujourd'hui, n'intègre pas les concepteurs pédagogiques dans la description des exigences et la conception des SHA. Les enseignants concepteurs éprouvent, à l'usage, beaucoup de difficultés pour adapter les expériences d'apprentissage aux apprenants. Dans la plupart des cas, les approches sont orientées « contenu » et ne pensent pas l'apprentissage comme un ensemble d'interactions entre enseignants, apprenants, collègues et activités [Berlanga 2007]. Tout cela peut provoquer le décrochage des utilisateurs de la formation, notamment des apprenants, en raison d'une efficacité pédagogique qui est souvent assez faible.

Pour dépasser les limites des STI et des SHA, le défi est double selon Berlanga. D'une part, il est nécessaire de concevoir des outils de création qui prennent en charge la

modélisation des conceptions de l'apprentissage sans prescrire toute approche particulière et, d'autre part, il est souhaitable que les ressources et les éléments utilisés dans les conceptions pédagogiques puissent être réutilisés et échangés entre les différents cours et systèmes [Berlanga 2007]. En outre, Murray a affirmé dans [Murray et al. 2003] que les outils d'édition ont été élaborés pour des environnements d'apprentissage parce qu'ils traitent toutes des préoccupations évoquées précédemment.

2.1.3 Conception et adaptation des scénarios pédagogiques

Baker [Baker 2007] a distingué trois types de modèles d'un EIAH : (1) le modèle comme composant logiciel de l'EIAH « *modèle-composant* », par exemple le modèle d'apprenant qui est un modèle cognitif de traitement des informations symboliques ; (2) le « *modèle-évaluation* » qui définit les changements d'états de connaissances humaines ; enfin (3) le « *modèle-conception* » qui fixe les objectifs pédagogiques principaux à atteindre dans le cadre de la conception des systèmes EIAH (*instructional design*). Ce dernier modèle, d'après Baker, « *doit permettre une réflexion dialogique ouverte, mettant en jeu des compétences diverses chez de multiples acteurs* » [Baker 2007]. L'auteur poursuit : « *de tels modèles doivent nécessairement être ouverts, délibérément et stratégiquement imprécis, des objets qui suscitent la réflexion* [Puustinen et al. 2006], *fonctionnant dans une collaboration entre chercheurs et les acteurs du terrain* » [Baker 2007].

En ingénierie des EIAH, les décisions et les pratiques des concepteurs pédagogiques sont fondamentales. Dans notre travail, nous considérons le scénario pédagogique comme un modèle suffisamment consistant et pertinent pour approcher l'ingénierie des EIAH. Cottier, Choquet et Tchounikine affirment que « *la préoccupation d'un enseignant qui souhaite utiliser l'informatique pour son enseignement n'est pas de faire de la recherche en EIAH, elle est de concevoir une situation d'enseignement qui réponde au problème pédagogique qu'il se pose. Comme le souligne Pastré, le « praticien est un opportuniste » [il] « ne cherche ni la complétude, ni la cohérence de son savoir, il cherche l'efficacité » [Pastré 1997] » » [Cottier et al. 2008]. Les auteurs poursuivent : « *concevoir des situations pédagogiques avec instruments nécessite de définir différents éléments : un but d'apprentissage, c'est-à-dire une ou plusieurs connaissances ; une tâche d'apprentissage, c'est-à-dire quelque chose à réaliser par les élèves (ou « apprenants ») au moyen d'actions ou d'opérations ; des acteurs soit, a minima, un enseignant et un apprenant ; un contexte d'apprentissage définissant un temps, un lieu, des artefacts, etc. ; une tâche d'enseignement, où l'on tente de mettre en adéquation les quatre termes précédents, de sorte que la connaissance enseignée, à travers la tâche à réaliser par les apprenants dans le**

contexte, puisse donner lieu à l'élaboration d'une nouvelle connaissance par l'apprenant [Tchounikine et Tricot 2008] » [Cottier et al. 2008].

Selon J-P. Pernin et A. Lejeune : « Aujourd'hui, la problématique de la conception des situations d'apprentissage par l'enseignant occupe une place stratégique dans le champ des TICE. Après s'être focalisées sur les questions de création, de mutualisation et de réutilisation de ressources, les recherches en ingénierie pédagogique [Paquette 2002] [Paquette 2004] [Pernin 2003] [Crozat 2002] insistent sur la nécessité de prendre davantage en compte l'activité de l'apprenant et replace la notion de scénario d'apprentissage au centre des débats. Un des objectifs poursuivis est de favoriser les pratiques de réutilisation, de mutualisation et d'échange entre professionnels de l'éducation, non pas uniquement en termes de ressources et de documents, mais également en termes de savoir-faire pédagogiques intégrés aux contextes d'apprentissage » [Pernin et Lejeune 2004a]. Ces travaux amènent à étudier le concept de scénario pédagogique, souvent employé dans les milieux francophones mais répondant à des définitions variées.

Selon Vantroys et Peter, « les scénarios pédagogiques provoquent un grand intérêt dans le domaine des EIAH. Ils fournissent des mécanismes pour la gestion et l'orchestration des activités à l'intérieur des unités d'apprentissage. Ils peuvent ainsi être au centre de l'ingénierie de formation et sont des moyens pour définir l'utilisation des outils et des objets pédagogiques au cours d'un module ou d'une tâche dans lesquels les utilisateurs sont impliqués. En fait, le bénéfice de l'utilisation de scénarios pédagogiques, comme indiqué par [Hummel et al. 2004], réside dans le fait que l'attention principale est mise sur les activités d'apprentissage qui doivent être réalisées pour atteindre les objectifs pédagogiques » [Vantroys et Peter 2005].

Charlier et Daele définissent le scénario pédagogique comme étant « le résultat du processus de conception d'une activité d'apprentissage. Dans un scénario, on trouve donc des objectifs, une planification des activités d'apprentissage, un horaire, une description des tâches devant être effectuées par les étudiants, des modalités d'évaluation le tout étant défini, agencé et organisé au cours d'un processus de design » [Charlier et Daele 2002].

Pernin et Lejeune définissent dans [Pernin et Lejeune 2004b] le scénario pédagogique comme étant « la description du déroulement d'une situation d'apprentissage en termes de rôles, d'activités et d'environnement nécessaire à sa mise en œuvre, mais aussi en termes de connaissances manipulées ». Ces auteurs ont distingué deux types de scénarios :

- Le scénario prédictif : défini comme établi a priori par un concepteur en vue de la mise en place d'une situation d'apprentissage ;

- Le scénario descriptif : défini comme un scénario qui *décrit a posteriori le déroulement de la situation d'apprentissage en y incluant en particulier les traces de l'activité des acteurs (essentiellement les apprenants) et leurs productions.*

Un scénario pédagogique peut être défini par un langage de design [Schneider 2006], en l'occurrence langage de modélisation pédagogique (EML pour *Educational Modeling Language*) proposé par [Koper 2001]. Rawlings et al. [Rawlings et al. 2002] définissent un EML comme un modèle sémantique d'informations décrivant le contenu et le déroulement d'une unité d'apprentissage d'un point de vue pédagogique afin de supporter la réutilisation et l'interopérabilité.

2.1.3.1 Deux approches de conception des scénarios pédagogiques

Formellement, un EML peut être décrit par un métamodèle qui est lui-même lié par une relation de conformité avec le scénario pédagogique considéré comme modèle. On distingue, dans ce cadre, deux approches de conception pédagogique [El-Kechaï 2008] [Choquet 2007] : la première est basée sur l'utilisation d'un métamodèle prédéfini pour exprimer le scénario pédagogique ; nous la désignons par "*approche interprétative*". La seconde est basée sur une construction du scénario et du métamodèle par abstraction depuis une source ; nous la désignons par "*approche constructive*". La figure 1 illustre ces deux approches.

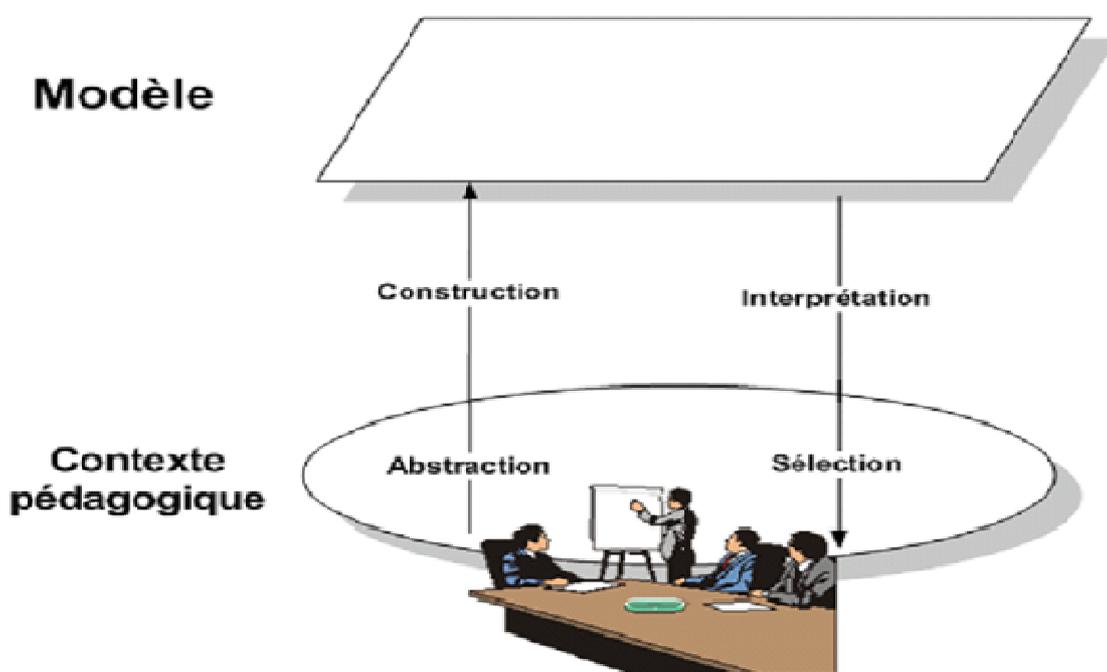


Figure 1 : Les approches interprétative et constructive de la conception pédagogique [El-Kechaï 2008].

2.1.3.1.1 Approche interprétative

Un processus «classique» de conception est considéré comme «approche interprétative», où un EML existant (tels que IMS-LD) est adopté pour spécifier des scénarios pédagogiques. Cet EML est souvent générique et développé par des experts de scénarisation autres que les concepteurs (principalement les enseignants) des scénarios. Les enseignants ne participent pas à l'implémentation du métamodèle pédagogique. Ce métamodèle peut être « réutilisable pour décrire des scénarios dans différents domaines et/ou pour différentes approches pédagogiques » [El-Kechaï 2008]. Les enseignants concepteurs doivent alors s'approprier la sémantique de cet EML afin de transformer le métamodèle de leur domaine spécifique en métamodèle de l'EML choisi. Cette approche est illustrée par la figure 2 :

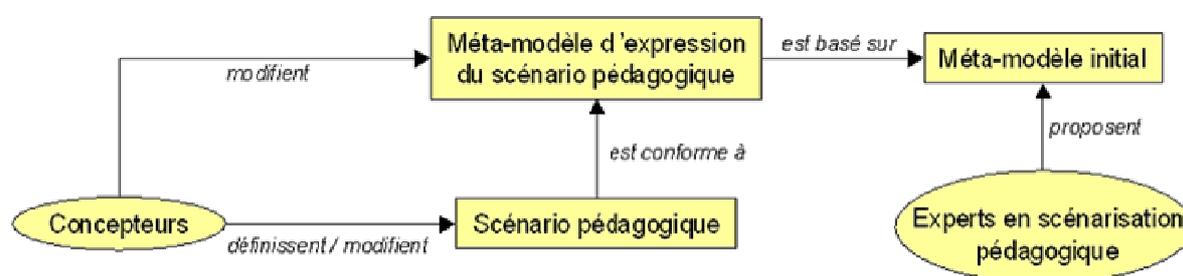


Figure 2 : Approche interprétative de la conception d'un scénario pédagogique [Choquet 2007].

Cependant, les praticiens et la littérature [Griffiths et al. 2005] [Dodero et al. 2008] confirment le manque d'appropriation de la sémantique des EML génériques et la difficulté d'utilisation des éditeurs disponibles dans la pratique [Ouraiba et al. 2011b]. Le risque au moment de choisir un métamodèle prédéfini, c'est que certaines particularités peuvent émerger *in situ* (dans la situation d'apprentissage réel). Ces particularités ne peuvent pas toujours être prévues au moment du design. Leur description ne peut pas être supportée par le scénario pédagogique prédictif parce que le métamodèle d'expression choisi ne le permet pas. Le métamodèle devrait donner aux concepteurs la possibilité d'adapter et faire évoluer leurs scénarios [El-Kechaï 2008]. En outre, l'utilisation d'un EML générique impose une structure de modélisation basée sur une métaphore particulière. Pour l'utiliser, il est nécessaire de modéliser le scénario en conformité avec cette structure [El-Kechaï 2008].

2.1.3.1.2 Approche constructive

Selon Van Rosmalen [Van Rosmalen 2008], il semble difficile d'utiliser en pratique les modèles de design pédagogique existants en dehors du contexte d'équipes spécialisées. Pour répondre aux insuffisances de l'approche interprétative, un processus plus itératif de conception « l'approche constructive » permet de construire un langage de modélisation pédagogique spécifique où les enseignants concepteurs, généralement aidés par des spécialistes de la modélisation, identifient et cristallisent les connaissances de leur domaine spécifique afin d'élaborer un métamodèle d'expression pédagogique (et donc, leur EML « spécifique au domaine »), et de l'utiliser pour spécifier leurs scénarios. Cette approche consiste à engager les concepteurs dans un processus de conception itératif, éventuellement soutenu par des phases de réingénierie de scénario pédagogique et aussi de métamodèles d'expression pédagogique pour disposer de concepts permettant de mieux caractériser le scénario pédagogique dans les cycles de vie ultérieurs. Cette approche est illustrée par la figure 3 :

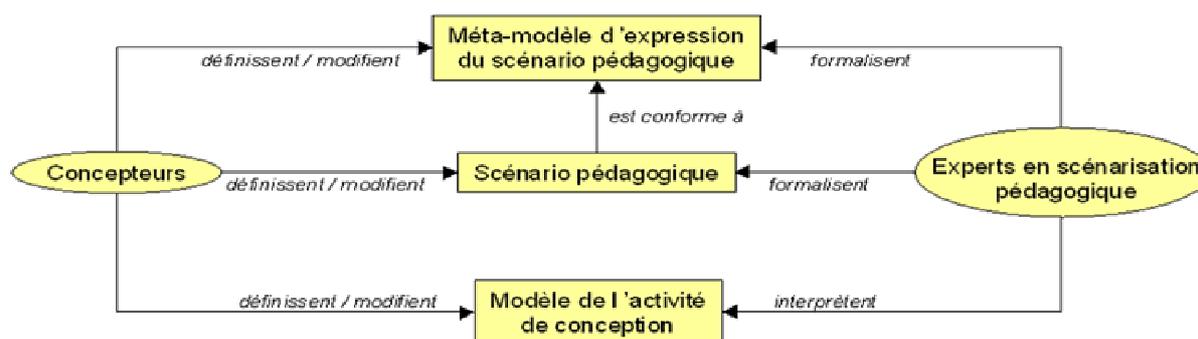


Figure 3 : Approche constructive de la conception d'un scénario pédagogique [Choquet 2007].

F. Henri [Henri 2007] a affirmé que : « *les plus grandes faiblesses en matière de design pédagogique sont de s'arrêter à l'étape de la " livraison " à l'étudiant du produit de la conception* ». En fait, l'activité de la conception d'un scénario pédagogique doit se poursuivre durant l'exécution. Elle ne doit pas être simplifiée comme un acte de modélisation préliminaire d'un artefact qui est exogène du contexte réel de son utilisation, elle doit être poursuivie dans l'activité des utilisateurs eux-mêmes [Cottier et El-Kechai 2009]. Cela nécessite le développement et l'utilisation de modèles qui sont endogènes à des contextes d'utilisation et qui peuvent évoluer en cohérence avec leur métamodèle.

2.1.3.2 Nécessité d'adaptation des scénarios pédagogiques

Dans des situations d'enseignement en classe, selon Cottier et Choquet : « *la pratique usuelle des enseignants et des formateurs en enseignement présentiel ne les amène pas à scénariser précisément l'activité pédagogique : si le contenu d'une formation est bien défini a priori, une simple idée générale et peu détaillée de l'organisation de l'activité pédagogique suffit à l'enseignant, son habileté et son expérience lui permettant d'adapter sa méthode et d'improviser en fonction du déroulement de l'activité* » [Cottier et Choquet 2005]. Ainsi, Dessus a affirmé que « *pendant la phase de planification en enseignement présentiel, l'enseignant met au jour une intention préalable, qui peut être distincte de son intention en action* » [Dessus 2005]. L'auteur poursuit « *l'enseignant ne peut pas planifier, mais, une fois dans sa classe, sa préoccupation majeure va consister à réagir aux élèves* [Huberman 1986], donc à replanifier. Par conséquent, tout enseignant sait justement opérer des changements importants par rapport à ses intentions initiales, selon les imprévus survenant en classe » [Dessus et K.Schneider 2006].

En enseignement médiatisé par les TIC, malgré les travaux visant à lever certains obstacles en matière d'EIAH, liés à la distance notamment, et pour faciliter la pratique des enseignants et des apprenants, « *les technologies aujourd'hui disponibles n'ont pas toujours la souplesse requise pour une utilisation dans le cadre éducatif qui demande bien souvent des adaptations rapides à des événements nouveaux et souvent imprévus* » [Cottier et al. 2008]. Les EIAH existants ne sont souvent pas bien adaptés aux scénarios effectifs conçus par les enseignants et/ou au contexte effectif d'apprentissage. F. Henri affirme que « *l'environnement d'apprentissage est construit progressivement (work in progress). Il se présente comme un ouvrage malléable, un instrument d'apprentissage autant pour l'enseignant-concepteur que pour l'apprenant. Il est modifié, amélioré de session en session. C'est une œuvre qui n'est jamais finie, qui s'élabore au fil de l'évolution de son contenu, des apprenants visés, des besoins et des compétences du concepteur* » [Henri et al. 2007]. Cette évolution renvoie selon nous à l'approche instrumentale proposée par Rabardel [Rabardel 1995] qui souligne la dimension créatrice, inventive et difficilement prédictible d'un sujet qui, engagé dans une activité, mobilisera les artefacts à sa disposition. Il existe en effet une forme de permanence dans l'activité du sujet qui veut que la conception se poursuive dans l'usage.

Cependant, comme le souligne Henri, « *l'enseignant n'est-il pas investi du devoir d'améliorer et de faire évoluer constamment les cours qu'il donne, les leçons qu'il enseigne et de ne jamais les considérer comme finis ?* » [Henri et al. 2007]. Pour qu'une expérience d'apprentissage atteigne son objectif efficacement, l'EIAH doit être ouvert afin de s'adapter à

des circonstances changeantes. Cette souplesse est nécessaire, voire indispensable pour dépasser les imprévus qui peuvent faire obstacle au bon déroulement des scénarios pédagogiques.

Ainsi, Henri et al. affirment que : « *un scénario pédagogique pourra être interprété différemment selon les acteurs. Le scénario peut jouer un rôle essentiel dans la mesure où, entre autres, il prend en compte ou non les buts poursuivis par les apprenants, exploite ou non leurs pratiques de référence, prévoit ou non des moments de régulation et rend ou non explicite le choix des activités et leurs sens* » [Henri et al. 2007]. La question de l'adaptation des scénarios pédagogiques devient alors cruciale. « *Quelles que soient les qualités didactiques d'un scénario pédagogique ou les compétences d'un étudiant, c'est leur ajustement qui est essentiel. Plusieurs dimensions – les objectifs, les activités, les rôles des acteurs, les usages des TIC – peuvent être ajustées ou négociées en même temps par les acteurs et former ainsi une configuration positive pour un apprenant à un moment donné dans un contexte donné. Un ajustement adéquat n'est pas donné d'emblée par les caractéristiques du scénario pédagogique, il n'existe qu'en contexte lorsqu'un apprenant tire parti des ressources offertes* » [Henri et al. 2007]. Ceci signifie la nécessité d'adapter en temps réel le scénario pédagogique selon le contexte courant, les ressources disponibles, etc. En outre, au moment de la description d'un scénario pédagogique, il est impossible d'identifier tous les paramètres qui entreront en jeu durant l'utilisation d'un EIAH. Cette utilisation dans des contextes réels d'enseignement n'est jamais rigide et linéaire, il est par conséquent très difficile de prévoir toutes les réactions possibles des apprenants et les différents événements qui peuvent émerger durant une session d'apprentissage [Zarraonandia et al. 2006a] [Yongwu 2005]. La question qui se pose n'est donc pas celle de la conception de « bons » ou de « mauvais » scénarios, mais celle de la manipulation par les enseignants de ces scénarios qui doivent s'adapter en cours d'utilisation dans le contexte effectif d'apprentissage.

Par ailleurs, des études en matière de conception ont montré que le concepteur ne peut compléter et préciser sa représentation mentale que progressivement [Bonnardel et Rech 1998]. En fait, le scénario réel d'apprentissage est unique et situé dans son propre contexte. En raison des chargements dynamiques et de spécificité de ce contexte, il est difficile donc de mener le scénario d'apprentissage exactement suivant le modèle de conception d'apprentissage prescrit, sans dévier [Yongwu 2005]. Cette déviation est gérée par l'enseignant usager qui joue le rôle de superviseur de la session d'apprentissage. Selon Dessus et K.Schneider : « *tout enseignant sait justement opérer des changements*

importants par rapport à ses intentions initiales, selon les imprévus survenant » [Dessus et K.Schneider 2006].

2.1.3.3 Quand et comment adapter un scénario pédagogique ?

L'adaptation d'un scénario pédagogique peut se faire lors de deux phases principales de son cycle de vie : soit pendant la conception (*at design-time*), soit pendant l'exécution (*at run-time*). Quant à l'adaptation pendant la conception, l'enseignant concepteur doit prendre en considération tous les cas possibles pour la mise en œuvre d'une situation d'apprentissage adaptative. La majorité des travaux [Burgos 2008] [Yongwu 2005] [Berlanga et Garcia 2005a] et [Towle et Halm 2005] sont basés sur IMS-LD (niveau B). Ces travaux essaient d'aider l'enseignant concepteur à définir des règles d'adaptation de type : « *if condition then action* », ceci en utilisant des *propriétés* et des *conditions*. Ces règles seront appliquées pendant l'exécution suivant les valeurs des propriétés qui permettent de préciser si les conditions sont satisfaites ou non.

Par contre, dans le cas d'adaptation pendant l'exécution, les enseignants et/ou le système doivent réagir selon les événements déclenchés par les apprenants en temps réel et tout au long du déroulement de leurs sessions d'apprentissage. Ce type de travaux consiste à orienter le déroulement des sessions implicitement par le système par exemple en se basant sur des agents logiciels [Van Rosmalen et al. 2006] ou explicitement par les enseignants qui peuvent entrer des ajustements nécessaires [Zarraonandia 2007]. De plus, Goodyear [Goodyear 1997] a affirmé qu'il est plausible que le système de livraison (artificiel ou humain) soit capable de prendre des décisions pédagogiques dynamiquement (en temps réel), en utilisant le produit de la conception pédagogique en tant que source des objectifs, des méthodes, des ressources et des contraintes.

Ainsi, nous nous focalisons dans cette section sur l'étude des phases et des approches possibles pour adapter des scénarios pédagogiques. Ces derniers peuvent être considérés comme des processus (le terme généralement utilisé en anglais est « *Learning Process* »). Les travaux effectués dans le domaine de la flexibilité des processus, notamment ceux de Schonenberg et al. [Schoenberg et al. 2007] [Schoenberg et al. 2008], nous semblent utiles pour mieux comprendre quand, comment et avec quelles approches, adapter des scénarios pédagogiques en fonction de l'évolution des circonstances des situations d'enseignement/apprentissage.

Schoenberg et al. ont en effet affirmé que dans les communautés qui travaillent sur les Workflows et les technologies liées aux processus métiers (*Business Process*), la « *flexibilité* » est considérée comme une qualité indispensable pour s'adapter au caractère

fluctuant des circonstances [Schonenberg et al. 2008]. Ces auteurs ont confirmé que la notion de flexibilité des processus est relativement simple à un niveau conceptuel mais difficile à traduire dans la pratique. Les auteurs affirment que les systèmes d'information à base de processus (PAISs : *Process-Aware Information Systems*) ne peuvent capturer qu'une abstraction du processus métier qu'ils supportent. Il existe en effet une différence entre la définition d'un processus et son déroulement dans « la vie réelle ». Cela exige une intervention humaine pour adapter le processus sur le système. En outre, dès qu'une différence est identifiée entre le processus prédictif et le processus réel, des moyens de minimisation de cet «écart» doivent être mis à disposition afin de s'assurer que le processus se déroule conformément aux attentes [Schonenberg et al. 2008].

Pour réaliser la flexibilité des processus, les auteurs proposent quatre approches. Nous les présentons ici (cf. figure 4).

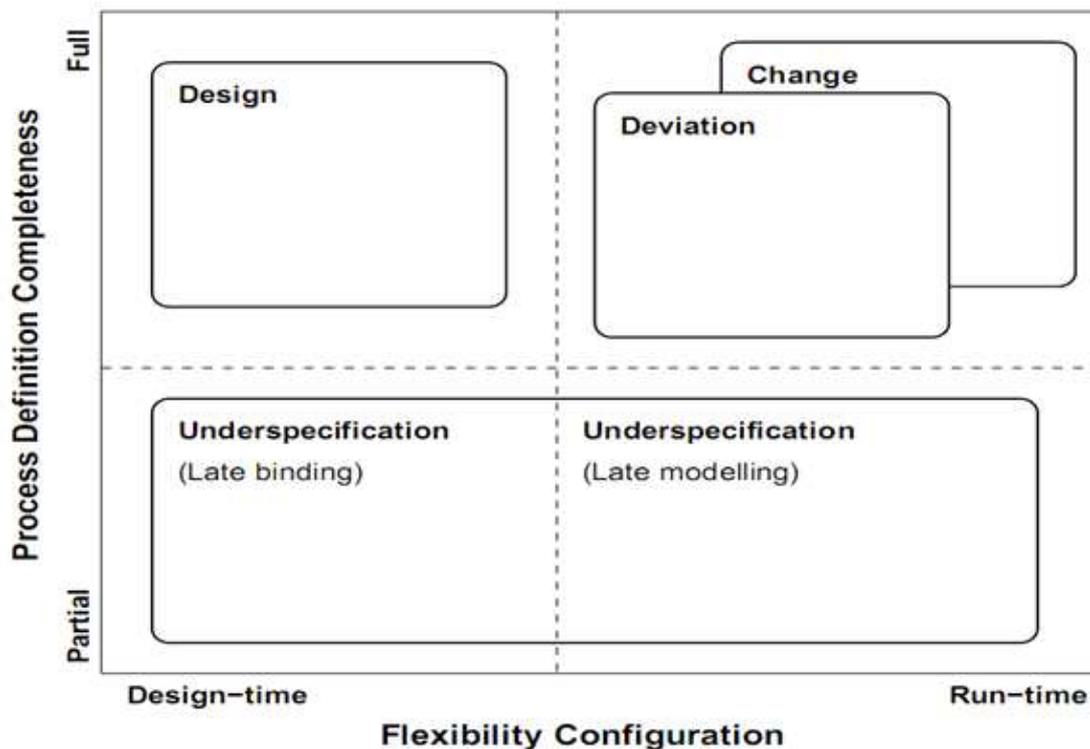


Figure 4 : Le spectre de la flexibilité d'un processus [Schonenberg et al. 2008].

2.1.3.3.1 Flexibilité par conception

Motivation d'utilisation. Dans le cas où un processus est exécuté dans des environnements variés, il est souhaitable d'incorporer les supports pour les différentes alternatives connues. Pendant l'exécution, le chemin le plus approprié peut être sélectionné parmi ceux qui ont été spécifiés pendant la conception initiale du processus.

Définition. La flexibilité par conception est la capacité d'incorporer les chemins alternatifs d'exécution dans une définition de processus pendant la phase de conception afin que la sélection du chemin d'exécution le plus approprié puisse être effectuée en temps réel pour chaque instance du processus.

Situation d'utilisation. Ce type de flexibilité peut être appliqué pour des processus pouvant avoir plus d'un chemin possible d'exécution.

Discussion. La spécification complète de tous les chemins possibles de l'exécution dans la définition d'un processus, pendant la phase de conception, peut être non souhaitable du point de vue de la complexité du modèle, ou même impossible en raison du nombre inconnu ou illimité de chemins possibles d'exécution.

2.1.3.3.2 Flexibilité par déviation

Motivation d'utilisation. Certaines instances du processus doivent être déviées temporairement par rapport à la séquence décrite dans la définition associée du processus, cela afin de s'adapter aux changements rencontrés dans l'environnement de l'exécution.

Définition. La flexibilité par déviation est la capacité pour une instance de processus de se dévier en temps réel du chemin d'exécution prescrit par le processus original sans altérer sa définition. La déviation ne peut qu'englober les modifications de la séquence des tâches d'exécution dans le modèle de processus pour une instance spécifique. Elle ne permet pas de modifications dans la définition du processus ou des tâches qui le composent.

Situation d'utilisation. Le concept de la déviation convient particulièrement à la spécification des définitions des processus qui sont destinés à guider les séquences possibles d'exécution plutôt que de restreindre les options qui sont disponibles (c'est-à-dire qu'elles sont de nature descriptive plutôt que prescriptive). Ces spécifications contiennent l'exécution souhaitée du processus. D'autres variantes sont aussi possibles.

Discussion. Des opérations de déviation peuvent être mises en œuvre de différentes manières. Néanmoins il devrait être possible d'identifier les déviations qui ont été faites pendant l'exécution du processus. En outre, des exigences supplémentaires peuvent être

imposées pour les opérateurs en termes, par exemple, d'annulation d'une tâche ou de création une instance supplémentaire pour une tâche.

2.1.3.3.3 Flexibilité par sous-spécification

Motivation d'utilisation. Lorsqu'une définition d'un processus est spécifiée, il pourrait être envisagé que lors de son exécution, des chemins d'exécution supplémentaires soient nécessaires. Ils doivent être traités conformément à la définition existante du processus. Dans le cas où tous les chemins d'exécution ne peuvent pas être définis en amont, il est utile de pouvoir exécuter une définition incomplète du processus quitte à ajouter dynamiquement des fragments de processus pour intégrer les alternatives manquantes.

Définition. La flexibilité par sous-spécification est la capacité d'exécuter incomplètement un processus. Ce type de flexibilité ne suppose pas de modifier en temps réel le modèle du processus, par contre, il nécessite de le compléter par la réalisation concrète de parties non encore définies, appelées « *point ouverts* ». Deux approches sont distinguées pour définir le contenu des points ouverts : (a) *liaison tardive (Late binding)* par sélection des fragments prédéfinis auparavant ; (b) *modélisation tardive (Late modelling)* par la construction de nouveaux fragments pour spécifier le contenu des points ouverts.

Situation d'utilisation. Le concept de sous-spécification convient principalement aux processus pour lesquels on sait clairement à l'avance à quels points spécifiques ils doivent être complétés. Cette approche de conception et de déploiement des processus est particulièrement utile lorsque des parties d'un processus sont conçues et contrôlées par différents groupes de travail. L'activité des concepteurs possède un certain degré de liberté tout en restant conforme au processus global.

Discussion. Les fragments de processus disponibles pour la spécification de contenu des points ouverts peuvent être stockés dans ce qu'on appelle un entrepôt. Ce dernier peut être utilisé pour la définition d'un ou de plusieurs processus ou juste pour la définition d'une tâche particulière ou d'un ensemble de tâches.

2.1.3.3.4 Flexibilité par changement

Motivation d'utilisation. Dans certains processus, des événements qui n'étaient pas prévus pendant la conception peuvent se produire pendant l'exécution. Ces événements ne peuvent parfois pas être traités par des déviations temporaires et nécessiteraient d'ajouter ou de supprimer des tâches ou des liens dans la définition du processus en se basant sur des règles permanentes. Cela peut requérir des changements au niveau du modèle du processus pour une ou plusieurs instances. Dans le cas où la mesure de changement est

plus importante, il peut être nécessaire de changer le modèle de processus pour toutes les instances en cours d'exécution comme pour celles qui seront exécutées ultérieurement. La définition d'un processus peut être changée par une extension ou une réduction de ses activités, ou aussi par une redéfinition des liens entre les activités.

Définition. Dans ce type de flexibilité, contrairement aux trois précédents, le modèle construit au moment de la conception doit être modifié pendant l'exécution. Une ou plusieurs instances devront par conséquent être transférées de l'ancien modèle vers le nouveau.

Situation d'utilisation. La flexibilité par le changement permet aux processus de s'adapter aux évolutions qui sont identifiées dans l'environnement de déroulement. Des changements peuvent être introduits au niveau du type de processus et/ou également au niveau de son instance.

Discussion. Weber et al. détaillent précisément les opérations de changement [Weber et al. 2007]. Ils proposent d'utiliser des patterns de changement de haut niveau plutôt que des primitives de changement de bas niveau et de donner des descriptions complètes des patterns identifiés.

2.2 Design Rationale

Nous présentons dans cette section l'approche de Design Rationale, sur laquelle nous nous sommes appuyés pour définir notre modèle rationnel des scénarios pédagogiques ouverts (cf. chapitre 4).

2.2.1 Qu'est-ce que le Design Rationale ?

D'après Lee [Lee 1997], une mémorisation du processus de conception et des décisions qui ont été prises est nécessaire pour être capable de comprendre, recréer et modifier une conception, mais cela n'est pas suffisant sans une compréhension de la logique de conception, le Design Rationale (DR). Le DR consiste à prendre en compte les raisons derrière une décision de conception, ainsi que sa justification, l'argumentation, les autres alternatives considérées, et les choix évalués qui ont conduit à la décision prise au terme du processus de conception de l'artefact [Hooey et Foyle 2007].

Sim et Duffy [Sim et Duffy 1994] définissait le DR comme le raisonnement et l'argumentation qui conduisent à la décision finale de la manière avec laquelle l'intention de conception est réalisée. L'intention de conception est l'effet ou le comportement attendu, qui est visé par le concepteur, que l'objet conçu doit atteindre pour remplir la fonction requise. L'objectif principal du DR est de soutenir les concepteurs en leur offrant des moyens pour mémoriser et communiquer l'argumentation et le raisonnement motivant le processus de conception [Horner et Atwood 2006]. McKerlie et MacLean affirmaient dans [McKerlie et MacLean 1993] que le DR consiste à travailler avec des représentations explicites, non seulement des solutions possibles de conception, mais aussi les raisons et les processus derrière ces solutions. En d'autres termes, la logique de conception explicite les décisions effectuées lors d'un processus de conception, et les raisons pour lesquelles ces décisions ont été prises [Jarczyk et al. 1992].

En outre, Lacaze [Lacaze 2004] affirmait que le DR agit comme une mémoire partagée du projet pour les différentes parties prenantes du projet. Il force les concepteurs à clarifier leurs décisions et à les argumenter suffisamment. Ceci a un impact direct sur la qualité future de l'artefact final [Newman et Marshall 1991]. Il permet effectivement de détecter les incompatibilités très tôt dans le processus de conception.

Ainsi, la conception rationalisée permet d'explorer de façon systématique les options de conception tout au long du processus de conception, de justifier les choix de conception et de garder la trace des décisions prises [Lacaze 2004], cela par la représentation des différentes solutions possibles dans un « *espace de conception* » [MacLean et al. 1991].

Pratiquement toutes les options possibles d'un problème donné (ou un objectif à atteindre) sont explorées comme le montre la figure 5. Notons que les méthodes de conception classiques ne permettent d'envisager qu'une solution (carré noir) à un problème donné (ou un objectif à atteindre) ; par contre, le DR permet d'évaluer les différentes solutions ou options (carrés gris), ce qui offre aux concepteurs la possibilité d'avoir un jugement qualitatif sur une solution [Lacaze 2005].

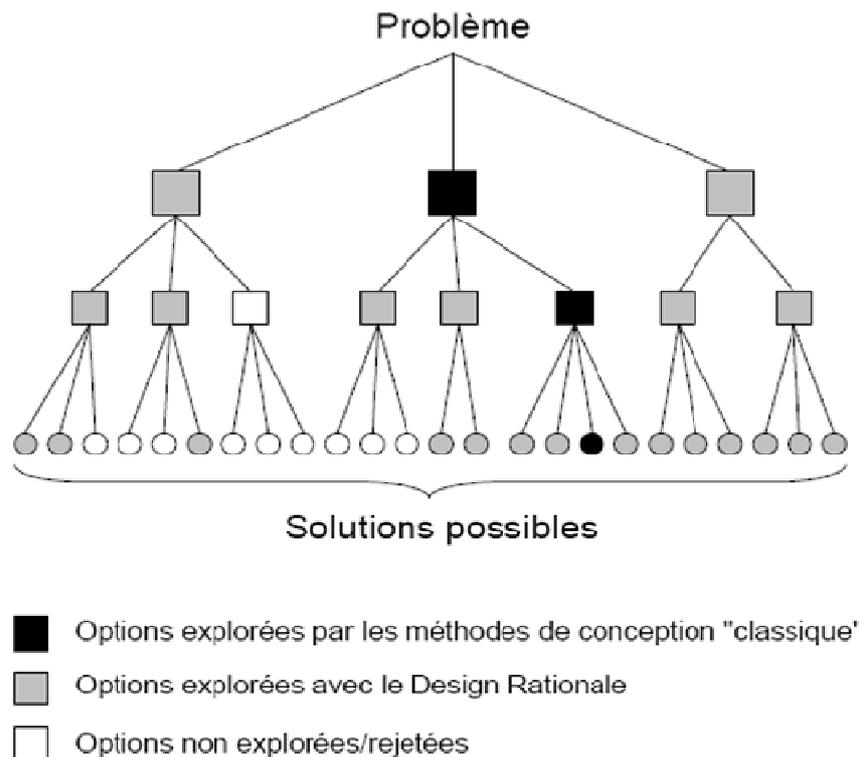


Figure 5 : Solutions envisagées pour un problème donné [Lacaze 2005].

Burge et Brown [Burge et Brown 1998] ont classé les DR en plusieurs types. Ces types ne sont pas mutuellement exclusifs ; certains systèmes peuvent prendre en compte plusieurs types à la fois. Les types de DR sont présentés ci-dessous [Burge et Brown 1998] :

- ❖ à *base d'argumentation*, le DR est principalement utilisé pour représenter les arguments qui définissent une conception [Garcia et al. 1993]. Ces arguments se constituent des questions soulevées, des réponses alternatives à ces questions, ainsi que des arguments pour et contre de chacune des alternatives.
- ❖ à *base d'historique*, le DR se constitue de la séquence d'événements qui s'est produite lors de la conception [Garcia, 1993]. Cette information pourrait être stockée de plusieurs façons telles que : entrées dans un ordinateur de conception, une

archive d'e-mails, ou d'autres types de documents contenant la trace capturée des actions effectuées au fil du temps.

- ❖ à *base de dispositif*, un modèle de dispositif lui-même est utilisé pour à la fois obtenir et présenter le DR [Gruber 1990]. Les explications de la conception sont produites en utilisant ce modèle afin de pouvoir simuler le comportement du dispositif. L'utilisateur pourra ainsi visualiser ce modèle et poser des questions sur sa conception et son comportement.
- ❖ à *base de processus*, la capture de DR est intégrée dans le processus de conception lui-même qui guide le format de raisonnement. Dans [Ganeshan et al. 1994], la description de la conception est modifiée uniquement par des changements et des améliorations des objectifs de conception, ainsi la capture du DR est effectuée dans le cadre du processus de conception.
- ❖ à *base de document actif*, le DR est pré-généré et stocké dans le système. Dans ce type de systèmes, le concepteur crée la conception et le système de DR génère sa logique en se basant sur les connaissances stockées du système. Pour chaque décision prise par l'utilisateur, le système la compare avec la décision qu'il aurait prise en fonction de ses connaissances. Si les actions de l'utilisateur ne correspondent pas aux recommandations du système, une option est proposée à l'utilisateur pour qu'il change sa décision ou modifie certains critères.

2.2.2 Apports du Design Rationale pour la conception

La logique de conception a été utilisée par les communautés de recherche de plusieurs domaines scientifiques tels que : génie logiciel, conception mécanique, intelligence artificielle, génie civil, gestion des connaissances, sciences cognitives, et interaction homme-machine.

Selon Burge et Brown [Burge et Brown 1998], le DR comprend beaucoup d'informations différentes, y compris l'historique du processus de conception et les raisons de chaque décision (alternative). Ces informations peuvent être utiles dans plusieurs aspects de la conception. Il s'agit notamment de la vérification de la conception, de son évaluation, de sa maintenance, de sa réutilisation, de son enseignement, de sa communication, de son assistance et enfin de sa documentation [Burge et Brown 1998].

- ❖ *Vérification de la conception* : l'objectif est d'utiliser la logique afin de vérifier que la conception répond aux exigences et à l'intention du concepteur. Cette vérification peut se produire à n'importe quel point dans le processus de conception.
- ❖ *Évaluation de la conception* : la logique est utilisée pour évaluer les conceptions (et des portions partielles des conceptions) et des choix de conception les uns par rapport aux autres.
- ❖ *Maintenance de la conception* : la logique de conception est utilisée pour déterminer quels choix ont été effectués lors du processus de conception afin de localiser les sources de problèmes de conception ou pour indiquer où des modifications doivent être apportées afin d'améliorer la conception. En gardant la trace des alternatives qui avaient été rejetées, le concepteur peut éviter de faire un choix qui a été rejeté auparavant.
- ❖ *Réutilisation de la conception* : la logique de conception est utilisée pour déterminer quelles parties de la conception peuvent être réutilisées, et dans certains cas, suggérer où et comment elles devraient être modifiées pour répondre à un nouvel ensemble d'exigences. Il est particulièrement important de laisser le concepteur savoir pourquoi les décisions ont été prises. Dans certains cas, ce qui peut sembler être une solution inefficace peut effectivement être critique pour la conception dans son ensemble. Sans présence de justification qui indique cela, la conception pourrait être modifiée d'une manière inadéquate.
- ❖ *Enseignement de la conception* : la logique de conception peut également être utile pour faciliter l'enseignement de nouveau personnel au sujet de la conception. Au delà de fournir un aperçu sur comment chaque choix de conception fonctionne, le DR montre pourquoi ce choix a été fait. Ceci véhicule plus d'informations qu'une description statique. Certains systèmes de logique de conception permettent à l'utilisateur de poser des questions sur la conception, ce qui est souvent une manière plus rapide et plus aisée pour apprendre à concevoir au lieu de se débattre avec de grandes quantités de documents de conception. Les supports d'apprentissage sont particulièrement importants lorsque les concepteurs experts ne sont pas disponibles pour enseigner.
- ❖ *Communication de la conception* : la présence de justification améliore la conception à la fois pendant et après le processus de design. En capturant les choix de conception et leurs raisons, ces informations peuvent être mises à la disposition d'autres personnes impliquées dans la conception afin de leur en

donner un aperçu et aussi de leur permettre d'apporter leur contribution dans le processus [Fischer et al. 1995]. Cela peut également fournir un moyen efficace pour détecter les conflits dans le travail de plusieurs concepteurs et de raccourcir le cycle de révision [Pena-Mora et al. 1995]. En capturant les raisons derrière les décisions de conception, la logique de conception peut être utilisée pour répondre aux questions des vérificateurs de la conception qui ont besoin de savoir pourquoi un choix particulier a été fait ou pourquoi un choix envisagé n'a pas été fait.

- ❖ *Assistance à la conception* : la logique de conception peut également fournir une aide durant le processus de conception. La capacité de vérifier et d'évaluer les choix de conception permet au concepteur de visualiser les résultats de leurs décisions de conception. La documentation de l'argumentation peut remplir plusieurs fonctions : elle clarifie les arguments en encourageant les concepteurs à documenter les informations et elle peut être évaluée pour s'assurer que tous les problèmes sont résolus et que des alternatives sont sélectionnées qui répondent aux exigences sans n'en ignorer aucune. Certaines caractéristiques qui offrent des bénéfices immédiats sont la vérification des contraintes ou de dépendance, où la logique est utilisée pour vérifier que la conception est correcte ; la simulation où le système permet au concepteur de vérifier l'impact des modifications de conception ; et l'atténuation des conflits, où le système cherche les violations des contraintes entre les différents concepteurs et les informe quand il y a un problème.
- ❖ *Documentation de la conception* : la logique de conception aide également à documenter la conception en offrant une image de l'historique de la conception et des raisons des choix de conception ainsi qu'un aperçu du produit final. Si la logique de conception est stockée sous une forme interprétable par l'ordinateur, elle peut être utilisée dans un système de génération de la documentation personnalisée en permettant de générer la documentation à partir de différentes perspectives, et dans certains systèmes, en permettant à l'utilisateur de poser des questions sur la conception. Certains systèmes utilisent la logique pour générer la documentation voulue par différents groupes de personnes.

2.2.3 Représentation du Design Rationale

Les modèles de représentation de la logique de conception doivent faciliter l'explicitation de la liste des solutions possibles (alternatives) avec leurs justifications en vue de résoudre efficacement un problème donné. Pour cela, la notation du DR doit être simple à comprendre et à utiliser. Plusieurs modèles de DR ont été proposés. Dans cette section,

nous citons rapidement quelques modèles de DR. Nous présentons par contre le modèle QOC en détail, car c'est le modèle que nous avons finalement retenu dans notre travail.

Bien que les structures d'argumentation sont issues des travaux de Stephen Toulmin dans les années cinquante [Toulmin 1958], l'origine de la logique de conception proprement dite peut être assimilée au développement de la notation argumentative IBIS (*Issue-Based Information System*) par Kunz et Rittel dans les années soixante-dix [Kunz et Rittel 1970]. IBIS est utilisé et développé ainsi par gIBIS (*graphical IBIS*) [Conklin et al. 1988] et itIBIS (*test-based IBIS*) [Conklin et Yakemovic 1991]. IBIS utilise un ensemble d'éléments de la logique (représentés comme des nœuds) tels que : les questions, les positions, les arguments, les résolutions et plusieurs types de relations pour lier les discussions à propos d'une question. Cependant, Selon Karsenty [Karsenty 2001], IBIS n'est pas destiné à faciliter la logique de conception mais plutôt à favoriser la coordination au sein d'un projet.

D'autres modèles sont apparus afin de réduire les difficultés d'IBIS. C'est le cas notamment de PHI (*Procedural Hierarchy of Issues*) [McCall 1987] et DRL (*Decision Representation Language*) [Lee 1991]. PHI étend IBIS aux questions non controversées et redéfinit les relations. PHI ajoute la relation sub-question pour signifier la dépendance entre la résolution de questions différentes. Quant au DRL, est un langage de représentation des décisions qui consiste à définir les principaux éléments comme les problèmes de décision, les alternatives, les objectifs, les exigences et les groupes. Il permet de découper la logique de conception à un niveau de précision plus fin. Il focalise davantage sur la représentation de la prise de décision et de sa justification au lieu de la logique de conception. Toutefois, DRL amène à représenter plus d'informations que nécessaire, ce qui rend les diagrammes rapidement illisibles [Lacaze 2004].

2.2.3.1 Le modèle QOC (*Questions, Options, and Criteria*)

Le modèle QOC (*Questions, Options, Criteria*), proposé par MacLean et al. [MacLean et al. 1991], est une notation semi-formelle qui permet la production d'une représentation graphique de la logique de conception. Selon MacLean et al. QOC peut être utilisé pour représenter « *l'espace de conception* » autour de l'artefact en cours de production, pour le situer dans un contexte plus large [McKerlie et MacLean 1993]. MacLean, Bellotti et Shum ont affirmé dans [MacLean, Bellotti et Shum 1993] qu'une caractéristique clé de cette approche est que le produit de la conception est conçu comme un espace de conception plutôt que comme un seul artefact. Cette approche contraste avec les méthodes traditionnelles de conception qui supposent que le produit final est une spécification ou un

artefact, incarnant certes les décisions des concepteurs, mais ne gardant que peu de traces des processus de raisonnement ayant conduits à sa réalisation.

Les diagrammes peuvent aider le concepteur à expliquer, à élaborer, à comparer et à examiner les idées de conception [McKerlie et MacLean 1993]. Comme le montre la figure 6, QOC permet de représenter le raisonnement de la conception comme un réseau de concepts principaux : « Questions », « Options », « Critères » [McKerlie et MacLean 1993]. Les questions identifient les problèmes clés dans l'espace de conception ; les options représentent des solutions possibles (alternatives) à ces questions ; et les critères décrivent de manière explicite les méthodes d'évaluation des options, telles que les exigences à satisfaire ou les propriétés souhaitées. Les options sont liées avec des critères positifs ou négatifs. Ces liens sont définis comme des évaluations. Graphiquement, une ligne continue entre un critère et une option signifie que le critère est favorable pour l'option, sinon il est défavorable. Le choix d'une option peut soulever un nouveau problème. Toujours sur le plan graphique, l'option qui est encadrée est celle qui a été retenue pour produire l'artefact final, c'est la décision de conception. De cette manière le concepteur pourra avoir une lecture visuelle de la logique de conception afin de comprendre les pour et les contre des différentes options [McKerlie et MacLean 1993].

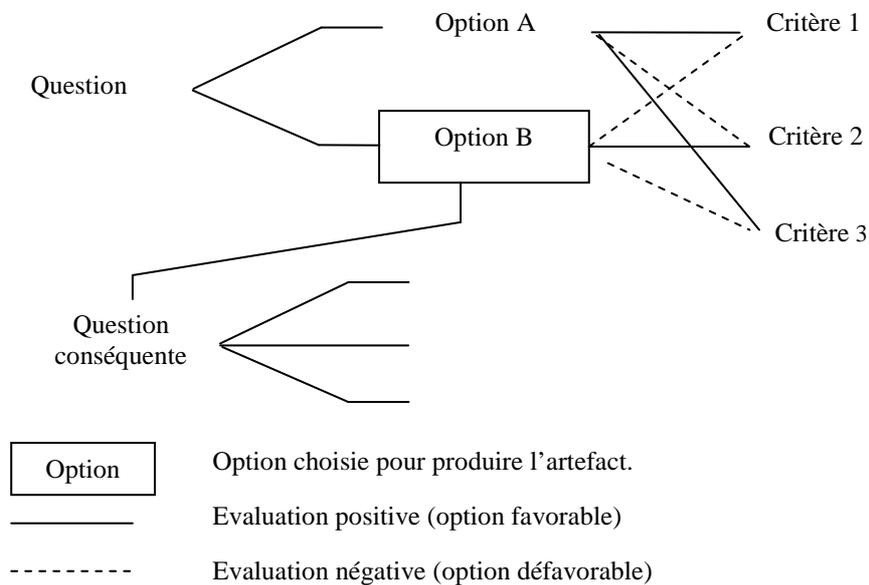


Figure 6 : La notation du modèle QOC (*Questions, Options, Criteria*).

Ainsi, cette représentation d'espace de conception fournit une justification succincte de l'artefact final issu de la conception, ceci en le plaçant dans un contexte plus large qui met en évidence sa différence et le pourquoi de sa conception. Une telle représentation est

IMS-LD est ainsi structurée autour de trois niveaux (A, B et C). Le niveau A représente le noyau de la spécification. Il fait référence à la description des composants qui configurent la spécification (Rôles, Activités et Environnement) et leur coordination via des éléments spécifiques (Method, Play, Act, Role-part). Les niveaux B et C de la spécification permettent de construire et d'établir des scénarios d'apprentissage adaptés ou personnalisés [Koper et Burgos 2005] [Tattersall et al. 2005] [Berlanga et Garcia 2005a] [Towle et Halm 2005]. Le niveau B complète le niveau A par des propriétés, des conditions et des règles qui agissent sur l'ensemble. À travers l'évaluation des expressions (si *Condition* alors *Action*), l'évolution du scénario d'apprentissage est prédéterminée. Ce qui permet de faire une personnalisation par rapport aux différents profils d'apprenants ou groupe d'apprenants. Le niveau C ajoute et fournit des notifications au niveau A (déclenchement des activités, par exemple l'envoi d'un mail au tuteur après la fin d'une activité d'apprenant).

Dans la littérature, on trouve plusieurs travaux s'appuyant sur IMS-LD pour développer des environnements d'apprentissage adaptables. Nous relevons ici les plus significatifs.

2.3.1.1.1 Approche à base d'exceptions en IMS-LD

Yongwu [Yongwu 2005] a constaté que certains processus d'apprentissage sont caractérisés par leurs incertitudes, qu'ils nécessitent des changements dynamiques et qu'il est impossible parfois de prescrire exactement les événements d'exécution. Pour cela, il a proposé une approche à base *d'exceptions* pour permettre aux enseignants de définir les règles de traitement des exceptions en utilisant le même formalisme de modélisation des scénarios d'apprentissage normaux, ceci pour assurer une progression lisse du processus d'apprentissage. Ce travail présente une approche pour gérer les exceptions dans une perspective de modélisation, c'est-à-dire dans la phase de conception (*at design time*). L'idée de base est de permettre aux concepteurs de modéliser, avec IMS-LD, des processus d'apprentissage dynamiques en anticipant des exceptions de genre prédictif. Pour traiter ces exceptions prédictives, le système proposé fournit des mécanismes en deux phases : la détection et la résolution.

En outre, d'après Yongwu, il est normal que la conception pédagogique comporte des définitions incomplètes ou inappropriées. Ceci implique la nécessité de mécanismes de traitement des exceptions. L'auteur a constaté que la communauté d'IMS-LD n'a pas donné une grande importance à ce point, et le player CopperCore ne fournit qu'un support limité pour traiter des exceptions et effectuer des modifications dynamiques du scénario d'apprentissage.

2.3.1.1.2 Conception de simples unités d'apprentissage adaptables avec IMS-LD

Towle et Halm [Towle et Halm 2005] présentent une analyse importante selon nous sur le potentiel d'IMS-LD à développer des unités d'apprentissage (UoLs) adaptées. De leur point de vue, l'adaptation est représentée entièrement dans l'UoL. Ils décrivent 4 champs possibles d'adaptation en IMS-LD : *environment*, *method*, *role* et *activity*. Ils présentent trois exemples de stratégies d'apprentissage adaptatif qui peuvent être modélisées par IMS-LD : *synchronous vs. asynchronous interactions*; *rule-example vs. example-rule*; et *variations in encouragement (feedback adaptation)*.

Towle et Halme mentionnent trois limites d'IMS-LD pour l'adaptation : *multiple rule interactions* ; *lack of enforced ordering* ; et *manifest-centred vs. server-centred*. Cette dernière signifie que le schéma de LD est « centré manifeste », cela signifie que toutes les informations nécessaires pour interagir avec l'unité d'apprentissage sont à l'intérieur du manifeste. Les problèmes de cette représentation selon les auteurs sont :

1. La difficulté engendrée par l'interaction entre plusieurs règles (impossibilité de considérer toutes les règles possibles en même temps par l'enseignant concepteur) ;
2. L'impossibilité de modifier l'unité d'apprentissage après la génération du manifeste ;
3. La redondance de stratégies d'adaptation (les mêmes stratégies peuvent être intégrées dans plusieurs manifestes) ;
4. Les connaissances concernant les objets d'apprentissage sont souvent intégrées dans le manifeste et non accessibles par des métadonnées pour l'utilisation dans de nouvelles stratégies.

Les auteurs proposent une solution pour dépasser ces problèmes qui consiste à utiliser un schéma « centré serveur » au lieu d'utiliser un schéma « centré manifeste » qui force l'adaptation statique. Ceci est possible si l'on extrait la logique d'adaptation du manifeste et qu'on utilise un *player LD* comme un client (ou agent) qui informe le serveur de ce que l'apprenant a fait. Le serveur, ensuite, envoie au client l'ID de l'activité suivante la plus appropriée.

2.3.1.1.3 Le projet aLFanet

Le projet aLFanet [Van Rosmalen et al. 2004] [Alfanet-Tool 2005] (*Active Learning For adaptive interNET 2002-2005*) est l'un des projets qui se focalisent principalement sur l'adaptation intelligente des environnements d'apprentissage basés sur des standards (IMS-LD, IMS-CP, IEEE-LOM, IMS-LIP, IMS-QTI), des systèmes multi-agents et des composants open source (dotLRN, CopperCore, Jade), ceci en décrivant l'adaptation dans tout le cycle de vie du processus d'apprentissage [Van Rosmalen et al. 2006]. L'approche est illustrée dans la figure 8. Elle représente « le cadre ouvert » pour aLFanet [Van Rosmalen et Boticario 2005].

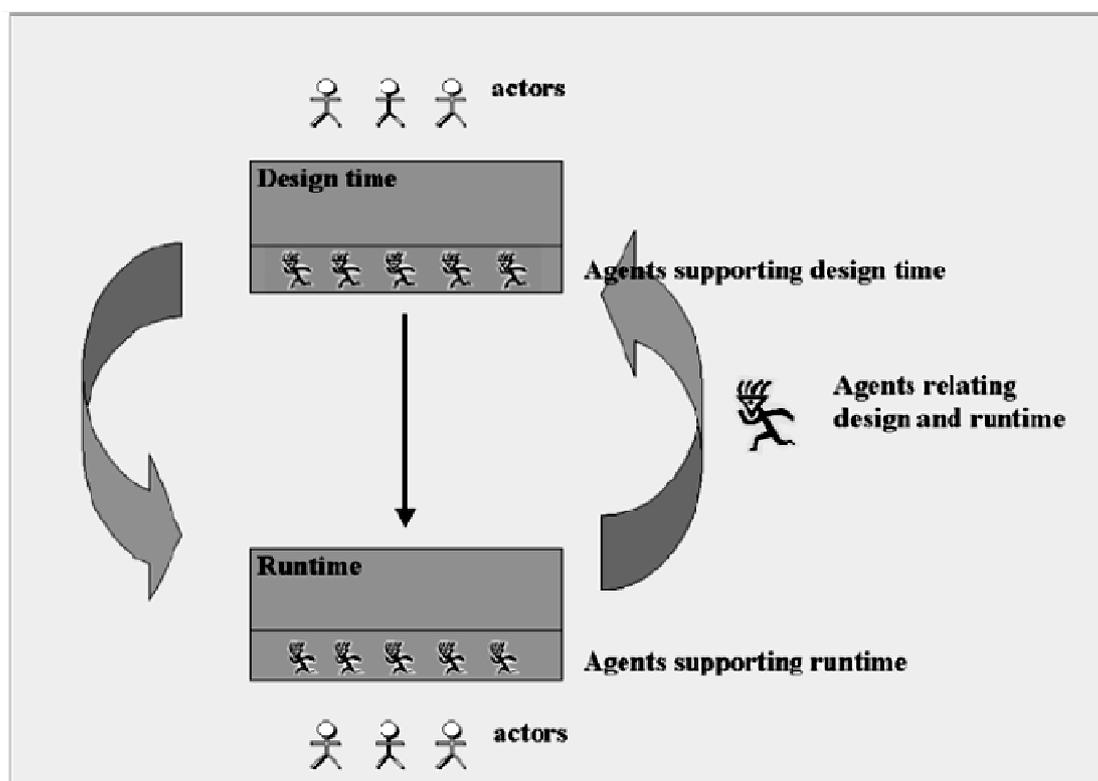


Figure 8 : Liaison d'adaptation en conception et l'adaptation en exécution [Van Rosmalen 2008].

Le projet Alfanet s'intéresse au problème de l'adaptation des scénarios pédagogiques, il s'agit de fournir des réponses différentes en fonction des caractéristiques des utilisateurs. Trois types d'adaptation aux apprenants et aux enseignants sont prévus [Alonso 2005] :

- *Adaptation « by instructional design »*, dont le but est de fournir des contenus, des activités et des services différents à l'apprenant selon ce que l'auteur du cours a spécifié à la conception. À l'exécution, le système vérifie si le modèle de l'utilisateur coïncide avec les spécifications de l'auteur et décide des cours, activités et services à fournir ;

- *Adaptation de l'interaction*, dont le but est de fournir une assistance à l'apprenant pendant l'interaction en se basant sur les techniques de l'intelligence artificielle. Cette assistance s'adresse aux auteurs et aux apprenants et comprend les services à utiliser, les contenus et les activités. Cela est fourni à l'utilisateur en prenant en compte l'information stockée dans les différents modèles (modèle de l'utilisateur, du groupe et des services) ;
- *Adaptation de la présentation*, qui consiste à présenter une interface utilisateur différente à chaque apprenant, selon son modèle. L'adaptation ne se fait pas seulement en fonction de ce que l'utilisateur a personnellement défini mais aussi en fonction de ce que le système a appris des précédentes interactions de cet utilisateur afin de lui proposer la présentation la plus efficace pour l'apprentissage.

Dans Alfabet, l'adaptation est ainsi spécifiée dès la conception grâce à IMS-LD par l'établissement d'un ensemble de règles à vérifier [Van Rosmalen et al. 2004]. Il ne s'agit pas de « scénarisation dynamique ». En outre, un éditeur de scénarios dépendant fortement du standard IMS-LD, appelé Alfabet-LD editor, a été développé. L'utilisation de cet éditeur exige une bonne connaissance d'IMS-LD.

2.3.1.1.3.1 Travaux de Van Rosmalen

Parmi les travaux contribuant significativement au projet aLFanet, on trouve les travaux de thèse de Van Rosmalen [Van Rosmalen 2008]. L'auteur a défini et implémenté deux modèles pour répondre à des exigences différentes. Le premier modèle porte sur la nécessité de soutenir les concepteurs dans leur tâche de design, ceci en se basant sur IMS-LD, pour formaliser la description des unités d'apprentissage afin de permettre la communication entre les différents acteurs (enseignant, tuteur et agent) [Van Rosmalen et Boticario 2005]. Le second est destiné à aider les tuteurs à la supervision et à l'orientation des apprenants dans leurs activités d'apprentissage pendant l'exécution. Plus précisément, le premier modèle facilite la tâche de conception en offrant un ensemble d'outils de conception et d'exécution qui sont liés pour permettre et faciliter un apprentissage adapté. Le second modèle allège la mission de tutorat en invoquant les connaissances et les compétences des apprenants suivis.

2.3.1.1.3.2 Travaux de Santos

L'approche proposée par O. Santos [Santos et al. 2005] se base également sur les standards pour adapter les environnements d'apprentissage. L'auteure a commencé ses travaux à partir du projet aLFanet et s'est intéressé dans un second temps à la question de l'acceptabilité [Santos, Boticario, Campo et Saneiro 2007] [Santos et al. 2007] [Santos et

[Boticario 2007](#)]. Pour dépasser les problèmes d'utilisation des standards (imposés) pour élaborer des scénarios adaptés pour les plateformes existantes, Santos et al. ont proposé une approche (ADAPTAPlan) pour réduire l'effort des enseignants en conception en les assistant dynamiquement pour générer automatiquement des design IMS-LD. Ceci est possible par le recueil des éléments du scénario pédagogique (tels que l'objectif pédagogique et les activités d'apprentissage avec les objets et les services nécessaires) qui nécessite l'expérience et l'expertise des enseignants. Selon les auteurs, cette approche permet l'utilisation de n'importe quel éditeur comme RELOAD [[Reload 2004](#)], CopperAuthor [[Vogten et Martens 2005](#)], ALFanet Authoring Tool [[Alfanet-Tool 2005](#)], etc., ou des outils plus spécifiques tels que COLLAGE [[Hernández-Leo et al. 2006](#)].

Santos et Boticario [[Santos et Boticario 2007](#)] affirment que l'adaptation du processus de fourniture des services selon les besoins, les préférences et les compétences des utilisateurs nécessite de: 1) définir les préférences d'interaction ; 2) comprendre le profil des utilisateurs grâce à des données collectées sur leurs interactions avec la plateforme ; 3) prendre en compte les équipements utilisés et le contexte en cours ; 4) supporter le scénario pédagogique pendant son exécution par les systèmes de recommandation ; 5) supporter des orientations psychologiques; 6) capitaliser des interactions pour ajuster des services. Pour répondre à toutes ces perspectives, les auteurs ont essayé d'enrichir le module d'adaptation de aLFanet pour sélectionner les contenus et les services appropriés et ajuster l'interface à l'utilisateur pour satisfaire les besoins et les préférences de l'apprenant. Ce module est intégré dans le cadre de OpenACS/dotLRN grâce à des technologies de Web services. Les auteurs ont également réalisé une extension de la spécification IMS-LD prenant en compte le support des processus psychologiques qui doivent être fournis aux étudiants handicapés dans les universités [[Santos, Boticario, Campo et Saneiro 2007](#)].

2.3.1.1.4 Approche mixant IMS-LD et les systèmes hypermédias adaptatifs

Berlanga [[Berlanga 2006](#)] a combiné des techniques des systèmes hypermédias adaptatifs (SHA) et des éléments d'IMS-LD, pour spécifier les unités d'apprentissage adaptables (*Adaptive Learning Design* (ADL)), ceci en permettant la réutilisation et l'échange des éléments d'IMS-LD et des règles d'adaptation entre les différents SHA. Ces règles sont définies en se basant sur les connaissances et le style d'apprentissage de l'apprenant ainsi que sur le style d'apprentissage de l'activité.

D'après Berlanga, ce couplage entre IMS-LD et SHA est rendu nécessaire du fait des lacunes des SHA. En les critiquant, il fait le constat que leur utilisation dans des classes réelles ou des environnements d'apprentissage à distance est rare. Parmi les raisons qu'il invoque : leur coût élevé de production, la complexité de leur développement, l'utilisation de

méthodes exclusives pour spécifier l'adaptation et certains éléments pédagogiques, leur manque d'interopérabilité entre les cours et les applications ainsi que l'absence de preuves crédibles d'avantages [Murray 2004].

Les limites des SHA sont, selon Berlanga et Garcia [Berlanga et García 2005a] :

- qu'il n'y a pas de prise en compte de l'*adaptation de la stratégie pédagogique* ;
- qu'il y a exclusion de l'*enseignant* dans la création et la conception des mécanismes d'adaptation ;
- qu'il n'y a pas d'*éditeurs* permettant aux enseignants de définir des opérations d'adaptation qui pourraient être nécessaires ;
- qu'il y a absence de mécanismes ou des possibilités pour *réutiliser* ou échanger des éléments pédagogiques, des contenus, des stratégies et des règles d'adaptation, etc.

La communauté des SHA a donc constaté le besoin d'un « *langage commun* » pour faciliter la « *compréhension* » de la sémantique des règles d'adaptation et des ressources pédagogiques.

Berlanga et García [Berlanga et García 2005b] ont distingué trois types de réutilisabilité des unités d'apprentissage adaptables (ALD) :

- ALD comme un *template*, où un ALD « vide » est fourni pour qu'il soit accompli (rempli) par des éléments désirés (ressources, propriétés, activités, conditions, etc.).
- ALD réutilisable, où un ALD est modifié selon les nouvelles configurations ou contextes ;
- Éléments de ALD réutilisables, où des composants spécifiques d'un ALD sont échangés avec d'autres ALDs.

Le but de cette approche s'appuyant sur la spécification IMS-LD est de surmonter les lacunes des SHA. Cependant, l'utilisation d'IMS-LD dans les SHA pour définir des unités d'apprentissage adaptables est restrictive. En plus d'imposer à l'enseignant (non expert) un langage et un environnement différents de ceux de son propre domaine métier, la conception des scénarios adaptables, plus complexes, n'est pas facile à faire.

Certains considèrent qu'IMS-LD ne permet pas de spécifier des adaptations complexes [Towle et Halm 2005].

2.3.1.1.5 Extension de la spécification IMS-LD pour permettre l'adaptation et l'intégration des unités d'apprentissage

Burgos [Burgos 2008] a analysé comment des méthodes classiques en apprentissage adapté (*adaptive learning*) peuvent être effectuées en utilisant IMS-LD.

Sur les huit types d'adaptation présentés dans (cf. section 2.1.1), Burgos a distingué trois niveaux en fonction de la possibilité de leur prise en charge :

- a) le flux d'apprentissage, le contenu d'apprentissage, l'évaluation et le support interactif de résolution des problèmes sont bien supportés, même s'ils pourraient être améliorés avec des structures spécifiques axés sur l'adaptation (c'est-à-dire, la modification de produit de conception en temps réel) ;
- b) le regroupement d'utilisateurs, l'adaptation d'interface, l'évaluation adaptative et la modification complète du cours en temps réel sont partiellement pris en charge ;
- c) certains types d'adaptations restent sans support en IMS-LD, ce sont les modifications dynamiques de la structure et la méthode d'apprentissage en temps réel, ainsi que le filtrage et la recherche adaptatifs des informations.

Selon l'auteur, une partie de ces lacunes incombe à l'état actuel des outils supportant IMS-LD, et non pas à la spécification elle-même. Par exemple, l'adaptation de l'interface en temps réel dans des players d'IMS-LD (tels que CopperCore [Vogten et Martens 2005], Reload Player [Reload 2004]) est encore insuffisante. Selon [Koper et Burgos 2005] l'adaptation de flux d'apprentissage peut être effectuée grâce à des éléments du niveau B : *properties, calculations, global elements, conditions* et *monitoring services*. Le cycle de vie de chaque unité d'apprentissage est constitué de trois phases [Koper et Tattersall 2005] : conception, publication et exécution. Burgos [Burgos 2008] a mentionné qu'avec les outils actuels, une fois que l'unité d'apprentissage est publiée, il n'est pas possible de changer la structure, la méthode ou la définition des éléments de base (tels que : *conditions, properties, etc.*) au sens d'IMS-LD.

L'auteur conclut que la spécification IMS-LD a besoin d'être restructurée, de modifier certains éléments et d'en intégrer de nouveaux. Ces deux actions (modification et extension) sont la clé pour améliorer l'expressivité pédagogique et l'intégration avec d'autres spécifications et systèmes de *e-learning*. Elles visent deux objectifs clairs dans la définition de l'IMS-LD : la personnalisation des processus d'apprentissage et l'interopérabilité.

2.3.1.1.6 Adaptation des unités d'apprentissage en temps réel par l'introduction des petites actions adaptatives

Zarraonandia [Zarraonandia 2007] [Zarraonandia et al. 2007] [Zarraonandia et al. 2006a] [Zarraonandia et al. 2006b] a proposé une approche très voisine de notre objet de recherche. Il essaie de projeter la pratique réelle des enseignants classiques dans le domaine de l'enseignement via des dispositifs informatiques. Sa proposition vise à augmenter le degré de liberté d'utilisation offert à l'enseignant quand il informatise un scénario d'apprentissage et à lui permettre d'introduire des variations sur le scénario d'apprentissage en temps réel. En d'autres termes, dans le but de permettre la réussite de l'expérience des apprenants vis-à-vis de l'objectif d'apprentissage, le scénario pédagogique est raffiné à travers son usage par des modifications permanentes grâce à l'approche de « la modélisation tardive ».

L'auteur décrit le cycle de vie de la conception itérative des processus d'apprentissage adaptables et propose une architecture pour la mise en œuvre des étapes d'exécution des processus décrits grâce à la spécification IMS-LD [IMS-LD 2003]. Il constate que le déroulement d'un processus d'apprentissage est en pratique assez flexible, car il n'est pas possible de prévoir toutes les réactions possibles de l'apprenant. Pendant l'exécution du scénario d'apprentissage, certaines adaptations doivent être appliquées afin que le processus soit affiné dans son déroulement. L'expérience acquise lors des exécutions passées est intégrée dans la définition du processus. Comme le montre la figure 9, Zarraonandia a décrit le cycle de vie itératif d'un scénario de la manière suivante [Zarraonandia et al. 2007] [Zarraonandia et al. 2006a] :

1. Le processus démarre une fois qu'un modèle initial du cours est défini et que son exécution commence (*conception initiale*).
2. L'enseignant observe les interactions des apprenants et introduit ensuite les adaptations nécessaires (*la conception continue dans l'exécution par l'enseignant*).
3. La réussite des adaptations appliquées doit être évaluée, et après la fin du processus, l'achèvement de l'objectif d'apprentissage doit être mesuré (*évaluation de la pertinence des adaptations faites suivant l'objectif pédagogique global*).
4. En se basant sur ces informations, la nouvelle version du scénario d'apprentissage doit être générée en intégrant les modifications réussies à la version initiale (*c'est-à-dire la réingénierie du scénario en intégrant les bonnes adaptations effectuées, qui sont considérées comme des fonctions constituées en genèse instrumentale*).
5. Cette nouvelle version passe par le même cycle dans un déroulement suivant jusqu'à ce qu'il n'y ait plus d'adaptation nécessaire (*raffinage du scénario jusqu'à sa stabilité*).

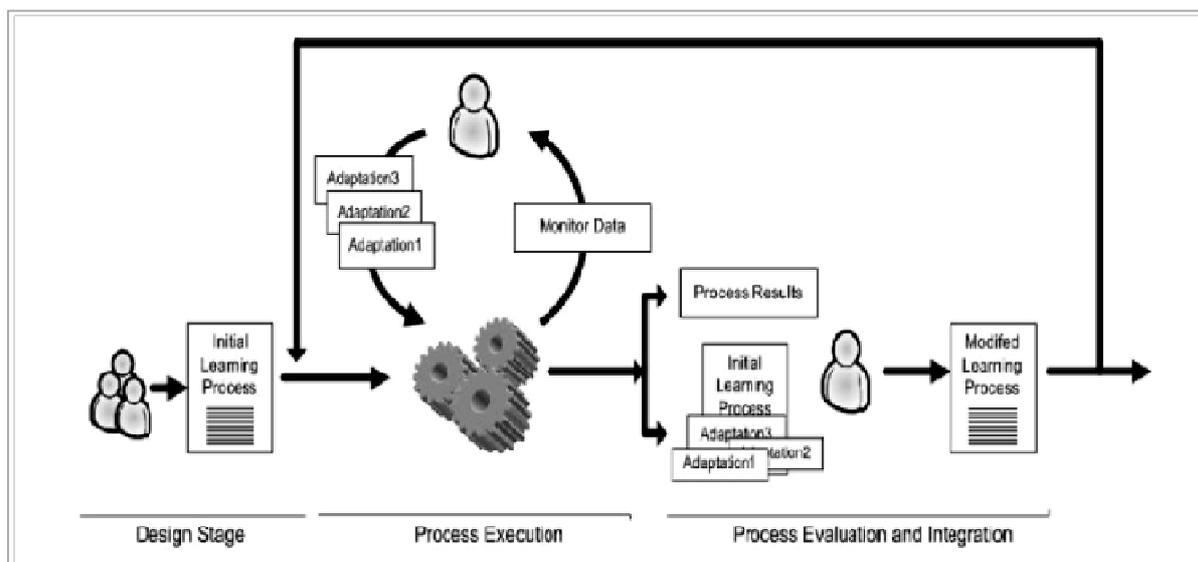


Figure 9 : Phases de conception itérative d'une unité d'apprentissage [Zarraonandia et al. 2007].

Comme technique d'adaptation, l'auteur introduit la notion de « *poke d'adaptation* » par la spécification, hors de la définition IMS-LD, de *petites actions d'adaptation* qui peuvent être appliquées, selon le contexte réel de l'exécution, sur un processus d'apprentissage préalablement défini. Les actions d'adaptation mises en place sont évaluées par rapport à leur objectif initial, en mesurant leur influence, et par conséquent, en donnant à l'enseignant la possibilité de les intégrer automatiquement dans le scénario original.

Pour l'implémentation de cette approche, une architecture d'un *LD player*, qui fournit les moyens d'exécution de la composition itérative du scénario spécifié avec IMS-LD, a été décrite. Le player est conçu comme une extension du moteur d'exécution CopperCore et est implémenté à l'aide d'un ensemble de patterns de conception et d'une approche de programmation orientée aspects. Ensuite, une fois l'unité d'apprentissage adaptée, elle doit être validée afin de garantir sa conformité avec la spécification IMS-LD. Pour cette raison, une ontologie est utilisée pour capturer la sémantique des éléments de la spécification du design pédagogique. L'observation de la progression du processus d'apprentissage est effectuée grâce au module « *ProgressWatcher* ». La figure 10 illustre la structure de *LD Player*.

L'approche proposée ici nous semble originale et pertinente. Elle décrit tout le cycle de vie d'une unité d'apprentissage adaptable, de manière très proche de la réalité des processus d'adaptation mis en œuvre en classe par un enseignant.

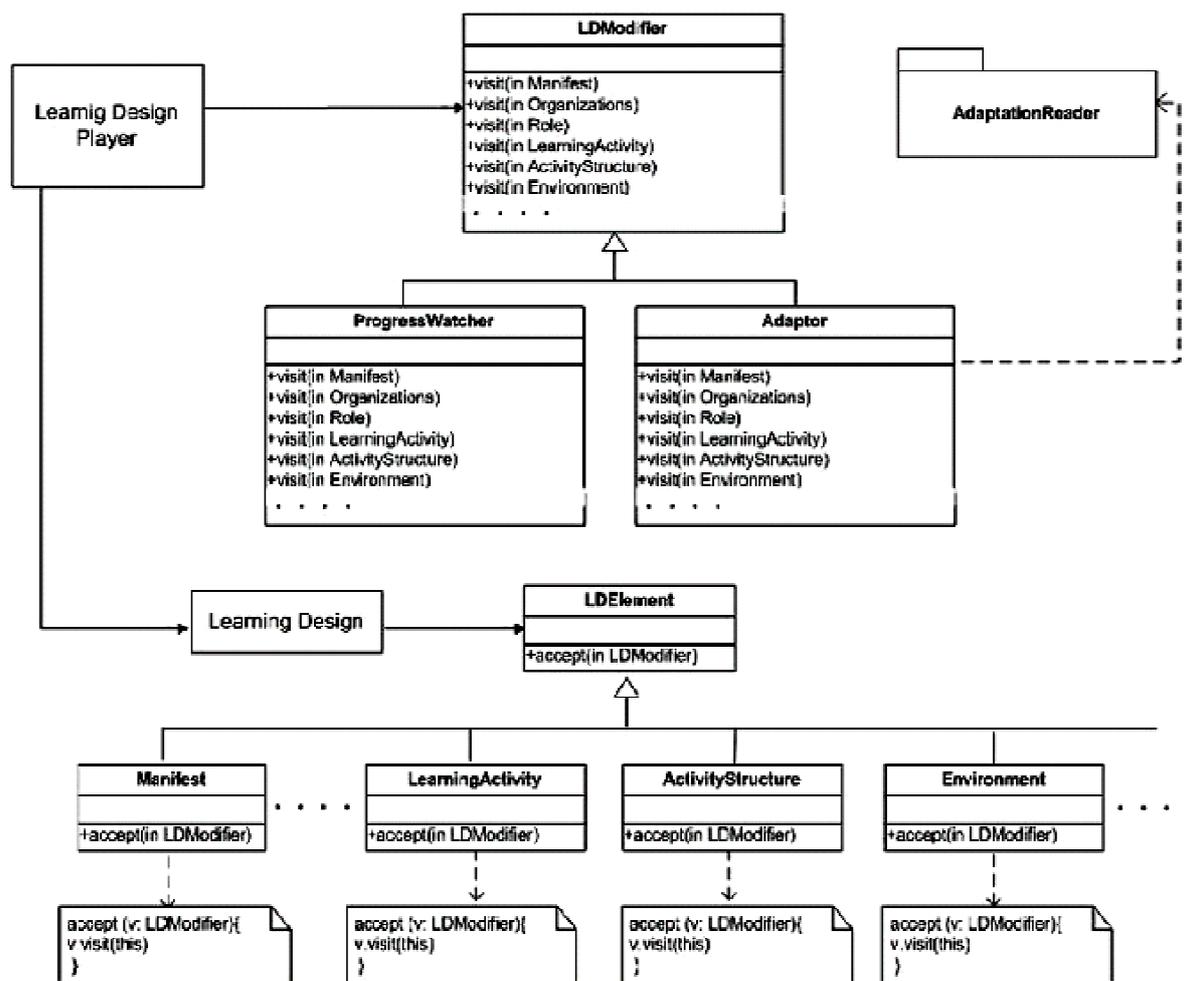


Figure 10 : La structure de LD Player [Zarraonandia et al. 2007].

2.3.1.2 Le langage de modélisation pédagogique LDL

LDL (*Learning Design Language*) [Ferraris et al. 2005] est défini pour répondre aux insuffisances du langage IMS-LD pour décrire des scénarios pédagogiques collaboratifs. Il est né du constat que les activités sont imprévisibles par nature, collaboratives et essentiellement fondées sur des interactions entre participants [Ferraris et Lejeune 2007]. LDL est basé sur une adaptation du Modèle de Participation [Martel 1998], dédié à la modélisation de ce que ses auteurs nomment l'espace de régulation d'une activité collective [Ferraris et al. 2005]. Le métamodèle de LDL, décrit par Martel et co-auteurs [Martel et al. 2007], repose sur les concepts d'enceinte, d'interaction et de position (cf. figure 11).

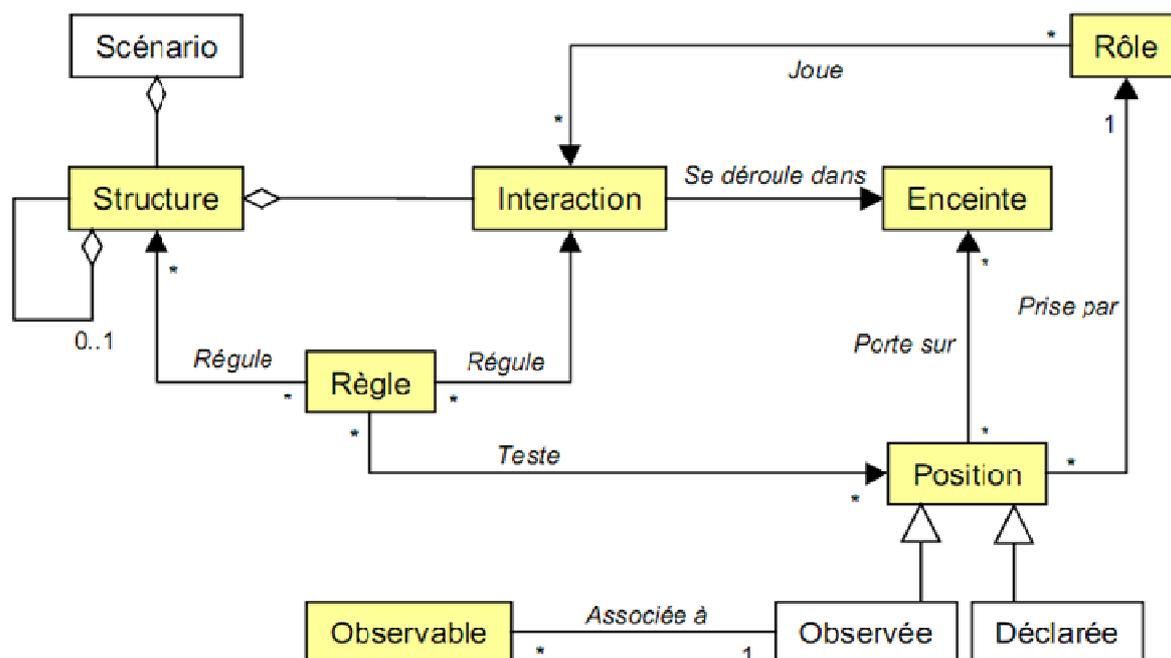


Figure 11 : Le métamodèle du langage LDL [Martel et al. 2007].

L'enceinte est un lieu virtuel dans lequel les acteurs vont pouvoir interagir et exprimer des positions. La spécification de positions permet de modéliser les dépendances entre les activités sans prendre en compte leur ordre. Ceci autorise la définition de scénarios adaptables et contrôlables par les participants ; les auteurs du langage ont affirmé [Martel et al. 2007] : « *introduire une certaine forme de malléabilité dans le déroulement des activités* ». De plus, il existe des positions particulières permettant l'observation de l'activité. Les observées peuvent alors être utilisées directement par le scénario (régulation), par les acteurs (par exemple par un tuteur pour superviser l'activité), ou par les concepteurs du scénario, afin d'en évaluer la qualité [Choquet 2007].

Toutefois, tel qu'il est observé dans [Choquet 2007], « le métamodèle de LDL repose lui aussi sur une métaphore imposée – la métaphore *spatiale* [Ferraris et al. 2007], certes beaucoup plus simple que celle d'IMS-LD, mais qui peut paraître tout aussi étrangère à des enseignants. Les auteurs du langage l'ont compris et se proposent [Ferraris et al. 2007] de développer un éditeur proposant des translations des concepts de LDL dans d'autres métaphores. Il nous semble que le problème de l'appropriation d'un langage de modélisation pédagogique par un concepteur ne repose pas uniquement sur l'adaptabilité de sa métaphore, mais bien plus sur la résonance que peut avoir sa syntaxe abstraite dans le métier et le contexte d'enseignement du concepteur : « *le processus de conception doit s'inscrire dans l'univers métier de l'enseignant plutôt que de proposer des passerelles terminologiques* » » [Choquet 2007].

2.3.1.3 Langage de modélisation pédagogique orienté perspectives (POEML)

Caeiro-Rodriguez et al. [Caeiro-Rodriguez et al. 2007] affirment que la flexibilité dans les unités pédagogiques représente une exigence clé afin de supporter les différents changements et alternatives. Les auteurs proposent un langage de modélisation pédagogique orienté perspectives (*Perspective-oriented Educational Modeling Language (PoEML)*). Ce langage est un EML organisé en plusieurs packages conformément à un métamodèle qui sépare la modélisation des unités d'apprentissage en fonction des perspectives et aspects identifiés. Son idée principale est que, au lieu de résoudre la modélisation du processus d'apprentissage dans son ensemble, il la décompose, le plus possible, en plusieurs parties indépendantes. Ensuite, chaque partie est modélisée séparément dans une démarche par étapes. En conséquence, selon les auteurs, la modélisation de l'ensemble est simplifiée et les modèles produits sont flexibles. La figure 12 illustre l'élément de scénario pédagogique et ses principales relations.

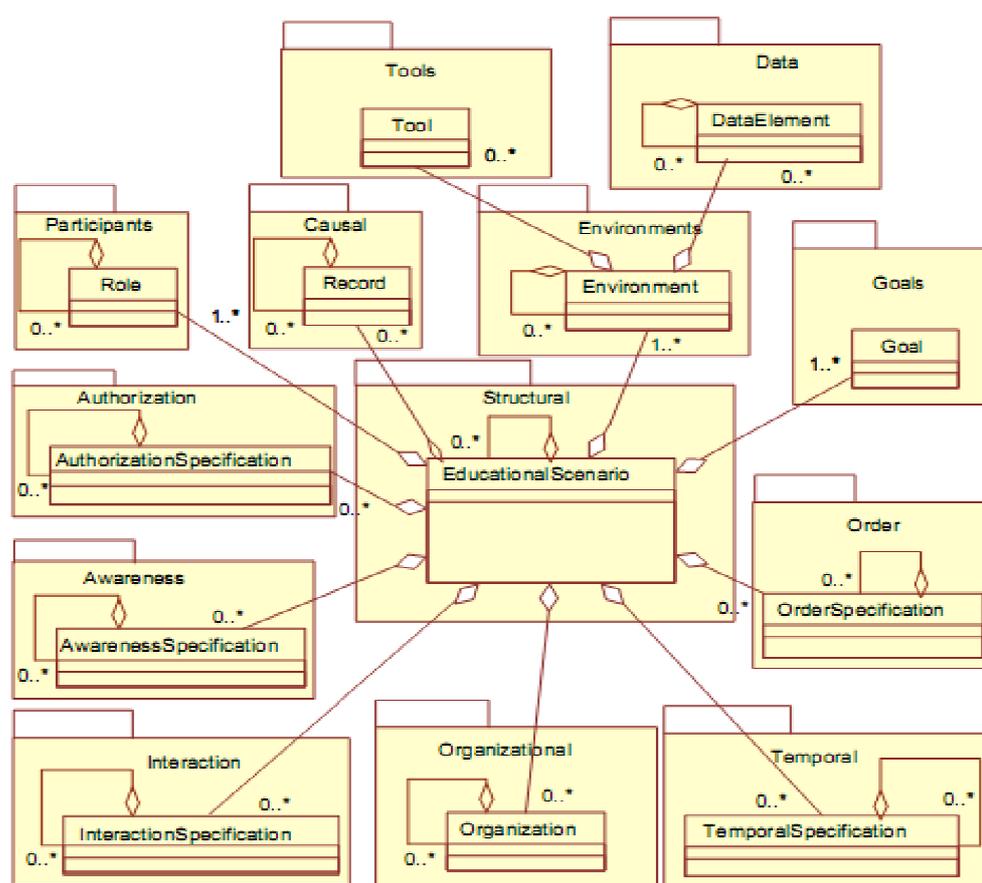


Figure 12 : Élément de scénario pédagogique et ses principaux composants [Caeiro-Rodriguez et al. 2007].

En insistant sur la nécessité de la flexibilité dans des unités pédagogiques, Caeiro-Rodriguez et al. se sont focalisés principalement sur les exigences suivantes : permettre la modélisation des différents alternatives pendant la phase de conception ; permettre la modélisation des nouveaux alternatives en temps réel ; et permettre le raffinement dynamique des modèles pendant l'exécution.

2.3.2 Approches par l'ingénierie dirigée par les modèles (IDM)

1.3.2.1 Présentation de l'IDM

Avant de développer les travaux basés sur l'IDM nous présentons en premier lieu les principes de cette approche du développement informatique : modélisation et métamodélisation, transformation des modèles, modélisation spécifique au domaine (DSM pour *Domain-specific Modeling*), etc. L'outillage d'Eclipse est également présenté : EMF (*Eclipse Modeling Framework*), GMF (*Graphical Modeling Framework*). Cette présentation nous semble importante car c'est l'univers méthodologique et technologique dans le quel nous avons inscrit nos travaux.

L'ingénierie dirigée par les modèles (IDM), en anglais *Model Driven Engineering* (MDE), est une approche spécifique du génie logiciel qui place le modèle au centre du processus de conception. Elle a pour objectif de définir un cadre théorique pour générer du code en utilisant des combinaisons et des transformations successives de modèles [Caron 2007a]. Ce courant s'est développé suite à la définition de l'approche MDA (*Model Driven Architecture*) fin 2000 par OMG (*Object Management Group*) [OMG 2006], en vue de fournir des solutions aux problèmes liés à l'émergence continue des technologies logicielles qui oblige les entreprises à migrer⁵ leurs systèmes logiciels à chaque émergence d'une technologie [Laforcade 2010]. Le but principal de MDA est de séparer les parties métiers de leur mise en œuvre technologique et ainsi de garantir l'interopérabilité des modèles fonctionnels pour différents choix d'implémentation. L'IDM est une approche plus globale et générale que MDA. Cette dernière concerne seulement les technologies OMG principalement, par contre, l'IDM consiste à appliquer les mêmes principes de MDA à tout espace technologique et à les généraliser. En fait, MDA est un processus de type IDM [Laforcade 2007a].

2.3.2.1.1 Principes d'IDM

Le premier principe est d'utiliser la modélisation et des modèles pour développer des systèmes logiciels. Le deuxième est la séparation entre les fonctionnalités d'un système

⁵ Une **migration**, en informatique, est le passage d'un état existant d'un système d'information ou d'une application vers une cible définie dans un projet ou un programme [Wikipédia, consultation en 2012].

d'information d'une entreprise et la mise en œuvre de ces fonctionnalités sur des plateformes technologiques spécifiques (EJB, CORBA, etc.). La spécification abstraite du système est devenu le principal atout dans le développement des logiciels : de nombreuses implémentations utilisant des technologies concrètes peuvent être dérivées de la même spécification abstraite. Cette approche est dite dirigée par les modèles, car elle fournit un moyen facilitant l'utilisation des modèles pour guider la compréhension, la conception, la construction, le déploiement, l'exploitation, la maintenance et la modification d'un système informatique [OMG 2006].

Le processus de développement des logiciels est basé sur des transformations automatiques ou semi-automatiques entre les modèles, à partir des modèles abstraits, centrés domaine métier et généralement informels (CIM : *Computation Independent Model*) vers des modèles spécifiques à une plateforme (PSM : *Platform Specific Model*).

En effet, l'IDM est séduisante pour approcher l'ingénierie des EIAH pour deux raisons principales [Choquet 2007] :

- ❖ elle préconise le développement de modèles productifs, ce qui aide le concepteur à maîtriser les choix de développement et d'implantation,
- ❖ elle permet de s'inscrire explicitement dans l'univers métier de l'application cible en définissant des langages spécifiques à son domaine.

L'IDM repose sur les principes suivants [Laforcade 2007b] :

i/ Capitalisation : parmi ses avantages on trouve la possibilité de réutilisation et de capitalisation des modèles et des pratiques (la transformation et les règles de transcription entre les modèles).

ii/ Abstraction : les modèles doivent être indépendants des technologies de mise en œuvre afin d'adapter la logique métier à différents contextes techniques et de permettre de faire plus facilement évoluer les applications vers de nouvelles technologies.

iii/ Modélisation : la modélisation est abordée selon une vision productive (pour générer le code final du logiciel pour une technologie de mise en œuvre donnée) par opposition à la traditionnelle vision contemplative (but de documentation, spécification, communication).

iv/ Séparation des préoccupations : l'IDM s'illustre généralement selon les deux principales préoccupations du métier et de la plateforme de mise en œuvre mais d'autres préoccupations sont possibles.

2.3.2.1.2 Les niveaux d'abstraction dans IDM

L'IDM est fondée sur une architecture à quatre niveaux [Bezivin et Blanc 2002], initialement proposée dans le cadre du MDA, de concret à abstrait : système (monde réel), modèle, métamodèle et méta-méta-modèle. La figure 13 ci-dessous illustre les niveaux d'abstraction d'IDM.

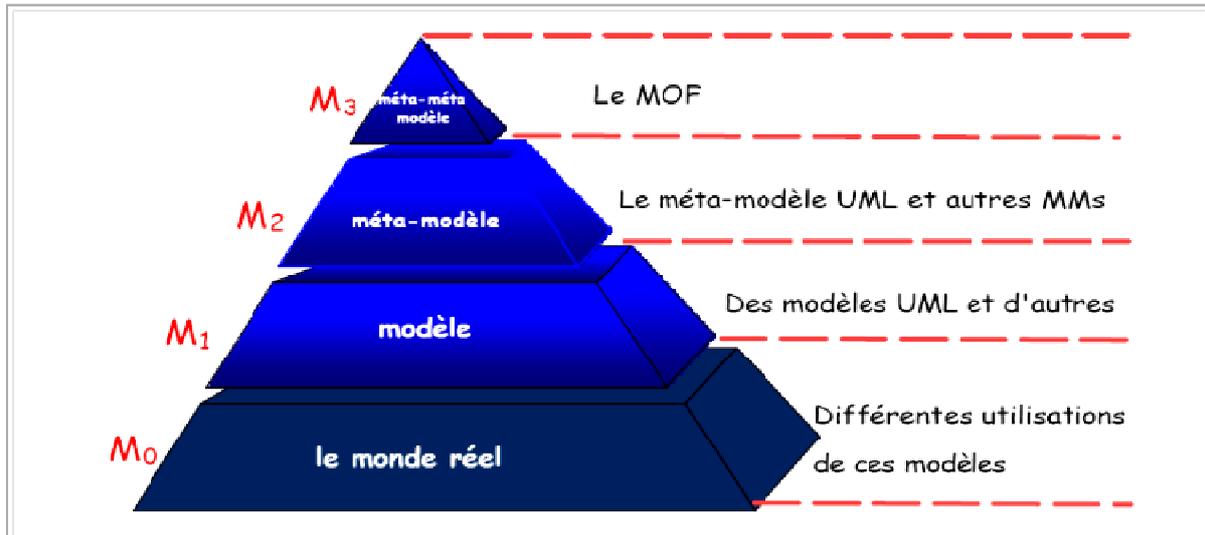


Figure 13 : Architecture à quatre niveaux [Bezivin et Blanc 2002].

- **M3** est le niveau supérieur : le méta-méta-modèle. Il donne les notions de base permettant l'expression des métamodèles (M2), et des modèles (M1). Il permet d'exprimer les règles de conformités qui lient les entités du niveau M1 à celles du niveau M2. M3 se définit lui-même [Caron 2007a].
- **M2** définit le métamodèle. MDA repose par exemple sur un métamodèle principal UML extensible par des profils. M2 permet de manipuler des modèles pour en préciser le sens et les rendre opérationnalisables, cela en définissant des règles de transformation, de fusion, de composition, ou de tissage des modèles [Caron 2007a].
- Le niveau **M1** définit les modèles. Chaque modèle doit être conforme à son métamodèle selon des règles de conformité définies au niveau du méta-méta-modèle. Un modèle doit être validé formellement pour favoriser son opérationnalisation. C'est cette validation qui distingue l'IDM des méthodes de modélisations antérieures.
- Le niveau **M0** représente le monde réel, c'est le système à modéliser.

Par conséquent, les modèles sont des concepts clés de l'IDM, utilisés tout au long du processus d'ingénierie. L'IDM offre la capacité de « projeter » la connaissance métier exprimée dans les modèles abstraits vers ceux qui sont concrets et dépendants d'une plateforme cible, cela en proposant une automatisation du processus d'ingénierie, par des transformations et des affinements successifs des modèles, jusqu'au code. L'IDM a pour objectif la production et la manipulation de modèles productifs, qui peuvent être opérationnalisés et manipulés par la machine, c'est-à-dire qui peuvent être utilisés pour générer – plus ou moins automatiquement – tout ou partie du code de l'application qu'ils représentent. Ces modèles doivent être *conformes* à des métamodèles explicites et formels et *représenter* sans ambiguïté un point de vue sur l'artefact à produire. La relation de « *conforme-à* » signifie que le modèle est défini formellement par son métamodèle à l'aide du langage associé, et que ce métamodèle est lui-même explicite, donc défini formellement par un méta-métamodèle. Ce même méta-métamodèle définit alors les règles de consistance et de cohérence de la relation « *conforme-à* » liant un modèle à son métamodèle. La relation de « *représente* » consiste à faire une simplification [Choquet 2007] [Caron 2007a].

2.3.2.1.3 Les types de modèles

Rappelons que l'IDM consiste à effectuer les différentes transformations des modèles en vue de générer tout ou une partie du code automatiquement. Elle définit trois niveaux de modèles (qui sont initialement proposés par MDA) :

- **CIM** (*Computer Independent Model*) : ce modèle comporte les besoins des utilisateurs recensés à partir du domaine métier ;
- **PIM** (*Platform Independent Model*) : ce modèle spécifie la partie métier d'une application sans utiliser le langage technique ;
- **PSM** (*Platform Specific Model*) : ce modèle spécifie l'application sur une plate-forme technologique donnée.

EI-Kechaï [EI-Kechaï 2008] relève que certains travaux de la communauté EIAH ont repris cette approche dans le cadre de la scénarisation pédagogique. Le cycle des transformations appliquées aux EIAH est illustré dans la figure 14.

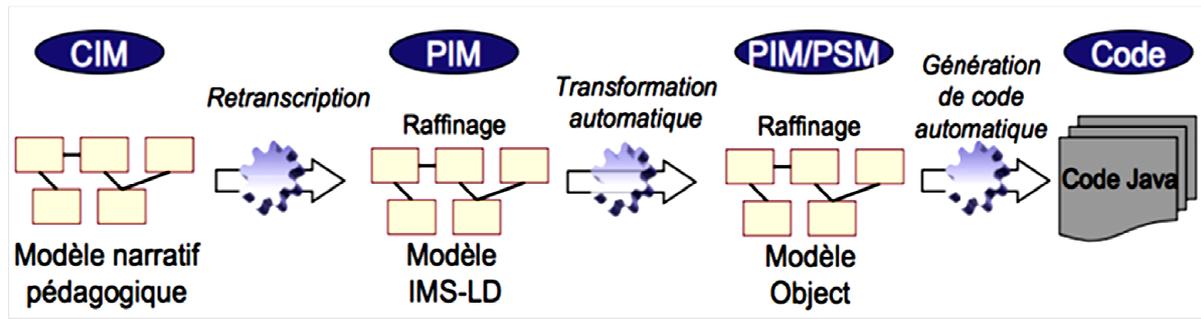


Figure 14 : Cycle de transformations et EIAH [Nodenot et al. 2005].

Toutefois, l'IDM est considérée comme trop simpliste [Favre et al. 2006]. Il est par exemple assez naïf de supposer que le métamodèle UML, généraliste, soit le meilleur support à la définition des PIM. C'est pourquoi certains [Bhanot et al. 2005] [Tolvanen 2006], toujours dans une approche IDM, préfèrent définir leurs modèles à l'aide de langages spécifiques à un domaine (DSML – *Domain Specific Modeling Language*) reposant sur un métamodèle plus simple mais plus ciblé sur le métier des experts du domaine. Notons qu'un domaine est défini par [Czarnecki et Eisenecker 2000] comme un espace de connaissance : précisé, afin de maximiser la satisfaction des besoins de ses acteurs, incluant un ensemble de concepts et une terminologie compris par les praticiens dans cet espace.

2.3.2.2 Modélisation spécifique au domaine : DSM

2.3.2.2.1 Qu'est-ce que le DSM ?

La modélisation spécifique au domaine, en anglais « *Domain Specific Modeling (DSM)* » est une approche, toujours dans le paradigme de l'IDM, qui est selon [Le Gac 2007], « *pragmatique beaucoup plus simple à comprendre et à mettre en œuvre que le MDA académique* ». DSM a été définie principalement pour les raisons suivantes :

- ✓ réduire la complexité de transformation des modèles et des pertes sémantiques qu'elle génère [Kelly et Tolvanen 2008],
- ✓ élever le niveau d'abstraction plus haut que la programmation afin de permettre la spécification des solutions dans un langage qui utilise directement les concepts et les règles du domaine spécifique du problème [Kelly et Tolvanen 2008]. En conséquence, « *les experts du domaine eux-mêmes peuvent comprendre, valider modifier, et souvent même développer des programmes en langage dédié* » [Jézéquel et al. 2012],

- ✓ « les langages dédiés améliorent la qualité, la productivité, la fiabilité, la maintenabilité, la portabilité et les possibilités de réutilisation » [Jézéquel et al. 2012],
- ✓ un langage dédié permet la communication entre le concepteur et la machine, et aussi entre les concepteurs eux-mêmes dans une perspective d'échange entre les membres d'une communauté de pratique.

Contrairement aux langages génériques comme UML, le principe de DSM est de développer un langage spécifique au domaine métier, un DSML⁶ adapté pour spécifier des artefacts qui instrumentent une activité spécifique dans un contexte (ou domaine) spécifique. Les DSMLs sont généralement de petite taille et doivent être facilement manipulables, transformables, combinables, etc. [Jézéquel et al. 2012]. Un DSML doit être formel, mais son métamodèle doit refléter le domaine des utilisateurs : le vocabulaire de modélisation utilisé est celui de leur propre domaine métier, dans lequel ils sont experts. De cette manière, les praticiens peuvent comprendre, valider et élaborer des DSML par le biais des concepts et des règles qui leur sont familières [Miao et al. 2008]. Il est souvent possible de valider et d'optimiser dans un univers métier plutôt que dans l'univers des langages génériques, où certains détails supplémentaires peuvent masquer des éléments importants [Deursen et al. 2000].

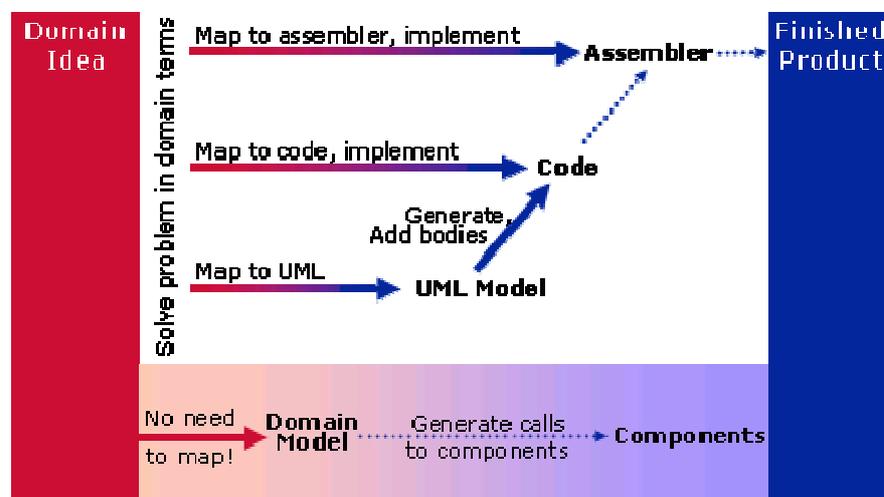


Figure 15 : Le DSM élève le niveau d'abstraction par rapport aux autres approches⁷.

Ainsi, des générateurs de code peuvent être développés pour transformer « directement » les modèles exprimés avec le DSML, sous un format interprété par l'humain, en un format interprété par la machine, spécifique à la plate-forme technologique cible. Comme le montre

⁶ DSML: *Domain Specific Modeling Language*.

⁷ <http://www.dsmforum.org/why.html>, dernière visite : Avril 2012.

la figure 15, les produits finaux peuvent être générés directement à partir des idées du domaine modélisées grâce au DSML à un niveau plus abstrait que ceux des autres approches traditionnelles : UML, Code, Assembleur. L'utilisation de ces derniers par des praticiens introduit de la complexité. Elle nécessite à chaque fois des retranscriptions des spécifications de l'univers métier vers le code, ce qui peut générer d'importantes pertes sémantiques et qui peut pénaliser la productivité et la qualité des produits finaux.

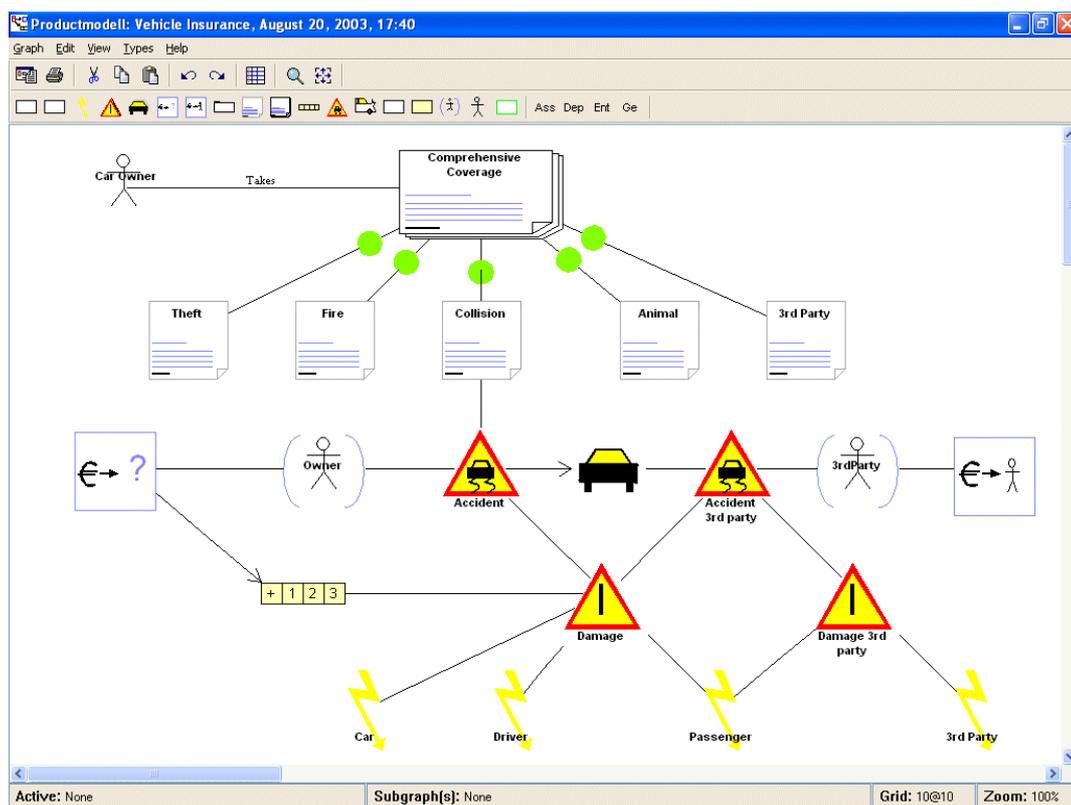


Figure 16 : Exemple de modélisation des produits financiers et d'assurance pour une application web J2EE avec un langage DSM basé sur des concepts d'experts de domaine [Tolvanen 2006].

La figure 16 montre un exemple d'un langage basé sur des concepts des experts du domaine des assurances [Tolvanen 2006]. Pour ce langage particulier, les concepts de modélisation sont liés à des produits financiers et d'assurance. Des concepts tels que « Risque », « Bonus », et « Dommage » capturent les faits pertinents dans le domaine des assurances. En utilisant ce langage, un expert en assurances, et donc a priori non-programmeur, spécifie des modèles pour définir les différents produits d'assurance. Des générateurs doivent effectuer soigneusement des transformations de ces conceptions vers le code d'une application web J2EE. De cette façon, le programmeur expert en Java peut construire un processus de *mapping* à partir du niveau de DSML vers le code en une seule fois, et cela sans que chacun n'ait à se préoccuper du domaine d'expertise de l'autre. On

peut dire que les modèles, utilisant des concepts des experts du domaine, sont au niveau le plus élevé d'abstraction et que la sortie générée pourra plus facilement migrer dans un autre langage d'implémentation [Tolvanen 2006].

2.3.2.2.1 Comment définir un DSML ?

Un DSL (*Domain-Specific Language*), au sens large de terme, peut être construit en utilisant un métamodèle ou en utilisant une grammaire [Langlois et al. 2007]. Généralement, les langages graphiques (DSML) sont décrits en utilisant des métamodèles et les langages textuels sont définis par leur grammaire [Feilkas, 2006]. Dans notre contexte d'étude, nous voulons définir un langage graphique, la méta-modélisation est donc utilisée. J-M Jézéquel la définit comme : « *l'activité consistant à définir le métamodèle d'un langage de modélisation. Elle vise donc à modéliser un langage, qui joue le rôle de système à modéliser* » [Jézéquel et al. 2012]. Un métamodèle est un modèle de modèles. AI-Watch définit un modèle comme : « *une abstraction qui permet de réduire la complexité en se focalisant sur certains aspects, de façon indépendante des aspects liés à l'implantation. Il est utilisé pour représenter et structurer des connaissances du domaine relevant de la pratique des concepteurs ainsi que de leurs relations. Il constitue donc le savoir acquis par l'étude ou l'expérience* » [AI-Watch 1995].

Un langage de modélisation se constitue d'une syntaxe abstraite, une syntaxe concrète et une sémantique de domaine.

- ❖ **La syntaxe abstraite** définit la structure conceptuelle du langage en spécifiant ses concepts, leurs relations et leurs propriétés, ainsi que les règles et les contraintes. Cette syntaxe peut être décrite, sous forme d'un métamodèle, par un langage de méta-modélisation. Plusieurs langages de méta-modélisation sont utilisés dans les différents environnements de DSM : Ecore dans EMF [Steinberg et al. 2008], MetaGME dans GME [Ledeczi et al. 2001], KM3 dans AMMA [Jouault et Bézivin 2006], Xcore dans XMF-Mosaic [Clark et al. 2004]. Tous ces langages reposent sur le méta-métamodèle MOF (*MetaObject Facility*) proposé par l'OMG [OMG 2006]. De plus, il existe des langages, comme OCL (*Object Constraint Language*) [OMG 2010], qui sont proposés pour enrichir la sémantique du métamodèle en définissant un ensemble de règles qui devront être respectés au niveau des modèles.
- ❖ **La syntaxe concrète** spécifie des symboles pour représenter des concepts abstraits, pour but de définir une notation qui peut être interprétée par l'humain et/ou la machine. La figure 17 [De Moura Filho 2007] illustre un exemple de trois syntaxes concrètes

différentes qui correspondent à la même syntaxe abstraite : (a) diagramme UML ; (b) notation arborescente ; et (c) une notation textuelle (XML).

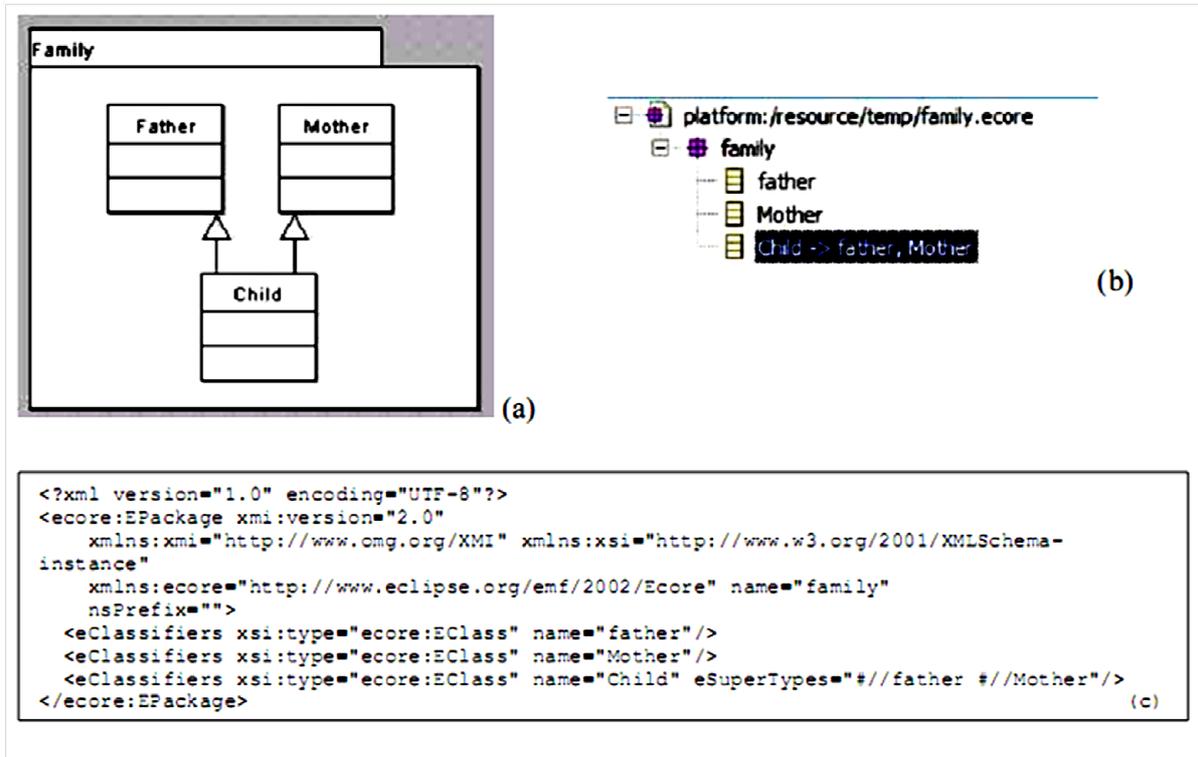


Figure 17 : Exemples des syntaxes concrètes : (a) par UML, (b) par arborescence et (c) textuelle (XML) [De Moura Filho 2007].

Selon Kelly et Tolvanen [Kelly et Tolvanen 2008], les formes de représentation, utilisées dans la plupart des DSML, sont graphiques et combinées avec le texte. Le choix d'une notation pour un DSML est typiquement en fonction de la représentation réelle des concepts du domaine.

❖ **La sémantique** d'un DSML consiste à préciser la signification non ambiguë de ses éléments. La définition d'une signification peut être effectuée par l'ajout d'un élément dans un modèle, ou par la création d'une liaison entre deux éléments [Kelly et Tolvanen 2008]. Le rapport entre la sémantique et les deux syntaxes est similaire à celui qui est entre le fond et la forme. Une sémantique est considéré comme formelle lorsqu'elle est exprimée dans un formalisme mathématique, et elle permet de vérifier la cohérence et la complétude du DSML [Jézéquel et al. 2012].

J-P Tolvanen [Tolvanen 2006] propose certaines étapes à suivre pour définir un DSML. Il affirme que la meilleure façon pour construire des langages de modélisation est de suivre un processus incrémental, ç-à-dire construire une petite partie du langage ; modéliser un petit

peu avec ; définir des générateurs ; apporter quelques modifications au langage ; modéliser un peu plus avec ; ainsi de suite, jusqu'à atteindre une stabilisation et une bonne satisfaction des futurs usagers de DSML.

L'auteur a divisé le processus en quatre phases, qui peuvent changer selon le contexte et les besoins :

- 1) identification des abstractions du domaine et leur fonctionnement ensemble ;
- 2) spécification des concepts du langage et leurs règles (définition de métamodèle) ;
- 3) création de la représentation visuelle du langage (choix de la notation) ;
- 4) définition des générateurs pour la vérification de modèle, de code et de documentation.

2.3.2.2.2 L'outillage d'Eclipse Modeling Project

Plusieurs environnements supportant le DSM existent actuellement, citons parmi ceux-ci les produits commerciaux : MetaEdit+ de MetaCase [Kelly et al. 1996], DSL Tools de Microsoft (basés sur des Software Factories©) [MS DSLT 2010], ainsi que les projets académiques tels que Topcased [Farail et al. 2006], AMMA-ATL [Jouault et al. 2006] et TIGER [Ehrig et al. 2005] [TIGER 2009], ou les projets open source comme le projet de modélisation d'Eclipse [EMP 2011] qui proposent deux Frameworks EMF/GMF.

Tous ces environnements proposent des outils supportant des techniques de méta-modélisation capables d'exprimer des vocabulaires spécifiques au domaine (syntaxes abstraites), et offrent des facilités pour construire des notations (syntaxes concrètes). Ces outils peuvent générer des éditeurs puissants et conviviaux qui sont dédiés pour les DSML, de manière à permettre aux concepteurs de spécifier visuellement les modèles à un haut niveau d'abstraction, et ainsi d'assurer la persistance des ces modèles afin de faciliter leur chargement et leur stockage dans un format interprété par la machine. Ce format est souvent indépendant de la notation de représentation utilisée. Dans la majorité des cas, cette notation est graphique [Laforcade 2010].

Dans notre travail nous avons choisi d'utiliser le projet open source d'Eclipse, « *Eclipse Modeling Project (EMP)* », qui se focalise sur l'évolution et la promotion des technologies de développement dirigé par les modèles au sein de la communauté Eclipse. Cela en fournissant un ensemble unifié de Frameworks et d'outils de modélisation, et aussi d'implémentations des standards [EMP 2011].

2.3.2.2.1 EMF (Eclipse Modeling Framework)

EMF⁸ est un Framework qui permet de définir des métamodèles décrivant le métier d'un domaine spécifique, et de générer ainsi le code des éditeurs qui permettent de créer des modèles conformes à ces métamodèles. Ces métamodèles sont, quant à eux, conformes à un méta-métamodèle conforme au méta-métamodèle MOF (*MetaObject Facility*) proposé par l'OMG [OMG 2006]. Ce méta-métamodèle, fournit par EMF, est appelé « Ecore ». La figure 18 montre une version simplifiée d'Ecore.

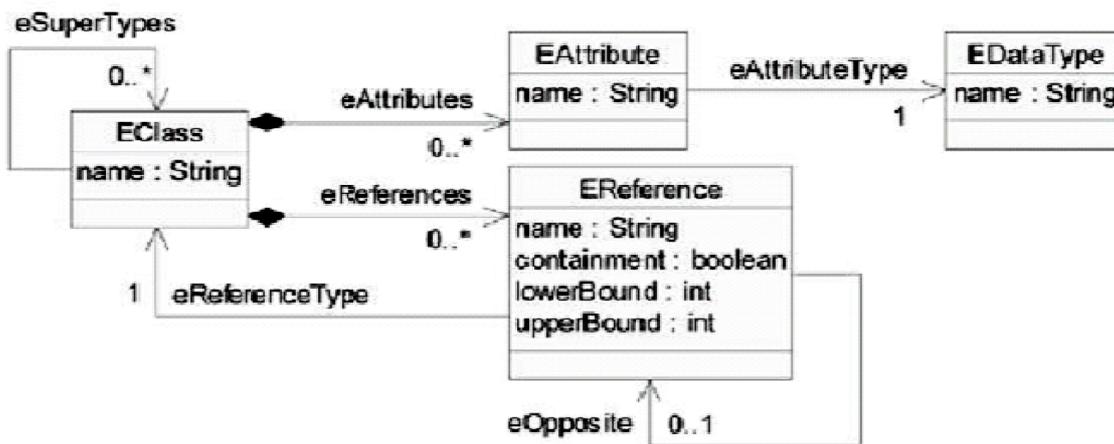


Figure 18 : Vue d'Ecore simplifiée en quatre classes [Vanworkmhoudt 2004].

Avant d'importer les métamodèles par EMF, et éventuellement les transformer en un format conforme à Ecore, ils peuvent être définis initialement en utilisant les formats d'entrée standards suivants [Miloud, 2010] : des interfaces Java annotées ; des modèles UML ; des schémas XML (XSD) ; des fichiers XMI (*XML Metadata Interchange*) ; ou des modèles Ecore définis par l'éditeur interne d'EMF.

En outre, comme le montre la figure 19, le processus EMF implique les composants principaux suivants [Miloud, 2010] :

- ❖ **emf.core** : se constitue de deux éléments, le métamodèle Ecore, qui est utilisé pour définir des modèles EMF (*.ecore), ainsi que le support d'exécution qui assure la persistance de ces métamodèles et la notification en cas de changement. Il fournit une API réflexive.
- ❖ **emf.codegen** : est le générateur de code Java de l'éditeur basique comportant une interface arborescente. Ce code généré est divisé en trois niveaux de plug-ins suivants.

⁸ <http://www.eclipse.org/emf/>, dernière consultation: 07/04/2012

- ❖ **emf.model** : comporte une implémentation Java qui est étroitement alignée avec le modèle Ecore. Il contient une API orientée-objet qui permet la persistance et la manipulation des objets définis dans le métamodèle métier⁹.
- ❖ **emf.edit** : comprend un ensemble de classes génériques et réutilisables pour la construction des éditeurs à partir des métamodèles EMF [Bézivin et al. 2005]. Ce composant fournit des classes permettant l'affichage et la modification des instances de métamodèles dans des éditeurs basiques comportant une interface arborescente. Ces éditeurs peuvent être perfectionnés par des spécialistes en programmation, en améliorant le code généré par l'ajout des fonctionnalités nécessaires.
- ❖ **emf.editor** : contient le code de l'interface de l'éditeur destinée à l'utilisateur.

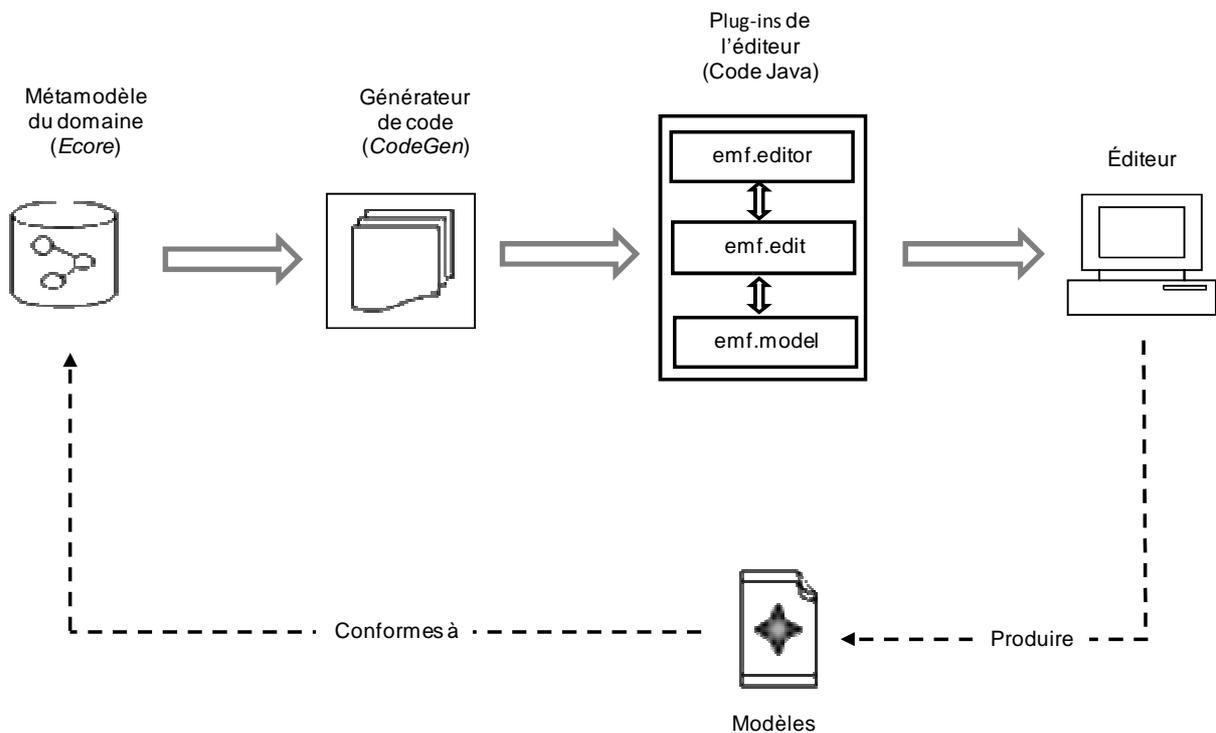


Figure 19 : Processus d'EMF pour développer un éditeur à partir d'un métamodèle du domaine.

⁹ <http://petergraff.sys-con.com/node/46848>

2.3.2.2.2.2 GMF (Graphical Modeling Framework)

GMF¹⁰ a été introduit comme une passerelle générative entre GEF (*Graphical Editing Framework*)¹¹ et EMF, dans le but de simplifier la création des éditeurs graphiques par GEF en se basant sur des modèles de domaine définis par EMF. GEF offre le support graphique nécessaire pour la construction d'un éditeur de diagrammes qui remplace la couche générée par EMF qui reste limitée graphiquement [Kelly 2004].

GMF se constitue de deux composants principaux : un outillage et une infrastructure d'exécution. Quant à l'outillage, il propose une approche dirigée par les modèles pour générer des éditeurs graphiques dans Eclipse. Les deux premiers sont ceux d'EMF. GMF utilise donc un modèle Ecore d'EMF (métamodèle du domaine, **.ecore*) pour définir la syntaxe abstraite d'un DSML, et créer ainsi le modèle générateur d'EMF (**.codegen*). Ainsi, GMF permet de décrire la syntaxe concrète de ce DSML en définissant le modèle graphique et le modèle de la palette d'outils. Le modèle de définition graphique (**.gmfgraph*) consiste à utiliser les composants de GEF (figures, nœuds, connexions, compartiments, etc.), pour spécifier une représentation graphique pour chacun des concepts du modèle de domaine, sans pour autant spécifier de liaisons directes avec des concepts du domaine. Quant au modèle de définition d'outillage (**.gmftool*), il consiste à définir une palette d'outils qui sera utilisée pour sélectionner et créer des représentations graphiques dans la zone d'édition des diagrammes. Ce modèle comporte un élément racine appelé « *Tool Registry* » qui se compose de l'élément « *Palette* » et de certains autres éléments, comme la « barre d'outils » et le « menu ».

GMF permet d'élaborer un modèle de *mapping* (**.gmfmap*) pour associer chacun des concepts du domaine avec sa représentation graphique et son outil dans la palette. Généralement, les nœuds, les connexions et les Labels de diagrammes, qui sont définis dans le modèle de définition graphique, sont mis en correspondance avec les concepts du modèle de domaine, et aussi avec des outils de la palette. Le modèle de *mapping* comporte un élément racine nommé « *mapping* » qui peut contenir les éléments suivants : *Canvas Mapping*, *Top Node Reference*, *Link Mapping*, *Audit Container*, *Metric Container*, et *Generic Style Selector*.

Ainsi, après avoir défini la liaison directe des concepts métier avec leurs outils d'instanciation et leurs représentations graphiques, le modèle générateur de GMF (**.gmfgen*) doit être créé à partir du modèle de *mapping* pour définir les paramètres de la génération de

¹⁰ <http://www.eclipse.org/gmf/>, dernière consultation : 07/04/2012

¹¹ <http://www.eclipse.org/gef/>, dernière consultation : 07/04/2012

code d'un éditeur graphique entièrement fonctionnel, qui est basé sur l'infrastructure d'exécution de GMF.

Cette infrastructure offre de nombreuses fonctionnalités que l'on aurait dû coder à la main si on utilisait directement EMF et GEF. Elle met à disposition :

- un ensemble de composants réutilisables pour le développement des éditeurs graphiques, tels que l'impression, l'exportation d'image, les barres d'outils, etc. ;
- un modèle standardisé pour décrire les éléments du diagramme, en séparant entre la sémantique du domaine et les éléments de la notation graphique ;
- une infrastructure de commande qui synchronise les commandes d'EMF avec celles de GEF ;
- un Framework extensible qui permet aux éditeurs graphiques d'être ouverts et extensibles.

La figure 20 donne un aperçu sur le processus GMF pour le développement des éditeurs graphiques.

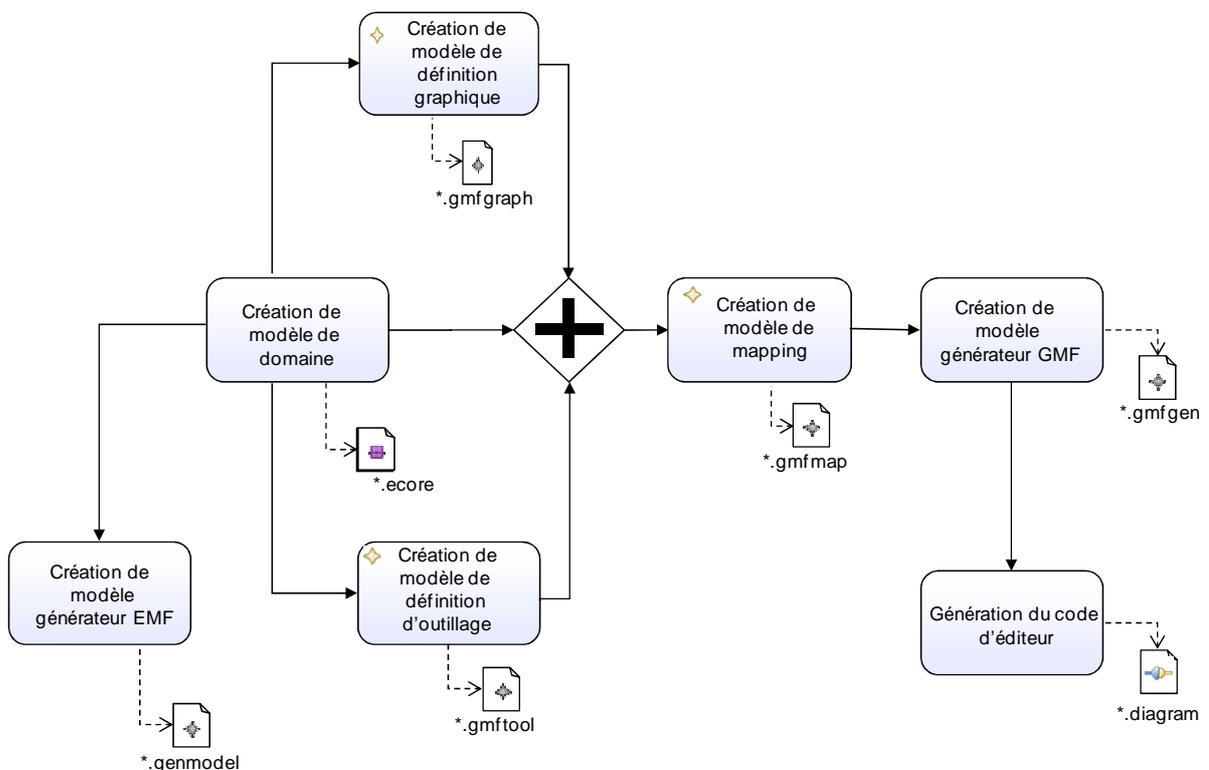


Figure 20 : Processus GMF pour le développement des éditeurs graphiques.

2.3.2.3 CPM (Cooperative Problem-based learning Metamodel)

Les travaux menés dans le laboratoire LIUPPA (Laboratoire Informatique de l'Université de Pau et des Pays de l'Adour) sur la modélisation des situations-problème coopératives s'inscrivent dans une approche IDM et DSM. Le résultat de ces travaux est un profil UML proposé pour définir CPM (Cooperative Problem-based learning Metamodel) [Laforcade et al. 2003] [Nodenot et al. 2003] [Laforcade 2004] comme langage graphique de conception des scénarios de type PBL (*Problem Based Learning*).

Le langage CPM est composé du métamodèle CPM (cf. figure 21) qui définit les concepts et leurs relations (syntaxe abstraite), et du profil CPM (cf. figure 22) qui définit le système de notation des concepts et relations (syntaxe concrète). Le profil utilise un élément principal, « le stéréotype », et exploite la richesse graphique proposée par UML.

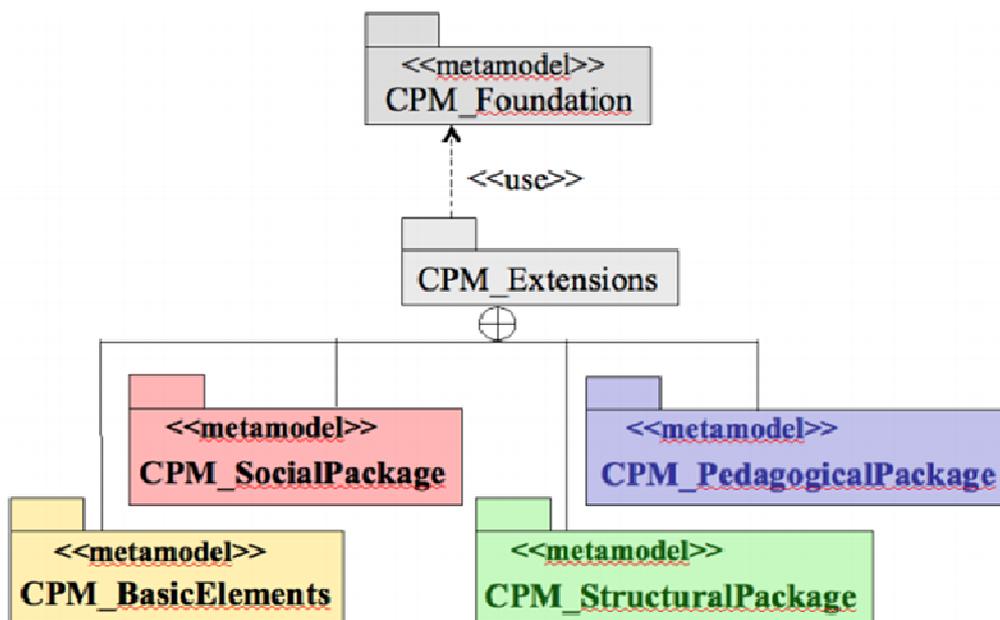


Figure 21 : Paquetage du métamodèle CPM [Laforcade 2004].

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

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

Figure 22 : Profil CPM (extraits) [Laforcade, 2004].

Dans ce travail, une technique de transformation de modèles de CPM (au niveau CIM) vers IMS-LD (au niveau PIM) est proposée. Pour extraire les informations nécessaires au modèle cible conforme à IMS-LD, La technique adoptée consiste à déterminer pour chaque concept d'IMS-LD la méta-classe UML et le stéréotype utilisés comme équivalents dans la spécification du même concept avec le langage CPM. Chaque composante d'IMS-LD est alors construite à partir des informations extraites de CPM. Toutefois, l'auteur [Laforcade 2005] a affirmé que certains éléments sont plus difficilement déductibles que d'autres.

2.3.2.4 Approche à base de Workflow pour permettre la flexibilité des processus d'apprentissage

Dans sa thèse [Vantroys 2003] effectuée au sein de l'équipe NOCE¹² du laboratoire LIFL (Lille), Vantroys a proposé une plate-forme d'exécution de scénarios pédagogiques en se basant sur une approche par Workflow puisque, d'après lui, les langages de structuration d'activités pédagogiques (EML) présentent des points communs avec la gestion des processus en général. Cette approche consiste à prendre en compte les aspects concernant la malléabilité pour réaliser la plateforme d'exécution qui permet une flexibilité pour autoriser une adaptation continue au contexte.

¹² NOCE : Nouveaux Outils pour la Coopération et l'Éducation.

En fait, l'objectif de Vantroys n'est pas de reconstruire une nouvelle plateforme mais de créer un « composant technique » réalisant l'exécution de scénarios et intégrable dans les LMS (Learning Management Systems) existants. L'auteur a basé son approche sur la programmation par composants logiciels pour construire l'application par réutilisation et intégration de composants existants.

Le composant technique implémenté est un moteur de Workflows, COW (pour Cooperative Open Workflow). Il permet de :

- Supporter les différents types d'enseignements : individuels et de groupes y compris au sein d'un même processus ;
- Supporter les activités collaboratives, c'est-à-dire permettre à des groupes d'utilisateurs d'intervenir dans une activité avec des rôles et des tâches différentes ;
- Supporter une redéfinition dynamique du processus d'apprentissage afin de pouvoir l'adapter aux étudiants et à l'environnement ;
- Favoriser la réutilisation des modèles de processus et d'activités y compris après une modification d'un modèle existant.

Pour supporter la flexibilité et la possibilité de modifier les processus d'apprentissage, ce travail combine une approche par métamodèle afin d'assurer la réflexivité, et l'approche par points ouverts afin de modifier le comportement du moteur. Le métamodèle défini est basé sur celui du langage de description de processus XPDL (*XML Process Definition Language*) qui est étendu afin de supporter la gestion du temps et les activités coopératives (via la notion de tâche) ainsi que pour favoriser la réutilisation des modèles d'activité.

Cependant, l'exécution de scénarios pédagogiques d'IMS-LD sur le moteur Workflow a nécessité de définir une projection des concepts de l'EML vers le langage de définition de processus et d'implanter un ensemble de services associés au moteur de Workflows pour gérer les propriétés qui ne sont pas liées à l'exécution proprement dite (tel que les objectifs pédagogiques). Outre la projection de IMS-LD, l'auteur a défini une architecture du moteur workflow pour la gestion de ces propriétés spécifiques aux EIAH et il a présenté un exemple de mise en œuvre : le Guide d'Apprentissage Interactif Numérique (GAIN).

Dans cette approche, le passage d'un EML vers le langage de définition de processus est fait par la comparaison des métamodèles. La transformation est ensuite réalisée par des filtres XSL. L'auteur a affirmé que cette transformation pourrait être simplifiée par exemple en utilisant un référentiel MOF (*Meta-Object Facility*) contenant les métamodèles des différents langages et QVT (*Query/View/Transformation*).

2.3.2.5 Le projet Bricoles : une approche dispositive

Dans le cadre du projet Bricoles de l'équipe NOCE du laboratoire LIFL (Lille) une approche « dispositive » [Caron 2007a] a été développée pour le développement d'EIAH. Le projet se propose d'instrumenter cette approche, qu'il appelle le « bricolage » pour construire de tels dispositifs : permettre à des enseignants d'adapter et de contextualiser des applications de manière à construire des dispositifs pédagogiques [Caron 2007a].

P.A. Caron a proposé une approche d'ingénierie des EIAH adaptée à des équipes d'enseignants pratiquant l'ingénierie pédagogique dans une perspective artisanale. Cette approche repose sur une infrastructure exploitant les ressources de l'ingénierie dirigée par les modèles (IDM) qui est composée d'une plateforme de méta-modélisation et de modélisation (ModX¹³), et d'un générateur de constructeur spécifique (GenDep).

D'après l'auteur, cette infrastructure favorise un processus de concrétisation permettant de passer de l'expression d'un modèle de dispositif en langage naturel à son expression dans un langage spécifique à la plateforme puis à sa construction dans le contexte de la plateforme. Pour implémenter cette infrastructure, l'auteur utilise les possibilités d'automatisation offertes par l'IDM. La démarche proposée consiste, au sein d'une plateforme de modélisation, à définir des métamodèles à transformer ou à fusionner. Ces métamodèles exprimés en MOF permettent d'abstraire les notions nécessaires à l'expression des modèles de dispositifs. La définition de règles de transformation ou de fusion entre métamodèles, puis la construction du générateur qui permet de contextualiser le dispositif en interagissant avec le greffon de services Web situé sur la plate-forme sur laquelle le dispositif est déployé. L'approche dispositive est illustrée par la figure 23.

Bricoles a démontré la faisabilité technique de l'IDM appliquée à la scénarisation pédagogique. L'approche « dispositive » conduit à étudier la contextualisation du scénario pédagogique par la prise en compte des spécificités de la plate-forme technique de diffusion. Cela s'est traduit par une fusion des métamodèles liés à l'expression pédagogique et aux possibilités techniques. Par contre, la génération du modèle dans le métamodèle associé à la plate-forme ne peut pas être une génération sans perte. Mais les différences, les incohérences qu'elle souligne entre les deux métamodèles, sont porteuses de sens et font la richesse de la démarche en amenant l'enseignant à mieux comprendre le spectre d'application et les possibilités de la plate-forme de formation qu'il utilise [Choquet 2007].

¹³ <http://noce.univ-lille1.fr/projets/ModX/>.

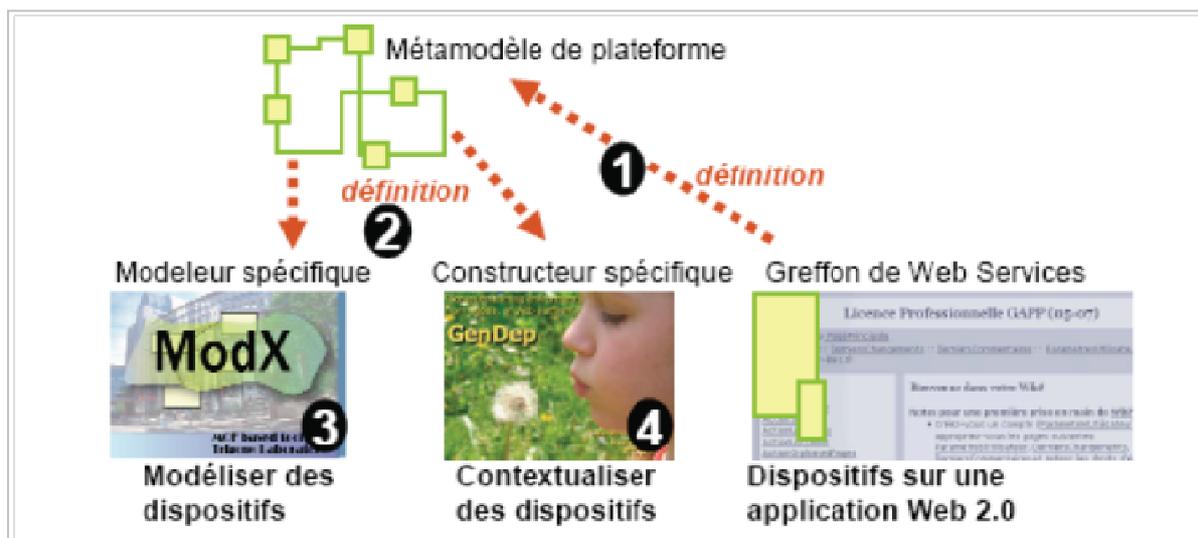


Figure 23 : Approche de conception du projet Bricoles [Caron 2007b].

Cependant, l'approche « dispositive » peut paraître complexe. Elle est une des premières applications concrètes de l'IDM à destination des enseignants. L'approche de modélisation est descriptive dans un premier temps, lors de la définition du métamodèle par le chercheur (informaticien), puis interprétative et productive, lors de la modélisation et de la contextualisation des dispositifs, phases de conception qui impliquent largement l'enseignant [Choquet 2007].

2.3.2.6 Instrumentation des situations d'apprentissage de Pédagogie par Projet Collectif (PPC)

Les travaux de thèse de F. Abdallah [Abdallah 2009] s'inscrivent dans le cadre des recherches menées au LIUM. Ce travail porte sur la spécification et l'instrumentation d'une situation d'apprentissage dans un contexte de Pédagogie par Projet Collectif (PPC). L'enseignant intervient dans tout le processus de spécification et d'opérationnalisation du scénario qui décrit la situation d'apprentissage PPC. La démarche adoptée par l'auteur est MDA en s'appuyant sur une méta-modélisation dédiée à la pédagogie par projet collectif. En vue de répondre à l'insuffisance en termes des langages de scénarisation pédagogique dans un contexte PPC, l'auteur a proposé un métamodèle PPC pour aider l'enseignant concepteur à élaborer des scénarios d'apprentissage. Après leur conception, les scénarios PPC sont déployés sur une plateforme de formation. Cette plateforme doit être choisie en fonction de son adéquation avec les scénarios en question. Le choix d'une plateforme cible pour l'opérationnalisation doit se baser sur une analyse qui doit être effectuée sur les fonctionnalités disponibles et permettre l'extraction et l'externalisation du modèle fonctionnel

de cette plateforme. L'opérationnalisation d'un scénario PPC s'effectue à travers « une démarche de transformation de modèles depuis le métamodèle de la situation d'apprentissage de la PPC vers celui de la plate-forme qui a été sélectionnée » [Abdallah 2009]. L'utilisation de MDA a permis à l'auteur de « satisfaire son hypothèse fondatrice de départ qu'une situation d'apprentissage de type PPC peut être médiatisée par la transformation du scénario PPC abstrait en un scénario PPC opérationnalisable sur la plate-forme cible » [Abdallah 2009]. F. Abdallah a testé sa proposition sur la plate-forme Moodle. En terme de réalisation, un outil informatique a été développé pour être utilisé par des enseignants-concepteurs en vue de leur faciliter la transformation des scénarios représentant des situations d'apprentissage de Pédagogie par Projet Collectif (PPC) vers des scénarios opérationnalisables sur MOODLE (cf. figure 24). Les règles de transformation sont définies par le langage ATL (*Atlas Transformation Language*).

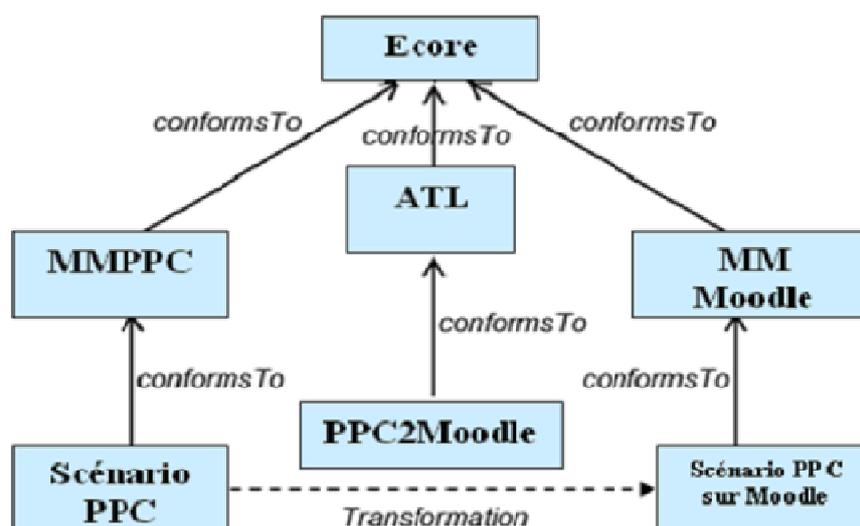


Figure 24 : Transformation du scénario PPC [Abdallah 2009].

2.3.2.7. Conception pédagogique dirigée par les modèles (MDLD)

Dans [Dodero et Diez 2006] les auteurs ont utilisé dans un premier temps les techniques d'IDM pour essayer d'augmenter le degré d'automatisation dans la création ou l'adaptation des ressources pédagogiques dans un contexte d'ingénierie pédagogique (*Instructional Engineering*) où plusieurs acteurs de différents domaines sont invoqués pour l'élaboration du système, ressources, artefacts, etc. Chacun de ces acteurs utilise son propre langage métier, son propre métamodèle. L'auteur affirme que la complexité des processus d'ingénierie pédagogique s'explique par l'existence de différents modèles et niveaux

d'abstraction, des contextes et des préoccupations différents. L'IDM permettant de considérer chaque aspect du processus du développement comme un modèle séparé, l'auteur propose de créer des ressources pédagogiques par la composition et la transformation des différents modèles instanciés de différents métamodèles.

Dans d'autres travaux de la même équipe [Dodero et al. 2008], les auteurs utilisent les techniques de transformation pour la conception pédagogique dirigée par les modèles. Ils montrent les limites des approches du design pédagogique existantes, en matière de distance entre l'utilisateur et les spécifications (telles que IMS-LD), ainsi que certaines lacunes des éditeurs existants de ressources d'apprentissage en matière de difficultés d'utilisation par les enseignants (non experts). Les auteurs favorisent les approches graphiques telles que MOT et CPM. En outre, ils distinguent trois caractéristiques pour classer des éditeurs : *design abstractions* (des abstractions utilisées pour la conception) ; *source vs. target model* (des modèles sources et cibles utilisés dans la transcription) ; et *multiple models* (fusion de plusieurs modèles).

Les auteurs distinguent 4 types de techniques de transformation : *shortcutting* ; *mapping* ; *refactoring* ; *extending*.

(a) *Shortcutting* : pour la génération des instances des unités d'apprentissage à partir des *templates* paramétrables tels que des patterns et des idiomes.

(b) *mapping* : différents genres de *mapping* peuvent être distingués :

1) *un modèle vers plusieurs* : où l'enseignant crée une unité d'apprentissage contenant les activités d'apprentissage et d'évaluation sans savoir qu'il y a plusieurs modèles nécessaires tels que : IMS-LD, QTI, etc.

2) *plusieurs modèles vers un seul (merging)* : par exemple la modélisation de différents aspects d'un même artefact par plusieurs acteurs.

3) *un vers un* : entre modèles ou métamodèles.

- *Mapping intra-langage* : d'un modèle à l'autre, suivant le même métamodèle (par exemple l'enrichissement d'un modèle en ajoutant des éléments tels que des activités);

- *Mapping inter-langage* : d'un métamodèle à l'autre en respectant la distance sémantique entre les concepts, (par exemple UML vers IMS-LD).

(c) *Refactoring* : est un mapping intra-langage, qui consiste à améliorer une unité d'apprentissage pour la rendre plus efficace et réutilisable sans changer son objectif original.

(d) *Extending* : est un mapping inter-langage, qui consiste à ajouter des nouveaux éléments au méta-modèle source ou cible, par exemple dans ASK-LDT [Sampson et al. 2006], l'utilisateur peut définir plusieurs types d'activités, au contraire d'IMS-LD.

Dodero et al. désignent leur contribution : *Model-Driven Learning Design* (MDLD). MDLD (en français, la conception pédagogique dirigée par les modèles) fournit des outils de spécification d'une unité d'apprentissage indépendamment de la plateforme qui va la supporter, permettant de transformer la spécification abstraite vers une spécification plus concrète correspondant à une plateforme particulière. MDLD consiste à intégrer, dans un environnement unique, les différentes abstractions non formelles pour spécifier une unité d'apprentissage, ceci pour dispenser des enseignants des aspects techniques de la conception souvent trop éloignés de leurs compétences.

Pour l'implémentation, l'auteur propose un outil de méta-modélisation basé sur Eclipse et le plug-in FXL qui fournit deux vues de développement à base de modèles :

- *la vue du modelleur de l'activité* : utilisation de plug-in de BPEL (*Business Process Execution Language*) pour modéliser des activités au niveau d'abstraction PIM ;
- *la vue du modelleur de transformation* : fournit une interface graphique pour éditer des lignes de transformation de PIM vers PSM.

L'environnement MDLD est basé sur l'utilisation des métamodèles et des transformations à base d'idiomes afin de manipuler des modèles multi-sources et de produire d'éventuelles unités d'apprentissage. Dodero et ses collègues affirment qu'à l'aide de l'environnement MDLD, le concepteur peut générer une unité d'apprentissage, conforme à IMS-LD, en combinant plusieurs standards de la conception pédagogique sans l'exigence de les connaître en détail. La figure 25 représente les étapes de processus de transformation pour générer le fichier de manifeste IMS-LD d'une unité d'apprentissage.

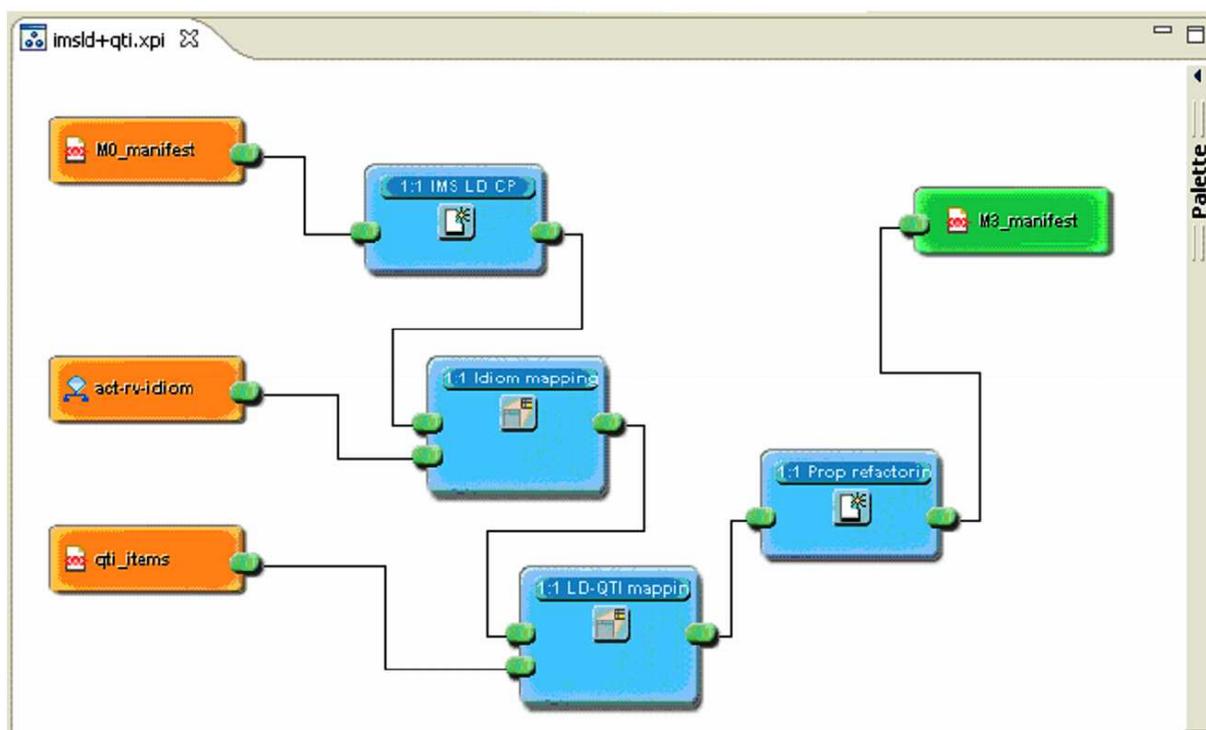


Figure 25 : Vue de modélisation de transformation de l'environnement MDLD
[Dodero et al. 2008].

Enfin, Dodero et al. constatent que les langages abstraits de modélisation ne sont pas tous adéquats pour la conception pédagogique, et qu'aucun d'eux n'est complètement satisfaisant et suffisant pour la modélisation des aspects complexes de la spécification IMS-LD. Ceci implique donc la nécessité de définir des DSL (*Domain-Specific Language*) pour le domaine pédagogique traité, qui peuvent étendre l'environnement MDLD.

2.3.2.8 Implication des praticiens dans la définition des langages spécifiques de conception pédagogique

L'IDM a permis de rendre envisageable l'automatisation de l'activité d'opérationnalisation des scénarios pédagogiques sur des plateformes de formation. Toutefois, les premiers résultats des expérimentations réalisées au sein des travaux de [Laforcade 2004] [Vantroys 2005] [Caron 2007a] [Dodero, 2008] [Abdallah, 2009] ont montré certaines limites. Ces travaux sont basés sur les transformations entre le domaine de conception et celui de l'opérationnalisation. La transformation entre deux domaines différents est difficile à mettre en œuvre en pratique [Laforcade et al. 2007] parce que certains concepts/rerelations d'un métamodèle n'ont pas forcément de concepts/rerelations "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 (certaines informations*

nécessaires au langage "métier" cible ne peuvent parfois être précisées : le scénario obtenu est sémantiquement incomplet) » [Abdallah 2009].

Ainsi, afin de dépasser les limites de transformation des modèles et de préserver la sémantique des scénarios pédagogiques lors de l'opérationnalisation, certains travaux se sont focalisés sur la définition des langages métiers des concepteurs dans une approche DSM. Nous évoquons ici deux travaux principaux : le projet MDEduc [de Moura Filho 2007] et l'approche de conception collective des scénarios pédagogiques [El-Kechaï 2008].

2.3.2.8.1 Le projet MDEduc

Le travail de thèse de C. de Moura Filho [de Moura Filho 2007] dans le cadre du projet MDEduc de l'équipe de recherche NOCE du laboratoire LIFL de l'Université Lille1, s'inscrit dans une approche d'ingénierie des EIAH fondée sur l'ingénierie dirigée par les modèles. L'auteur partage la même position sur la scénarisation pédagogique que [Choquet 2007] en considérant qu'il faut permettre aux enseignants de mieux s'impliquer dans le processus de développement d'un EIAH en les aidant à définir un langage de modélisation pédagogique adapté à leurs pratiques, à leurs contextes et à leurs objectifs métiers. Dans ce travail, un tel langage est décrit par un métamodèle qui est défini en collaboration avec des enseignants concepteurs des scénarios pédagogiques. Cette approche vise ici encore à surmonter les difficultés qu'un enseignant néophyte peut rencontrer en utilisant des langages et des outils génériques ou en communiquant avec des développeurs logiciels. Elle s'appuie sur le concept de « patron pédagogique » de conception [De Moura Filho et Derycke 2007]. Ces patrons sont écrits en langage naturel et utilisés pour définir le CIM d'un scénario. La génération du dispositif s'effectue suivant l'approche MDA, le CIM servira à la création de métamodèle PIM après un ensemble d'interactions avec le concepteur et un éditeur dédié à la personnalisation du patron. Le métamodèle formel, permettant l'expression de scénarios de niveau PIM, est aussi utilisé pour définir un PSM par transformation, puis le dispositif de formation correspondant est généré.

Cette approche a abouti à la création d'un éditeur logiciel, nommé « *Pedagogical Patterns Editor* » (ou PP Editor), qui permet l'édition de patrons pédagogiques et l'extraction de modèles pédagogiques formels à partir des scénarios d'apprentissage édités dans cet environnement. Cela tout en autorisant les basculements entre des représentations informelles proches du langage naturel et des représentations plus formelles.

La figure 26 (extraite de [de Moura Filho et Derycke 2007]), représente la construction du métamodèle (partie droite de la figure) suite à des interactions de type édition/sélection menées sur le patron (partie gauche de la figure).

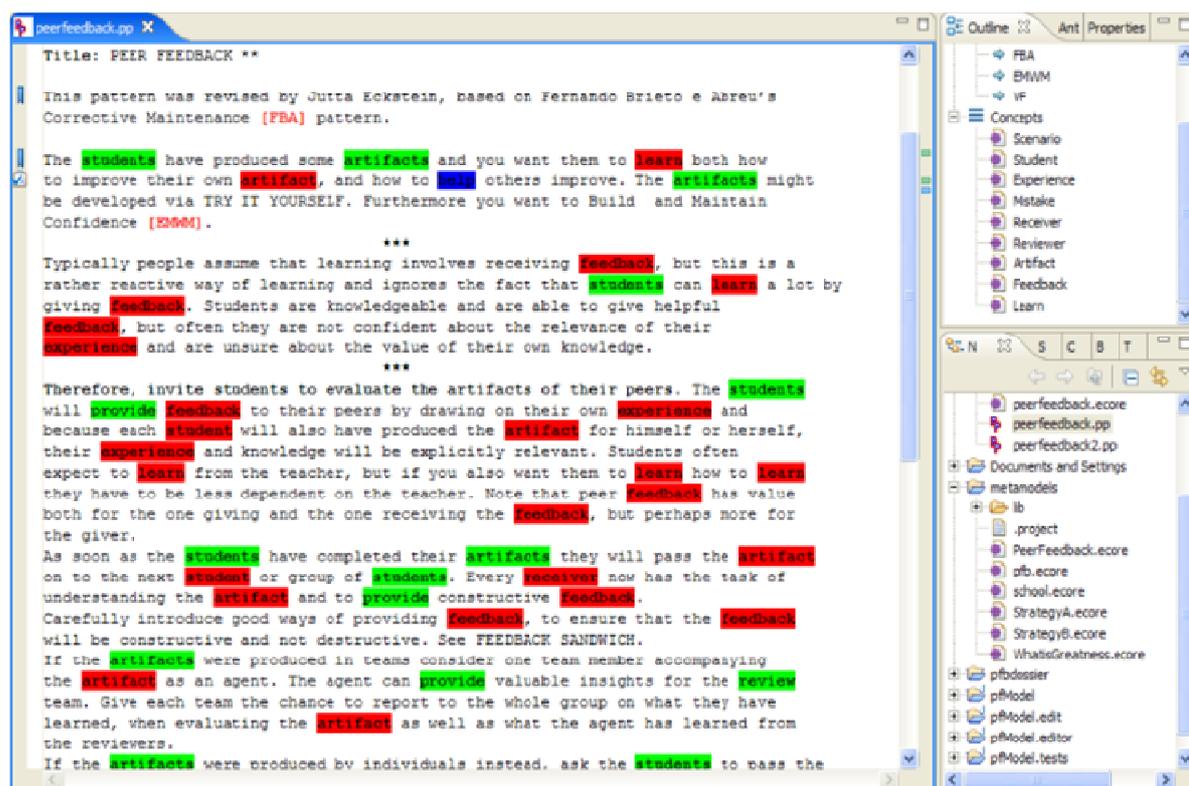


Figure 26 : Construction de scénarios à base de patrons pédagogiques
[De Moura Filho 2007].

2.3.2.8.2 Conception collective des scénarios pédagogiques : approche par méta-modélisation située

Le travail de thèse de H. El-Kechaï [El-Kechaï 2008] s'inscrit dans le cadre du projet REDiM. Ce travail vise à intégrer les enseignants dans le processus de conception des scénarios pédagogiques opérationnalisés dans l'EIAH. Selon l'auteure, une des phases essentielles du processus de conception d'un scénario est « *l'élaboration de son métamodèle d'expression pédagogique. Celui-ci définit un langage de modélisation pédagogique* », qui permet de spécifier formellement le scénario pour répondre à plusieurs objectifs : opérationnalisation, échange, réutilisation, interopérabilité.

Ainsi, H. El-Kechaï considère que le caractère idiosyncrasique des situations d'apprentissage, ainsi que le caractère hétérogène des futurs utilisateurs (enseignants et apprenants) de l'EIAH, justifient le développement et la mise en œuvre de démarches spécifiques de conception. C'est sur cette base que l'auteure a proposé une approche "constructive" de métamodèles d'expression et de scénarios pédagogiques. Cette approche a comme particularité d'impliquer dans la conception les utilisateurs finaux (essentiellement les enseignants) de l'EIAH désignés par usagers-concepteurs.

Pour mener sur le terrain ses propositions, El-Kechaï a travaillé avec un ensemble de concepteurs qui étaient impliqués dans une approche participative de construction d'un EIAH pour un Centre de Formation par Apprentissage, dans le cadre du projet LEA (Livret Electronique d'Apprentissage) [Cottier 2008]. Cette expérimentation a amené les concepteurs à réfléchir à leurs besoins par la description des scénarios à implémenter. Le processus de conception a amené ses acteurs à produire et mobiliser une quantité importante d'Objets Intermédiaires (OI). Ces objets sont les traces du raisonnement de conception sous forme de notes, de représentations graphiques, de schémas (voir le coté gauche de la figure 19), et sont considérés comme un moyen de lire une activité de conception. Le contenu et la nature de ces différents Objets Intermédiaires peut alors être analysé pour dégager les principales caractéristiques de l'activité concernée et formaliser un métamodèle d'expression pédagogique (voir le coté droit de la figure 27).

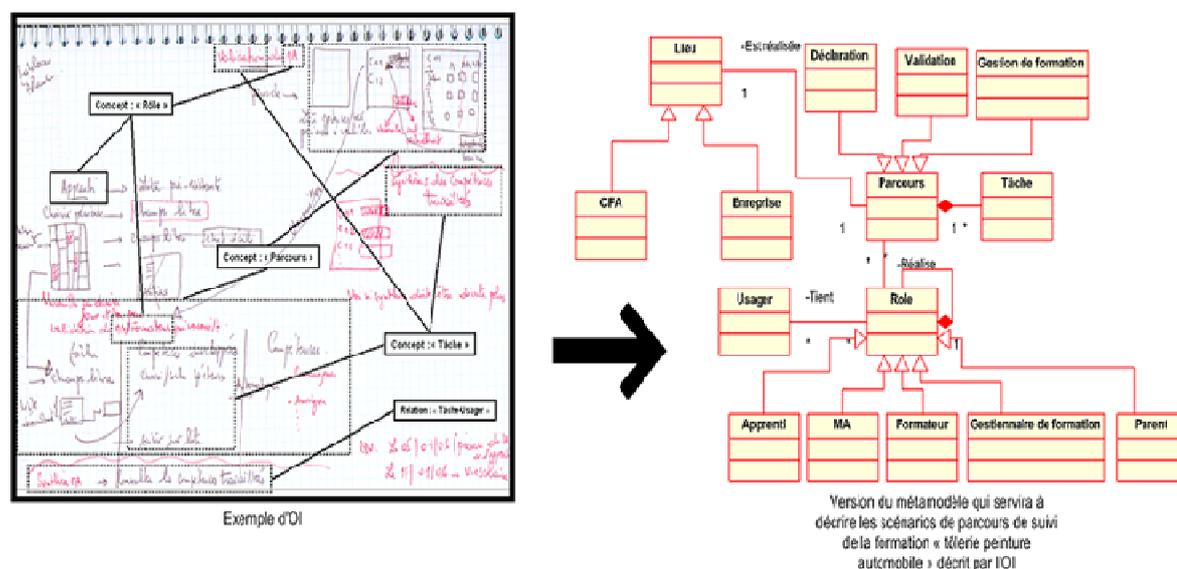


Figure 27 : Modélisation sur la base du contenu de l'OI [El-Kechaï 2008].

2.3.3 Approches liées au Web sémantique

2.3.3.1 Approche à base des documents virtuels adaptatifs

Dans ses travaux de thèse [Tetchueng 2008], Tetchueng propose une approche pluridisciplinaire de développement des EIAH adaptatifs et contextuels, fondée sur l'étude et la formalisation du savoir-faire et des pratiques réelles des enseignants, dans un modèle de scénario, en utilisant une théorie didactique : la théorie anthropologique didactique du savoir de Chevallard. Le modèle de scénario est conçu en articulant le savoir-faire et les pratiques

réelles d'enseignement des sciences selon une approche par résolution de problème : la théorie de Chevallard et les modèles de tâches hiérarchiques. Un modèle générique des scénarios prend en compte un maximum de situations d'apprentissage. À partir de ce modèle, l'EIAH peut générer dynamiquement une situation d'apprentissage particulière à mettre en œuvre pour un apprenant donné dans un contexte spécifique. Afin de faciliter l'acquisition de ce modèle par les enseignants, un outil auteur de gestion des scénarios a été développé. L'EIAH final est vu comme un document virtuel adaptatif (DVA). Il est construit à partir d'un moteur de composition flexible de ressources dénommé SCARCE fondé sur le web sémantique et les DVA.

2.3.3.1 Représentation des connaissances par des technologies de Web sémantique

Kravčik et Gašević [Kravčik et Gašević 2006] proposent une approche à base de web sémantique pour résoudre les problèmes de la réutilisation et de l'adaptation des environnements d'apprentissage. Cette approche consiste à représenter les connaissances de design pédagogique et d'adaptation de l'apprentissage en deux types : *déclaratives* et *procédurales*. Le premier type consiste à décrire le *domaine*, l'*utilisateur* et le *contexte* d'apprentissage. Le second est important pour spécifier des *activités* d'apprentissage et définir les *stratégies d'adaptation*. Cela permet la génération des instances concrètes de façon dynamique et selon le contexte d'usage tout en autorisant leur réutilisation.

Les auteurs ont critiqué les standards en terme d'adaptation, affirmant que IMS-LD permet seulement de très simples adaptations. Ils constatent également que les connaissances procédurales sont encodées et fixées dans les environnements d'apprentissage, ce qui pose des difficultés aux enseignants pour les adapter suivant leurs besoins dans les différents contextes d'enseignement.

Comme proposition, les auteurs utilisent les ontologies pour représenter les différents types de connaissances pertinentes pour l'apprentissage adaptatif personnalisé. Pour l'implémentation, une ontologie des services Web à base d'OWL (OWL-S) est utilisée pour intégrer le design pédagogique et les processus de composition des ressources d'apprentissage. En outre, pour surmonter le problème de partage des règles d'adaptation, les auteurs ont proposé d'utiliser deux langages : le langage de marquage des règles (*RuleML*) ou le langage des règles de Web sémantique (*SWRL*).

Ce travail alimente les critiques des standards, notamment en ce qui concerne la réutilisation et l'adaptation des instances des unités d'apprentissage selon le contexte réel d'usage. Les auteurs ont choisi une approche utilisant des technologies du Web sémantique.

2.3.4. Analyse et discussion des travaux existants

Un nombre significatif de travaux de recherche se sont donc intéressés à la conception et l'adaptation des scénarios pédagogiques. Concernant l'adaptation par les standards (en particulier niveau B et C de IMS-LD) on trouve plusieurs travaux : [Burgo 2006], [Yongwu 2005], le projet aLFanet [Van Rosmalen et al. 2004] [Santos et al. 2005], [Berlanga 2006], ainsi que [Towle et Halm 2005]. Ces travaux ont montré que IMS-LD fournit des méthodes statiques pour implémenter des stratégies simples d'adaptation des unités d'apprentissage en se basant sur des règles de type « *if-then* », et non pas des adaptations complexes telles que l'interaction entre des règles multiples ou l'ordonnement imposé [Kravčik et Gašević 2006].

Selon Burgo [Burgo 2006] la spécification IMS-LD a besoin d'une restructuration et d'une modification et d'un ajout de certains éléments. Yongwu [Yongwu 2005] a constaté quant à lui que la communauté IMS-LD n'a pas donné une grande importance à la gestion des exceptions parues en exécution.

Le projet aLFanet [Van Rosmalen et al. 2006] propose une approche qui consiste à combiner des standards indépendants les uns des autres, afin de développer des environnements d'apprentissage adaptables. Les résultats de ce projet sont intéressants mais les outils développés demeurent difficilement accessibles aux néophytes.

L'approche ADL hybridant IMS-LD et les SHA [Berlanga et Garcia, 2005b] paraît quant à elle restrictive. Au-delà de l'imposition à l'enseignant (non expert) d'un langage, IMS-LD, et d'un environnement (*LD Editor*) différents de ceux de son propre domaine métier, la conception de scénarios adaptables, plus complexes, est là aussi loin de lui être abordable.

Aux limites, bien connues maintenant, de l'appropriation des standards par les enseignants, on peut ajouter celles concernant la question de l'adaptation pendant la conception et l'exécution. La définition des stratégies d'adaptation en amont ne peut pas en effet couvrir toutes les situations possibles. La prédétermination de l'ensemble des scénarios d'usage est impossible puisque, comme l'indique l'approche instrumentale et les nombreux travaux menés sur les usages, la conception d'un instrument se poursuit dans son usage [Béguin 2005]. [Towle et Halm 2005 p25] ont listé les limites de l'adaptation par IMS-LD pendant la conception (cf. section 2.3.1.1.2). Van Rosmalen et Boticario [Van Rosmalen et Boticario 2005] ont affirmé que les limites de l'adaptation pendant l'exécution reviennent principalement à la difficulté de contrôler les apprenants durant toutes leurs sessions d'apprentissage, ainsi qu'à la difficulté de combiner l'adaptation effectuée au moment du déroulement avec l'adaptation effectuée au moment de la conception.

Enfin, la plupart des outils actuellement disponibles sont plus adaptés à l'informaticien qu'à l'enseignant. Ils sont centrés essentiellement sur les compétences des informaticiens plutôt que sur celles des enseignants. Ils sont peu transparents. Ces outils ne permettent pas d'opérationnaliser les niveaux B et C d'IMS-LD. Il y a un consensus dans la littérature et entre les praticiens pour estimer que les non-experts ne peuvent pas concevoir les unités d'apprentissage en utilisant des outils actuellement disponibles. Ils sont trop complexes, nécessitent un utilisateur final qui maîtrise bien la structure et les formalismes de la spécification. En outre, ces outils ne fournissent pas de soutien à l'auteur non-expert au niveau de l'application des règles de conception de l'apprentissage, pour capturer la connaissance de l'auteur dans les unités d'apprentissage rationnelles d'un point de vue pédagogique [Sodhi et al. 2007] [Dodero et al. 2008].

Partant de critiques des standards en terme de réutilisation et d'adaptation des instances des unités d'apprentissage selon le contexte réel d'usage, Kravčik et Gašević [Kravčik et Gašević 2006] ont choisi d'exploiter les technologies du Web sémantique, utilisant une ontologie des services Web à base d'OWL (OWL-S) pour générer des instances concrètes, dynamiquement selon le contexte d'usage, tout en garantissant leur réutilisation.

Caeiro-Rodriguez et al. [Caeiro-Rodriguez et al. 2007] ont proposé PoEML en vue d'introduire plus de flexibilité dans des unités d'apprentissage. Ils ont optimisé les EML en proposant une solution orientée perspectives pour traiter chaque préoccupation indépendamment des autres. C'est une bonne approche qui permet de décomposer les problèmes en « sous problèmes » plus faciles à résoudre. Néanmoins, la question de l'appropriation de PoEML et de ses outils supports par des enseignants néophytes reste posée.

Quant au langage LDL, même si ses auteurs ont voulu : « *introduire une certaine forme de malléabilité dans le déroulement des activités* » [Martel et al. 2007], Choquet [Choquet 2007] affirme que « *le métamodèle de LDL repose lui aussi sur une métaphore imposée [...] certes beaucoup plus simple que celle d'IMS-LD, mais qui peut paraître tout aussi étrangère à des enseignants* ». La terminologie utilisée n'est pas celle de leur propre domaine métier.

L'approche de [Tetchueng 2008] est intéressante car elle prend en considération le savoir-faire et les pratiques réelles des enseignants. Grâce à la technique des documents virtuels, cette approche peut aider l'enseignant dans la recherche des scénarios les plus adaptés à la situation d'apprentissage à mettre en œuvre. Elle aborde donc l'adaptation des scénarios en amont en posant ainsi le problème : comment aider l'enseignant à adapter son scénario en temps réel ?

Zarraounandia [Zarraounandia, 2007] quant à lui part du principe que la conception d'une unité d'apprentissage est raffinée par son usage, pendant l'exécution, par modélisation tardive. Cette approche est focalisée sur la réutilisabilité. Une architecture est développée pour adapter automatiquement les unités d'apprentissage à leur contexte réel d'exécution par l'interprétation, *en temps réel*, de petites actions, des « pokes », adaptatifs, qui sont spécifiées séparément au fichier de définition des unités d'apprentissage. C'est une approche qui nous semble intéressante car elle aborde l'ensemble du cycle de vie de l'unité d'apprentissage adaptable en permettant la capitalisation des bonnes pratiques des enseignants pour la réingénierie. Le stockage des « pokes » adaptatifs permet la réutilisation et le partage des stratégies d'adaptation. Pour autant, cette approche s'appuie également sur IMS-LD, ce impose à nouveau aux enseignants de se familiariser avec les concepts et les outils de cette spécification pour définir et utiliser les unités d'apprentissage et les « pokes » d'adaptation.

L'IDM a confirmé les hypothèses posées sur la possibilité d'automatiser l'activité d'opérationnalisation des scénarios pédagogiques sur des plateformes de formation. Les premiers résultats des expérimentations réalisées au sein des travaux de [Laforcade 2004] [Vantroys 2005] [Caron 2007a] [Dodero et al. 2008] [Abdallah 2009] ont montré toutefois certaines limites.

Vantroys [Vantroys, 2003] a développé un composant « COW » comme moteur Workflow qui permet d'exécuter, sous forme de processus, des scénarios pédagogiques représentés par IMS-LD. Mais la projection des concepts d'IMS-LD vers le langage de définition de processus provoque une perte sémantique. Par exemple, l'élément « objectif » n'existe pas dans le langage traité par COW. Cette projection est réalisée par des filtres XSL. Nous partageons l'idée de cette approche sur l'adaptation des scénarios qui s'inspire des travaux issus de la flexibilité des Workflows.

Pour opérationnaliser des scénarios CPM, [Laforcade 2005] a proposé une technique de transformation de modèles de niveau CIM (conformes au métamodèle de CPM) vers le niveau PIM (conformes au métamodèle d'IMS-LD), qui consiste à définir pour chaque concept d'IMS-LD son correspondant dans CPM en précisant la méta-classe UML et le stéréotype utilisés. Toutefois, l'auteur a affirmé que certains éléments sont plus difficilement déductibles que d'autres, ce qui signifie une perte sémantique.

F. Abdallah [Abdallah 2009] a adopté une approche MDA pour réaliser l'opérationnalisation des scénarios PPC à travers une démarche de transformation des modèles depuis le métamodèle de la situation d'apprentissage de la PPC vers celui d'une

plate-forme cible. Il est pertinent de spécifier des scénarios à un niveau abstrait indépendant de la complexité technique des plateformes, mais la transformation induit souvent ici aussi à des pertes sémantiques.

L'approche « dispositive » proposée par [Caron 2007a] peut paraître complexe. Elle est une des premières applications concrètes de l'IDM à destination des enseignants. L'approche de modélisation est descriptive dans un premier temps, lors de la définition du métamodèle par l'informaticien, puis interprétative et productive, lors de la modélisation et de la contextualisation des dispositifs, phases de conception qui impliquent largement l'enseignant [Choquet 2007]. De plus, la génération du modèle dans le métamodèle associé à la plate-forme ne peut pas être une génération sans perte sémantique.

Quant à [Dodero et al. 2008], il constate les limites des approches du design pédagogique existantes en terme de distance entre l'utilisateur et les spécifications, ainsi que les éditeurs utilisés. Il propose d'utiliser des techniques de transformation pour la conception pédagogique dirigée par les modèles. Dodero trouve dans l'IDM et les approches graphiques le potentiel qui permet aux concepteurs de travailler à un niveau abstrait, loin de la complexité technique. Il propose de modéliser les scénarios par un langage de modélisation des processus (BPEL), puis de les transformer en IMS-LD et QTI pour qu'ils soient exécutables. La limite ici est la même que pour les autres approches IDM qui utilise la transformation entre le domaine de conception et celui de l'opérationnalisation : il y a une perte sémantique certaine pendant les transformations depuis ou vers un langage de définition des processus. Il n'est par exemple pas possible de représenter le concept « rôle » par BEPL. En outre, la transformation entre deux domaines différents est difficile à mettre en œuvre en pratique [Laforcade et al. 2007] parce que certains concepts/rerelations d'un métamodèle (cible ou source) n'ont pas forcément de concepts/rerelations "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 (certaines informations nécessaires au langage "métier" cible ne peuvent parfois être précisées : le scénario obtenu est sémantiquement incomplet)* » [Abdallah 2009].

Afin de dépasser les limites de transformation des modèles et de préserver la sémantique totale des scénarios pédagogiques, [Dodero et al. 2008] constate la nécessité de définir des DSL (*Domain-Specific Language*) pour les domaines éducatifs des enseignants. Nous trouvons pour notre part cette piste très pertinente. Choquet [Choquet 2007] la confirme : « *le processus de conception doit s'inscrire dans l'univers métier de l'enseignant plutôt que de proposer des passerelles terminologiques* ». Certains travaux se sont donc focalisés sur la définition de langages métiers pour les enseignants. Nous évoquons ici deux travaux

principaux : le projet MDEduc [de Moura Filho 2007] et l'approche de conception collective des scénarios pédagogiques [El-Kechaï 2008]. Les auteurs et nous partageons la même position sur la conception pédagogique en considérant qu'il faut permettre aux enseignants de mieux s'impliquer dans le processus de développement d'un EIAH, notamment en les aidant à définir un langage de modélisation pédagogique adapté à leurs pratiques, à leurs contextes et à leurs objectifs métiers. Ces travaux s'inscrivent alors dans une approche DSM. Ils n'abordent toutefois pas les aspects d'adaptation dynamique des scénarios. De plus, les outils spécifiques destinés à favoriser l'expressivité chez les enseignants (tels que ECOS [Choquet 2007] [El-Kechaï 2008]), ne fournissent pas toujours des modèles productifs, suffisamment riches, qui peuvent être opérationnalisés et adaptés dynamiquement. Il existe en effet un manque d'expressivité des métamodèles définis.

Les principales conclusions que nous avons pu tirer de notre analyse et discussion de l'ensemble des travaux existants se synthétisent en deux points :

- ❖ Les langages génériques de modélisation pédagogique, même s'ils permettent l'adaptation des scénarios, restent difficiles à comprendre et à utiliser réellement par des praticiens. De plus, les outils supports ne permettent souvent pas de produire des scénarios opérationnalisables (productifs).
- ❖ Les travaux menés à base d'IDM sont en lien direct avec notre objet d'étude. Certains travaux s'appuient sur la fusion et la transformation des modèles qui provoquent une perte sémantique. Cela a poussé d'autres travaux à proposer des approches et des techniques permettant d'aider les praticiens dans la définition de métamodèles décrivant les langages de leurs propres domaines. Ces travaux s'inscrivent dans une approche DSM. Ils n'abordent toutefois pas les aspects d'adaptation des scénarios.

3

Problématique et approche méthodologique

Sommaire

- 3.1 Définition de notre problématique de recherche
- 3.2 Hypothèses et objectifs de recherche
- 3.3 Méthodologie de notre recherche

Ce chapitre détaille notre problématique, présente nos objectifs de recherche en s'appuyant sur le positionnement scientifique. Des questions de recherche sont posées et l'approche méthodologique est présentée.

3.1 Définition de notre problématique de recherche

La conception d'un EIAH représente un effort significatif. Coûteuse en temps comme en moyens techniques et humains, elle amène généralement les concepteurs à faire des choix en termes de pédagogie, de modalités d'interaction, de technologies, etc. Depuis maintenant plusieurs dizaines d'années, de nombreux artefacts logiciels ont été développés dans les laboratoires pour explorer de nouvelles modalités d'apprentissage. L'une des difficultés identifiées est de transférer ces propositions expérimentales à une échelle plus large [Nodenot 2005] [Tchounikine 2006] [Pernin 2007]. Malgré des volontés institutionnelles et politiques d'intégrer les EIAH dans des formations de différents niveaux, cette intégration reste très faible [Previt 03]. Parmi les raisons principales on évoque des démarches d'ingénierie essentiellement « technocentriques », décomposées en étapes qui segmentent le temps, les acteurs et le travail à accomplir. Elles cristallisent une discontinuité entre les processus de conception et d'usage, entre les concepteurs et les utilisateurs.

Cottier et al. mettent en exergue un constat qui nous semble fondamental, « *les technologies aujourd'hui disponibles n'ont pas toujours la souplesse requise pour une utilisation dans le cadre éducatif qui demande bien souvent des adaptations rapides à des événements nouveaux et souvent imprévus* » [Cottier et al. 2008]. Certes, plusieurs types¹⁴ d'EIAH ont été proposés ces trois dernières décennies, notamment les STI et les SHA. La majorité de ces systèmes demeurent toutefois inaccessibles en raison de leur sur-automatisation qui tend à exclure, ce que [Rabardel 1995] appelle, l'intuition, le jugement subjectif, les connaissances tacites, l'imagination et l'intentionnalité chez les usagers, en l'occurrence des enseignants. Cottier et El-Kechaï ont confirmé que : « *en matière d'ingénierie des EIAH, la parole et la pratique du concepteur pédagogique sont fondamentales* » [Cottier et El-Kechaï 2009]. Il nous semble nécessaire alors d'impliquer les enseignants dans les processus d'ingénierie des EIAH en vue d'éviter les difficultés qu'ils rencontrent en utilisant les systèmes proposés comme clés en main [RAP 1999]. En ce sens, [Murray 2004] [Ainsworth et al. 2003] [Half et al. 2003] [Van Joolingen et de Jong 1996] [Munro et al. 1997] [Merrill 2003] ont affirmé que les outils d'édition peuvent éviter la nécessité des compétences techniques pour permettre au plus grand nombre de personnes à participer au processus de développement. Ils peuvent permettre ainsi la réutilisation et l'adaptation des contenus pédagogiques. Murray a évoqué les apports de ces outils d'édition, notamment en termes d'augmentation de la productivité par la réduction du temps,

¹⁴ Pour des éléments généraux sur les EIAH et leur évolution consulter [Wenger 1987], [Brusilovsky 1997], [Grandbastien 2006].

des efforts et du cout nécessaires pour développer à chaque fois un EIAH pour un ensemble défini de situations pédagogiques.

Mais offrir un éditeur à un enseignant en vue d'effectuer une conception préalable (prédéfinition) de scénarios pédagogiques n'est pas toujours suffisant. Certaines situations d'apprentissage ont un caractère idiosyncrasique selon [El-Kechaï 2008] car bien souvent, les apprenants ont des profils hétérogènes. Dans la pratique, il est parfois difficile, voire impossible, de créer une description complète d'un scénario d'apprentissage avant le début de son exécution [Zarraounandia 2006b]. El-Kechaï considère que « *les études menées sur les situations de conception reconnaissent depuis de nombreuses années son caractère « mal défini » (ou “mal structuré”)* » [El-Kechaï 2008]. Cette dernière souligne le fait que le concepteur dispose, au départ, d'une représentation mentale incomplète et imprécise de l'objet à concevoir [Eastman 1969] [Simon 1973]. Ce n'est que progressivement que le concepteur complète et précise sa représentation mentale [Bonnardel et Rech 1998]. La question posée par l'ingénierie d'un scénario n'est donc pas seulement de produire un modèle correspondant à une spécification donnée, mais d'élaborer un modèle adaptable, souvent en cours d'usage, dans des contextes d'apprentissage qui sont par définition instables ou en tout cas mouvants. Un scénario doit pouvoir selon nous être adapté, au moment de son exécution, par l'enseignant, exactement comme dans la classe lorsqu'un enseignant adapte naturellement son cours aux évènements qui surviennent.

On peut considérer un scénario pédagogique comme une entité qui évolue de façon continue en fonction des contextes dans lesquels il se déroule. Il doit donc dépendre fortement des circonstances de son utilisation. Ceci implique que l'enseignant doit pouvoir le gérer et le contrôler tout au long de son cycle de vie. Pour cela, un certain nombre de langages de modélisation pédagogique (EML) ont été proposés, tels que IMS-LD [IMS-LD 2003], ainsi que leurs outils supports comme *Reload Editor* [Relaod 2004]. La littérature le souligne bien [Griffiths et al. 2005] [Dodero et al. 2008], les enseignants éprouvent beaucoup de difficultés à manipuler les EML génériques et leurs outils supports pour concevoir et adapter les scénarios pédagogiques, dynamiquement, suivant l'évolution des contextes réels d'apprentissage. Il existe actuellement un profond fossé conceptuel entre ces instruments disponibles et les besoins réels des enseignants praticiens. La sémantique de ces langages est difficilement compréhensible et les outils restent inaccessibles car ils sont techniquement complexes pour des usagers praticiens.

D'après P. Dessus et K.Schneider, les EML sont destinés normalement à rationaliser le travail de l'enseignant et le rendre plus efficace. L'auteur a critiqué les EML : « *Ils présupposent que toute situation d'enseignement est statique, découpable et aisément descriptible, ce qui est démenti par les résultats de la recherche sur l'enseignement* » [Dessus et K.Schneider 2006]. Les auteurs poursuivent : un EML est « *un formalisme, un objet rigide, qui met l'accent sur la phase de planification, en la figeant, sans permettre une adaptation ou une modification en temps réel ou a posteriori de ses objets, et ce d'autant moins qu'il est véhiculé par un environnement informatique* ». De plus, « *scénariser l'activité d'enseignement peut la rendre figée et peu adaptée aux véritables contraintes de la situation* » [Dessus et K.Schneider 2006]. Enfin, les auteurs ont conclu qu'il est nécessaire que les EML « *prennent en compte le caractère dynamique et contextualisé de toute situation d'enseignement, sans jouer sur l'interopérabilité, permettent la saisie en direct des séquences d'enseignement pour un plus grand contrôle de l'enseignant (user empowerment, [Kynigos 2004]) sur les séquences produites* » [Dessus et K.Schneider 2006].

En outre, les EML, tels que la spécification IMS-LD [IMS-LD 2003], sont présentés comme des moyens imposés et génériques. Imposés car ils cadrent la description des scénarios selon leurs propres métaphores. Génériques car ils provoquent une perte de particularisme des contextes pédagogiques des enseignants [Cottier et El-Kechaï 2009].

En effet, le premier ensemble de travaux que nous avons cité dans notre positionnement scientifique (cf. section 2.3.1.1) s'intéresse particulièrement à IMS-LD et ses outils. Ces travaux s'inscrivent dans une approche interprétative de la conception pédagogique, ils visent à fournir des moyens nécessaires à l'enseignant pour lui permettre la conception et l'adaptation des scénarios pédagogiques. Au delà de ses lacunes en termes de difficulté d'appropriation de la sémantique, les principales limites d'IMS-LD en termes d'adaptation sont expliquées dans [Towle et Halm 2005]. Selon ces auteurs, IMS-LD, avec ses niveaux B et C, utilise des « propriétés » et des « conditions » pour définir des règles simples d'adaptation. Mais vue l'impossibilité qu'il y a à anticiper l'activité humaine, continuer à concevoir un scénario pendant son exécution semble plus efficace que l'adaptation par règles. De plus, la logique et les stratégies d'adaptation ne sont pas séparées du design pédagogique, ce qui empêche le partage et la réutilisation de l'expérience et de savoir-faire en terme d'adaptation entre les enseignants.

Quant aux outils de modélisation existants, ils sont difficiles à utiliser dans la pratique et ne fournissent pas des scénarios opérationnels. La plupart d'entre eux sont centrés sur les compétences des informaticiens. Ils ne sont pas intuitifs et ne conviennent pas aux usages

habituels et aux connaissances des enseignants. Par exemple, des outils supports d'IMS-LD ne permettent pas d'opérationnaliser le niveau «B» dynamique et le niveau «C» événementiel de cette spécification afin d'adapter la structure et la méthode d'apprentissage en temps réel.

Les difficultés que les enseignants praticiens rencontrent en utilisant des EML génériques et des outils complexes ont motivé de nombreux travaux (cf. section 2.3.2) s'appuyant sur l'IDM comme cadre théorique et pratique en vue d'éviter de proposer aux enseignants des langages ayant une sémantique (ou une métaphore) difficile à comprendre et des outils complexes et peu intuitifs. Ces travaux sont en lien direct avec notre objet d'étude. Certains s'appuient sur la fusion et la transformation des modèles qui impliquent toutefois une perte sémantique. Cela a poussé d'autres travaux, toujours dans le cadre d'IDM, à suivre une démarche DSM en vue d'aider les praticiens dans la définition de métamodèles décrivant les langages propres à leurs domaines métiers (cf. section 2.3.2.8). Ces travaux s'inscrivent dans une approche constructive de conception pédagogique. Ils n'abordent toutefois pas les aspects d'adaptation dynamique des scénarios.

Ces constats nous amènent à poser une question principale et un ensemble de questions sous-jacentes :

Comment instrumenter les enseignants par des langages compréhensibles et des outils conviviaux leur permettant de concevoir et d'adapter des scénarios pédagogiques en fonction de l'évolution des contextes, à un niveau élevé d'abstraction ?

- Par quelles méthodes ?*
- Selon quel processus ?*
- Avec quels moyens ? Quels outils ? Quelles techniques ?*

3.2 Hypothèses et objectifs de recherche

Nous considérons que l'ingénierie des EIAH doit s'opérer dans une perspective anthropocentrique qui s'appuie « *sur les compétences existantes des utilisateurs et cherche à les développer* » [Rabardel 1995]. D'après Rabardel, une telle technologie « *cherche à augmenter les degrés de liberté laissés aux opérateurs pour définir leurs propres objectifs et activités de travail. Le contrôle s'exerce dans le sens homme technologie et non dans le sens inverse* » [Rabardel 1995]. En ingénierie pédagogique, la parole et le pouvoir d'agir doivent être donnés aux usagers, les enseignants, qui peuvent jouer différents rôles : concepteurs et/ou tuteurs. Pour cela, nous considérons qu'il est nécessaire de définir un cadre d'ingénierie capable de permettre aux enseignants d'assumer leurs rôles, de manière à avoir la main sur les EIAH. C'est ce que nous désignons par « ouverture ». Henri affirme que cette approche « *ouvre en effet une perspective élargie et productive pour penser les pratiques de conception en formation à distance, notamment celles qui s'appuient sur le "design pédagogique". Les plus grandes faiblesses en matière de design pédagogique sont de s'arrêter à l'étape de la " livraison " à l'étudiant du produit de la conception* » [Henri 2007]. Les enseignants impliqués dans le déroulement de la session d'apprentissage doivent donc être capables de développer leurs propres instruments en agissant sur les artefacts qui leur sont proposés suite à la conception préalable, cela à travers un double processus : la réflexivité par la compréhension de la situation (*instrumentation*) et la catachrèse par l'adaptation et le détournement des artefacts, suivant le contexte et les besoins de l'activité (*instrumentalisation*), une genèse instrumentale [Rabardel 1995].

Dans notre travail, nous prenons le parti de considérer un EIAH existant et de chercher le meilleur moyen de le rendre ouvert pour ses utilisateurs enseignants. Nous avons choisi alors le scénario pédagogique comme un modèle suffisamment consistant et pertinent pour approcher l'ouverture des EIAH aux enseignants. Le scénario pédagogique en tant que modèle intermédiaire, d'après Baker, « *doit permettre une réflexion dialogique ouverte, mettant en jeu des compétences diverses chez de multiples acteurs* ». D'où, nous formulons notre première hypothèse comme suit :

- L'ouverture d'un EIAH aux enseignants est possible si l'on permet que la conception des scénarios pédagogiques puisse être continue, y compris en phase d'exécution, grâce à une adaptation en temps réel.

Pour ce faire, nous adoptons l'ingénierie dirigée par les modèles (IDM), et plus particulièrement l'approche DSM, comme cadre théorique et technique. Nous considérons en effet que cette ingénierie permet d'impliquer les enseignants dans le processus

d'instrumentation consistant à fournir des moyens conceptuels et informatiques (méthodes, langages, outils, techniques, etc.), pour leur faciliter la conception et l'adaptation des scénarios pédagogiques. Notre seconde hypothèse est :

- ❑ Il est possible de permettre à l'enseignant de concevoir et d'adapter des scénarios pédagogiques en lui proposant un éditeur basé sur le langage spécifique à son domaine métier.

Nous voulons ainsi dépasser les limites des travaux de la littérature et répondre à nos questions de recherche de façon opérationnelle. Afin d'ouvrir un EIAH existant aux enseignants utilisateurs, nous avons considéré qu'il fallait atteindre les objectifs suivants :

- ❖ Permettre à l'enseignant de poursuivre la conception des scénarios pédagogiques pendant leur exécution à travers l'adaptation en temps réel.
- ❖ Permettre la capitalisation, réutilisation et partage de savoir-faire en termes de décisions de conception et d'adaptation entre les membres de la communauté de pratique des enseignants.
- ❖ Éviter aux enseignants l'utilisation des langages génériques comme IMS-LD et des outils existants afin de surmonter les barrières conceptuelles et techniques.
- ❖ Collaborer avec des enseignants praticiens pour externaliser et formaliser leur métier sous forme d'un métamodèle d'expression des scénarios pédagogiques selon une approche constructive de conception. Cette collaboration doit s'appuyer sur l'EIAH mis en place comme vecteur de communication.
- ❖ Instrumenter l'enseignant par un éditeur dédié basé sur un ADSEGML¹⁵ permettant de travailler à un niveau élevé d'abstraction (*format interprété par l'humain dans un univers métier*), et de générer *directement* le format interprété par la machine (*univers technique*).

¹⁵ ADSEGML: *Adaptive Domain-Specific Graphical Educational Modeling Language*.

3.3 Méthodologie de notre recherche

L'objectif général de nos travaux est de proposer une approche pour rendre un EIAH ouvert à ses usagers. Dans notre contexte de recherche, nous nous intéressons en particulier à l'enseignant concepteur que nous considérons comme expert de son domaine [Ouraiba et al. 2011c]. Pour nous, il n'est pas juste un usager mais plutôt un concepteur impliqué dans le processus d'ingénierie de l'EIAH afin notamment d'exprimer ses besoins métiers et d'explicitier ses pratiques usuelles. Etant donné qu'il joue un rôle central dans le cycle de vie des scénarios pédagogiques, ce processus d'ingénierie pour rendre un EIAH ouvert vise à lui fournir des instruments logiciels. Ces derniers permettent de continuer la conception des scénarios pédagogiques pendant l'usage de l'EIAH, par l'adaptation du scénario pédagogique en fonction du contexte d'utilisation.

Pour ce faire, notre démarche de recherche a consisté à mener un travail s'appuyant sur des approches qui relèvent de différentes disciplines : les sciences humaines et sociales pour avoir des inspirations et des réflexions à propos de notre objet d'étude, ainsi que le génie logiciel et l'informatique pour proposer des solutions concrètes consistant à faciliter l'activité des enseignants usagers des EIAH.

Nous avons alors suivi une méthodologie de travail (cf. figure 28) qui prend appui sur des études théoriques et sur une étude pratique. Notre étude théorique a pour but de bien comprendre les aspects de la conception continue des scénarios pédagogiques à travers l'adaptation dynamique selon le contexte réel d'exécution. Nous avons donc effectué une étude bibliographique sur les différentes thématiques impliquées dans notre travail : la genèse instrumentale, l'ingénierie des EIAH, la conception pédagogique, l'adaptation, le Design Rationale, et l'IDM/DSM.

Cette étude pluridisciplinaire a abouti principalement à nos contributions scientifiques qui visent à atteindre nos objectifs de recherche évoqués dans la section précédente (cf. section 3.2). Nos trois propositions théoriques sont :

- ❖ un modèle rationnel des scénarios pédagogiques ouverts ;
- ❖ un processus itératif et incrémental d'ingénierie des SPO ;
- ❖ une méthode dirigée par les modèles et spécifique au domaine pour instrumenter le processus d'ingénierie des SPO d'un EIAH.

Quant à notre première proposition, nous nous sommes inspirés principalement de Design Rationale qui est une approche du génie logiciel consistant à définir la structure d'un artefact en plusieurs variantes. Cela nous a permis de définir une structure générique des SPO sous forme d'un modèle afin d'assurer la consistance et la pertinence dans toutes les situations d'usage possibles. Notre seconde proposition est définie dans le but de guider les enseignants concepteurs à travers le processus continu de conception et d'adaptation des SPO structurés suivant le modèle rationnel proposé. Dans notre troisième proposition, nous nous sommes appuyés sur l'IDM/DSM comme cadre théorique et technique pour définir une méthode dirigée par les modèles et spécifique au domaine métier. Cette méthode consiste à fournir des instruments dédiés (langage métier et éditeur graphique) aux enseignants concepteurs pour leur permettre d'effectuer la conception et l'adaptation des SPO à un niveau abstrait. Or, grâce à des techniques de méta-modélisation on considère un SPO comme un modèle en conformité à un métamodèle défini à base du modèle rationnel des SPO, et qui décrit le langage métier (ADSGEML) des enseignants. L'outillage d'IDM/DSM permet de définir ce métamodèle et de développer par conséquent un éditeur graphique dédié, contenant la sémantique spécifique (concepts, règles, etc.) du domaine métier traité.

Afin de vérifier nos propositions dans des situations réelles d'utilisation, nous avons effectué une étude pratique sur le domaine métier d'un EIAH existant nommé « Hop3x ». Ce dernier a été conçu pour dérouler des sessions de travaux pratiques de la programmation Java. L'objectif ultime est de rendre cet EIAH ouvert aux enseignants qui veulent l'utiliser. Nous avons donc appliqué notre méthode pour instrumenter le processus d'ingénierie et de réingénierie des sessions d'apprentissage ouvertes. Nous avons donc fait appel aux concepteurs initiaux de Hop3x en tant qu'experts de domaine, dans le but d'extraire et de cristalliser les connaissances métiers et d'étudier les possibilités d'adaptation des sessions. Par conséquent, l'ADSGEML et son éditeur graphique ont été élaborés pour permettre aux enseignants, même non concepteurs initiaux de cet EIAH, de concevoir et d'adapter des sessions d'apprentissage ouvertes à un niveau élevé d'abstraction. Pour l'implémentation nous avons utilisé les frameworks EMF/GMF (*Eclipse Modeling Framework / Graphical Modeling Framework*), puisqu'ils facilitent le développement d'éditeurs graphiques conformément à un métamodèle structuré, qui décrit l'ADSGEML des enseignants usagers de Hop3x.

La figure 28 suivante illustre notre méthodologie de recherche.

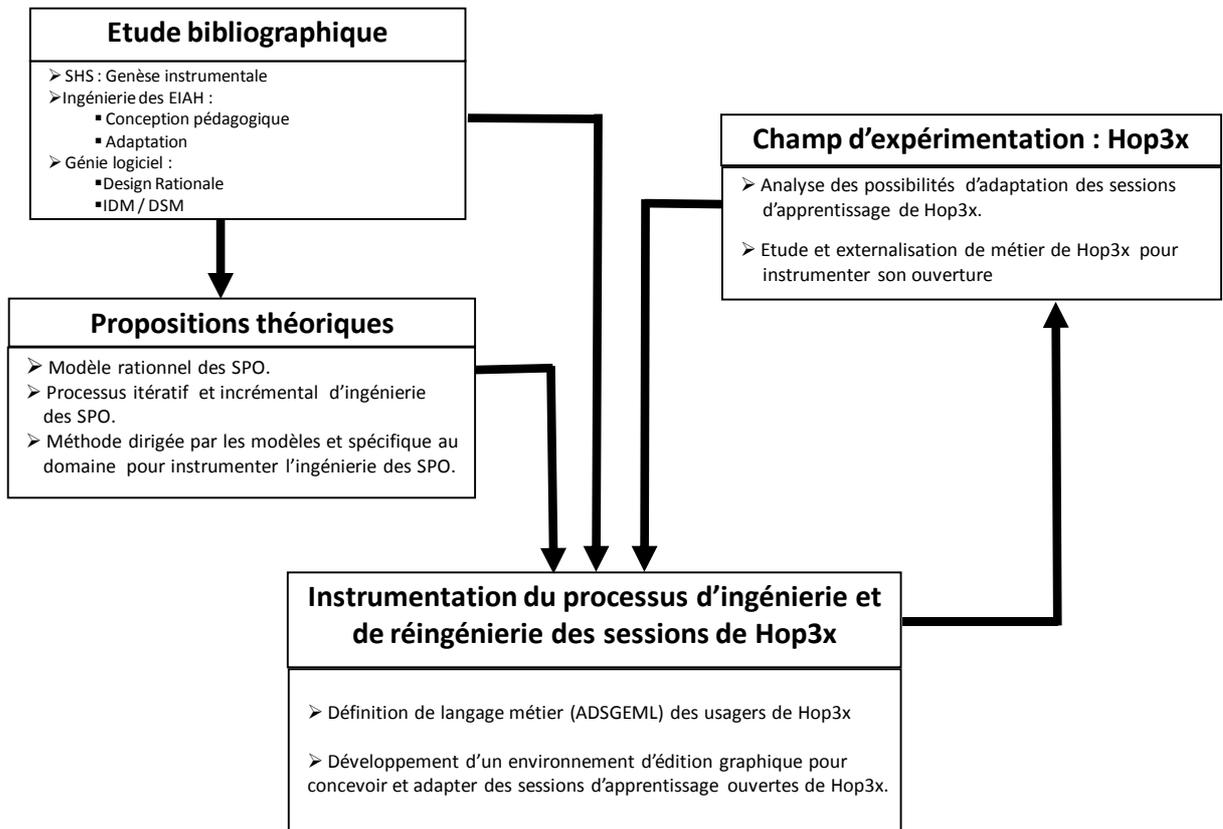


Figure 28 : Notre méthodologie de recherche.

II. Deuxième partie :

Propositions scientifiques

4

Modèle Rationnel des SPO

Sommaire

- 4.1 Introduction
- 4.2 Notion d'un scénario pédagogique ouvert (SPO)
- 4.3 Modèle rationnel des SPO
- 4.4 Métamodèle implémentant le modèle rationnel des SPO
- 4.5 Conclusion

4.1 Introduction

La logique d'adaptation dans les EIAH n'est généralement pas explicitée et formalisée ; quand elle est spécifiée, c'est à un niveau assez technique, ce qui la rend difficile à contrôler et maintenir. Nous voulons par notre travail proposer une approche permettant de spécifier la logique d'adaptation à un niveau « méta », et de la séparer par rapport à l'architecture interne du système qui est complexe. Nous proposons pour cela de concevoir, à un niveau abstrait, des scénarios pédagogiques ouverts (SPO) opérationnalisables sur l'EIAH, et de continuer leur conception pendant l'exécution à travers des adaptations dynamiques.

Ce chapitre est consacré à notre première proposition scientifique : le modèle rationnel des SPO. Nous précisons dans un premier temps ce que nous entendons par SPO. Nous présentons ensuite le modèle rationnel des SPO que nous proposons, puis le métamodèle qui implémente notre modèle rationnel des SPO.

4.2 Notion d'un scénario pédagogique ouvert (SPO)

L'origine des scénarios remonte aux sciences de management, planification militaire et théâtre [Jarke et al. 1998], mais ils ont suscité un grand intérêt selon Wild [Wild 2006] dans le domaine de génie logiciel depuis quelques années. Wild a affirmé qu'il est largement convenu dans la littérature que les scénarios sont assez concrets et suffisants pour communiquer l'essence d'une situation concernant une préoccupation ou un besoin sans qu'il soit nécessaire d'apprendre en détail les techniques complexes de modélisation [Wild 2006].

Carroll a relevé [Carroll 2000] cinq avantages à l'utilisation des scénarios dans des contextes de conception :

- 1) ils évoquent la réflexion dans le contexte du travail de conception ;
- 2) ils sont à la fois concrets et flexibles ;
- 3) ils offrent de multiples points de vue ;
- 4) ils peuvent être abstraits et classés ;
- 5) ils promeuvent le travail orienté communication.

La notion de scénario pédagogique a connu un fort essor, tout particulièrement dans le domaine des EIAH, avec l'apparition de l'approche centrée activité, où l'enseignant est désormais considéré comme « scénariste ». Le bénéfice de l'utilisation des scénarios pédagogiques pour approcher l'ingénierie des EIAH, comme l'indique Hummel [Hummel et al. 2004], réside dans le fait que l'attention principale est mise sur les activités d'apprentissage qui doivent être réalisées pour atteindre les objectifs pédagogiques [Vantroys et Peter 2005].

Cependant, Park et Lee évoquent une question centrale et persistante qui concerne la mise à disposition d'environnements pédagogiques et les conditions sous-jacentes qui peuvent satisfaire « individuellement » des différents objectifs éducatifs et des capacités d'apprentissage [Park et Lee 2001]. Les approches pédagogiques qui sont destinées à répondre aux besoins des apprenants, individuellement, relèvent de ce que l'on désigne par « *enseignement adaptatif* » (en anglais « *Adaptive instruction* ») [Como et Snow 1986]. Plus spécifiquement, l'enseignement adaptatif se réfère à des interventions éducatives visant à bien assimiler les différences individuelles chez les apprenants tout en aidant chaque apprenant à développer les connaissances et les compétences nécessaires pour apprendre une tâche.

D'après Park et Lee [Park et Lee 2001] :

« *Adaptive instruction is generally characterized as an educational approach that incorporates **alternative** procedures and strategies for instruction and resource utilization and has the built-in **flexibility** to permit students to take various routes to, and amounts of time for, learning [Wang et Lindvall 1984]* ».

La flexibilité d'un scénario pédagogique peut donc être assurée par la définition de différentes alternatives. Quant à la notion de flexibilité, on trouve dans la littérature des travaux qui relèvent du domaine de la gestion des processus de travail (Workflow), [Schonenberg et al. 2007] considèrent que l'une des caractéristiques clés de déroulement efficace des processus est leur capacité de faire face à la fois aux changements prévus et imprévus de l'environnement de leur exécution. Cette qualité des processus - qualifiée de flexibilité - reflète leur capacité de s'adapter en faisant varier des parties du processus concernées par le changement en cours, cela tout en conservant la structure essentielle des parties qui ne sont pas concernées par ce changement [Schonenberg et al. 2007].

En outre, les auteurs ont distingué quatre types de flexibilité d'un processus que nous avons présentés dans (cf. section 2.1.3.3) [Schonenberg et al. 2007] :

- ❖ *Flexibilité par conception* : pour le traitement des changements prévus dans le contexte d'exécution, où les stratégies de soutien peuvent être définies en amont, au moment de la conception. (Nous qualifions ce type de flexibilité de proactif. En revanche, les trois types suivants sont réactifs).
- ❖ *Flexibilité par déviation* : pour la manipulation des comportements imprévus occasionnels, où la différence avec les comportements attendus est minimale.
- ❖ *Flexibilité par sous-spécification* : pour le traitement des changements prévus dans le contexte d'exécution, où les stratégies ne peuvent pas être définies au moment de la conception initiale, parce que la stratégie finale n'est pas connue à l'avance ou qu'elle n'est pas applicable.
- ❖ *Flexibilité par changement* : pour la manipulation des comportements imprévus occasionnellement, où les différences nécessitent des adaptations de processus, ou bien pour la manipulation des comportements souvent imprévus.

Schonenberg et al. concluent en affirmant que la flexibilité des processus vise alors à améliorer leur capacité à s'adapter aux évolutions des contextes d'exécution sans nécessiter une redéfinition complète des spécifications initiales. Par conséquent, « *un processus ne peut être considéré comme flexible que s'il est possible de le changer sans avoir besoin de le remplacer complètement* » [Schonenberg et al. 2007].

Naturellement, dans les situations de conception, le concepteur dispose, au départ, d'une représentation mentale incomplète et imprécise de l'objet à concevoir [Eastman 1969] [Simon 1973]. L'incomplétude de la représentation laisse selon P. Rabardel [Rabardel 1995] l'artefact inachevé et ainsi « **ouvert** » sur de multiples possibles. Rabardel affirme qu'il s'agit d'une « *incomplétude construite, recherchée, gérée et maintenue, en tant que telle, par un sujet compétent voire expert, en tant que moyen de gestion de la complexité des situations* :

- *au plan synchronique, l'incomplétude fournit le "jeu" nécessaire pour que les différentes modalités de régulation de l'action puissent être articulées, coordonnées en temps réel afin de gérer la singularité imprévisible des situations ;*
- *au plan diachronique, l'incomplétude permet le traitement de problèmes de niveaux différents, à des moments différents. Le flou maintenu à un niveau de traitement limite les contraintes répercutées à d'autres niveaux ».*

La conception des scénarios pédagogiques requiert des modalités de conception qui ne sont pas classiques [Cottier et El-Kechaï 2009]. Les spécifications ne doivent pas s'arrêter à la phase d'instanciation, elles doivent être continuées durant l'exécution en permettant aux usagers de faire les ajustements nécessaires. Notre choix d'adopter l'approche instrumentale [Rabardel 1995] comme source d'inspiration se trouve ici conforté. Cette approche ouvre en effet une perspective élargie et productive pour étudier les pratiques de conception en design pédagogique [Henri 2007]. Cette approche suggère une continuation de la conception des artefacts dans l'usage. Les usagers doivent être capables d'agir sur les artefacts préétablis (scénarios prédictifs) et d'élaborer leurs propres instruments(scénarios effectifs). Les artefacts proposés doivent être suffisamment ouverts pour permettre un double processus de genèse instrumentale : instrumentation et instrumentalisation.

Rabardel affirme que : « *une certaine anticipation des usages fonde certes la conception des artefacts, mais leur complexité est telle, la diversité des possibilités est si grande, que seule une faible partie d'entre elles peut être anticipée. C'est dans la mise en œuvre des systèmes que les instrumentalités potentielles émergent, se révèlent ou s'inventent, sont conçues le plus souvent par les utilisateurs eux-mêmes* » [Rabardel 1995].

Dans la pratique, l'enseignant ne suit pas aveuglément des scénarios prédictifs qui ne constituent qu'une base de départ pour lui [Zarraounandia 2006b]. L'attitude des apprenants ne peut pas être prévue dans le détail pendant la phase de conception, mais elle peut être supervisée et ajustée pendant le déroulement de la session sur la plateforme d'apprentissage. Selon Rogalski [Rogalski 2003], l'enseignant dans son activité doit gérer un « *environnement dynamique ouvert* ». « Dynamique » parce que l'apprentissage des apprenants évolue même sans intervention de l'enseignant, on parle alors d'évolution spontanée. « Ouvert » parce que l'enseignant ne peut prévoir ni l'évolution spontanée des apprenants ni l'effet de ses interventions [Roditi 2003].

Nous considérons alors qu'il est nécessaire de penser à concevoir des scénarios que l'on pourrait qualifier d'ouverts, donnant la possibilité aux usagers d'effectuer eux-mêmes la conception et l'adaptation. À notre sens, la question posée par l'ingénierie d'un scénario pédagogique n'est donc plus seulement de produire un modèle correspondant à une spécification donnée, mais d'élaborer un modèle adaptable, souvent en cours d'usage, en fonction de l'évolution du contexte et des besoins des usagers. L'activité de conception est donc située et continue dans le processus d'usage. Elle ne se réduit pas à l'acte de modélisation préalable de l'artefact qui est exogène au contexte réel de son usage, elle se poursuit dans l'activité des usagers eux-mêmes [Cottier et EL-Kechai 2009]. Cela passe par l'élaboration puis l'utilisation de modèles endogènes aux contextes d'usage.

Les scénarios que nous considérons alors comme ouverts visent à ce que les apprenants apprennent avec beaucoup plus de souplesse et de liberté en choisissant par exemple le cheminement convenable en fonction de leurs contextes d'apprentissage correspondants. Un problème inévitable surgit alors car dans certains cas une liberté totale ou trop importante peut nuire à l'objectif pédagogique, il y a nécessité de protéger peu ou prou un noyau insécable du scénario. Selon Pernin et Lejeune : « *un scénario ouvert ou adaptable décrit dans les grandes lignes les activités à réaliser et permet de déléguer aux acteurs humains les choix ne pouvant être anticipés sans nuire à la qualité des objectifs d'apprentissage poursuivis* » [Pernin et Lejeune 2004].

À notre sens, pendant le déroulement d'un scénario ouvert, certains éléments obligatoires doivent être pris en compte, mais leur organisation est libre. Par exemple, les rôles doivent réaliser des activités pour atteindre leur but en toute liberté. Les rôles doivent être libres d'agir dans le cadre des règles, le déroulement du scénario peut toujours diverger et connaître des variantes (ou alternatives selon [Park et Lee 2001]). Prévoir un certain nombre

d'éléments obligatoires est cependant nécessaire, c'est ce qui permet de conserver une cohérence du scénario sans avoir à être trop dirigiste [Ouraiba 2010].

Dans le domaine des scripts d'apprentissage collaboratif, Dillenbourg et Tchounikine [Dillenbourg et Tchounikine 2007] [Tchounikine 2008] ont confirmé qu'un script doit être flexible, c'est-à-dire adaptable par les apprenants et/ou les enseignants, pour éviter de décrire plus que le nécessaire (*overscripting*), ce qui inhibe des mécanismes naturels d'interaction et limite la collaboration entre les membres des groupes. Nous partageons l'idée de ces auteurs qui consiste à distinguer deux parties dans un script flexible (scénario ouvert à notre sens), contraintes extrinsèques et contraintes intrinsèques. La partie extrinsèque définit l'espace dans lequel le script doit pouvoir être adaptable par les enseignants et / ou étudiants. La partie intrinsèque crée les limites de la flexibilité, c'est-à-dire ce qui ne peut pas être modifié sans que le script perde sa « *raison d'être* ».

Sur notre terrain de la scénarisation pédagogique, de manière analogue à [Dillenbourg et Tchounikine 2007] [Tchounikine 2008], nous distinguons deux catégories d'éléments constitutifs d'un scénario pédagogique ouvert. La première catégorie est rigide au contraire de la seconde qui est adaptable en comportant des éléments modifiables (points ouverts). Cette dernière catégorie comporte des points de variation qui permettent de spécifier différentes manières opportunes de dérouler une session d'apprentissage que l'on désigne par « variantes ». Un point de variation peut être n'importe quelle entité du scénario pédagogique : une activité ou étape d'apprentissage, un rôle, un attribut ou une propriété, etc. La représentation des différentes variantes d'une session d'apprentissage consiste à les combiner dans un seul artefact intégré, nommé « *scénario pédagogique ouvert - SPO* ». Grâce à cet artefact l'enseignant concepteur peut rassembler plusieurs variantes d'un même scénario qui correspondent à certaines variations du contexte d'utilisation.

Quant à la notion de contexte, Simondon qualifie le contexte d'utilisation comme : « *un milieu que l'être technique crée autour de lui-même et qui le conditionne comme il est conditionné par lui. Ce milieu à la fois technique et naturel peut être nommé milieu associé* » [Simondon 1958]. Notons que, Zarraonandia [Zarraonandia et al. 2006] le confirme aussi plusieurs définitions et différentes notions du terme contexte et « *context-aware* » peuvent être trouvées dans la littérature de l'informatique [Pasco 1998] [Schilit 1994] [Chen 2000] [Dey 2001]. La définition de Dey est la plus partagée : « *le contexte correspond à toutes les informations qui peuvent être utilisées pour caractériser la situation d'une entité. Une entité est une personne, un lieu, ou l'objet que l'on considère pertinent pour l'interaction entre un utilisateur et une application, y compris l'utilisateur et l'application eux-mêmes* » [Dey 2001].

Dans le domaine des EIAH, Pernin définit le contexte comme : « *une configuration particulière, organisationnelle et temporelle, des moyens humains et matériels, construite à partir des moyens offerts de façon générale par le dispositif de formation* » [Pernin 2007]. À notre sens, le contexte d'enseignement/apprentissage peut être caractérisé par les informations qui influencent l'exécution de SPO. Ces informations décrivent l'environnement et les différentes circonstances de la situation d'enseignement/apprentissage en considération, telles que : les contraintes à respecter, les indicateurs calculés avant, lors et/ou après le déroulement d'une session d'apprentissage, les ressources disponibles et les supports mis en disposition, les compétences et habitudes des enseignants impliqués, les profils des étudiants concernés (informations biographiques, pré-requis et compétences, motivations, préférences, etc.), les conditions d'accessibilité, le type d'institution de formation et sa culture, etc.

4.3 Modèle rationnel des SPO

Afin d'assurer les différents types de flexibilité dans un scénario pédagogique ouvert (SPO) (cf. section 2.1.3.3), nous avons défini un modèle générique des SPO qui s'appuie sur l'approche du Design Rationale (cf. section 2.2). C'est la raison pour laquelle nous qualifions notre modèle de SPO de « rationnel ». Notons que le Design Rationale a de nombreux apports pour les processus de conception en général. Ces apports concernent notamment la vérification de la conception, son évaluation, sa maintenance, sa réutilisation, son enseignement, sa communication, son assistance et enfin sa documentation [Burge et Brown 1998].

Cependant, contrairement aux autres domaines, notamment les IHM (Interface Homme-Machine), le Design Rationale demeure moins utilisé dans l'ingénierie des EIAH. Nous pensons que le Design Rationale peut être adopté comme un cadre de travail pour justifier les décisions prises au cours du processus de conception des scénarios pédagogiques ouverts, cela afin de permettre la compréhension, la justification, l'adaptation, et/ou la réutilisation de la production de la conception [Ouraiba et al. 2011 a] [Ouraiba et al. 2011b]. De plus, étant donné qu'un SPO est le produit d'un processus de conception, il est intéressant de capter les connaissances exploitées au cours de ce processus et de les capitaliser pour leur partage et leur réutilisation. Cela permet de comprendre comment et pourquoi chacun des éléments du SPO a été défini.

En outre, Gruber et Russell [Gruber et Russell 1990] ont affirmé que le Design Rationale consiste à répondre à une préoccupation par la prise en compte du contexte du problème à résoudre de façon claire et concise. Par conséquent, les justifications des choix ne sont pas absolues, ce sont des déclarations endogènes à un contexte donné. Elles devront être basées sur un nombre important de connaissances sur le processus de conception, le domaine, et le contexte du problème à résoudre.

Quant aux formalismes du Design Rationale, plusieurs ont été proposés (cf. section 2.2.3), nous citons à titre d'exemples : IBIS [Kunz et Rittel 1970], PHI [McCall 1987], DRL [Lee 1991], QOC [Maclean et al. 1991], etc. Bien qu'il existe des revendications de plusieurs avantages d'utilisation de ces formalismes pour capturer la logique de conception, il n'est pas clair toutefois comment ces modèles peuvent être adoptés par les praticiens et combien leur utilisation coûte dans la réalité [Tang et al. 2006]. Karsenty [Karsenty 2001] a confirmé que les études effectuées en la matière ont conclu que les concepteurs préfèrent une méthode qui nécessite moins d'effort pour capitaliser la logique de conception tout en favorisant la possibilité de sa réutilisation, ces exigences des praticiens ont naturellement conduit à opter pour le modèle QOC¹⁶ (cf. section 2.2.3.1) que nous trouvons en effet assez simple et efficace pour représenter la logique de la conception pédagogique [Ouraiba et al. 2011a] [Ouraiba et al. 2011b].

Nous rappelons brièvement ici le principe de QOC. C'est une notation semi-formelle qui permet de produire une présentation graphique de Design Rationale [MacLean et al. 1991], où les diagrammes peuvent aider le concepteur à expliquer, à élaborer, à comparer et à examiner les idées de conception et de questions [McKerlie et MacLean 1993]. Le principe est le suivant : pour répondre à une « question », le concepteur est censé définir un ensemble de solutions possibles, « options ». Chaque option est spécifiée pour satisfaire un ensemble d'exigences ou des propriétés souhaitées décrites par des « critères ».

Villiot-Leclercq souligne que « *les choix pédagogiques de l'enseignant sont rarement explicités [...]* » [Villiot-Leclercq 2007]. Il est difficile alors de comprendre réellement les critères qui l'ont poussé à prendre telle décision plus qu'une autre dans un contexte d'enseignement/apprentissage donné. Nous nous sommes inspirés alors de QOC pour proposer un *modèle rationnel des scénarios pédagogiques ouverts* (cf. figure 29). Ce modèle graphique aide à capturer, à un niveau abstrait, la logique de conception pédagogique de l'enseignant dans le but d'améliorer la qualité des SPO qu'il produit [Ouraiba et al. 2011a] [Ouraiba et al. 2011b]. Cela grâce à l'argumentation des décisions par l'explication des

¹⁶ QOC : Questions, Options, Criteria.

critères considérés afin de capturer les variantes évaluées pour éviter la duplication des efforts dans les futurs cycles de vie du SPO.

En effet nous voulons, par ce modèle rationnel, permettre aux enseignants d'avoir des scénarios qui s'adaptent à la particularité de chaque situation d'apprentissage (des variantes spécifiques aux contextes d'apprentissages); et aussi de garder un caractère ouvert au scénario pour qu'il puisse intégrer d'éventuelles nouvelles variantes autres que celles déjà répertoriées.

Nous considérons alors « **l'objectif d'apprentissage** » ciblé par un SPO de la même manière que l'élément *question* est considéré dans le QOC. Afin d'atteindre cet objectif l'enseignant concepteur doit définir deux couches dans la structure de SPO. La première couche est statique, elle doit être pré-spécifiée pendant la phase de conception initiale. La seconde est adaptable, elle comporte l'ensemble des variantes que le SPO peut avoir suivant les différents contextes d'enseignement/d'apprentissage.

- ❖ **La couche statique.** Elle est appelée *intrinsèque* par [Dillenbourg et Tchounikine 2007]. Cette partie basique du SPO est relative à sa raison d'être. Elle contient un ensemble de contraintes et d'éléments à définir impérativement pour préciser le noyau insécable du SPO afin qu'il conserve sa cohérence. Selon Roditi [Roditi 2003], l'enseignant garantit une ligne directrice forte qui lui permettra de rester dans ce que Rogalski appelle « *l'enveloppe des trajectoires acceptables du déroulement* » [Rogalski 2003].
- ❖ **La couche des variantes.** La multitude des contextes possibles d'enseignement/apprentissage dans lesquels le SPO pourra être exécuté nécessitent à l'enseignant concepteur de définir un certain nombre de « **variantes** » qui correspond au concept *options* de QOC. Un SPO doit être défini de manière à être personnalisé pour répondre aux exigences spécifiques d'un contexte particulier en prenant en compte uniquement la variante la plus pertinente pour le contexte en considération. En outre, à la différence de QOC, nous avons introduit le concept intermédiaire « **contexte** » dans le but d'associer chacune des variantes avec son contexte de pertinence qui décrit l'ensemble des circonstances convenables pour son exécution parfaite. Dans notre modèle chaque contexte de pertinence est caractérisé par un ensemble d'informations. Ces « **informations** », considérées dans QOC comme des *critères*, peuvent être de types différents et hétérogènes, par exemple : des indicateurs pédagogiques calculés, les profils des apprenants impliqués, etc.

Toutefois, dans la réalité, les contextes possibles d'enseignement/apprentissage ne sont pas souvent prédictibles. De ce point de vue, nous avons distingué deux parties contextuelles dans cette couche des variantes : une partie contextuelle « anticipée » et une autre « ouverte ».

- ❑ **la partie contextuelle anticipée.** Les éléments des variantes de cette partie sont « pré-spécifiés » en fonction des contextes prévus d'exécution du SPO. Cette partie contient des multi-représentations structurelles et sémantiques du SPO où l'enseignant concepteur doit spécifier explicitement en amont un ensemble d'alternatives (variantes) prévues du SPO en fonction des contextes qui peuvent être anticipés. Un modèle d'un SPO peut contenir de nombreuses variantes correspondant à des décisions qui doivent être spécifiées préalablement pour donner une logique à la conception d'un tel SPO. Dans un contexte donné, l'enseignant peut définir les adaptations possibles de son SPO en précisant la variante la plus appropriée à déployer, (cf. figure 29, variante 3 par exemple). Une fois toutes ces décisions définies, le SPO peut être individualisé, selon un contexte particulier, en supprimant les variantes qui ne sont plus pertinentes, conduisant à un SPO contextualisé. En effet, cela permet d'avoir une conception plus fine des SPO en produisant à chaque fois une spécification exécutable pour satisfaire un ensemble d'exigences spécifiques de la situation d'apprentissage considérée.
- ❑ **la partie contextuelle ouverte.** C'est la partie des exceptions où le SPO est exécuté dans des circonstances inattendues qui définissent un contexte imprévu. Comme nous l'évoquons dans les chapitres précédents, les contextes réels d'apprentissage sont par définition mouvants et peu prédictibles. Les informations qui définissent ces contextes associés à des variantes restent encore imprévues, elles ne peuvent pas toujours être connues en amont. De nouvelles variantes vont alors émerger en temps réel dans cette partie du SPO. Dans la figure 29, ces variantes sont cadrées par une ligne pointillée (cf. figure 29, variante 4). La spécification de leurs éléments est alors reportée au moment du déroulement des sessions d'apprentissage sous forme de points ouverts. Dans la plupart des cas, il n'est pas aisé, voire impossible, de modéliser complètement un SPO. En raison de l'absence d'un modèle complet, l'adaptation dynamique doit être alors possible durant l'exécution pour permettre de continuer la conception du SPO pendant son usage. En fait, certains éléments ne peuvent être définis complètement que dans l'action (*in situ*) comme par exemple les ressources disponibles qui peuvent être utilisées dans une activité, etc. A la fin du processus d'apprentissage, le scénario effectif résultant pourra être capitalisé et, dans certains cas, nourrir la réingénierie du scénario prédictif par l'ajout d'une nouvelle variante anticipée, caractérisée par les informations décrivant son contexte de pertinence, cela dans une perspective de Design Rationale.

Dans chaque variante (anticipée ou émergente) l'enseignant peut définir des comportements désirables et/ou indésirables en spécifiant respectivement ce que l'apprenant doit faire et/ou ce qu'il ne doit pas faire. La figure 29 illustre notre modèle rationnel des SPO.

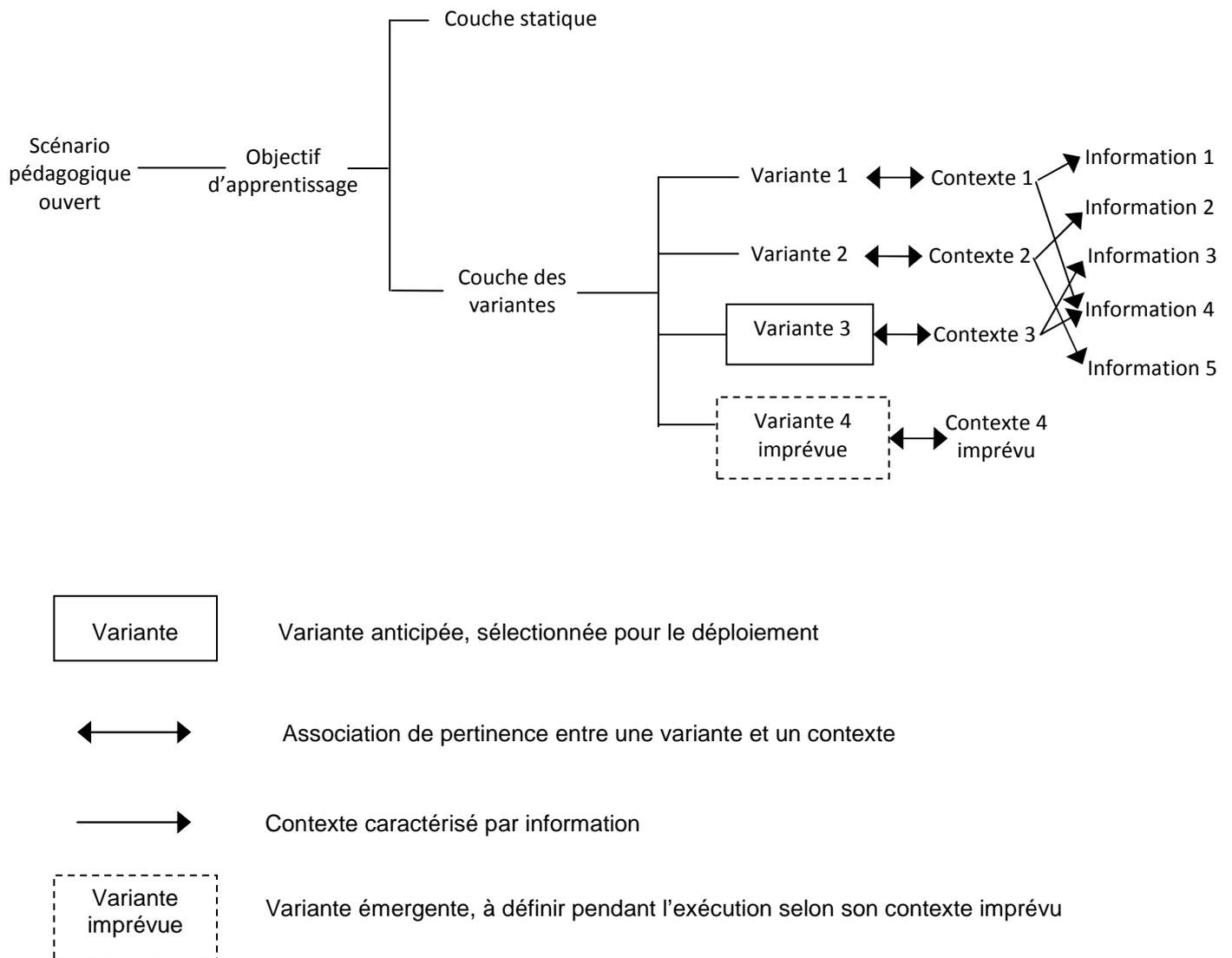


Figure 29 : Modèle rationnel des SPO [Ouraiba et al. 2011b].

Afin d'illustrer ce modèle nous donnons ici l'exemple de la mise en place d'une session d'apprentissage à la demande d'une université avec la plateforme de formation Moodle.

❖ **Énoncé de la situation d'enseignement/apprentissage :**

Pour permettre aux étudiants de s'initier aux bases de données dans un domaine disciplinaire particulier (ex : commerce, informatique, communication, etc.), un module de bases de données est proposé. Ce module doit être défini pour un parcours particulier (ex: parcours classique ou en alternance). Dans chaque parcours on proposera un seul module. Pour chaque module on trouvera au minimum deux séquences (ex : cours simple, cours avancé, TD, TP, etc.). Dans chaque séquence on définira un ensemble d'activités (ex : lire un document, rechercher des informations sur le Web, faire un document synthétisant cette recherche à déposer sur la plateforme, répondre à un quizz, etc.). Chaque séquence est destinée à un groupe de formation.

Un tuteur va animer la session avec un chat et un forum (il doit fixer un RDV pour une session de formation avec les apprenants/groupes d'apprenants via la messagerie), valider les documents déposés par les apprenants, consulter les informations concernant des activités de ses apprenants (nombre de connexions, messages envoyés, documents consultés, documents envoyés, statistiques d'avancement dans les modules, etc.).

❖ **Catégorisation des éléments selon le modèle rationnel des SPO :**

Le SPO est mentionné ici comme « *module de BDD* », son objectif d'apprentissage est donc « *d'initier les étudiants à créer et utiliser des BDD dans leurs domaines respectifs* ». Quant aux variantes, elles sont considérées ici comme des « *séquences* », deux sont exigées au minimum. Donc pendant la phase de conception du SPO, deux variantes doivent être pré-spécifiées en amont, où chacune se constitue d'un ensemble d'activités telles que : « *lire un document, rechercher des informations sur le web, faire un document synthétisant cette recherche à déposer sur la plateforme, répondre à un quizz, etc.* ». Chacune des variantes correspond à un contexte particulier. Les informations qui caractérisent le contexte ici sont : « *domaine disciplinaire, parcours suivi, groupe d'apprenants concernés* ». Quant aux éléments impératifs de la couche statique qui doivent être pré-spécifiés, on constate que « *un tuteur doit être impliqué pendant le déroulement de la session, il doit valider les documents déposés par des apprenants et il doit consulter les informations concernant leur activités, un chat et un forum doivent être utilisés également* ».

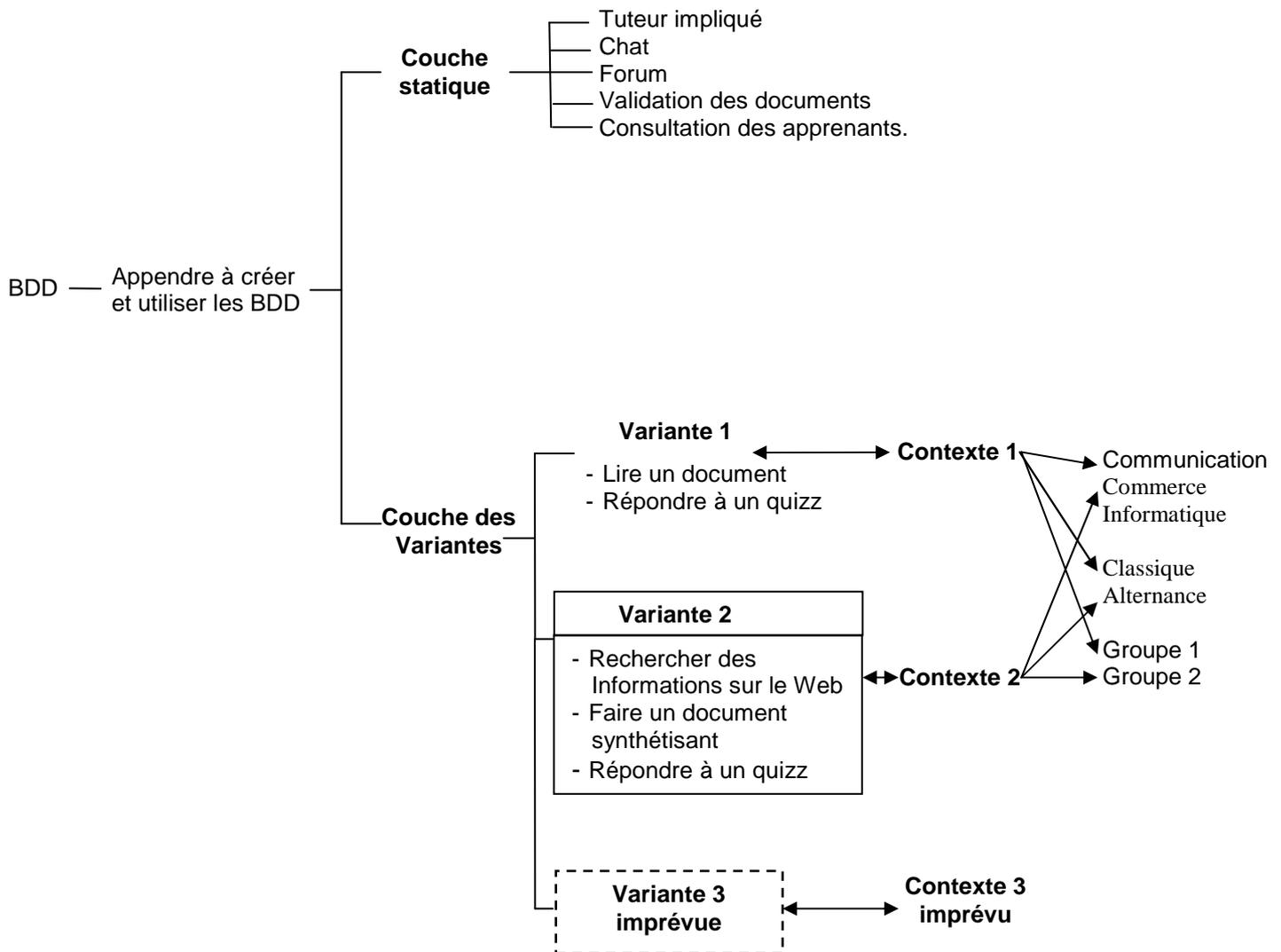


Figure 30 : Exemple du résultat de conception d'un SPO.

Cette figure 30 montre un exemple de conception d'un SPO. La variante 2 est encadrée, le résultat de cette conception est donc destiné à être déployé dans le contexte 2 qui concerne le groupe 2 d'apprenants qui suivent une formation par alternance dans le domaine du commerce. Quant aux activités de la variante 3, elles ne peuvent pas être anticipées car le contexte d'exécution est encore imprévu.

Toutefois, il n'est pas évident pour des enseignants concepteurs de structurer explicitement les scénarios narratifs selon le modèle rationnel des SPO, et de les opérationnaliser sur la plateforme d'apprentissage. Pour cela, nous nous intéressons dans notre objet d'étude à fournir des moyens nécessaires pour faciliter la conception et

l'opérationnalisation des scénarios à un niveau abstrait. A cet effet, nous avons pensé à formaliser notre modèle rationnel des SPO comme métamodèle, qui servira à la construction des instruments dédiés facilitant la conception et l'adaptation des SPO.

4.4 Métamodèle implémentant le modèle rationnel des SPO

Grâce à la formalisation de la logique de conception pédagogique, des lacunes et des contradictions de conception pourront être identifiées très tôt. Cette formalisation aidera notamment à l'indexation et à la récupération des décisions de conception. Pour cela, nous avons implémenté le modèle rationnel des SPO sous forme d'un métamodèle ayant pour but de supporter concrètement notre proposition et de proposer une notation des SPO dans un format interprétable par la machine. Ce métamodèle nous permettra, dans une approche IDM, de développer des outils d'édition assistant des enseignants concepteurs dans la conception et l'adaptation des SPO. Nous proposons donc ce métamodèle comme un cadre favorisant la compréhension de la logique d'adaptation des SPO par des enseignants concepteurs. Il constitue pour eux un guide et une base de raisonnement afin de :

- a) Assurer la consistance des scénarios pédagogiques pendant l'adaptation :
 - consistance sémantique : afin d'assurer une adaptation efficiente des SPO et garder la simplicité requise pour faciliter le passage à une échelle plus grande, ce métamodèle consiste à isoler la logique d'adaptation des autres composants de l'EIAH, cela en permettant à l'enseignant de catégoriser les éléments de son SPO, en termes de variabilité, selon les différents contextes d'exécution, à un niveau « méta », indépendamment des aspects techniques de l'architecture interne du système d'apprentissage.
 - consistance syntaxique : le métamodèle assure la conformité et la cohérence des SPO.
- b) Permettre l'explicitation des choix pédagogiques : grâce à ses concepts génériques, le métamodèle permet de spécifier les différentes dimensions classiques d'un scénario pédagogique, en termes d'objectifs d'apprentissage, d'acteurs, d'activité, de ressources, etc.
- c) Permettre la continuation de la conception des SPO pendant l'exécution, suivant les différentes approches d'adaptation : proactive par conception, réactive par déviation, par la sous-spécification, ou par changement.

- d) Assister les enseignants concepteurs dans la prise de décisions afin de faciliter le choix des variantes appropriées, tout en assurant :
- l'abstraction nécessaire pour permettre aux enseignants de prendre les décisions d'adaptation à un niveau abstrait indépendant des détails non pertinents, cela afin de pouvoir distinguer les aspects d'adaptation par rapport à ceux de conception. Cette abstraction permet en outre de favoriser la capitalisation et la réutilisation des décisions d'adaptation.
 - la justification des décisions par des informations caractérisant le contexte de pertinence de chaque variante.

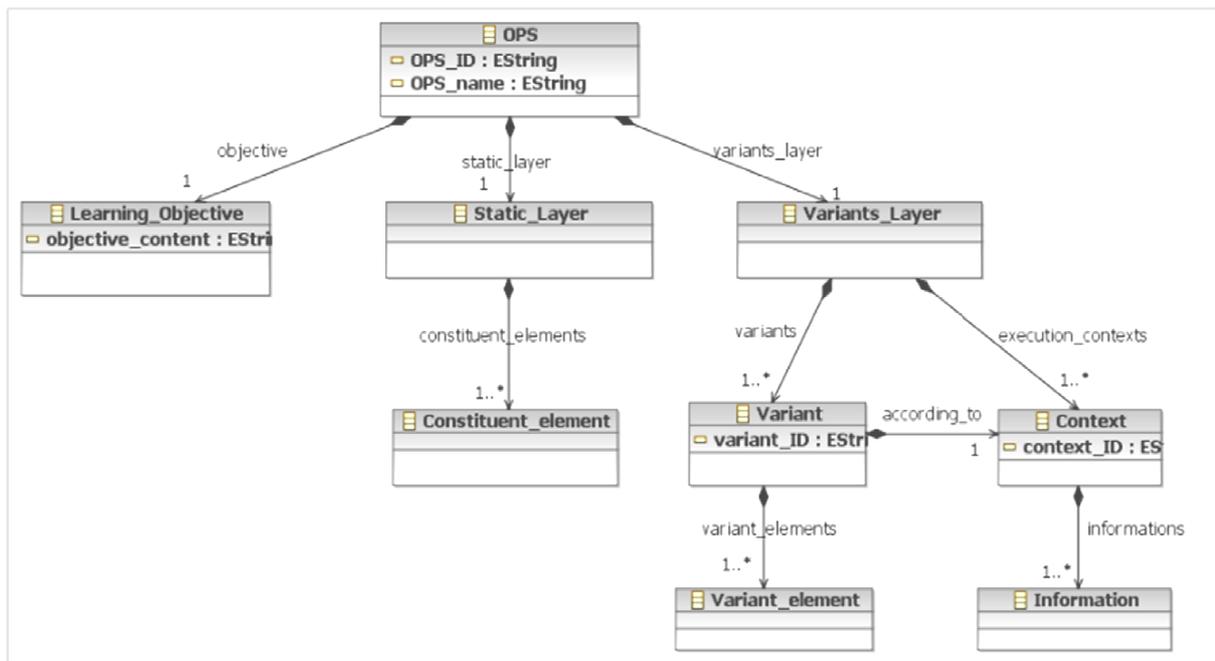


Figure 31 : Le métamodèle qui implémente le modèle rationnel des SPO.

Nous avons implémenté ce métamodèle dans le but de formaliser comment et pourquoi un SPO opérationnel doit être adapté. Cela grâce à l'identification des possibilités de variantes de ce SPO selon l'évolution de son contexte d'exécution, ainsi que la définition des relations qui associent chacune des variantes avec son contexte approprié. Nous avons utilisé le Framework EMF pour construire ce métamodèle, se réservant ainsi la possibilité de construire des éditeurs conformes à ce métamodèle. Sa description est définie principalement par un ensemble de concepts abstraits et des liens de composition (cf. figure

31). Nous avons choisi des mécanismes simples pour modéliser les aspects de la variabilité et les contextes en vue de faciliter l'appropriation de ce métamodèle par les enseignants praticiens lors de son tissage avec le métamodèle métier de ces derniers (cf. section 6.3.4).

Le concept « *OPS* » est l'élément racine de tout scénario pédagogique ouvert. Les deux attributs « *OPS_ID* » et « *OPS_name* » servent à identifier exclusivement le SPO pour des raisons d'indexation en vue d'éviter toute ambiguïté lors de son stockage, sa recherche, sa réutilisation, etc.

Les trois éléments principaux qui constituent un SPO sont définis par les concepts : « *Learning_Objective* », « *Static_Layer* », et « *Variants_Layer* ».

Le premier concept « *Learning_Objective* » correspond à l'objectif d'apprentissage que l'enseignant concepteur veut atteindre en définissant le SPO. Ce concept est lié au concept racine du SPO par une relation de composition avec une cardinalité de « 1 », car un SPO cible un seul objectif général d'apprentissage mais plusieurs variantes sont possibles pour l'atteindre. L'attribut « *objective_content* » permet de préciser l'énoncé de cet objectif.

Le second concept, « *Static_Layer* », correspond à la couche statique du SPO qui est liée également par une relation de composition au concept racine du SPO avec une cardinalité de « 1 », car un SPO ne peut comporter qu'une seule couche statique qui peut, quant à elle, être constituée de plusieurs éléments fixes qui peuvent être spécifiés grâce au concept « *Constituent_element* », qui est quant à lui, lié au concept « *Static_Layer* » par une relation de composition de cardinalité « 1..* ».

Le troisième concept « *Variants_Layer* », correspond à la partie flexible du scénario. C'est la couche des variantes et de leurs contextes appropriés. Chaque variante est représentée par le concept « *Variant* », tandis que le concept « *Context* » correspond aux différents contextes possibles d'exécution du SPO en question. Chacun de ces deux derniers concepts est lié au concept « *Variants_Layer* » par une relation de composition de cardinalité « 1..* », car cette couche de SPO peut comporter plusieurs variantes associées respectivement avec leurs contextes pertinents d'exécution. En outre, le concept « *Variant* » est lié au concept « *Context* » par la relation intitulée « *according_to* » portant une cardinalité de « 1 », car chaque variante doit être définie selon son propre contexte unique d'exécution. Or, chacun des concepts « *Variant* » et « *Context* » comporte respectivement un attribut d'identification « *variant_ID* » et « *context_ID* » à utiliser pour indexer les variantes et leurs contextes appropriés lors de leur stockage dans un entrepôt, afin de faciliter leur recherche et réutilisation. Quant au concept « *Variant_element* », il correspond aux éléments qui

composent une variante. Ce concept est donc lié par une relation de composition au concept « *Variant* » avec une cardinalité de « 1..* », car une variante du SPO peut contenir un seul ou plusieurs éléments. Le concept « *Information* » est défini pour spécifier des informations qui caractérisent des contextes. Ce concept est alors lié par une relation de composition au concept « *Context* » avec une cardinalité de « 1..* », car un contexte d'exécution peut être caractérisé par une seule information ou plusieurs.

Dans ce métamodèle des SPO, les concepts : « *Constituent_element* » de la couche statique, « *Variant_element* » de variante, et « *Information* » de contexte, sont génériques pour permettre aux concepteurs de les spécifier en fonction des concepts métiers de leur propre domaine spécifique (ex : activité, séquence, ressource, tuteur, apprenant, indicateur,...etc.), ceci lors du tissage de ce métamodèle générique avec leur métamodèle métier spécifique pour avoir un métamodèle métier permettant d'exprimer des SPO (cf. section 6.3.4).

4.5 Conclusion

Dans ce chapitre nous avons défini, en premier lieu, les scénarios pédagogiques ouverts. Nous avons présenté le modèle rationnel que nous proposons pour structurer les SPO en termes de variabilités selon les contextes d'exécution. Ce modèle est élaboré en s'appuyant sur des études théoriques, notamment le Design Rationale. Il permet aux enseignants d'avoir des scénarios qui s'adaptent à la particularité de chaque situation d'apprentissage (des variantes spécifiques aux contextes d'apprentissages) et de garder un caractère ouvert au scénario pour qu'il puisse intégrer d'éventuelles nouvelles variantes autres que celles déjà répertoriées. Nous présentons dans le chapitre suivant le processus itératif et incrémental d'ingénierie des SPO qui sont structurés suivant le modèle rationnel proposé.

5

Processus itératif et incrémental d'ingénierie des SPO

Sommaire

5.1 Introduction

5.2 Phases du processus itératif et incrémental d'ingénierie des SPO

5.3 Conclusion

5.1 Introduction

En matière d'ingénierie pédagogique, la question n'est pas celle d'élaborer un bon ou un mauvais artefact. D'après Simondon : « *un bon outil n'est pas seulement celui qui est bien façonné et bien taillé. Le sens de l'objet technique est son fonctionnement* » [Simondon 1958]. L'idée est que la conception d'un objet technique ne se complète que dans son usage. Selon Simondon, l'adoption des objets techniques par les usagers s'élabore dans ce qu'il désigne comme une « individualisation ». Il affirme : « *L'individualisation des êtres techniques est la condition du progrès technique* » [Simondon 1958]. L'ingénierie d'un SPO consiste donc à élaborer un artefact individualisable (adaptable) souvent en fonction de son contexte d'utilisation. La capitalisation des individualisations, effectuées à travers des cycles de vie, permet de raffiner et améliorer sa conception.

Un SPO est alors une entité en évolution continue selon les contextes dans lesquels il est exécuté. Son état doit dépendre fortement des circonstances de son utilisation. Bannon et Bodker soulignent que : « *les artefacts évoluent sans cesse, ils reflètent un état historique de la pratique des utilisateurs et en même temps ils modèlent cette pratique. Les opérations développées par les utilisateurs sont ensuite, à la génération suivante, incorporées dans l'artefact* » [Bannon et Bodker 1991]. D'ailleurs, l'approche de Design Rationale consiste à documenter le processus de conception en offrant une image de l'historique des choix effectués et de leurs raisons ainsi qu'un aperçu du produit final de la conception. Tout au long du cycle de vie d'un artefact, les concepteurs modifient les descriptions des objets, ils clarifient et affinent les fonctions prévues et les exigences. Ce sont des connaissances précieuses de conception. Le Design Rationale est en effet une explication du comment et du pourquoi un artefact est conçu.

Notre approche consiste à aider l'enseignant à concevoir et adapter des SPO à travers toutes les phases de leurs cycles de vie en respectant notre modèle rationnel présenté dans le chapitre précédant (cf. chapitre 4). Nous définissons alors ce cycle de vie comme un processus itératif et incrémental. Nous présentons ses phases dans le reste de ce chapitre, ainsi qu'une représentation formelle suivant les conventions graphiques du profil UML pour ECA (*Entreprise Collaboration Architecture*).

5.2 Processus itératif et incrémental d'ingénierie des SPO

Notre approche d'instrumentation est anthropocentrique, centrée sur des enseignants qui peuvent jouer différents rôles : concepteurs et/ou tuteurs. Nous définissons un processus itératif et incrémental d'ingénierie des SPO autour de l'enseignant [Ouraiba 2010]. Comme le montre la figure 32, ce processus est composé de quatre phases principales : conception initiale, déploiement, exécution et réingénierie. L'enseignant occupe une place au cœur de ce processus, où il joue un rôle pivot. Nous considérons l'enseignant comme sujet acteur de ses activités. Dans ses pratiques réelles, l'enseignant ne s'investit pas uniquement en amont, il sait qu'il aura à adapter ses prévisions à un contexte donné.

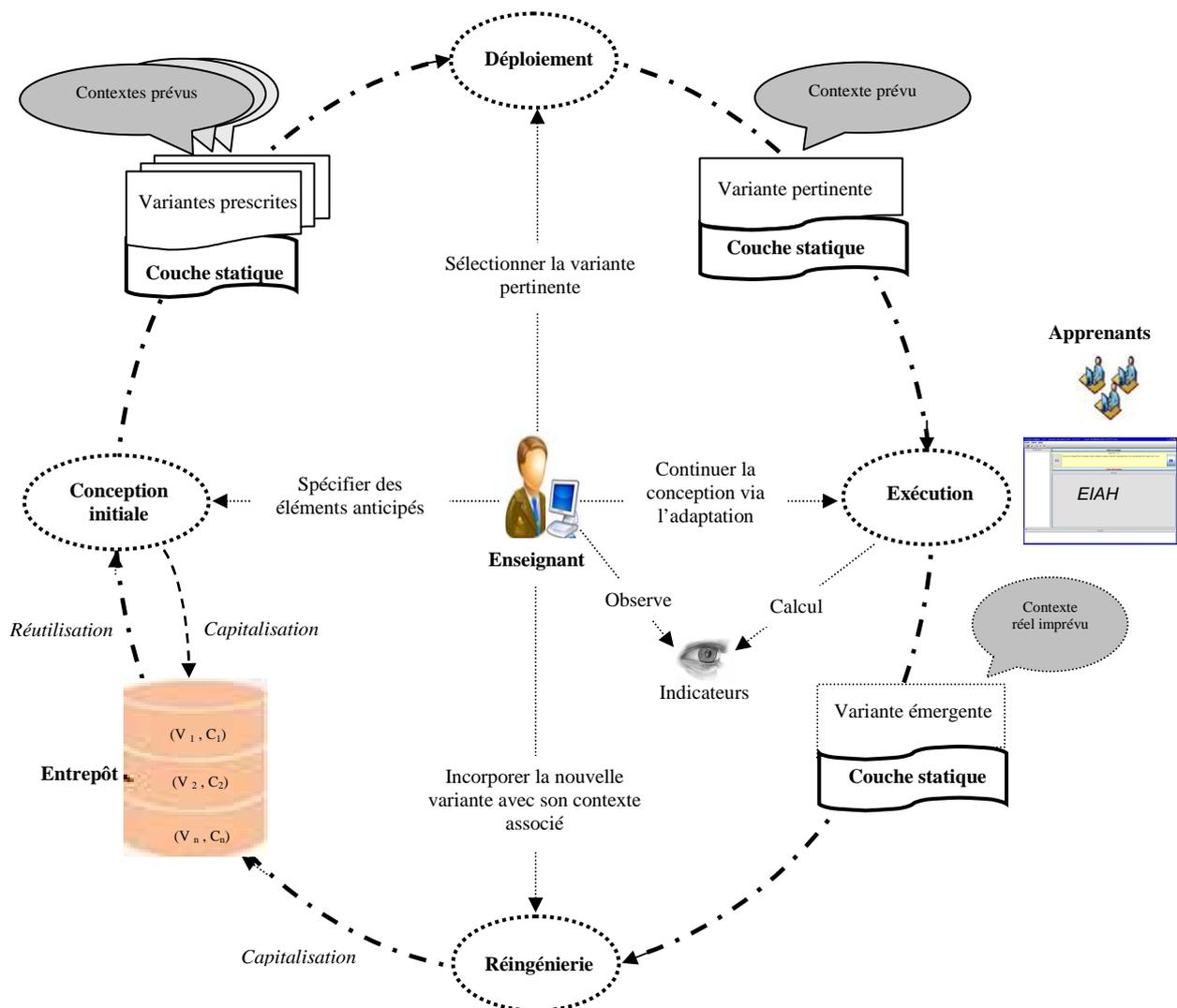


Figure 32 : Processus itératif et incrémental d'ingénierie d'un SPO.

5.2.1 Conception initiale

Pour atteindre un objectif d'apprentissage, les méthodes classiques de conception pédagogique ne permettent d'envisager facilement qu'une seule variante de scénario pédagogique ; par contre, l'approche de Design Rationale consiste à explorer de façon systématique les différentes variantes d'un SPO, de justifier les choix de conception en spécifiant des contextes d'exécution appropriés et de capitaliser des décisions prises. De cette façon, toutes les variantes d'un SPO, qui peuvent être anticipées pendant la conception initiale, sont pratiquement explorées et représentées dans ce que MacLean [MacLean et al. 1991] a appelé « *un espace de conception* », ce qui offre en effet à l'enseignant concepteur une vision plus exhaustive sur le SPO et une prise de décision qualitative.

Pendant la phase de conception initiale d'un SPO, l'enseignant concepteur effectue généralement une description incomplète car il ne peut avoir qu'une idée générale sur le déroulement de la session, il est difficile, voire impossible de le prédire de manière exhaustive. Comme le montre la figure 32, l'enseignant concepteur spécifie en amont un ensemble d'éléments qu'il peut anticiper. Certains de ces éléments resteront souvent rigides (non adaptables), ce sont les constituants de la couche statique, alors que les autres sont des éléments de la couche des variantes, que l'enseignant concepteur peut utiliser pour spécifier et combiner plusieurs variantes prescrites avec des contextes prévus dans un seul SPO. Ce dernier est adaptable selon le contexte d'exécution au moment de déploiement. L'enseignant concepteur peut aussi réutiliser des variantes capitalisées, issues des cycles antérieurs, stockées dans l'entrepôt. Nous considérons alors ici que l'anticipation des variantes en fonction des contextes prévus permet une flexibilité par la conception [Schonenberg et al. 2008] que nous admettons comme approche *proactive* d'adaptation. Avant leur déploiement, les variantes prédictives validées et leurs contextes associés devront donc être capitalisés dans un entrepôt dédié. Quant aux variantes imprévues, l'enseignant laisse certains éléments non définis sous forme de points ouverts jusqu'à la phase d'exécution. Ces éléments ne peuvent pas être connus avant le début du déroulement de la session d'apprentissage.

5.2.2 Déploiement

Un SPO doit être personnalisé avant son exécution dans un contexte donné où les acteurs impliqués tirent parti des ressources offertes et réalisent leurs activités. La personnalisation d'un SPO s'effectue alors au moment du déploiement de sa définition abstraite en spécification opérationnelle (interprétée par la machine). Au cours de cette phase,

l'enseignant sélectionne la variante la plus pertinente parmi l'ensemble des variantes pré-spécifiées pour la mettre en œuvre dans le contexte courant avec lequel elle est associée dans l'entrepôt. Le scénario est donc configuré en supprimant les variantes qui ne sont plus pertinentes pour le contexte en cours, par contre les éléments de la couche statique sont toujours intégrés et eux aussi sont concernés par l'exécution avec la variante choisie. Le scénario généré est donc adapté au contexte de son utilisation : il est "*situé*" [El-Kechaiï 2008]. Ce scénario individualisé peut être considéré comme une conception plus fine pour produire une spécification exécutable convenable à un ensemble particulier d'exigences.

Cependant, le SPO avec les variantes prescrites constitue un univers de contraintes et de possibles en même temps, avec lequel les utilisateurs peuvent agir dynamiquement selon le contexte réel de déroulement de la session d'apprentissage.

5.2.3 Exécution

C'est la phase d'utilisation de l'EIAH par les acteurs impliqués dans la session d'apprentissage, où la couche statique et la variante choisie du SPO s'exécutent dans le contexte réel de l'apprentissage. Ce contexte d'exécution est caractérisé par des informations, de différentes natures, et qui décrivent les circonstances réelles qui peuvent provoquer l'adaptation du SPO durant son déroulement. Dans la pratique, un enseignant ne suit pas exactement les scénarios pédagogiques prédictifs qui ne constituent qu'une base de départ pour lui. Il poursuit la conception du SPO selon les besoins et les événements qui émergent dynamiquement. Contrairement au schéma classique de conception qui distingue temporellement conception et usage, « *le processus de conception ne s'arrête pas au seuil de l'usage, il se poursuit au cours de celui-ci en genèses instrumentales* » [Rabardel 1995]. Un SPO est affiné pendant son exécution. L'enseignant doit pouvoir agir en temps réel sur la spécification préétablie en la restructurant et en la personnalisant pour satisfaire ses intentions et élaborer ses propres instruments dans un acte de *scénarisation dynamique*. Ceci peut se faire notamment suivant l'observation de progression des apprenants au cours du déroulement de la session, où il peut introduire les ajustements nécessaires afin de résoudre des problèmes spécifiques qui émergent, de renforcer l'apprentissage de certains concepts, et plus généralement, de garantir la réalisation des objectifs d'apprentissage d'origine [Zarraonandia et al. 2006a]. L'ajustement ne concerne pas seulement les modifications mais aussi la suppression ou l'ajout de nouveaux éléments nécessaires au bon déroulement de la session d'apprentissage. L'adaptation peut être alors effectuée suivant les trois types de flexibilité *réactive* (cf. section 2.1.3.3) :

- la déviation d'instance du SPO en cours d'exécution en adaptant la variante déployée sans capitaliser les modifications effectuées : *flexibilité par déviation* ;
- le changement de modèle du SPO, tant au niveau du type et d'instance, ceci via incorporation en temps réel des variantes émergentes dans la structure de modèle : *flexibilité par changement* ;
- la complétude de définition de la variante déployée du SPO, soit par la sous-spécification via liaison tardive, en utilisant uniquement des éléments définis dans d'autres variantes déjà prescrites ; soit par la sous-spécification via modélisation tardive de nouveaux éléments en spécifiant le contenu des points laissés ouverts jusqu'à l'exécution. Par conséquent, la sous-spécification consiste à garder les éléments nouvellement définis, ce qui provoque l'émergence de nouvelles variantes du SPO : *flexibilité par sous-spécification*.

L'ensemble des adaptations effectuées constitue une nouvelle variante émergente qui peut être incorporée, avec son contexte associé, dans la structure du SPO à travers un processus de réingénierie.

5.2.4 Réingénierie

La réingénierie d'un SPO n'est pas aisée en réalité pour l'enseignant qui doit prendre des décisions d'ordre pédagogique. En utilisant les informations collectées (indicateurs) à partir des traces de déroulement de la session d'apprentissage, l'enseignant concepteur peut évaluer des adaptations effectuées et analyser les variantes émergentes dans le SPO descriptif en vue d'identifier les problèmes rencontrés et d'envisager les améliorations nécessaires. La nouvelle version du SPO doit être constituée en intégrant des adaptations éprouvées et considérées comme positives, parce qu'elles améliorent la qualité du scénario prédictif, ces adaptations seront ainsi prises en considération dans des futurs cycles de vie. Cette intégration reste sous la maîtrise de l'enseignant qui décide si une nouvelle variante émergente permet d'améliorer ou non le scénario. Dans le cas où cette variante correspond à un nouveau contexte pour lequel elle est pertinente, ou qu'elle est meilleure qu'une existante, l'enseignant peut l'intégrer dans la deuxième couche des variantes du SPO. Elle sera donc stockée avec son contexte associé dans l'entrepôt. Si l'enseignant remarque qu'une variante se répète dans la majorité des contextes, il peut la considérer comme impérative pour les futurs cycles vie du SPO, et l'incorporer dans la couche statique du scénario. Les variantes qui ne sont jamais, ou très rarement utilisées, peuvent être supprimées par l'enseignant pour éviter le problème de débordement causé par la surcharge de l'entrepôt des variantes.

Afin de donner une représentation formelle de ce processus, nous utilisons les conventions graphiques du profil UML pour ECA (*Entreprise Collaboration Architecture*) présenté par le consortium OMG [OMG 2004]. Ce profil est destiné à supporter la modélisation dans une démarche de développement dirigée par les modèles utilisant le cadre ODP-RM [Nagase et al. 2004]. Les phases de ce processus, au sens d'ODP-RM, ont toutes des objectifs de production ou d'administration d'artefacts produits dans une communauté d'ingénierie des SPO.

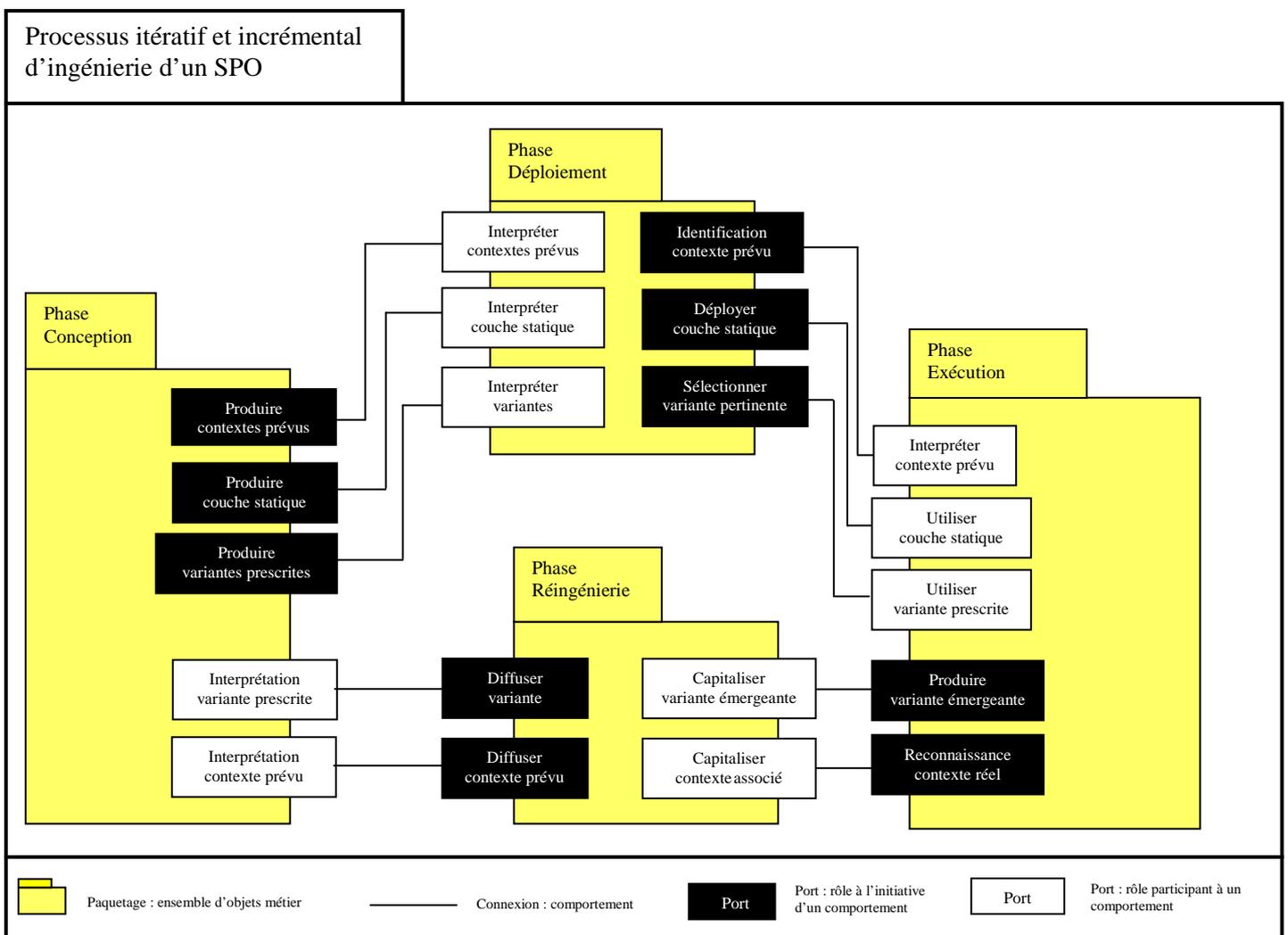


Figure 33 : Diagramme de paquetages UML représentant le processus itératif et incrémental d'ingénierie d'un SPO.

Comme le montre la figure 33, chaque phase est représentée par un paquetage. Une phase produit et consomme des artefacts. Ces derniers caractérisent des flux de communication entre les différentes phases. Chaque flux est représenté sous forme d'une interaction entre deux phases (une phase émettrice et une phase réceptrice).

- ❖ La phase de conception a pour objectif de spécifier des SPO en étant guidé par le modèle rationnel que nous avons proposé. Pour concevoir un SPO, l'enseignant peut donc soit produire, à nouveau, des variantes et leurs contextes prévus, ou bien réutiliser celles capitalisées lors des phases de réingénierie des cycles antérieurs.
- ❖ La phase de déploiement consiste à interpréter les spécifications produites lors de la phase de conception. Ces spécifications concernent la couche statique du SPO, ses variantes anticipées et leurs contextes prévus. Une fois ces spécifications interprétées, les éléments de la couche statique sont déployés avec une variante du SPO. Cette dernière est jugée comme la plus appropriée car son contexte associé est identique à celui qui est identifié lors de la reconnaissance du contexte prévu pour l'exécution du SPO. Cette reconnaissance peut être effectuée par l'interprétation de toute information qui caractérise et influence la situation en considération, par exemple, via l'identification des profils des apprenants impliqués, des ressources disponibles, des indicateurs qui peuvent être calculés en amont ou juste au moment du déploiement du SPO, etc.
- ❖ La phase d'exécution représente l'ensemble des activités des acteurs d'une session d'apprentissage. Elle a pour objectif d'utiliser les instances des éléments déployés du SPO. Ces instances concernent alors les éléments de la couche statique et ceux de la variante prescrite qui est déployée selon le contexte identifié. Dans la réalité, le contexte effectif d'exécution est généralement différent de celui qui a été interprété (prévu) au début de déroulement de la session. En conséquence, cela nécessite des adaptations en temps réel qui peuvent amener à l'émergence d'une nouvelle variante du SPO.
- ❖ La phase de réingénierie consiste à capitaliser les variantes émergentes produites lors d'exécution et leurs contextes associés dans le but d'altérer le SPO prédictif pour des futures conceptions. Cette phase comporte un processus de gestion des ressources dont l'objectif principal est de gérer les variantes et les contextes en les indexant et stockant dans un entrepôt dédié qui facilitera la recherche, la sélection et la réutilisation aux enseignants concepteurs à travers toutes les phases de cycles de vie de SPO.

5.3 Conclusion

La définition des SPO selon notre modèle rationnel permet de prendre en compte les différentes approches d'adaptation : soit « proactive », soit « réactive ». Proactive par la conception, en spécifiant *a priori* un ensemble de variantes anticipées. Réactive par la déviation à la volée des instances de la variante déployée, par le changement de la structure abstraite (au niveau modèle) et/ou d'instance du SPO en temps réel, par la complétude de définition du SPO en intégrant des éléments prescrits via la liaison tardive, ou par la spécification de nouveaux éléments via la modélisation tardive.

En outre, l'approche d'ingénierie des SPO à base de variantes peut réduire les efforts pour les enseignants concepteurs, améliorer la communication de l'expertise et favoriser la réutilisation systématique des pratiques communes ou éprouvées dans un domaine éducatif spécifique.

Cependant, les enseignants praticiens n'ont généralement pas de grandes compétences techniques et/ou de modélisation informatique. Ce processus doit donc être instrumenté par une méthode et des outils, intriqués dans leur domaine métier, qui les aident à concevoir, adapter et gérer les SPO à un haut niveau d'abstraction.

Nous présentons dans le chapitre suivant une méthode s'appuyant sur IDM/DSM visant à instrumenter les enseignants. Elle consiste à fournir un langage spécifique au domaine métier et un éditeur graphique dédié visant à faciliter la conception des SPO conformément à notre modèle rationnel (cf. chapitre 4), tout en respectant le processus d'ingénierie présenté (cf. chapitre 5).

6

Méthode d'instrumentation pour ouvrir les scénarios pédagogiques d'un EIAH

Sommaire

6.1 Introduction

6.2 Vision IDM / DSM sur l'ingénierie des SPO

6.3 Méthode d'instrumentation dirigée par les modèles et spécifique au domaine métier
pour ouvrir les scénarios pédagogiques d'un EIAH

6.4 Conclusion

6. 1 Introduction

En pratique, l'évolution continue des scénarios, que nous qualifions d'ouverts (SPO), tout au long de son cycle de vie, pose plusieurs défis à l'enseignant. Il doit être capable de préconiser et de spécifier toutes les parties du SPO (éléments de la couche statique, éléments adaptables et éléments ouverts de la couche des variantes), et de définir les stratégies d'adaptation permettant d'individualiser le scénario afin d'améliorer la qualité d'apprentissage, d'opérationnaliser dynamiquement sur la plate-forme les adaptations spécifiées à un niveau abstrait; tout ceci en incluant la perception des informations du contexte et la sélection ou la définition, à la volée, d'une stratégie d'adaptation convenable. L'enseignant doit ainsi pouvoir évaluer la pertinence des adaptations effectuées suivant l'objectif d'apprentissage du SPO, et décider la capitalisation de ces adaptations quand elles constituent une connaissance stratégique très importante et réutilisable, dans le cadre de développements futurs de nouveaux scénarios ou d'améliorations des scénarios existants dans une phase de réingénierie.

L'objectif de notre travail, nous le rappelons ici, est de définir un cadre d'ingénierie des EIAH ouverts capable de permettre aux enseignants d'assumer leur rôle pivot à travers toutes les phases du processus d'ingénierie des SPO. La spécification des SPO conformément au modèle rationnel et respectant le processus d'ingénierie définis dans les deux chapitres précédents, nécessite donc de disposer d'instruments facilitant la tâche de l'enseignant concepteur qui n'est pas nécessairement un spécialiste en informatique. Nous proposons dans ce chapitre une méthode d'instrumentation centrée enseignants pour ouvrir un EIAH à l'ingénierie des SPO. Autrement dit, une méthode qui fournit des instruments aux enseignants concepteurs pour leur faciliter la conception et l'adaptation des SPO. Pour ce faire, nous nous appuyons sur les aspects de l'ingénierie dirigée par les modèles (IDM) et en particulier la modélisation spécifique au domaine (DSM). Cette méthode est donc dirigée par les modèles et spécifique au domaine métier des enseignants concepteurs, dans le but de leur offrir des artefacts aisément accessibles. Avant de présenter notre méthode en détail, nous illustrons, en premier lieu, la projection de l'ingénierie des SPO sur les niveaux d'OMG selon le paradigme IDM/DSM.

6.2 Vision IDM/DSM sur l'ingénierie des SPO

Dans le but d'instrumenter le processus d'ingénierie des SPO (cf. Chapitre 5), nous adoptons IDM/DSM comme cadre théorique et pratique. Nous essayons de tirer profit de potentiel de ce paradigme tel qu'il est étudié en génie logiciel. Nous considérons alors la scénarisation pédagogique comme une activité de « modélisation » de la situation d'apprentissage. Il s'agit de produire des SPO sous forme de modèles pour qu'ils puissent être adaptés à tout moment, à niveau modèle. Nous considérons alors, pour qu'un SPO soit véritablement conçu et manipulé par un enseignant praticien, qu'il doit être considéré comme un modèle spécifique à son domaine éducatif (*Domain-Specific Model*), exprimé en conformité avec un métamodèle qui décrit un DSEML (*Domain-Specific Educational Modeling Language*) enraciné dans les pratiques métiers de l'enseignant [Ouraiba et al. 2010]. Ce métamodèle capture l'abstraction des différents aspects pertinents du domaine, il a pour rôle de maintenir la cohérence des SPO et d'assurer leur robustesse, en prenant en compte le fait qu'ils peuvent être conçus et modifiés dynamiquement (même en temps réel). Dans son activité de conception pédagogique, l'enseignant utilise un éditeur dédié qui embarque la sémantique de son propre domaine, cristallisée dans le métamodèle d'expression pédagogique.

L'outillage d'IDM/DSM permet de fournir des supports dédiés aux enseignants les aidant à assumer leur rôle de concepteurs et de tuteurs en continuant la conception des SPO à travers des adaptations dynamiques pendant le déroulement de la session d'apprentissage. En outre, les techniques d'IDM et DSM permettent de :

- définir le SPO avec une représentation spécifique au domaine métier et de l'opérationnaliser ainsi sans perte sémantique sur le système de formation. Le DSM va dans le sens de permettre aux enseignants de spécifier, comprendre, adapter et valider des SPO par eux-mêmes à un niveau abstrait du domaine ;
- générer des SPO concis et réutilisables à la fois. Concis car ils sont spécifiés par un langage dédié et dépouillé des concepts ou métaphores perturbants que la plupart des langages génériques imposent. Réutilisables car ils sont constitués de variantes qui correspondent aux différents contextes d'utilisation ;
- incarner les connaissances et le savoir-faire du domaine dans le DSEML. Ce qui assure leur capitalisation, partage et réutilisation ;

- communiquer avec d'autres acteurs notamment les développeurs. Le DSEML et les SPO sont des moyens intermédiaires de communication entre les développeurs et les enseignants qui sont considérés comme experts de domaine ;
- assurer un bon niveau de productivité, de fiabilité, de maintenabilité, et de portabilité à travers les phases du processus d'ingénierie des SPO.

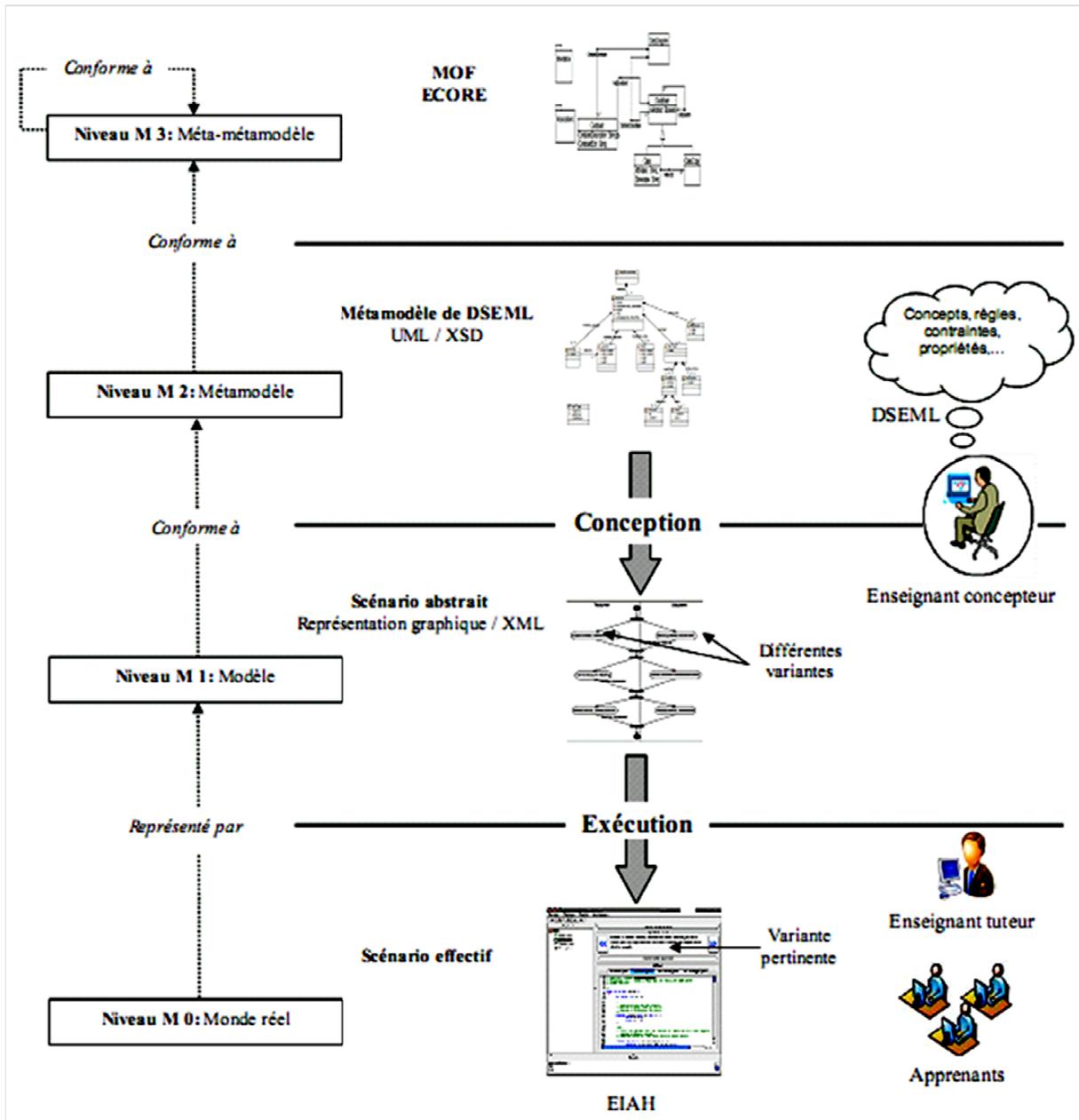


Figure 34 : Vue en niveaux d'OMG de l'ingénierie des scénarios pédagogiques ouverts [Ouraiba et al. 2011b].

La figure 34 illustre la projection de l'ingénierie des SPO sur les niveaux d'abstraction de l'OMG [OMG 2006]. Elle montre comment nous adoptons le paradigme IDM/DSM en vue de fournir des supports nécessaires aux enseignants pendant la conception et l'exécution des SPO. Nous considérons alors que le DSEML doit être décrit par un métamodèle spécifié au niveau M2, et conforme à un méta-métamodèle (MOF-*MetaObject Facility*) défini au niveau M3. Ce métamodèle est le modèle des SPO. Il doit formaliser la sémantique du domaine éducatif de l'enseignant en décrivant son langage métier (vocabulaire, règles, contraintes, etc.). Les SPO abstraits doivent être alors spécifiés au niveau M1 comme modèles en conformité avec le métamodèle du DSEML.

Pendant la conception, les éléments statiques et les variantes prévues d'un SPO, peuvent être définis par le biais d'instanciation des éléments génériques du métamodèle. Pendant l'exécution, les éléments statiques et la variante sélectionnée comme pertinente pour le contexte en considération, vont être opérationnalisés sur le système d'apprentissage au niveau M0 où ils seront exécutés dans le contexte réel de la situation d'apprentissage.

6.3 Méthode d'instrumentation dirigée par les modèles et spécifique au domaine métier pour ouvrir des scénarios pédagogiques d'un EIAH

Nous présentons dans cette section notre méthode d'instrumentation visant à ouvrir les scénarios pédagogiques d'un EIAH. Le but est d'offrir aux enseignants usagers d'un EIAH des supports conviviaux permettant la conception et l'adaptation dynamique des SPO suivant les évolutions du contexte d'enseignement/apprentissage [Ouraiba et al. 2011a] [Ouraiba et al. 2011b] [Ouraiba et al. 2011d].

Cette méthode, s'appuyant sur l'approche IDM/DSM, implique deux types de rôles : l'expert du domaine (l'enseignant) et le spécialiste en méta-modélisation et développement (l'informaticien). La collaboration entre ces deux rôles vise à identifier les pratiques des enseignants et à externaliser ainsi leur métier de la manière la plus efficace possible, pour développer ensuite des instruments dédiés à leur domaine éducatif. Cette instrumentation consiste à fournir concrètement un DSEML et un éditeur pour permettre l'expression des scénarios pédagogiques dans l'univers métier des enseignants. Quant au DSEML, sa définition nécessite de bien préciser deux syntaxes : abstraite et concrète. La syntaxe abstraite est réifiée par le métamodèle dont un ensemble de concepts (vocabulaire) et de règles métier qui sont familiers pour des enseignants. Ces concepts reliés et ces règles reflètent la sémantique du domaine spécifique. La syntaxe concrète définit une représentation graphique de chacun des concepts. Afin d'éviter toute forme d'ambiguïté ou de non pertinence, la spécification de ces deux syntaxes doit être appuyée sur un consensus issu de la négociation entre les deux rôles impliqués. Cela permet notamment à l'informaticien de formaliser efficacement les abstractions du domaine. En se basant sur ces deux syntaxes, il peut développer un éditeur des scénarios, ainsi générés sous forme de modèles conformes au métamodèle qui réifie la syntaxe abstraite du DSEML. L'enseignant peut utiliser cet éditeur dédié au domaine pour produire des scénarios pédagogiques, à un niveau abstrait. Cet éditeur embarque les concepts et les règles que les enseignants utilisent habituellement dans leur propre domaine éducatif. La figure 35 illustre les différents rôles et leurs activités dans le processus de définition du DSEML et dans le développement de l'éditeur dédié.

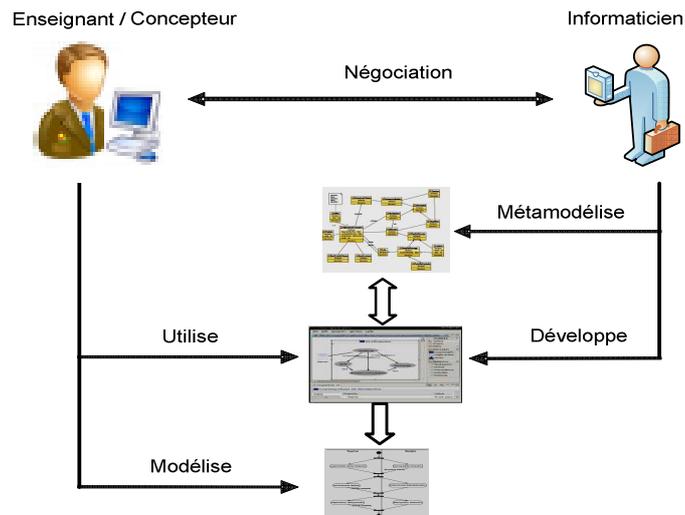


Figure 35 : Collaboration entre l'enseignant concepteur et l'informaticien.

Notre méthode, dirigée par les modèles et spécifique au domaine métier, consiste à construire des instruments progressivement (cf. figure 35), de façon partagée et négociée, sans négliger la compréhension du changement des habitudes et des modalités de travail, que les instruments proposés et la nouvelle logique d'utilisation de l'EIAH vont produire chez les enseignants. Une étude récente, intitulée « *Empirical Assessment of MDE in Industry* » [Hutchinson et al. 2011], confirme que le déploiement d'IDM, y compris du DSM évidemment, « a pu poser des problèmes lorsque celui-ci a été fait de manière autocratique » [Jézéquel et al. 2012].

Notre méthode est incrémentale et articule deux processus successifs d'instrumentation, où des enseignants praticiens sont assez impliqués, ceci afin de garantir, au final, l'utilisabilité et l'acceptabilité des instruments proposés. Le premier processus de notre méthode vise deux buts principaux : (a) montrer aux enseignants concepteurs leur « métier réifié » dans un outil d'édition des scénarios pédagogiques ; (b) examiner la pertinence et la capacité de ces outils « dédiés » aux domaines métiers des enseignants.

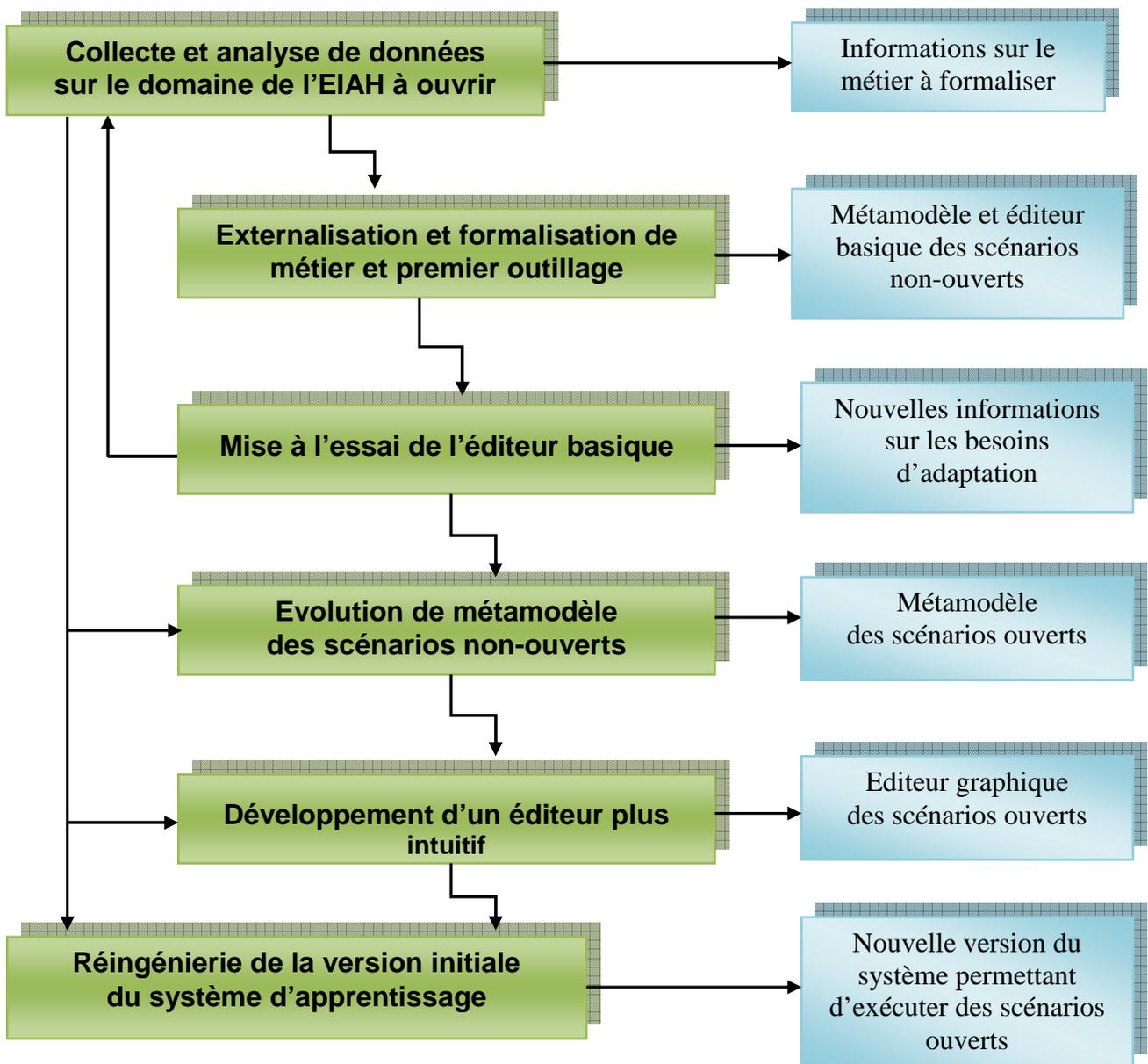
Pour ce faire, un premier éditeur basique doit être développé en s'appuyant sur un métamodèle initial qui décrit le DSEML des enseignants concepteurs sans prendre en compte les aspects d'ouverture et d'adaptation des scénarios pédagogiques. Cet éditeur peut ne disposer que d'une interface-utilisateur basique (par exemple arborescente) pour la conception des scénarios non-ouverts. Il n'est pas destiné à être pérenne : son rôle est de réifier le métier des enseignants usagers de l'EIAH considéré pour servir de base au second

processus visant à prendre en compte les aspects d'ouverture et d'adaptation dynamique des scénarios en utilisant des outils dédiés.

Un nouvel éditeur, plus intuitif et convivial que le premier, doit être développé. Cet éditeur doit disposer d'une notation graphique des concepts métiers, et permettre la production des SPO opérationnels sur le système d'apprentissage cible. Cet éditeur a pour vocation de donner plus d'ouverture aux scénarios en permettant une adaptation efficace et efficiente, y compris à la volée, pendant le déroulement de la session d'apprentissage. Le développement de cet éditeur dédié doit se baser sur un métamodèle enrichi par des concepts de modèle rationnels des SPO que nous avons proposé (cf. chapitre 4). Ce métamodèle décrit le langage métier (ADSGEML) des enseignants concepteurs permet de spécifier graphiquement des SPO à un niveau plus haut d'abstraction.

Notre méthode se déroule sur plusieurs étapes (cf. figure 36) :

- (1) collecte d'informations sur le domaine métier considéré. Ces informations, fondamentales, sont utilisées dans toutes les étapes ultérieures ;
- (2) définition d'un métamodèle initial décrivant le DSEML d'expression des scénarios non-ouverts et développement d'un premier éditeur basique ne comportant que des concepts métiers principaux ;
- (3) mise à l'essai de l'éditeur afin d'affirmer la pertinence et le potentiel des instruments spécifiques au domaine métier, et d'identifier les besoins réels des enseignants en termes d'ouverture des scénarios pédagogiques, de capitalisation et de réutilisation des stratégies d'adaptation ;
- (4) évolution du métamodèle initial vers un nouveau, enrichi, qui décrit l'ADSGEML. Cette évolution ne peut être effectuée qu'après un consensus issu d'une négociation entre l'informaticien spécialiste en méta-modélisation et les enseignants experts de leur domaine. Elle consiste à tisser le métamodèle rationnel des SPO (cf. chapitre 4) et le métamodèle qui réifie le métier des scénarios de l'EIAH à ouvrir ;
- (5) développement d'un second éditeur plus intuitif comportant une notation graphique et permettant de produire des SPO opérationnels et de les adapter dynamiquement à travers toutes les phases de leurs processus d'ingénierie ;
- (6) réingénierie du système d'apprentissage qui n'est pas toujours nécessaire, elle consiste à ajouter et/ou améliorer certaines fonctionnalités.



Légende

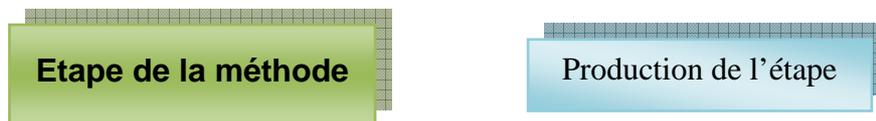


Figure 36 : Méthode d'instrumentation pour l'ouverture des scénarios pédagogiques d'un EIAH.

6.3.1. Collecte d'informations sur le domaine métier de l'EIAH à ouvrir

Cette étape exploratoire a pour objectif de collecter le maximum d'informations sur le domaine des enseignants pour lesquels nous voulons ouvrir l'EIAH. Ces informations devront être suffisamment pertinentes en vue de :

- ❖ cerner la sémantique propre au domaine métier des enseignants usagers de l'EIAH à ouvrir. Cette sémantique est définie par un ensemble de concepts, de propriétés, de règles et de contraintes à identifier pour offrir aux enseignants, dans un premier temps, des instruments basiques (le DSEML et un éditeur spécifique) permettant d'exprimer des scénarios non-ouverts tels qu'ils existent initialement dans l'EIAH, c'est-à-dire sans prendre en compte les aspects d'ouverture des scénarios. Ces instruments spécifiques au domaine sont proposés pour dépasser les difficultés rencontrées en utilisant les EML génériques et leurs outils complexes.
- ❖ identifier les cas d'utilisation offerts par la version initiale de l'EIAH et les analyser afin de pouvoir confronter son potentiel d'usage aux pratiques réelles effectuées par les usagers praticiens. Cette confrontation peut permettre de déterminer les fonctionnalités nécessaires à améliorer ou à intégrer à travers un processus de réingénierie.

Afin de recueillir les informations nécessaires nous proposons à l'informaticien d'utiliser trois méthodes en fonction des objectifs de récolte de données (cf. figure 37) :

- (a) **Observation des situations réelles d'utilisation de l'EIAH** — Une expérimentation exploratoire sur des séances d'utilisation de système d'apprentissage, avec étudiants et enseignants, permet de mettre à jour et comprendre les pratiques des usagers et le déroulement effectif des scénarios pédagogiques.
- (b) **Entretiens avec des enseignants praticiens** — Ces entretiens peuvent être menés avant et/ou après le déroulement des sessions d'apprentissage. Ils permettent d'interroger les enseignants praticiens (scénaristes et/ou tuteurs) dans le but de mesurer leurs satisfactions et leurs besoins d'amélioration du dispositif d'apprentissage existant. Ces entretiens permettent également d'effectuer des négociations avec les enseignants afin d'aboutir à des consensus à propos d'interprétation des concepts et des règles du domaine métier.
- (c) **Fouille de fichiers existants dans la base de données de l'EIAH** — L'analyse des fichiers de données existants dans le système permet de découvrir la structure générique des scénarios pédagogiques « opérationnalisables » sur le système existant. D'un point

de vue IDM/DSM, les données collectées sur cette structure générique permettent de définir un métamodèle des scénarios pédagogiques existants (non encore ouverts) qui décrit le DSEML des enseignants concepteurs.

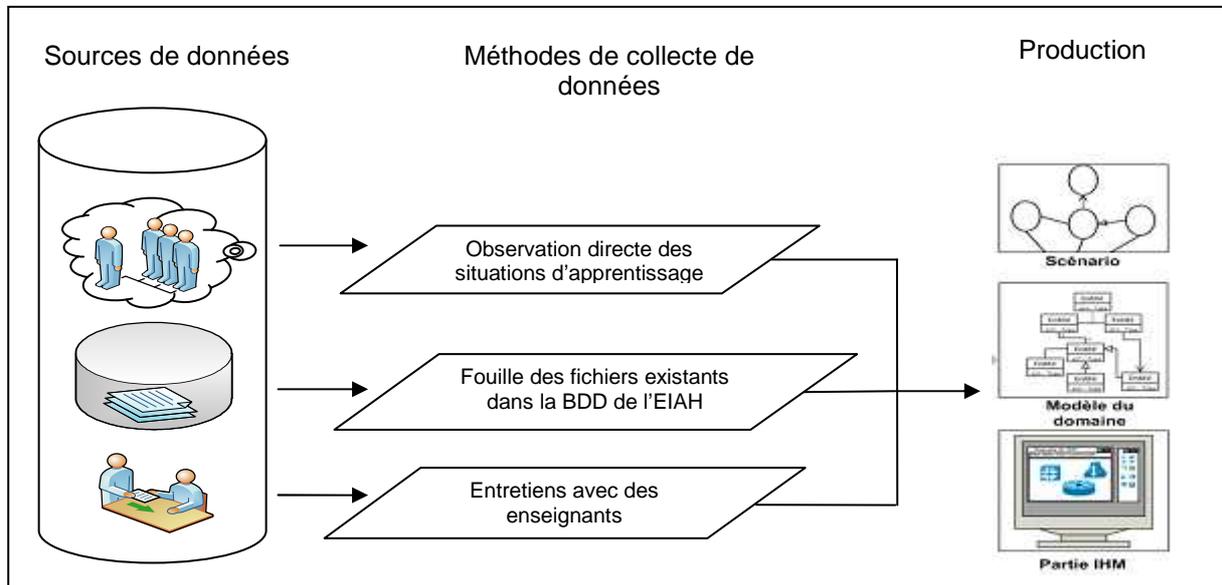


Figure 37 : Méthodes pour collecter les données du domaine métier.

6.3.2 Définition de DSEML d'expression des scénarios pédagogiques existants et développement d'un premier éditeur dédié

Cette étape consiste à formaliser le métier des enseignants concepteurs, sous forme de métamodèle, pour les instrumenter par un support dédié. La figure 39 décrit le processus de génération de code d'un éditeur spécifique à partir d'un métamodèle d'expression pédagogique qui formalise le domaine métier des enseignants concepteurs. Ce métamodèle décrit le DSEML utilisé par des enseignants pour concevoir les scénarios déjà existants dans l'EIAH, qui demeurent à ce stade non-ouverts. Il ne comporte que des concepts déjà formalisés dans les scénarios existants, sans prendre en compte les aspects d'adaptation et d'ouverture. Ce métamodèle réifie en effet la structure générique des scénarios pédagogiques existants en se basant sur les informations obtenues grâce à la fouille de fichiers disponibles dans la base de données du système, mais aussi aux entretiens et à la négociation avec les enseignants concepteurs pour parvenir à un consensus, tant sur l'ensemble des concepts que sur leur utilisation (cf. figure 38).

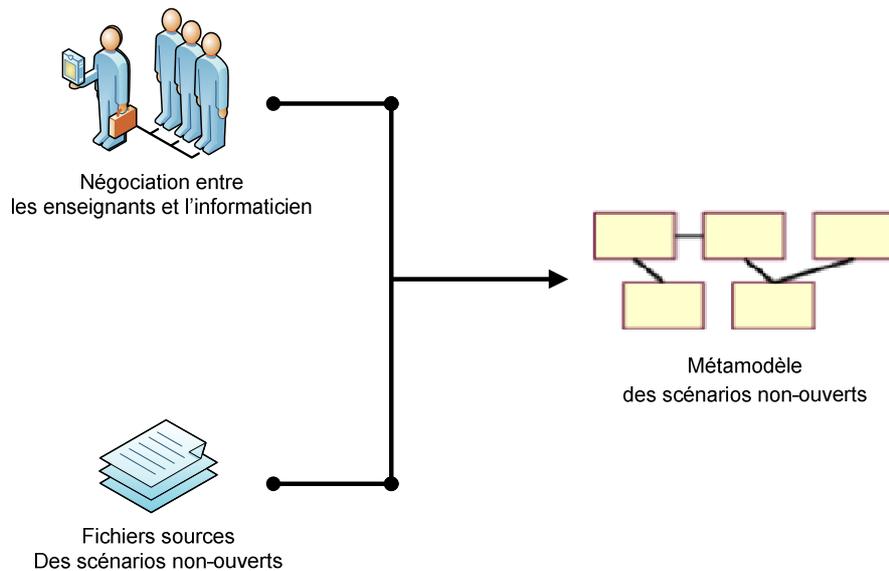


Figure 38 : Définition de métamodèle des scénarios pédagogiques existants.

Le métamodèle métier défini dans cette étape sert à développer un éditeur de scénarios linéaires (non-ouverts). Cet éditeur, basique, peut être prototypé rapidement avec les outillages d'IDM/DSM. Il est destiné à réifier les connaissances du domaine et les processus sous-jacents de conception de scénarios pédagogiques pour l'EIAH à ouvrir aux enseignants. Il constitue le premier artefact de conception produit. Il a vocation à être un vecteur de discussion et de négociation pour définir en quoi l'EIAH existant devra être ouvert.

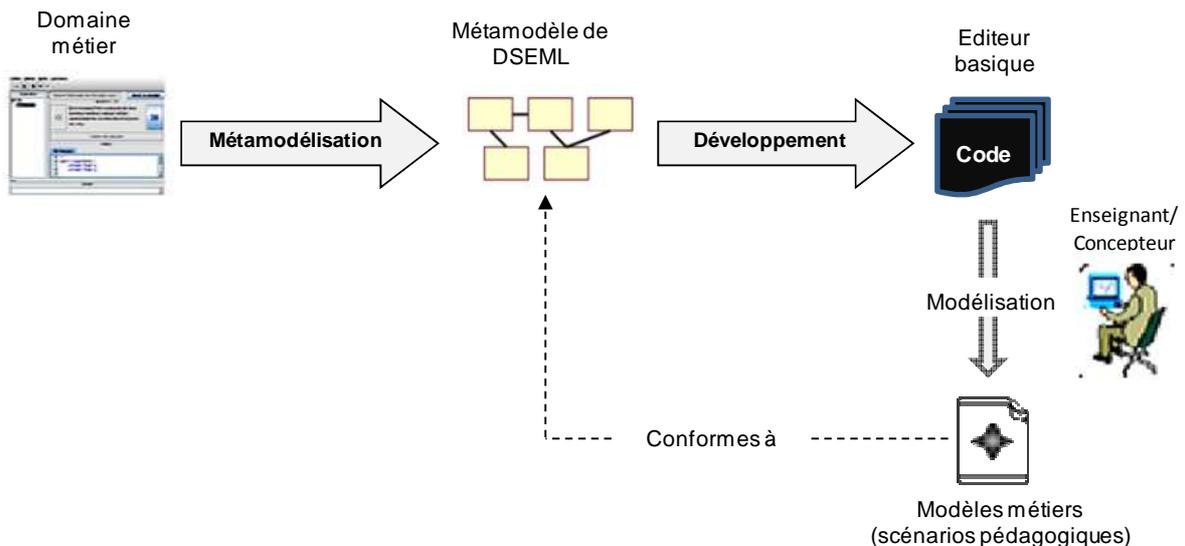


Figure 39 : Processus de développement d'éditeur à partir de métamodèle métier.

Cependant, le DSEML et son éditeur support, construits à cette étape, permettent à l'enseignant concepteur de travailler dans un univers métier, mais pas encore de produire des scénarios ouverts qui sont adaptables suivant la variation de contexte d'enseignement/apprentissage. Dans le cas où l'enseignant concepteur veut atteindre le même objectif d'apprentissage dans différents contextes, il est obligé de définir plusieurs scénarios (non-ouverts). La figure 40 illustre cela en faisant la projection des concepts de DSEML, scénario pédagogique et contexte d'apprentissage sur les niveaux d'OMG.

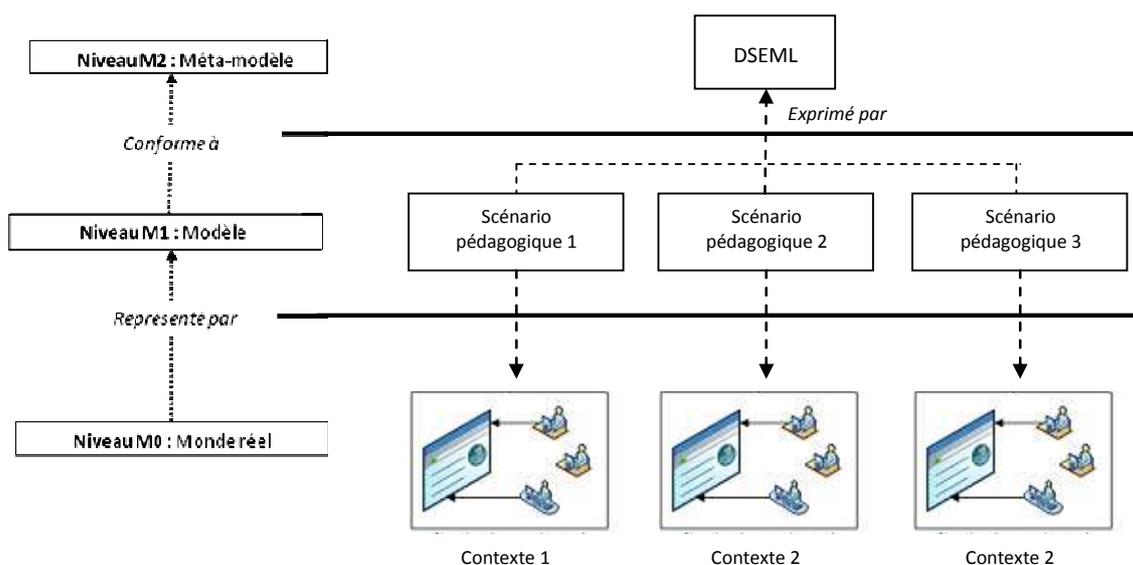


Figure 40 : Projection des concepts de DSEML, scénario pédagogique et contexte sur les niveaux d'OMG.

6.3.3 Mise à l'essai du premier éditeur dédié

Cette étape consiste à faire utiliser l'éditeur basique par les enseignants concepteurs pour exprimer les scénarios non-ouverts par le DSEML. Un des objectifs dans cette étape est de vérifier si le DSEML et l'éditeur sont suffisamment intuitifs et accessibles pour permettre l'autonomie des enseignants concepteurs.

Mettre à l'essai l'éditeur spécifique a pour but de montrer aux enseignants praticiens leur « métier réifié » dans un outil d'édition des scénarios pédagogiques, et de les amener à valider ce langage métier (DSEML) en le pratiquant. Ce travail permet de stabiliser la définition de la syntaxe réifiée dans le métamodèle, de faire évoluer ce dernier en émettant de nouveaux besoins en termes d'adaptation dynamique pour l'ouverture des scénarios. C'est pendant cette phase que les enseignants concepteurs vont émettre le souhait de définir plusieurs variantes du même scénario en fonction de différents contextes d'apprentissage.

6.3.4 Evolution de DSEML existant vers ADSEML d'expression des SPO

Cette étape est cruciale. Comme le montre la figure 41, elle consiste à produire le métamodèle des scénarios ouverts après une négociation entre les informaticiens (spécialistes en métamodélisation) et les enseignants usagers de l'EIAH (experts du domaine), sur la base du métamodèle initial (élaboré lors de l'étape 6.3.2) et du métamodèle rationnel d'un SPO (cf. chapitre 4). Cette négociation doit permettre un compromis entre les enseignants et les informaticiens afin d'aboutir à une interprétation qualitative et un enrichissement pertinent des éléments du métamodèle initial selon les éléments de modèle rationnel des SPO. Par exemple, spécifier ce qu'est une variante, identifier les éléments constituant la couche statique, etc. C'est avec les enseignants que les spécialistes en métamodélisation tissent donc le métamodèle initial avec le modèle rationnel des SPO, sur la base de décisions purement d'ordre métier. Le résultat de cette phase est donc un métamodèle des scénarios ouverts, qui décrit la syntaxe abstraite du langage spécifique de modélisation graphique des scénarios ouverts (ADSEML pour *Adaptive Domain-Specific Graphical Educational Modeling Language*).

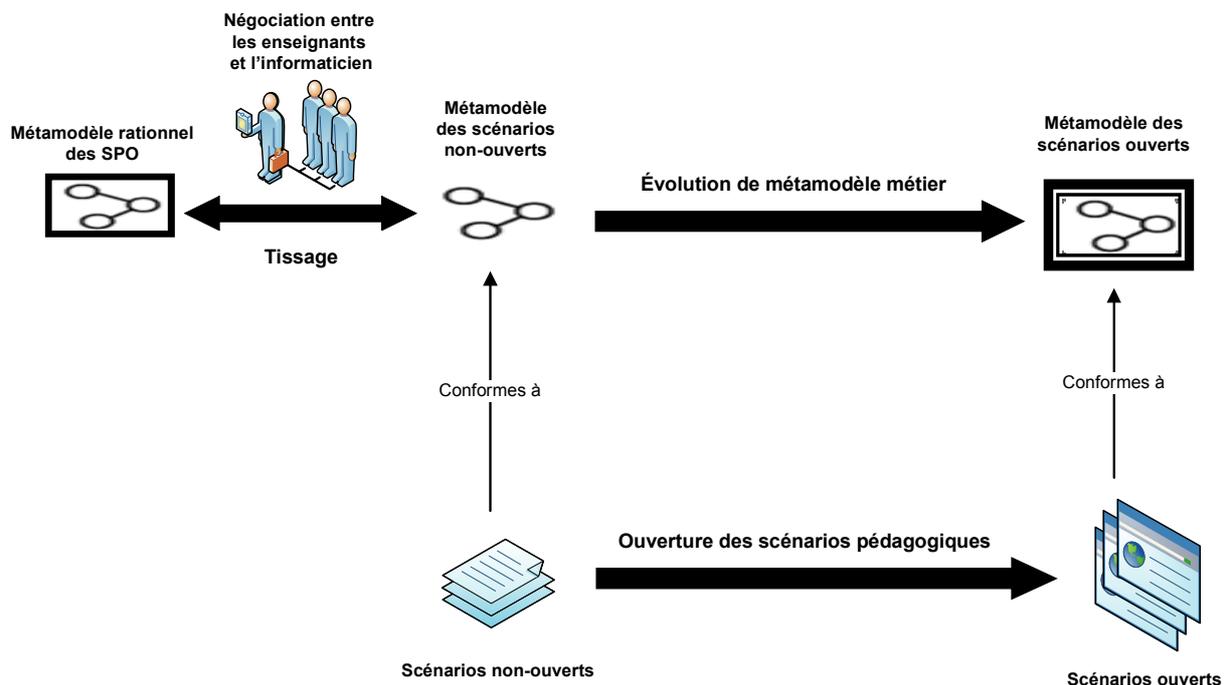


Figure 41 : Evolution de métamodèle initial après négociation de l'informaticien avec les enseignants.

Ainsi, pour atteindre le même objectif d'apprentissage dans différents contextes d'enseignement/apprentissage, le métamodèle décrivant l'ADSGEML permet de modéliser un seul scénario ouvert (SPO) qui comporte plusieurs variantes. Cela évite à l'enseignant concepteur de redéfinir à chaque fois un scénario (non-ouvert) pour chaque contexte. La figure 42 illustre cela en faisant la projection des concepts de SPO, variante, contexte et ADSGEML sur les niveaux d'OMG.

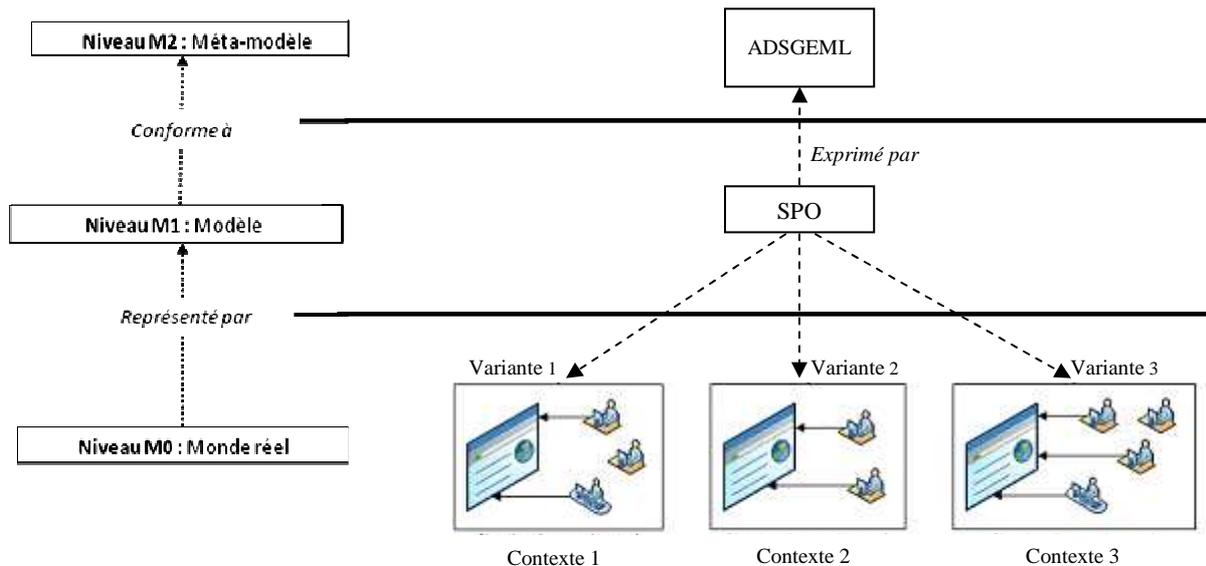


Figure 42 : Projection des concepts d'ADSGEML, SPO, variante et contexte sur les niveaux d'OMG.

6.3.5 Développement dirigé par les modèles d'un éditeur graphique des SPO

Les outillages de DSM, tels que « EMF/GMF », permettent actuellement de construire, à partir des langages dédiés, des éditeurs aisément accessibles par des usagers néophytes. Ces éditeurs permettent aux concepteurs d'un domaine métier de spécifier graphiquement, à un niveau élevé d'abstraction, des modèles de leur propre domaine et de les déployer sous un format interprétable par la machine.

Cette étape de notre méthode d'instrumentation consiste alors à développer un éditeur graphique comme environnement support à l'ADSGEML. Cet environnement de modélisation est proposé pour faciliter aux enseignants la conception et l'adaptation des scénarios ouverts. Le processus de développement de cet éditeur est dirigé par les modèles.

Il consiste à proposer des facilités de construction d'une notation graphique définissant la syntaxe concrète de l'ADSGEML pour implémenter la syntaxe abstraite cristallisée dans le métamodèle évolué. La définition de ces deux syntaxes doit se baser sur les consensus issus des négociations itératives effectuées avec les enseignants usagers de l'EIAH afin de s'assurer de la pertinence des choix effectués, notamment en termes de spécification des liaisons entre les concepts du domaine et leurs représentations graphiques respectives. Ceci en vue de favoriser l'acceptabilité et l'utilisabilité de l'ADSGEML et de son éditeur support chez les enseignants concepteurs.

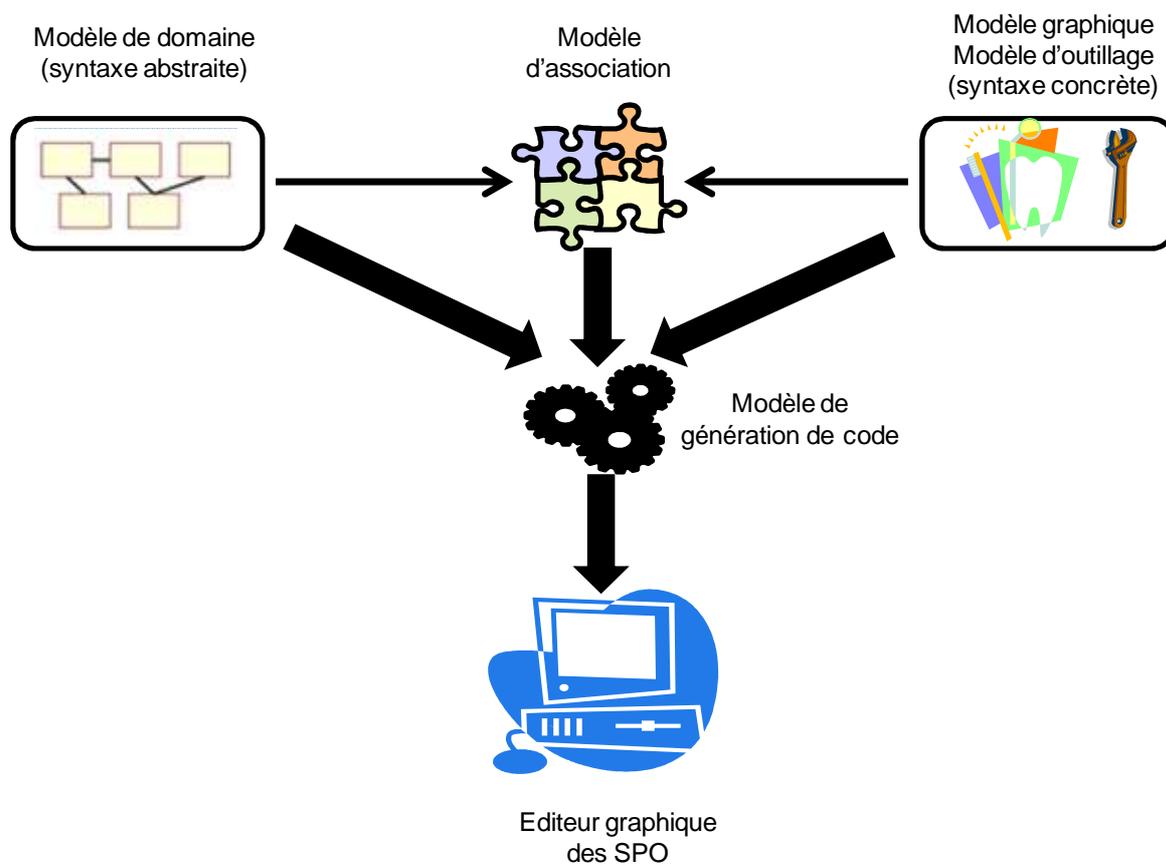


Figure 43 : Processus dirigé par les modèles pour le développement d'un éditeur graphique des SPO.

Comme le montre la figure 43, plusieurs modèles doivent être construits pour définir les deux syntaxes :

- ❖ La syntaxe abstraite doit être définie par le modèle du domaine métier, c'est-à-dire le métamodèle évolué qui décrit l'ADSGEML. Il permet d'exprimer des scénarios pédagogiques ouverts. Ce métamodèle est considéré comme l'élément central dans

le processus de définition des langages métiers et de développement des outils supports. C'est pourquoi il est primordial d'apporter un soin particulier à sa définition. Cette dernière requiert d'externaliser les concepts du domaine de l'EIAH qui sont pertinents pour la conception des SPO. Ces concepts doivent être déjà familiers et utilisés couramment par les enseignants afin de cristalliser une sémantique adaptée.

- ❖ La syntaxe concrète doit être définie par deux modèles, le premier est celui de définitions graphiques des différents composants à utiliser dans l'espace d'édition, le second modèle définit la palette d'outils de l'éditeur. Les éléments de ces deux modèles sont définis pour représenter graphiquement les concepts du modèle du domaine. La définition de cette syntaxe concrète a pour objectif de rendre plus facile et simple la modélisation des SPO suivant la syntaxe abstraite, c'est-à-dire en conformité au métamodèle métier évolué.
- ❖ Enfin, un autre modèle doit être défini pour assurer l'association (*mapping*) entre les éléments des trois premiers modèles : chaque concept du métamodèle doit être associé à un outil de la palette qui permet d'ajouter un composant graphique dans l'espace d'édition des SPO. Ce modèle d'association sert à produire un modèle de génération de code qui permet de paramétrer les détails d'implémentation du code final de l'éditeur.

6.3.6 Réingénierie de système d'apprentissage

Les pratiques réelles des usagers dépassent généralement le potentiel des fonctionnalités disponibles initialement dans le système d'apprentissage. L'ouverture de l'EIAH nécessite également d'effectuer la réingénierie de ce système au-delà de l'instrumentation des enseignants concepteurs par un ADSEGEML et un éditeur dédié. Cette réingénierie a pour but d'adapter les fonctionnalités du système d'apprentissage existant afin de donner les moyens aux enseignants d'adapter dynamiquement les scénarios embarqués. Cela passe par l'étude de la version disponible du système en se basant sur des informations collectées (cf. section 6.3.1). Ensuite, essayer de proposer des solutions par le perfectionnement des fonctionnalités existantes et/ou l'intégration de nouvelles fonctionnalités nécessaires à la facilitation des tâches des utilisateurs.

Les décisions d'adaptation d'un SPO s'appuient sur les caractéristiques du contexte dans lequel ce SPO devra être (ou est) exécuté. Cela nécessite que l'EIAH reconnaisse et

interprète ce contexte. La caractérisation et la reconnaissance des contextes d'exécution, de même que la compréhension des traces générées, ne sont pas faciles pour des enseignants. Un certain nombre de fonctionnalités doivent être alors ajoutées à l'EIAH pour assurer la collecte et la formalisation des informations caractérisant le contexte, afin de les capitaliser et d'alimenter la définition du SPO. Elles doivent être compréhensibles pour l'enseignant et interprétables par le système. Or, ces informations peuvent être de natures hétérogènes et obtenues depuis différentes sources. Elles peuvent être par exemple, les profils des apprenants impliqués, les indicateurs sur des activités d'apprentissage, etc.

Cette étape de réingénierie consiste également à ajouter du code dans le système pour implémenter des mécanismes permettant de :

- ❑ interpréter dynamiquement et sans perte sémantique, des SPO produits par l'éditeur graphique ;
- ❑ assurer l'opérationnalisation en temps réel d'adaptation des variantes que l'enseignant réalise au niveau abstrait de l'éditeur graphique ;
- ❑ charger le contenu de la variante appropriée pour chaque contexte ;
- ❑ basculer entre les différentes variantes, pour pouvoir les suivre simultanément.

6.4 Conclusion

Dans ce chapitre, nous avons présenté une méthode pour instrumenter l'ingénierie des SPO conformément à notre modèle rationnel que nous avons présenté dans le chapitre 4. Les enseignants sont impliqués directement à travers les étapes de cette méthode, qui est définie comme itérative et incrémentale, dans le but de s'assurer progressivement de la pertinence des instruments qui leur sont proposés. Dans un premier temps, elle consiste à formaliser le métier du domaine de l'EIAH sous forme d'un métamodèle décrivant un langage dédié d'expression pédagogique (DSEML) sans prendre en compte les aspects d'ouverture des scénarios, et à construire un éditeur basique qui permet de produire des scénarios non-ouverts, conformes au métamodèle initial. Dans un second temps, afin de prendre en compte les aspects d'ouverture, le métamodèle initial doit évoluer par tissage avec le métamodèle qui implémente le modèle rationnel des SPO proposé (cf. section 4.4). Le métamodèle obtenu décrit alors un langage dédié (ADSGEML) d'expression des SPO, et permet de conduire le développement dirigé par les modèles d'un éditeur graphique qui permet la conception et l'adaptation des SPO.

Ainsi, afin de favoriser une meilleure ouverture de l'EIAH pour faciliter sa diffusion à une échelle plus large, l'ADSGEML et l'éditeur construits peuvent être employés, suivant une approche interprétative, par d'autres enseignants concepteurs qui n'ont pas été impliqués dans leur développement. L'EIAH est devenu ouvert pour ces enseignants, la conception est facilitée et les SPO sont adaptés dynamiquement en utilisant l'éditeur dédié qui embarque l'ADSGEML.

Dans notre contexte de recherche, nous nous intéressons particulièrement à l'expression des scénarios pédagogiques en utilisant un langage de modélisation pédagogique (EML). Pour que l'EIAH soit ouvert aux différentes possibilités d'utilisation en fonction de différents contextes d'usages, nous considérons que les scénarios déployés sur cet EIAH doivent être eux aussi ouverts suivant plusieurs variantes de déroulement. Le langage de modélisation doit permettre la conception de ces scénarios ouverts. Mais nous considérons également que l'ADSGEML doit être évolutif pour maintenir l'ouverture de l'EIAH, afin de pouvoir prendre en compte de nouveaux besoins et exigences qui apparaîtront ultérieurement, car les pratiques métiers des enseignants changent avec le temps ; l'EIAH doit pouvoir accompagner ces changements par l'évolution de l'ADSGEML en intégrant de nouveaux concepts métiers dans le métamodèle.

La partie suivante présente l'application de nos propositions scientifiques dans le domaine métier d'un EIAH nommé « Hop3x », utilisé pour déployer des sessions de travaux pratiques en programmation orientée objet.

III. Troisième partie :

Etude de cas

7

Application sur Hop3x

Sommaire

7.1 L'environnement Hop3x

7.2 Application de notre méthode d'instrumentation

pour ouvrir des sessions d'apprentissage de Hop3x

7.3 Apports de notre approche pour Hop3x

Cette troisième partie est consacrée à l'application de nos propositions théoriques dans un champ d'expérimentation afin de les évaluer et de les raffiner. Ce champ d'expérimentation concerne le domaine d'un EIAH déjà existant nommé «Hop3x» [Desprès et Jacoboni 2010]. Hop3x a été développé préalablement par deux enseignants chercheurs en informatique au LIUM, dans le but de déployer à distance des sessions de travaux pratiques pour enseigner les langages de la programmation orientée-objet, tels que Java et Ruby. Nous nous focalisons ici sur la programmation Java.

Par la mise en œuvre de nos propositions sur Hop3x, nous visons à créer les conditions du déploiement, de l'appropriation et de l'utilisation de cet EIAH en vue de favoriser sa diffusion auprès d'autres enseignants partageant les mêmes connaissances métier sur l'enseignement de la programmation orientée-objet. Notre but est donc de rendre Hop3x ouvert aux enseignants souhaitant l'utiliser en leur facilitant la conception et l'adaptation dynamique des sessions d'apprentissage [Ouraiba et al. 2010] [Ouraiba et al. 2011c].

Nous présentons en premier lieu dans cette partie l'environnement Hop3x et nos motivations à le choisir comme cas d'application. Nous détaillons les étapes d'application de notre méthode d'ouverture des sessions d'apprentissage et d'instrumentation de leur processus de conception et d'adaptation.

7.1 L'environnement Hop3x

7.1.1 Architecture et fonctionnement de Hop3x

Hop3x est un système multiplateformes, il peut être installé sur différents systèmes d'exploitation. Il est composé des applications suivantes :

- « **Interface-Apprenant** » : permet aux apprenants d'écrire, éditer, compiler et exécuter le code de leurs programmes. Un outil de communication audio est mis à disposition pour permettre de demander une aide auprès du tuteur (le menu *Assistance*).
- « **Interface-Tuteur** » : permet de gérer un groupe d'apprenants à distance de façon synchrone. Il offre au tuteur des moyens de supervision (monitoring) des apprenants, comme les indicateurs et le code des programmes saisi par chaque apprenant en temps réel. Des outils de communication permettent au tuteur d'envoyer un message à un apprenant, ou lancer une conversation audio, pour lui donner des conseils et des recommandations.
- « **Serveur** » : contient un ensemble de fichiers XML qui définissent le contenu des sessions d'apprentissage déployées sur Hop3x. Le serveur gère la connexion et la synchronisation des communications entre les différentes parties du système. Il collecte les traces d'interaction des utilisateurs et les envoie à un module complémentaire pour le calcul des indicateurs en temps réel.
- « **Module complémentaire de calcul d'indicateurs** » : convertit les données brutes des traces en format UTL [Choquet et Iksal 2007] pour calculer les indicateurs grâce à l'outil DCL4UTL [Pham Thi Ngoc 2010].

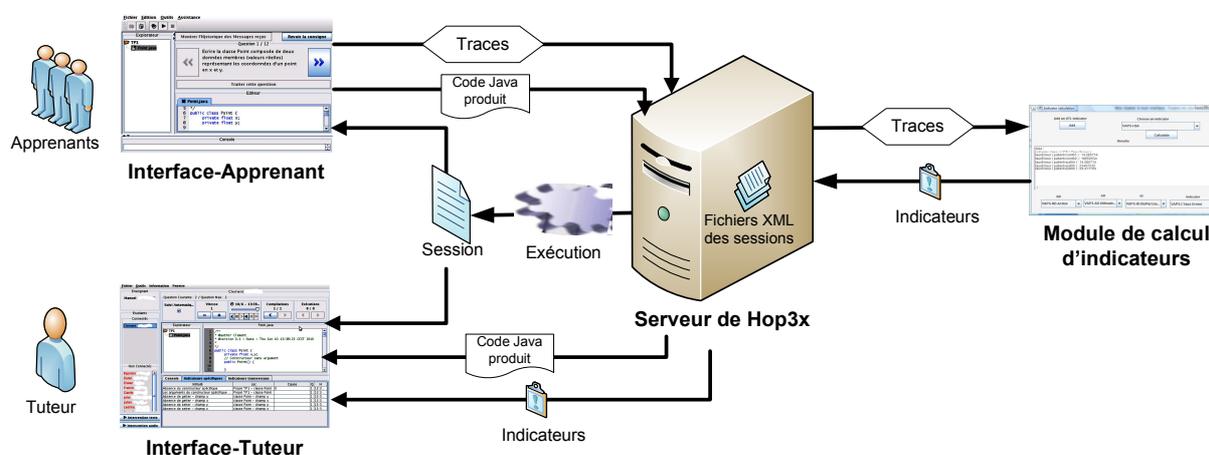


Figure 44 : Architecture du système Hop3x.

En utilisant l'interface-Apprenant (cf. figure 45), les apprenants doivent d'abord lire les consignes de l'énoncé avant de commencer à répondre aux questions : écrire et compiler du code Java pour résoudre le problème posé. L'ensemble des questions est présenté sous forme d'une séquence qui n'est pas statique. Les apprenants peuvent parcourir la liste des questions et choisir l'ordre à suivre. Lorsqu'un apprenant éprouve des difficultés, il peut demander l'aide auprès du tuteur via l'interface de communication audio.

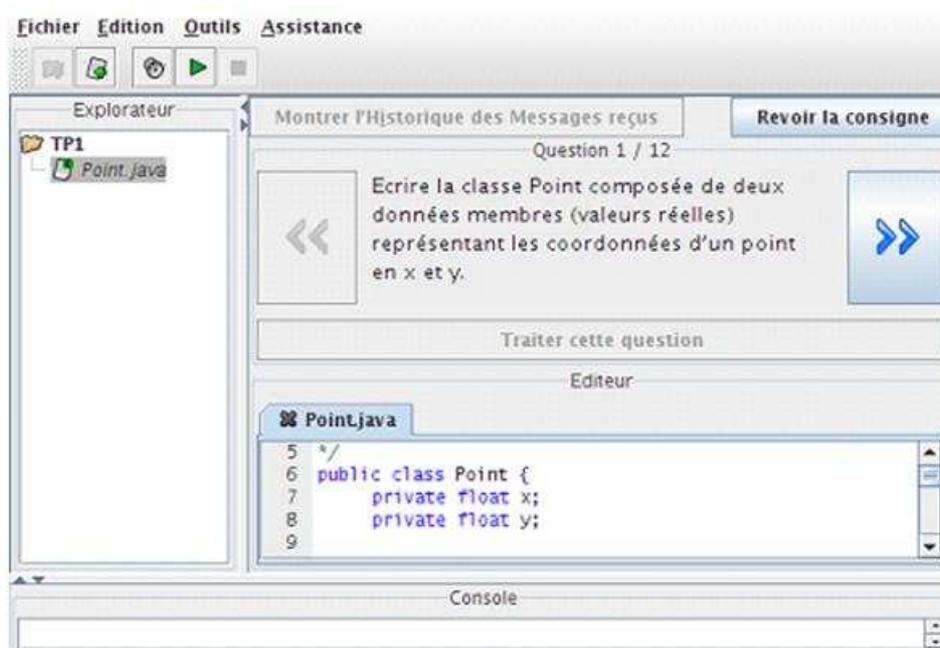


Figure 45 : L'interface-Apprenant de Hop3x.

Le tuteur utilise un ensemble de fonctionnalités permettant le contrôle des sessions et le suivi des apprenants durant leurs activités (cf. figure 46). Ces fonctionnalités permettent de voir la liste des apprenants connectés ou non, de voir en temps réel le code écrit par chaque apprenant, de rejouer la session de chaque apprenant, d'être alerté en temps réel par des indicateurs calculés sur la base des traces, de faire des interventions audio ou textuelles proactives ou réactives auprès des apprenants, de consulter l'historique de ses interventions, etc. Les interventions *proactives* se font sur la base du code Java écrit par un apprenant et en se basant sur les indicateurs d'activité. Les interventions *réactives* répondent aux demandes d'aide des apprenants.

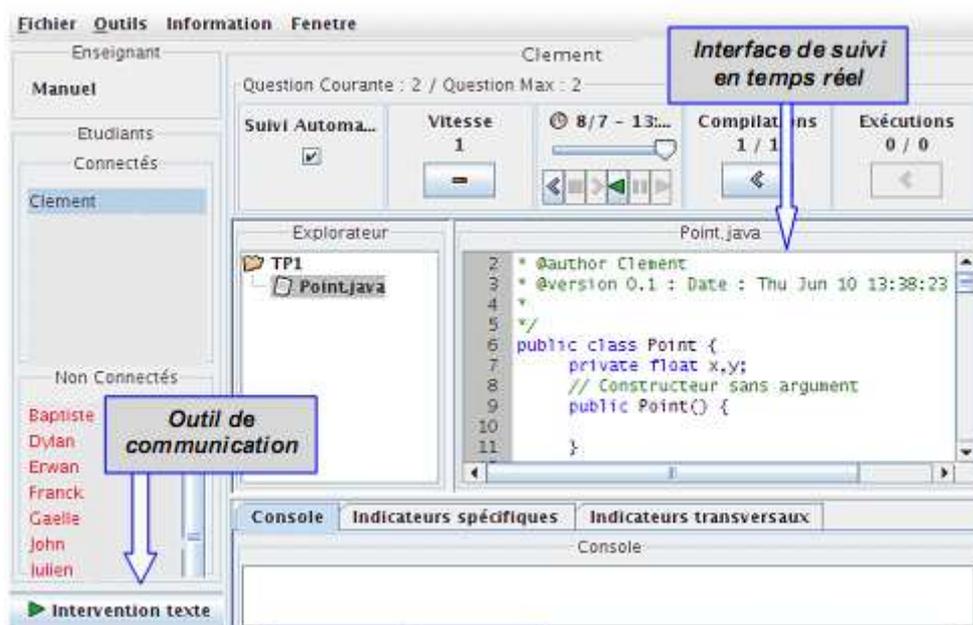


Figure 46 : L'interface-Tuteur de Hop3x.

7.1.2 Pourquoi le choix de Hop3x ?

La motivation qui nous amène à choisir Hop3x comme champ d'expérimentation, est qu'il constitue une bonne opportunité pour mener une étude d'ouverture d'un EIAH dans un contexte pédagogique et institutionnel réel. L'objectif de notre étude est de nous interroger sur les possibilités d'ouverture d'un EIAH et vérifier l'apport possible de nos propositions théoriques.

Nous avons choisi Hop3x car il s'agit d'un produit de notre laboratoire et que son code source est disponible. C'est un EIAH encore en développement. Il est ainsi possible d'inscrire concrètement notre approche dans le processus de conception et de réingénierie en cours. Nous pouvons ainsi travailler à l'amélioration de son appropriation et de son potentiel d'utilisation. Les actuels usagers concepteurs de Hop3x cherchent à l'ouvrir pour favoriser son utilisabilité à une échelle plus large. Ils préfèrent aussi avoir la possibilité de définir plusieurs variantes de la même session d'apprentissage. De plus, les structures et les contenus des sessions ne sont pas implantés dans le système, ce qui permet d'externaliser la conception des sessions.

Le contexte était donc favorable. Ceci a facilité la collecte de données sur les pratiques, de même que la réingénierie de l'EIAH. Nous avons pu collecter des données par des méthodes variées : fouille des fichiers du système, observations directes, entretiens ; dans le

but d'identifier les concepts utilisés, leurs relations et les différentes contraintes qui définissent les connaissances et le savoir faire du domaine métier de Hop3x.

7. 2 Application de notre méthode d'instrumentation pour ouvrir des sessions d'apprentissage de Hop3x

Notre objectif est d'ouvrir Hop3x le faisant passer d'un prototype uniquement accessible à ses concepteurs initiaux à un EIAH diffusé et utilisé au sein d'une communauté d'enseignants élargie. Nous avons appliqué notre méthode d'instrumentation qui consiste à étudier la situation existante de Hop3x puis à proposer des solutions pour l'ouvrir à la conception et l'adaptation des sessions d'apprentissage.

Notre méthode d'instrumentation (cf. chapitre 6) repose sur l'IDM/DSM. Elle implique des enseignants utilisateurs de Hop3x, considérés ici comme des experts du domaine. Elle est structurée en plusieurs étapes qui amènent progressivement à externaliser le métier de Hop3x et le cristalliser sous forme de métamodèle. Ce dernier décrit un langage d'expression des différents aspects des sessions ouvertes (*ADSGEML - Adaptive Domain-Specific Graphical Educational Modeling Language*). Il est exploité pour développer un environnement d'édition graphique associé permettant la conception et l'adaptation dynamique des sessions ouvertes à un niveau élevé d'abstraction, sans préoccupations techniques ni appropriation d'un outil générique.

Notre méthode ne vise pas seulement à aider les enseignants à produire des sessions opérationnelles et à les adapter dynamiquement, elle consiste aussi à effectuer la réingénierie du système en intégrant certaines fonctionnalités qui facilitent la tâche des usagers, ainsi que des mécanismes nécessaires à l'opérationnalisation des adaptations réalisées en temps réel dans l'outil d'édition.

7. 2.1 Collecte d'informations sur le domaine de Hop3x

À cette étape de notre méthode d'instrumentation, le but est d'explorer la structure des sessions d'apprentissage et de comprendre l'activité développée par les enseignants concepteurs de Hop3x. Il est alors possible de relever les besoins en termes de conception et d'adaptation des sessions d'apprentissage. Cela en vue de cristalliser le métier dans un langage de modélisation pédagogique spécifique au domaine pédagogique de Hop3x.

Afin de collecter le maximum d'informations pertinentes nous avons utilisé trois méthodes, différentes mais complémentaires :

- ❑ la fouille des fichiers XML existants dans le serveur de Hop3x ;
- ❑ l'observation des usagers de Hop3x en activité ;
- ❑ des entretiens auprès des concepteurs initiaux de Hop3x et des tuteurs impliqués dans l'utilisation de Hop3X.

7. 2.1.1 Fouille des fichiers XML existants dans le serveur de Hop3x

Les informations des sessions d'apprentissage sont stockées dans un ensemble de fichiers XML dans le serveur de Hop3x. Le système utilise ces fichiers comme sources de données pour exécuter les sessions. Toutefois, ces fichiers sont décrits manuellement par les enseignants concepteurs de Hop3x, et sans se baser sur une DTD ou un schéma (XSD) pour les valider et garantir leur persistance. Nous avons donc fouillé ces fichiers avec deux objectifs principaux :

- la découverte de la structure générique des sessions exécutables sur Hop3x,
- l'identification des connaissances métier déjà formalisées dans ces fichiers.

On distingue cinq types de fichiers XML (cf. figure 47) :

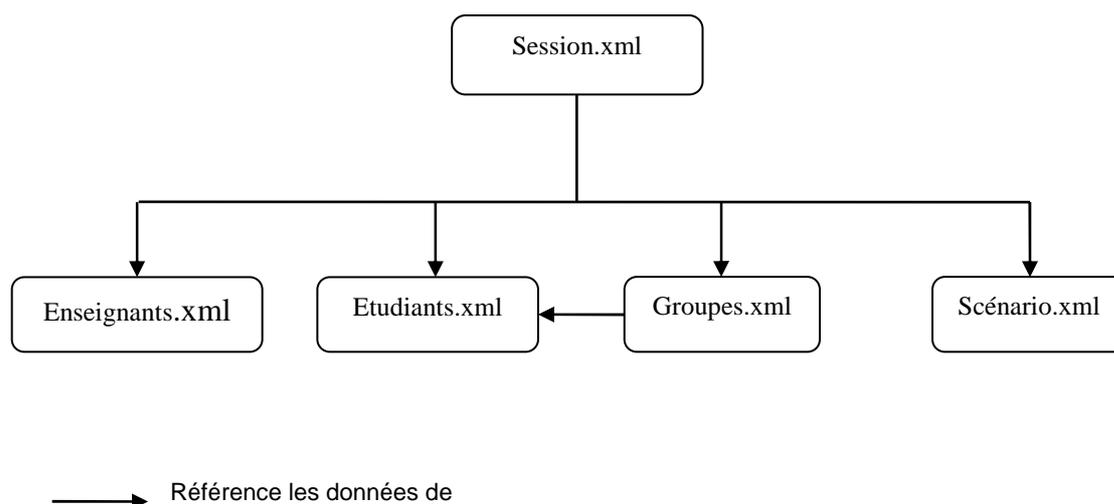


Figure 47 : Relation entre les différents fichiers XML constituant une session Hop3x.

Ces fichiers sont présentés dans la suite avec des extraits qui ont été rendus anonymes.

- ❖ Le premier fichier principal est intitulé « **Sessions.xml** » (cf. figure 48). Il contient l'ensemble des sessions d'apprentissage en spécifiant pour chacune un ensemble d'attributs : son titre, le langage de programmation utilisé, l'instant de son début et de sa fin, les noms des fichiers XML qui définissent le scénario de TP, les noms des apprenants et/ou des groupes d'apprenants et des tuteurs impliqués.

```
<SESSIONS xmlns:xsi="http://lium.fr" xmlns:hop3x="http://lium.fr">
  <SESSION LANGAGE="Java" SCENARIO="tp1.xml" NOM="TP1" MODE="AUTOMATIQUE" DEBUT="10" FIN="13" CONFIRMER="false">
    <GROUPE>GA</GROUPE>
    <GROUPE>GC</GROUPE>
    <GROUPE>GB</GROUPE>
    <ETUDIANT>Etud 1</ETUDIANT>
    <ETUDIANT>Etud 8</ETUDIANT>
    <ENSEIGNANT>Prof 1</ENSEIGNANT>
    <ENSEIGNANT>Prof 3</ENSEIGNANT>
  </SESSION>
  <SESSION LANGAGE="Java" SCENARIO="tp2.xml" NOM="TP2" MODE="AUTOMATIQUE" DEBUT="15" FIN="18" CONFIRMER="false">
    <GROUPE>GC</GROUPE>
    <ETUDIANT>Etud 6</ETUDIANT>
    <ENSEIGNANT>Prof 1</ENSEIGNANT>
    <ENSEIGNANT>Prof 3</ENSEIGNANT>
  </SESSION>
</SESSIONS>
```

Figure 48 : Extrait du fichier XML des sessions dans Hop3x.

- ❖ Le fichier « **Groupes.xml** » définit la composition des groupes d'apprenants (cf. figure 49). Il contient l'ensemble des groupes. Chaque groupe contient les login (*utilisateur*) des apprenants qui le constituent. Pour chacune des sessions le serveur utilise le nom de groupe pour identifier les apprenants impliqués.

```
<GROUPES>
  <GROUPE NOM="GA">
    <UTILISATEUR>Etud1</UTILISATEUR>
    <UTILISATEUR>Etud 7</UTILISATEUR>
    <UTILISATEUR>Etud 10</UTILISATEUR>
  </GROUPE>
  <GROUPE NOM="GB">
    <UTILISATEUR>Etud 9</UTILISATEUR>
    <UTILISATEUR>Etud 4</UTILISATEUR>
  </GROUPE>
  <GROUPE NOM="GC">
    <UTILISATEUR>Etud 3</UTILISATEUR>
    <UTILISATEUR>Etud 5</UTILISATEUR>
  </GROUPE>
</GROUPES>
```

Figure 49 : Extrait du fichier XML des groupes dans Hop3x.

- ❖ Les données des apprenants sont stockées dans un autre fichier XML nommé « **Etudiants.xml** » (cf. figure 50), ces données concernent le nom, le prénom, le login (*utilisateur*) et mot de passe (*MDP*) de chaque apprenant.

```
<ETUDIANTS>
<ETUDIANT NOM="Etud 1" PRENOM="Etud 1">
<UTILISATEUR>Etud 1</UTILISATEUR>
<MDP>E1</MDP>
</ETUDIANT>
<ETUDIANT NOM="Etud 2" PRENOM="Etud 2">
<UTILISATEUR>Etud2</UTILISATEUR>
<MDP>E2</MDP>
</ETUDIANT>
<ETUDIANT NOM="Etud 3" PRENOM="Etud 3">
<UTILISATEUR>Etud 3</UTILISATEUR>
<MDP>E3</MDP>
</ETUDIANT>
<ETUDIANT NOM="Etud 4" PRENOM="Etud 4">
<UTILISATEUR>Etud 4</UTILISATEUR>
<MDP>E4</MDP>
</ETUDIANT>
</ETUDIANTS>
```

Figure 50 : Extrait du fichier XML des apprenants dans Hop3x.

- ❖ Les données des tuteurs sont stockées dans un fichier XML nommé « **Enseignants.xml** » (cf. figure 51). Ces données concernent le nom, le prénom, le login (*utilisateur*) et le mot de passe (*MDP*) de chaque tuteur.

```

<ENSEIGNANTS>

<ENSEIGNANT NOM="Prof 1" PRENOM="Prof 1">
<UTILISATEUR>Prof 1</UTILISATEUR>
<MDP>p1</MDP>
</ENSEIGNANT>

<ENSEIGNANT NOM="Prof 2" PRENOM="Prof 2">
<UTILISATEUR>Prof 2</UTILISATEUR>
<MDP>p2</MDP>
</ENSEIGNANT>

<ENSEIGNANT NOM="Prof 3" PRENOM="Prof 3">
<UTILISATEUR>Prof 3</UTILISATEUR>
<MDP>p3</MDP>
</ENSEIGNANT>

<ENSEIGNANT NOM="Prof 1" PRENOM="Prof 4">
<UTILISATEUR>Prof 4</UTILISATEUR>
<MDP>p4</MDP>
</ENSEIGNANT>

<ENSEIGNANT NOM="Prof 1" PRENOM="Prof 5">
<UTILISATEUR>Prof 5</UTILISATEUR>
<MDP>p5</MDP>
</ENSEIGNANT>

</ENSEIGNANTS>

```

Figure 51 : Extrait du fichier XML des tuteurs dans Hop3x.

```

<SUJET>
<CONSIGNES>
<![CDATA[
<font face="Comic Sans MS"><u>A la fin du TP</u>, r&eacute;cuperez les sources dans data/workspace et g&eacute;ner&eacute;rez
la javadoc que vous enverrez zipp&eacute;e à Prof@univ-lemans.fr</font></div>
]]>
</CONSIGNES>

<QUESTION NUMBER="1">
<ENONCE> <![CDATA[<font face="Comic Sans MS">Ecrire la classe Point compos&eacute;e de deux champs
(valeurs r&eacute;elles) repr&eacute;sentant les coordonn&eacute;es d&#8217;un
point en x et y.</font>]]> </ENONCE>
</QUESTION>

<QUESTION NUMBER="2">
<ENONCE> <![CDATA[<font face="Comic Sans MS">Red&eacute;finir la m&eacute;thode <u>equals</u> pour
la classe Point. </font><br>]]> </ENONCE>
</QUESTION>

<QUESTION NUMBER="3">
<ENONCE> <![CDATA[<font face="Comic Sans MS">Ajouter un
comportement, que l'on appellera <u>distance</u>, permettant &agrave; un objet de la classe Point
de calculer la distance qui le
s&eacute;pare d&#8217;un autre
point.</font>]]> </ENONCE>
</QUESTION>

<QUESTION NUMBER="4">
<ENONCE> <![CDATA[<font face="Comic Sans MS">Un triangle &eacute;tant d&eacute;fini par trois points, Ecrire la classe Triangle.</font>]]>
</ENONCE>
</QUESTION>

<QUESTION NUMBER="5">
<ENONCE> <![CDATA[<font face="Comic Sans MS">Ajouter un
comportement, que l'on appellera <u>perimetre</u>, permettant &agrave; un objet de la classe Triangle de calculer son
p&eacute;rim&egrave;tre. </font>]]> </ENONCE>
</QUESTION>

<QUESTION NUMBER="6">
<ENONCE> <![CDATA[<font face="Comic Sans MS">Ajouter un
comportement, que l'on appellera <u>surface</u>, permettant de calculer la surface d'un triangle</font>]]></ENONCE>
</QUESTION>
</SUJET>

```

Figure 52 : Extrait du fichier XML d'un scénario de TP dans Hop3x.

- ❖ Le scénario d'apprentissage d'une session est spécifié dans un fichier qui porte son nom (par exemple « TP1.xml »). Ce fichier définit le sujet de TP à résoudre par les apprenants. Il contient les consignes à respecter et un énoncé constitué d'un ensemble de questions (cf. figure 52).

7.2.1.2 Expérimentation en vue d'observer l'activité des usagers de Hop3x

Nous avons effectué cette expérimentation pour observer en direct les comportements des usagers de Hop3x (essentiellement les enseignants tuteurs). Nous précisons dans ce qui suit le contexte de déroulement de cette expérimentation et nos objectifs. Nous décrivons la session d'apprentissage mise en place et les observations que nous avons pu recueillir lors de son exécution.

7.2.1.2.1 Contexte de l'expérimentation

En collaboration avec d'autres chercheurs de LIUM travaillant sur le calcul des indicateurs et l'autorégulation de l'activité du tuteur, nous avons mené une expérimentation au sein du département SRC de l'IUT de Laval entre la fin mai et le début juin 2010.

Le corpus de notre étude se compose de deux enseignants concepteurs, deux enseignants tuteurs et six groupes comprenant chacun 15 étudiants de niveau Bac+1. Ces étudiants préparent un DUT SRC qui a pour objectif de former des techniciens supérieurs aptes à concevoir et gérer des outils numériques de communication. Ces étudiants sont novices en programmation Java, mais ils ont suivi, en première partie du module, un cours d'introduction intitulé « Introduction à la programmation orientée objet » où ils ont assisté aux cours magistraux et se sont servis de tutoriaux. Hop3x est mis à leur disposition pour leur permettre de résoudre des exercices de TP structurés sous forme de sessions. Trois heures sont planifiées pour répondre à 12 questions. Le but est de mettre en œuvre les différents concepts dans des situations concrètes. La session mise en place est créée par deux enseignants (les concepteurs) du département d'informatique de l'université du Maine. Les enseignants du département SRC de l'IUT de Laval ont joué le rôle de tuteurs durant les sessions. Ces tuteurs sont spécialistes des nouvelles technologies mais certains restent néophytes quant à leur utilisation dans le cadre d'activités d'enseignement. Le but ici est de découvrir la capacité de déploiement de Hop3x à une échelle plus large, ceci en mettant la lumière sur les difficultés de son utilisation par d'autres enseignants qui n'ont pas participé à son développement.

Une salle équipée de 15 postes (ordinateurs, casques et micros) a été utilisée. Chaque apprenant a disposé d'un poste. Le tuteur a travaillé à distance en utilisant un ordinateur avec micro et haut-parleurs. La partie serveur de Hop3x a été déployée sur le serveur de l'IUT. La partie Interface-Apprenant a été mise en ligne et chaque apprenant l'a installée sur son ordinateur. Le tuteur a installé également la partie Interface-Tuteur sur son ordinateur.

7.2.1.2.2 Objectifs de l'expérimentation

À travers cette expérimentation, réalisée avec d'autres chercheurs de LIUM, nos objectifs visaient particulièrement à :

- ❑ identifier et comprendre les pratiques développées avec Hop3x dans un cadre expérimental somme tout proche du réel,
- ❑ analyser le déroulement effectif des sessions Hop3x,
- ❑ cerner les besoins des enseignants en termes d'adaptation en temps réel des sessions d'apprentissage,
- ❑ distinguer entre les éléments de la session lesquels sont adaptables et lesquels restent fixes pendant le déroulement,
- ❑ identifier les fonctionnalités nécessaires à intégrer dans la version initiale du système, pour faciliter certaines tâches aux tuteurs.

7.2.1.2.3 La session d'apprentissage mise en place

Les sessions d'apprentissage que nous étudions ici concernent la première séance de travaux pratiques (TP) du module de la programmation orientée-objet (POO) en Java. Dans chaque session un seul tuteur et un groupe d'étudiants ont été impliqués. Les étudiants avaient 3 heures pour répondre à 12 questions à propos des notions de base de la POO Java telles que : classe, objet, instance, héritage, encapsulation, polymorphisme, etc. L'énoncé de cette session se constitue des 12 questions suivantes :

1. Ecrire la classe *Point* composée de deux champs (valeurs réelles) représentant les coordonnées d'un point en x et y.
2. Redéfinir la méthode *equals* pour la classe *Point*.
3. Ajouter un comportement, que l'on appellera *distance*, permettant à un objet de la classe *Point* de calculer la distance qui le sépare d'un autre point.

4. Un triangle étant défini par trois points, écrire la classe *Triangle*.
5. Ajouter un comportement, que l'on appellera *perimetre*, permettant à un objet de la classe *Triangle* de calculer son périmètre.
6. Ajouter un comportement, que l'on appellera *surface*, permettant de calculer la surface d'un triangle.
6. 2. Ecrire la classe *Droite*, une droite étant définie par deux réels (a et b) représentant les coefficients de l'équation de droite $y=ax+b$.
8. Faire le nécessaire pour qu'il soit possible de créer une droite à partir de deux points (c'est-à-dire la droite qui passe par ces deux points).
9. Ajouter, à la classe *Point*, un comportement, que l'on appellera *estSurDroite*, permettant à un point de tester s'il se trouve sur une droite.
10. Ajouter, à la classe *Droite*, un comportement, que l'on appellera *estParallele*, permettant à une droite de tester si elle est parallèle à une autre.
11. Ajouter, à la classe *Droite*, un comportement, que l'on appellera *intersection*, permettant à une droite de calculer son intersection avec une autre droite.
12. Faire le nécessaire pour qu'il soit possible de créer un *Triangle* à partir de trois droites.

7.2.1.2.4 Observations recueillies

Au début du déroulement de la session, les interventions tutorales sont essentiellement effectuées en réponse à des problèmes d'ordre technique, les apprenants se familiarisent d'abord avec le système avant de répondre aux questions. Certains d'entre eux ont par exemple sollicité le tuteur puis annulé immédiatement leur requête parce qu'ils ne savaient pas formuler précisément leur besoin. Une fonctionnalité permettant aux apprenants la spécification de la cause de la demande d'aide aurait sans doute été pertinente pour faciliter au tuteur la réponse aux demandes. Nous avons également observé certains détournements du système par les usagers comme, par exemple, demander de l'aide au tuteur en rédigeant une demande comme un commentaire de code. La raison est que la fonctionnalité de demande textuelle d'aide n'existe pas dans « l'interface-apprenant ».

Dans certaines situations le tuteur était obligé de refaire la même intervention à plusieurs reprises ou bien de changer la modalité d'intervention s'il remarquait qu'elle n'avait pas d'effet sur l'apprenant concerné. Ces interventions pouvaient concerner un apprenant qui ne comprenait pas ou bien plusieurs ayant les mêmes demandes.

Durant le déroulement des sessions, les tuteurs suivaient ce qui était prédéfini par le concepteur en guidant les apprenants en fonction des objectifs pédagogiques visés. Certaines situations auraient nécessité des modifications de la structure ou du contenu des sessions, ce que le système ne permet pas. Pour surmonter cette limite, les tuteurs ont pu ajouter des questions aux étudiants concernés par le biais des outils de communication, sans modifier pour autant la structure de la session, c'est une adaptation par déviation (cf. section 2.1.3.3.2). En outre, les tuteurs pouvaient utiliser les outils de communication audio ou la messagerie pour donner des exemples, des renvois vers des sites de documentation, des rappels de solutions déjà vues au cours, des pistes de solutions, des explications des questions, etc. Des apprenants ont par exemple demandé de l'aide pour comprendre certaines questions car les énoncés étaient incomplets : par exemple l'absence d'une formule mathématique dans la question 8. Dans ce cas, les tuteurs ont donné explicitement la formule mathématique nécessaire via la messagerie.

Si le tuteur peut en temps réel découvrir le code écrit, il peut aussi suivre des apprenants grâce deux types d'indicateurs [Lekira 2010] : - des indicateurs spécifiques liés au sujet du TP (par exemple le respect des règles d'encapsulation) - des indicateurs transversaux liés aux bonnes pratiques (utilisation des commentaires *JavaDoc*, etc.) ou à la qualité du déroulement de la session (temps passé par un apprenant sur une question, etc.). En pratique, les tuteurs utilisent essentiellement les indicateurs spécifiques. Pendant l'exécution, les tuteurs ont par ailleurs ressenti le manque de certains indicateurs non disponibles, comme par exemple un indicateur révélant les apprenants inactifs.

7.2.1.3 Entretiens

À chaque étape de notre méthode d'instrumentation, nous avons réalisé des entretiens nous permettant de recueillir auprès de différents acteurs des informations sur les différentes facettes du métier embarqué dans Hop3x. Ceci nous a permis de recueillir deux types de données : des informations que nous ne pouvions pas obtenir autrement que par entretien de face à face ; des informations permettant de compléter et de comprendre, de donner sens, à d'autres données recueillies par d'autres méthodes sans interprétation sémantique dans l'univers métier de Hop3x.

7.2.1.3.1 Entretien avec les concepteurs initiaux de Hop3x

Nous avons mené nos premiers entretiens auprès des concepteurs initiaux de Hop3x. Ce sont des enseignants/chercheurs en informatique qui ont développé Hop3x pour leurs propres besoins d'enseignement des travaux pratiques en programmation orientée-objet d'une part, et d'autre part pour tester leurs hypothèses de recherche sur le tutorat [Lekira et al. 2011]. Ils jouent un double rôle : concepteurs pédagogiques lorsqu'ils définissent le contenu des sessions d'apprentissage, tuteurs pendant le déroulement des sessions. C'est la raison pour laquelle nous considérons ces enseignants comme des experts du domaine spécifique de Hop3x.

L'objectif de nos premiers entretiens avec ces concepteurs était de découvrir l'architecture et le fonctionnement du système Hop3x (cf. sa section d'avant). Suite à la fouille des fichiers XML existants sur le serveur, nous avons interrogé les concepteurs de Hop3x sur la raison d'existence de chaque élément. Leurs réponses nous ont permis de formaliser ces éléments dans un métamodèle (cf. étape suivante) qui cristallise les connaissances métiers déjà explicitées. Ce métamodèle initial sert à construire dans un premier temps un éditeur basique des sessions non-ouvertes.

Par ailleurs, nous avons posé un ensemble de questions ouvertes aux enseignants concepteurs. Le but était de récolter explicitement les informations sur les circonstances de conception des sessions exécutables sur Hop3x. Nous nous sommes focalisés sur les pratiques exercées et les instruments utilisés dans le processus de création de la structure et du contenu de ces sessions. En réponse, les enseignants ont confirmé qu'ils n'utilisaient ni une méthodologie formelle, au sens propre de terme, ni un outil spécifique. Ils ont créé manuellement des fichiers XML pour définir la structure et le contenu des sessions en utilisant un éditeur de texte. La méthode n'est bien évidemment pas appropriée pour tous les enseignants qui souhaiteraient utiliser Hop3x pour mettre en place des sessions d'apprentissage, des compétences informatiques étant nécessaires pour formaliser et opérationnaliser les sessions.

Mais nous avons aussi constaté que les concepteurs des sessions utilisent implicitement des concepts métier qui n'existent pas dans les fichiers XML. Nous avons ainsi pu identifier : « l'objectif de la session », « l'objectif de chaque question » et « les tâches liées à chaque question ». Les enseignants nous ont dit qu'ils ciblent un objectif d'apprentissage par la définition de chaque session, voire des sous objectifs pour chaque question, malgré le fait qu'ils ne les expriment pas explicitement. De plus, ils spécifient trois types de tâches pour chaque question : tâches obligatoires, tâches souhaitées, et tâches interdites. La définition

de ces tâches doit dépendre de chaque apprenant ou groupe d'apprenant, selon le contexte d'apprentissage. La définition explicite des caractéristiques de chaque question, y compris son objectif, est nécessaire pour plusieurs causes, par exemple pour guider les tuteurs dans l'adaptation dynamique pendant l'exécution, ainsi que pour des raisons de capitalisation et réutilisation des questions, voire des variantes.

Les enseignants concepteurs ont affirmé qu'ils ne peuvent pas prévoir en amont tous les éléments qui constituent le contenu d'une session. Ce n'est que durant le déroulement et en fonction de la progression des activités des apprenants que les adaptations peuvent être effectuées en fonction de l'objectif d'apprentissage ciblé et du contexte (apprenant concerné, temps restant, etc.). La première fois, ces adaptations sont imprévisibles mais dans les cycles suivants, elles peuvent être considérées comme prévisibles si les enseignants concepteurs décident de les capitaliser à travers un processus de réingénierie. Par exemple, après le déroulement de la session de TP déployée dans notre expérimentation, et après avoir constaté que les tuteurs ont donné aux apprenants une formule mathématique qui manquait à la question 8, pour les aider à avancer, les enseignants concepteurs l'ont ajoutée dans le contenu même de la session. Elle est considérée comme une adaptation améliorante de la question 8.

Un autre objectif de nos entretiens avec les experts de Hop3x visait à faire une distinction entre les éléments de la session qui sont adaptables et ceux qui sont fixes. À cet effet, nous avons conduit des négociations sur les choix à prendre pour améliorer le métier formalisé afin de rendre les sessions ouvertes. Nous avons donc pu identifier des éléments statiques d'une session et ceux qui peuvent être adaptables. Pour les premiers on distingue : les caractéristiques individuelles de la session (nom, langage, mode, confirmation, temps de début et de fin), ainsi que la liste des tuteurs qui peuvent être impliqués. Pour les seconds, on distingue l'ensemble des questions et leurs caractéristiques. Il apparaît par ailleurs que l'enseignant peut adapter la session suivant trois éléments : les indicateurs sur l'activité d'apprentissage, les apprenants et/ou les groupes impliqués. Ce sont les éléments qui caractérisent le contexte d'exécution de la session.

Nous nous appuyons sur toutes ces informations pour faire évoluer le métamodèle initial des sessions non-ouvertes vers un nouveau qui permet d'exprimer les sessions ouvertes.

7.2.1.3.2 Entretien avec les tuteurs impliqués dans l'expérimentation

Lors de notre expérimentation, nous avons mené nos observations essentiellement auprès des tuteurs. Ce sont en effet leurs comportements durant l'exécution des sessions que nous cherchions à comprendre, analyser et interpréter afin de :

- ❑ cerner les besoins en termes d'adaptation des sessions en temps réel,
- ❑ déterminer d'autres concepts métiers qui ne sont pas formalisés dans les fichiers XML,
- ❑ identifier les fonctionnalités nécessaires à intégrer dans la version initiale du système pour faciliter certaines tâches assez récurrentes.

Nous leur avons donc posé un ensemble de questions déliées à leurs interventions, au fur et à mesure et à posteriori du déroulement des sessions. Cela nous a permis de confirmer et d'approfondir notre observation et de préconiser des fonctionnalités nécessaires à intégrer dans le système pour promouvoir son ouverture.

Nous avons pu remarquer que les tuteurs ne pouvaient pas mettre à jour le contenu de la session pendant son déroulement en utilisant la version initiale du système. Les modifications apparaissant nécessaires ne peuvent être effectuées qu'à travers les interventions tuteurales introduites via les outils de communication (audio ou messagerie). Elles relèvent donc de l'approche d'adaptation par déviation (cf. section 2.1.3.3.2), c'est-à-dire, changement de l'instance de la session sans modifier sa définition et son contenu, tels que l'ajout ou la suppression des questions en temps réel. Ce sont des adaptations éphémères qui ne provoquent pas un changement de la structure de la session dans le serveur, donc pas de capitalisation et de réutilisation des adaptations considérées comme positives.

7.2.2 Définition du métamodèle des sessions non-ouvertes et prototypage d'un premier éditeur spécifique

Lors de cette étape, notre but était de construire un premier outil pour réifier le métier de Hop3x. Pour ce faire, nous avons utilisé EMF (*Eclipse Modeling Framework*)¹⁷, qui facilite la génération de code pour la construction d'outils basés sur des métamodèles structurés [Steinberg et al. 08]. Nous avons donc défini un métamodèle décrivant le DSEML¹⁸ des sessions Hop3x, qui a permis ensuite le prototypage d'un premier éditeur de sessions spécifique au domaine métier de Hop3x (cf. figure 53). Cet éditeur spécifique ne permet pas de produire des sessions d'apprentissage ouvertes, mais il est construit pour valider le métamodèle qui formalise le métier.

¹⁷ www.eclipse.org/emf, dernier visite Avril 2012.

¹⁸ DSEML: *Domain Specific Educational Modeling Language*.

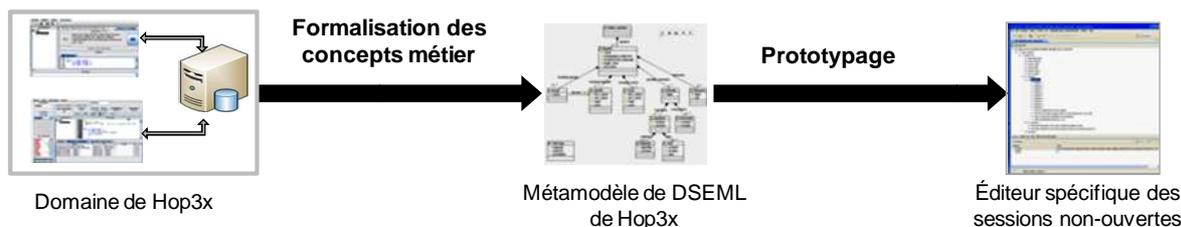


Figure 53 : Processus d'élaboration de DSEML de Hop3x et d'un éditeur spécifique.

Le métamodèle a été défini comme un modèle conforme à ECORE (le méta-métamodèle d'EMF). Pour ce faire, nous avons utilisé les informations récupérées à partir des fichiers XML existants dans le serveur Hop3x, après, toutefois, avoir conclu un compromis sur les significations et les raisons d'être de chaque élément et attribut de ces fichiers XML et avoir validé l'ensemble, lors des entretiens que nous avons menés auprès des enseignants créateurs de ces fichiers. La figure 54 présente le métamodèle élaboré pour cristalliser les connaissances métier déjà formalisées dans les fichiers XML.

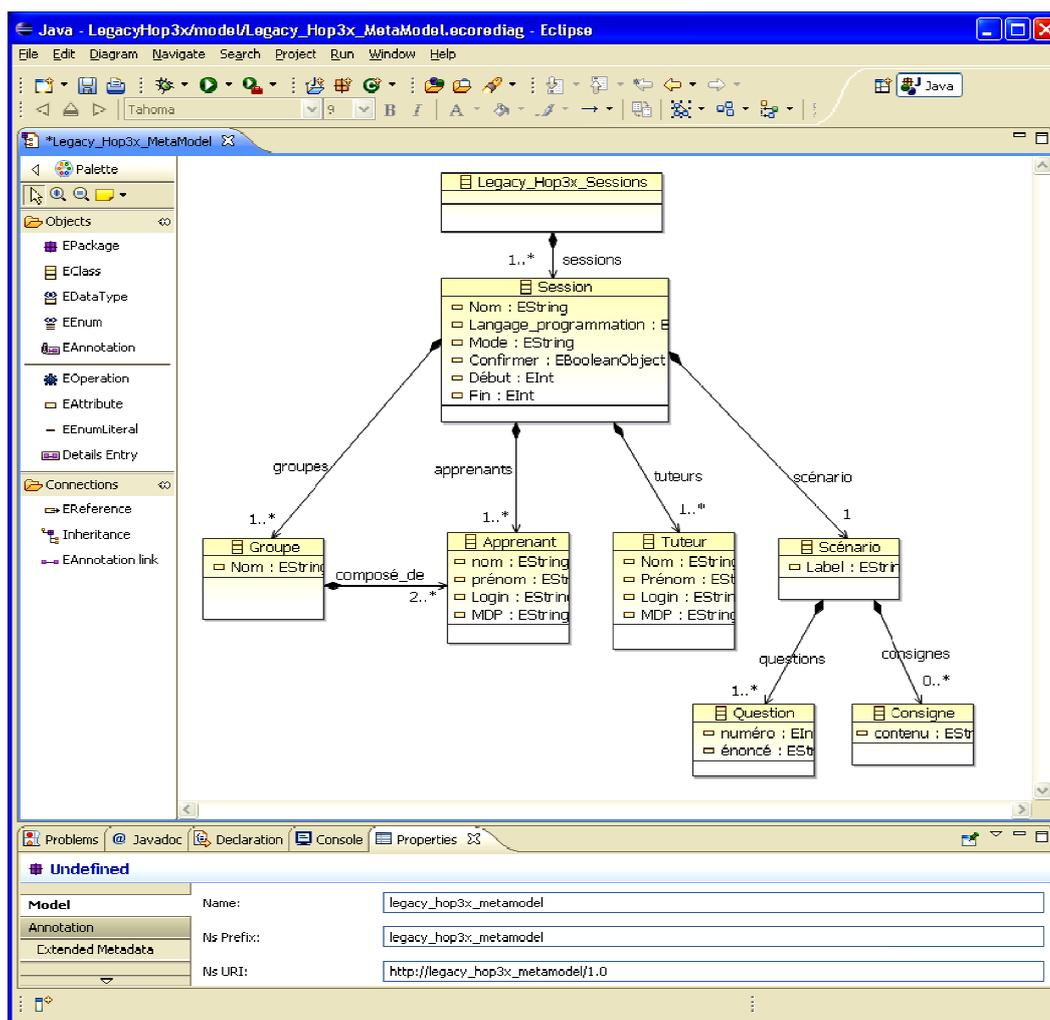


Figure 54 : Métamodèle des sessions Hop3x non-ouvertes.

En nous basant sur ce métamodèle et grâce à l'outillage d'EMF nous avons généré un éditeur basique (cf. figure 55). Il fournit une vue d'arborescence des modèles qui sont en l'occurrence des sessions Hop3x.

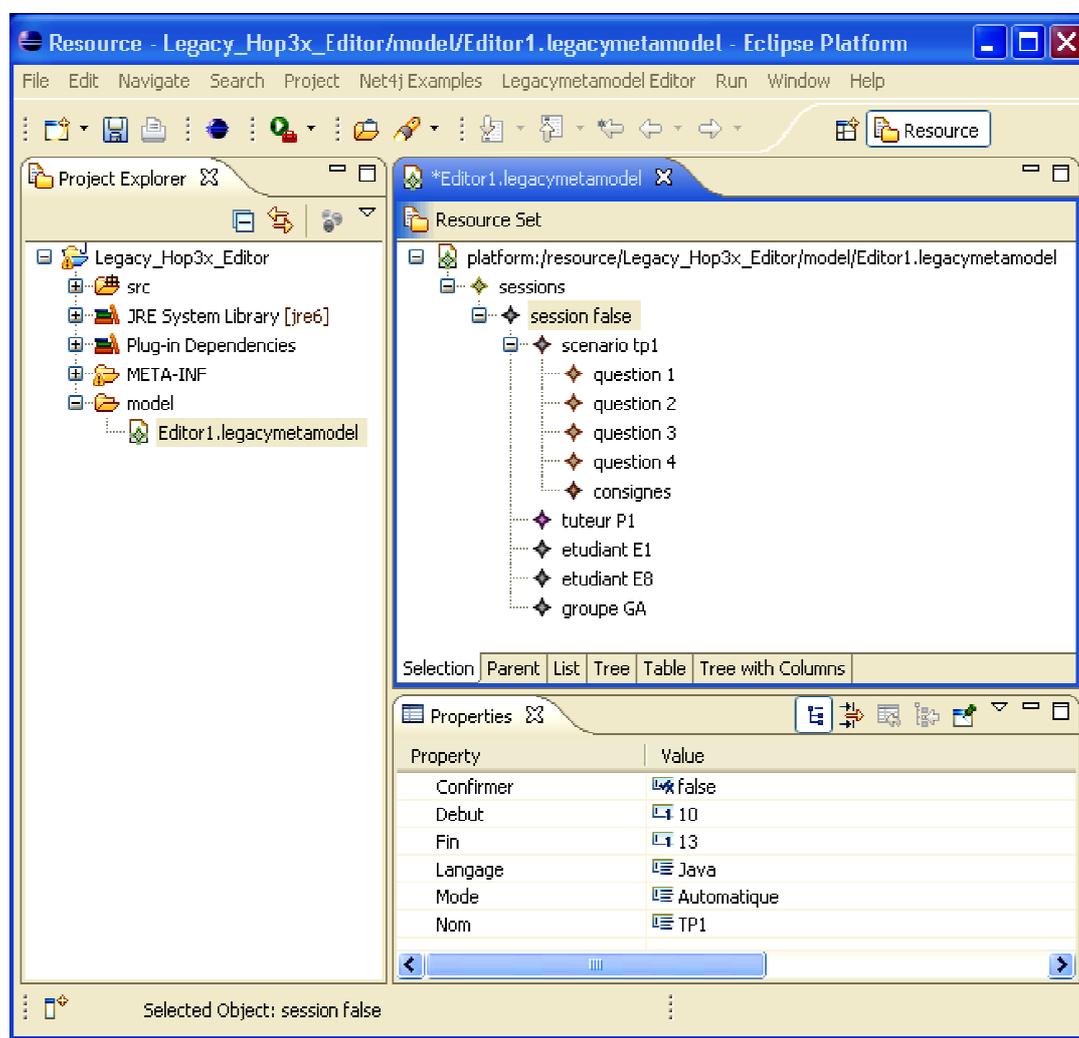


Figure 55 : Editeur basique généré par l'outillage d'EMF.

7.2.3 Évolution du métamodèle initial vers un autre des sessions ouvertes

L'objectif de cette étape est de définir un métamodèle qui prend en compte les aspects d'ouverture des sessions d'apprentissage de Hop3x. Les sessions doivent être structurées en variantes conformément au modèle rationnel des SPO que nous avons proposé (cf. chapitre 3). Pour ce faire, nous avons fait évoluer le métamodèle, défini initialement pour décrire le DSEML (cf. section 7.2.2), vers un autre métamodèle plus enrichi qui permet d'exprimer graphiquement les sessions ouvertes (cf. la figure 56). Ce métamodèle décrit le langage spécifique de modélisation graphique des sessions ouvertes de Hop3x (ADSGEML - *Adaptive Domain Specific Graphical Educational Modeling Language*).

L'évolution du métamodèle a consisté à mener une catégorisation des concepts du domaine spécifique de Hop3x selon les dimensions du modèle rationnel des SPO. Cette catégorisation est purement d'ordre métier, elle s'appuie sur des décisions prises lors des entretiens que nous avons menés avec les concepteurs de Hop3x, avec lesquels nous avons conclu des compromis sur la signification de chaque concept métier de Hop3x en termes de primitives du modèle rationnel (couche statique, couche des variantes, objectif d'apprentissage, etc.). Techniquement, cette évolution est réalisée par un tissage du métamodèle qui formalise le métier existant de Hop3x, avec le métamodèle qui implémente le modèle rationnel des SPO. Cette opération manuelle est effectuée via la définition des relations d'héritage entre les primitives du métamodèle rationnel et les concepts du domaine de Hop3x.

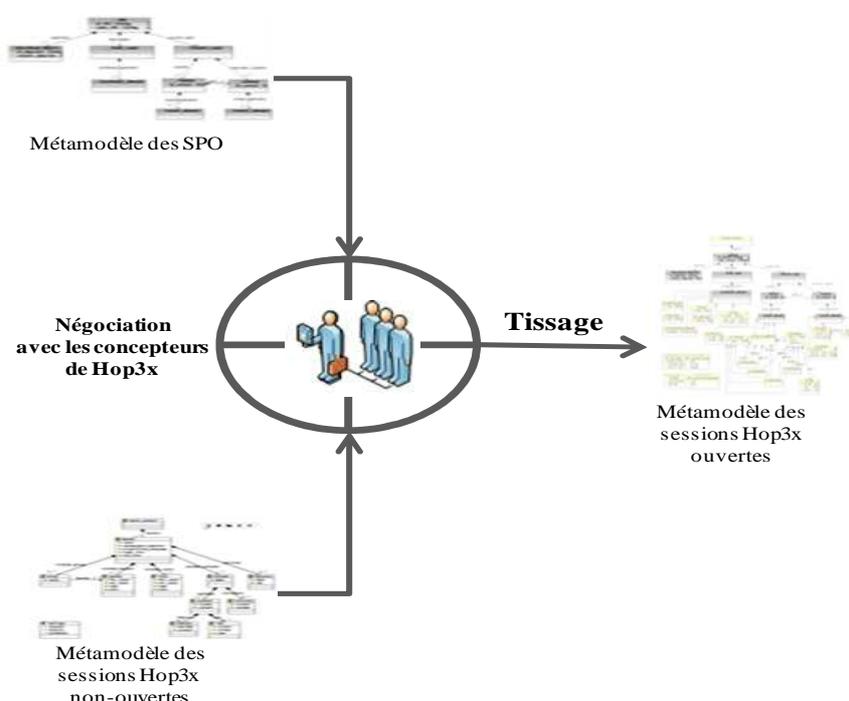


Figure 56 : Processus d'évolution du métamodèle initial en métamodèle des sessions ouvertes.

Pour implémenter le métamodèle évolué, nous avons utilisé EMF qui offre un éditeur graphique permettant de créer des métamodèles conformes à Ecore. Sachant qu'Ecore est le méta-métamodèle d'EMF (l'équivalent du méta-métamodèle MOF de l'OMG). Comme le montre la figure 57, nous avons défini le métamodèle évolué sous forme d'un diagramme de classe. Nous l'avons nommé « *MM_Open_Hop3x.ecore* ». Il constitue le point de départ du processus de développement dirigé par les modèles de l'environnement d'édition graphique des sessions Hop3x ouvertes.

Les concepts du métamodèle évolué (cf. figure 57) sont issus de quatre sources. Nous distinguons :

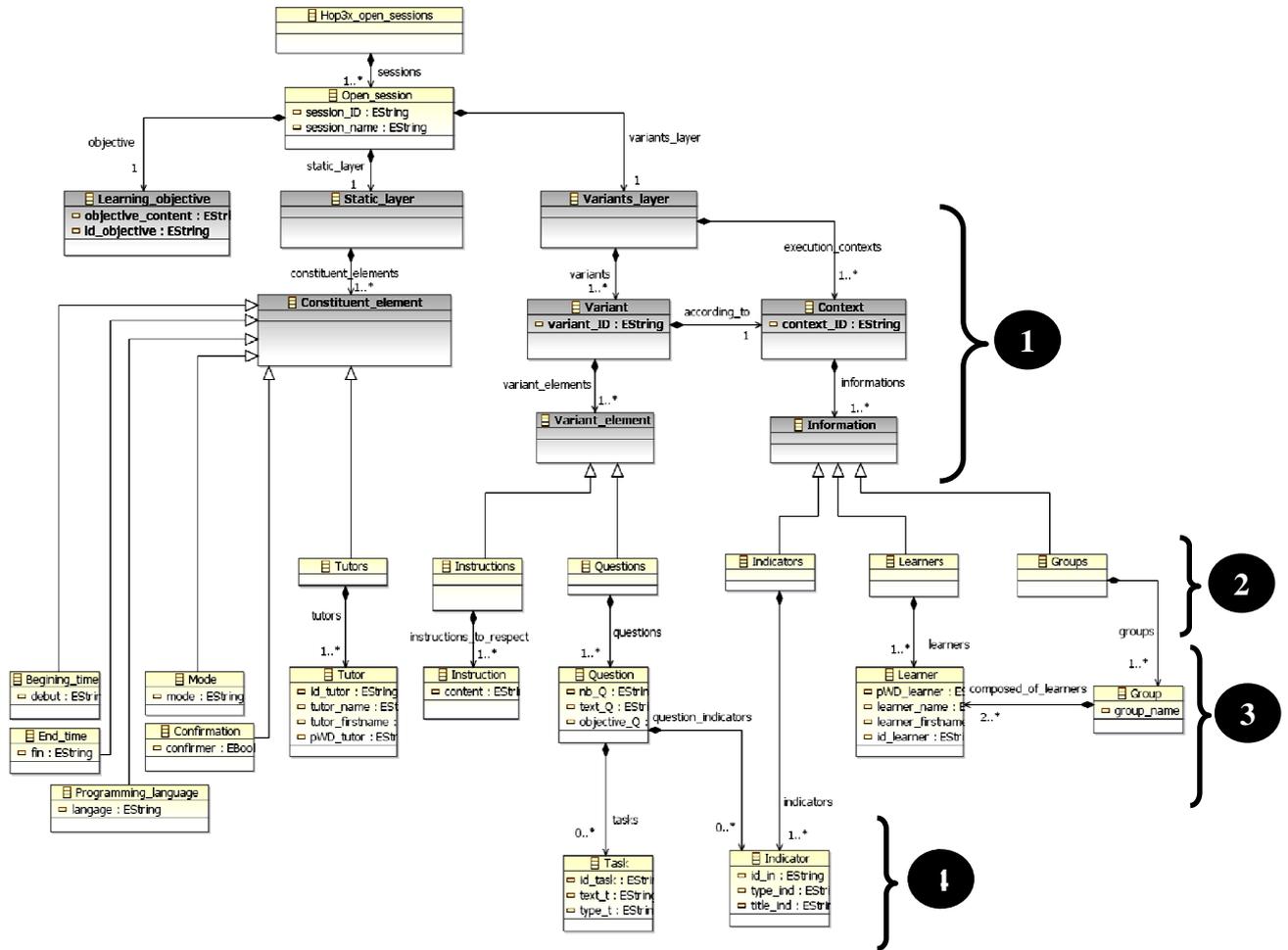
- (1) Les concepts du modèle rationnel des SPO, qui consistent à définir les dimensions permettant de réorganiser la structure des sessions Hop3x pour les rendre ouvertes.
- (2) Les concepts ajoutés pour répondre aux exigences liées au processus de développement de l'environnement d'édition graphique des sessions ouvertes (cf. section 7.2.4).
- (3) Les concepts qui définissent les connaissances métiers de Hop3x déjà formalisées.
- (4) De nouveaux concepts du domaine de Hop3x, que nous avons définis pour formaliser certaines connaissances métiers déjà utilisées par des enseignants dans leurs pratiques, elles sont « implicites » au sens où formellement elles ne sont pas stockées dans les fichiers XML. L'intégration de ces concepts permet d'enrichir le métamodèle et d'améliorer son expressivité pédagogique.

Le concept « *Hop3x_sessions* » est défini pour répondre principalement aux exigences techniques de développement de l'outil d'édition graphique des sessions.

Quant au concept « *Open_session* », il correspond à la session ouverte à modéliser. Il s'agit du nœud racine pour tous les éléments appartenant à cette session. Ce concept a deux propriétés : un nom "*session_name*" tel qu'il a été défini auparavant dans le fichier XML des sessions déjà existant, et un identifiant "*sessions_ID*" ajouté pour distinguer des sessions portant les noms identiques.

Ainsi, conformément au modèle rationnel, une session doit cibler un seul objectif global d'apprentissage et être constituée de deux couches, l'une statique et l'autre contenant les variantes. Trois concepts ont été donc définis : « *Learning_objective* », « *Static_layer* », « *Variants_layer* » ; liés avec le concept « *Open_session* » par des relations de composition avec la cardinalité « 1 » pour chacune.

Le concept « *Learning_objective* » a quant à lui une propriété "*objective_content*" pour permettre de définir l'objectif d'apprentissage en texte narratif. Ce concept n'a pas été formalisé dans l'état existant de Hop3x. Les enseignants concepteurs des sessions ne l'exprimaient pas explicitement, alors qu'ils le prenaient souvent en considération. Sa spécification explicite permet de capitaliser les connaissances métiers liées aux intentions et raisons de définition de chacune des sessions, voire de l'ensemble des variantes qui la constituent.



Légende :

- 1** ⇒ rationnel des SPO
- 2** ⇒ initial de Hop3x
- 3** ⇒ initial de Hop3x
- 4** ⇒ alisés du domaine de Hop3x

Figure 57 : Le métamodèle évolué qui décrit l'ADSGEML de Hop3x.

Le concept « *Static_layer* » est le nœud racine pour les éléments qui constituent la couche statique de la session. Il est donc relié avec le concept « *Constituent_element* » par une relation de composition avec la cardinalité « 1..* ». En se basant sur les décisions que nous avons prises avec les enseignants concepteurs de Hop3x, nous distinguons un certain nombre d'éléments de la session qui ne changent pas pendant son déroulement. Ce sont globalement les caractéristiques de la session : le temps de début et de fin de la session, son mode de lancement et d'arrêt (automatique ou manuel), la confirmation de son lancement et de son arrêt si son mode est automatique, le langage de programmation à utiliser pour répondre aux questions, et le(s) tuteur(s) qui peuvent être impliqué(s). Ces éléments, qui constituent la couche statique des sessions ouvertes de Hop3x, sont définis respectivement par les concepts suivant : « *Begining_time* », « *End_time* », « *Mode* », « *Confirmation* », « *Programming_language* », « *Tutors* ». Ces concepts sont liés par une relation d'héritage avec le concept « *Constituent_element* », car chacun d'eux est un élément constituant particulier. Le concept « *Tutors* » est relié avec un autre concept « *Tutor* » par une relation de composition portant la cardinalité « 1..* ».

Le concept « *Variants_layer* » est le nœud racine des éléments qui composent la couche des variantes des sessions de Hop3x. Il est composé de deux concepts : « *variant* » et « *Context* », qui correspondent respectivement aux variantes de la session et aux différents contextes de son déroulement. Les deux relations de composition qui les relient au concept « *Variants_layer* » portent la cardinalité « 1..* », car la couche des variantes doit contenir au moins une variante considérée comme unique pour tous les apprenants. Cette situation peut exister tout au début du cycle de vie de la session, quand l'enseignant concepteur définit initialement une seule variante qui pourra subir des adaptations durant l'exécution, afin de répondre aux besoins contextuels des apprenants. De cette situation, de nouvelles variantes pourront émerger suivant les contextes réels des apprenants. De plus, Le concept « *variant* » est relié au concept « *Context* » par la relation « *according_to* » portant la cardinalité « 1 », dans le but d'associer chaque variante à son propre contexte de pertinence.

Une variante peut être définie par un ou plusieurs éléments. C'est la raison pour laquelle le concept « *Variant_element* » est lié par une relation de composition avec le concept « *Variant* » avec une cardinalité « 1..* ». Les éléments qui constituent une variante d'une session Hop3x ouverte sont les questions, auxquelles les apprenants doivent répondre, et les consignes, que les étudiants doivent respecter lorsqu'ils répondent aux questions. Nous avons défini deux concepts (« *Questions* » et « *Instructions* ») qui sont liés au concept « *Variant_element* » par deux relations d'héritage.

Le concept « *Instructions* » correspond donc à l'ensemble des consignes. Le concept « *Instruction* » est défini pour caractériser chaque consigne. La relation de composition entre ces deux concepts porte la cardinalité « 1..* », car chaque variante de la session doit avoir au moins une consigne pour orienter le travail des apprenants.

Afin de caractériser chaque question, nous avons défini le concept « *Question* » dont les attributs “*nb_Q*” et “*text_Q*”, qui correspondent respectivement au numéro et à l'énoncé de la question. Le concept « *Question* » est lié au concept « *Questions* » par une relation de composition avec une cardinalité « 1..* », car une variante doit comporter au moins une question ou plusieurs. En outre, nous avons formalisé d'autres connaissances métiers de Hop3x en nous basant sur les informations collectées lors de nos observations et suite aux entretiens menés avec des enseignants concepteurs de Hop3x. Nous avons donc ajouté l'attribut “*Objective_Q*” au concept « *Question* », car nous avons constaté que lorsque l'enseignant définit chaque question, il cible un objectif spécifique (sous-objectif d'apprentissage de la session). Il prédit ainsi trois types de tâches possibles : ce que les apprenants doivent faire (tâches obligatoires), ou ce qu'ils doivent penser à faire (tâches souhaitables), ou ce qu'ils ne doivent pas faire (tâches interdites). Cela nous a amené à définir un concept « *task* », qui porte un premier attribut : “ *text_t* ” ; pour permettre à l'enseignant d'exprimer narrativement le contenu de la tâche, et un second : “ *type_t* ” pour la spécification du type de la tâche (obligatoire, souhaitée, ou interdite). De plus, l'enseignant peut identifier un ensemble d'indicateurs spécifiques permettant de suivre l'activité des apprenants au moment de répondre à une question. C'est la raison pour laquelle nous avons créé une relation entre le concept « *Question* » et « *Indicator* » avec une cardinalité « 0..* ».

Le concept « *Context* » est le nœud racine de toutes les entités du domaine de Hop3x qui peuvent contenir les informations sur les contextes d'exécution des sessions d'apprentissage. Grâce à nos négociations avec des enseignants concepteurs de Hop3x, nous avons pu distinguer trois entités qui peuvent caractériser le contexte pertinent de chaque variante : les apprenants impliqués, les groupes d'apprenants, et/ou les indicateurs sur les activités d'apprentissage. Nous avons donc défini respectivement trois concepts : « *Learner* », « *Group* », et « *Indicator* », qui composent respectivement trois autres concepts « *Learners* », « *Groups* », et « *Indicators* », qui sont eux-aussi liés par des relations d'héritage avec le concept « *Information* » car chacun comporte les informations qui permettent d'identifier le contexte de déroulement de la session.

7.2.4 Développement dirigé par les modèles d'un environnement d'édition graphique dédié à Hop3x

Nous distinguons cinq parties composantes de la structure d'une session ouverte de Hop3x. Ces parties contiennent des informations sur : les dimensions globales de la session, les variantes, les apprenants, les groupes d'apprenants et les tuteurs. Nous avons donc choisi de développer l'environnement d'édition des sessions ouvertes de Hop3x comme un ensemble de plug-ins d'éditeurs graphiques dans le but de permettre de modéliser séparément chacune des parties des sessions. Plusieurs raisons essentielles nous ont poussés à opter pour ce choix :

- L'exécution d'une session nécessite que le système Hop3x utilise des informations qui se trouvent dans différents fichiers XML. Chacun de ces fichiers comporte le contenu d'une partie de la session.
- Pour éviter des problèmes d'interface de l'éditeur, car le positionnement d'un nombre important d'éléments de la session dans la zone de modélisation peut influencer l'ergonomie conviviale de l'outil.
- La modélisation de chaque partie de la session séparément par un éditeur dédié facilite la tâche pour l'enseignant concepteur pendant le processus continu de conception.
- La conception de certaines parties de la session (essentiellement des variantes) se poursuit pendant l'exécution. Il est donc pertinent de permettre à l'enseignant concepteur d'utiliser que les éditeurs de ces parties.

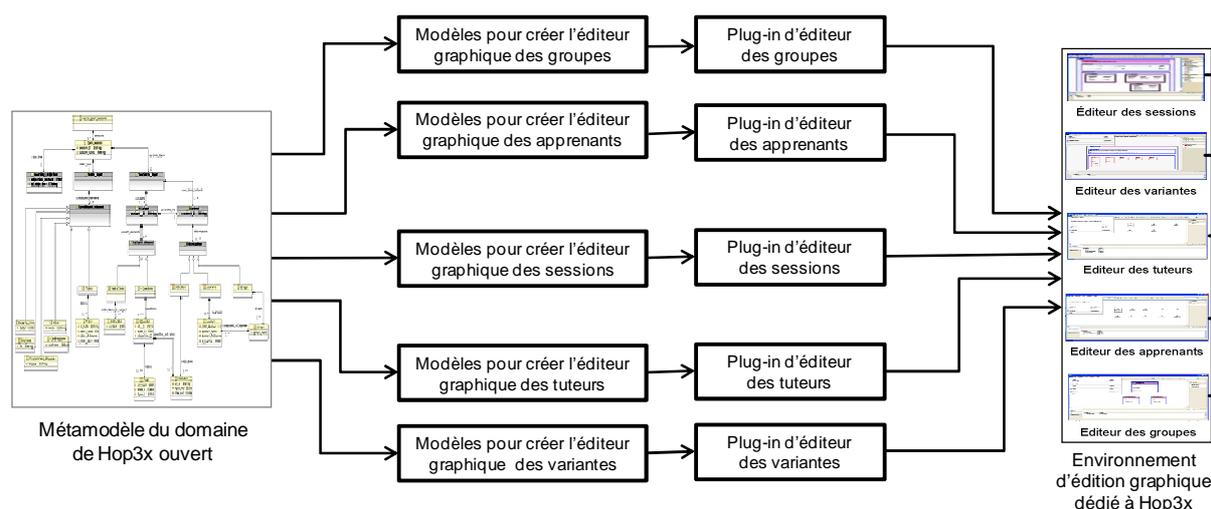


Figure 58 : Les processus dirigés par les modèles pour le développement d'un environnement d'édition graphique dédié à Hop3x.

Le framework GMF (*Graphical Modeling Framework*) (cf. section 2.3.2.2.2) permet de produire le code d'un ou de plusieurs éditeurs à partir d'un modèle de domaine suivant un processus dirigé par les modèles. Ces modèles sont : le modèle de domaine (*.ecore) défini par EMF, le modèle de définition graphique (*.gmfgraph), le modèle de la palette d'outils (*.gmftool), le modèle d'association (mapping) entre les éléments de ces trois modèles (*.gmfmap), le modèle de génération de code (*.gmfgen).

Nous avons employé l'outillage de GMF pour générer le code des plug-ins des éditeurs graphiques de différentes parties des sessions ouvertes de Hop3x. Cela à partir d'un seul métamodèle qui décrit l'ADSGEML de Hop3x ouvert (cf. figure 58). Ce métamodèle a été défini par EMF comme un modèle de domaine conforme à Ecore : « *MM_Open_Hop3x.ecore* ». Il a été utilisé également pour la création d'un modèle générateur EMF « *MM_Open_Hop3x.genmodel* », nécessaire à la génération de deux plug-ins : « *Opening_Hop3x.edit* » et « *Opening_Hop3x.editor* ». Ces deux derniers constituent la couche de bas niveau qui gère les modèles produits par les éditeurs développés et assure leur conformité au métamodèle d'ADSGEML de Hop3x ouvert.

La figure 59 illustre l'arborescence de notre projet GMF, qui comporte les différents modèles définis dans les processus de développement des plug-ins des éditeurs. La présentation en détail de tous ces processus est assez conséquente. Comme ces processus sont identiques dans la démarche, nous ne présentons par la suite (cf. section 7.2.4.1) qu'un seul processus de développement, celui de l'éditeur principal des sessions.

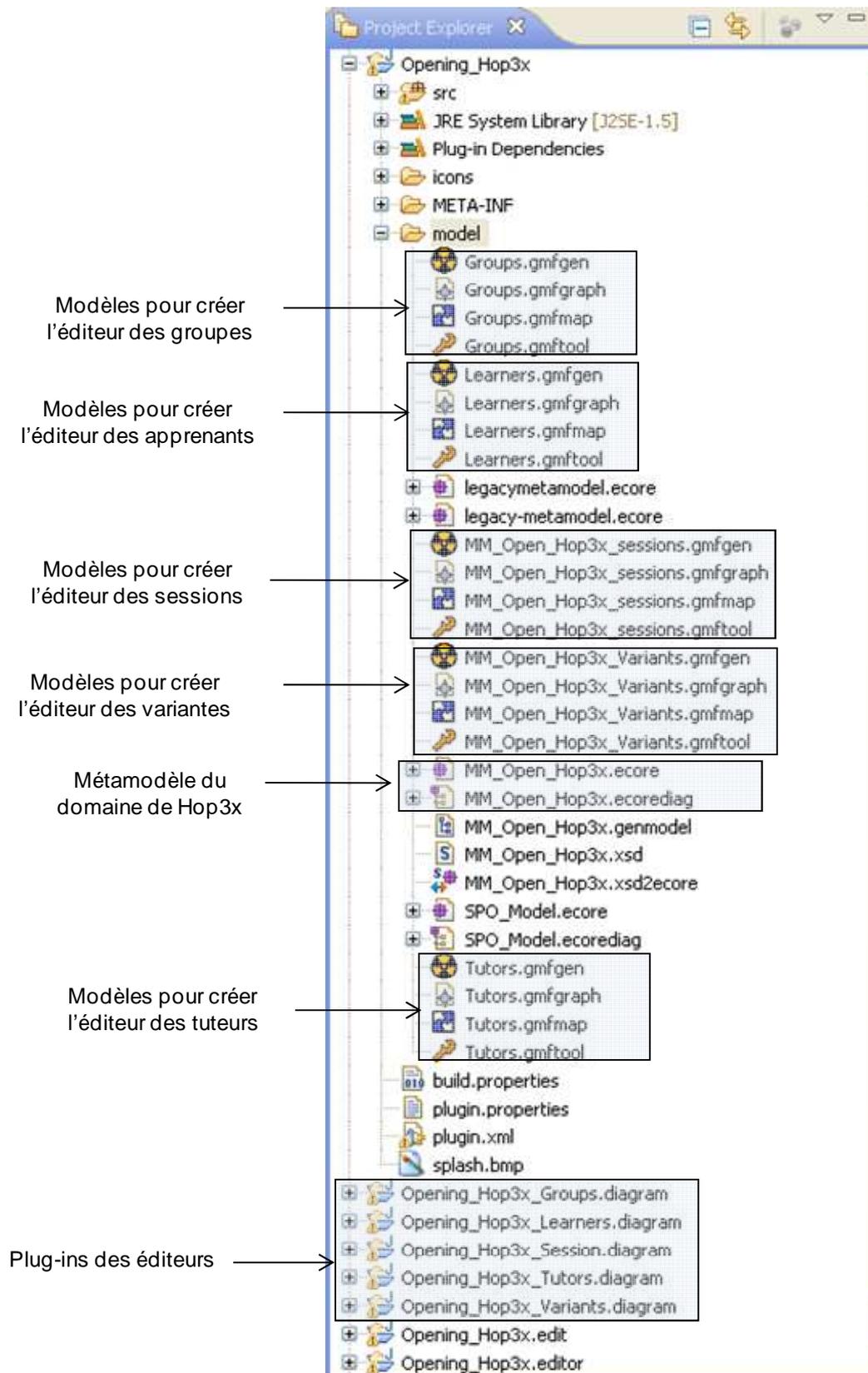


Figure 59 : Capture d'écran illustrant l'ensemble des modèles GMF définis pour le développement des différents éditeurs de Hop3x.

7.2.4.1 Développement de l'éditeur des sessions

La figure 60 présente le processus GMF que nous avons conduit pour développer l'éditeur qui sera utilisé pour la spécification des dimensions globales d'une session ouverte de Hop3x. Ce processus se constitue d'un certain nombre d'étapes consistant à créer des modèles qui permettent la construction de l'éditeur graphique.

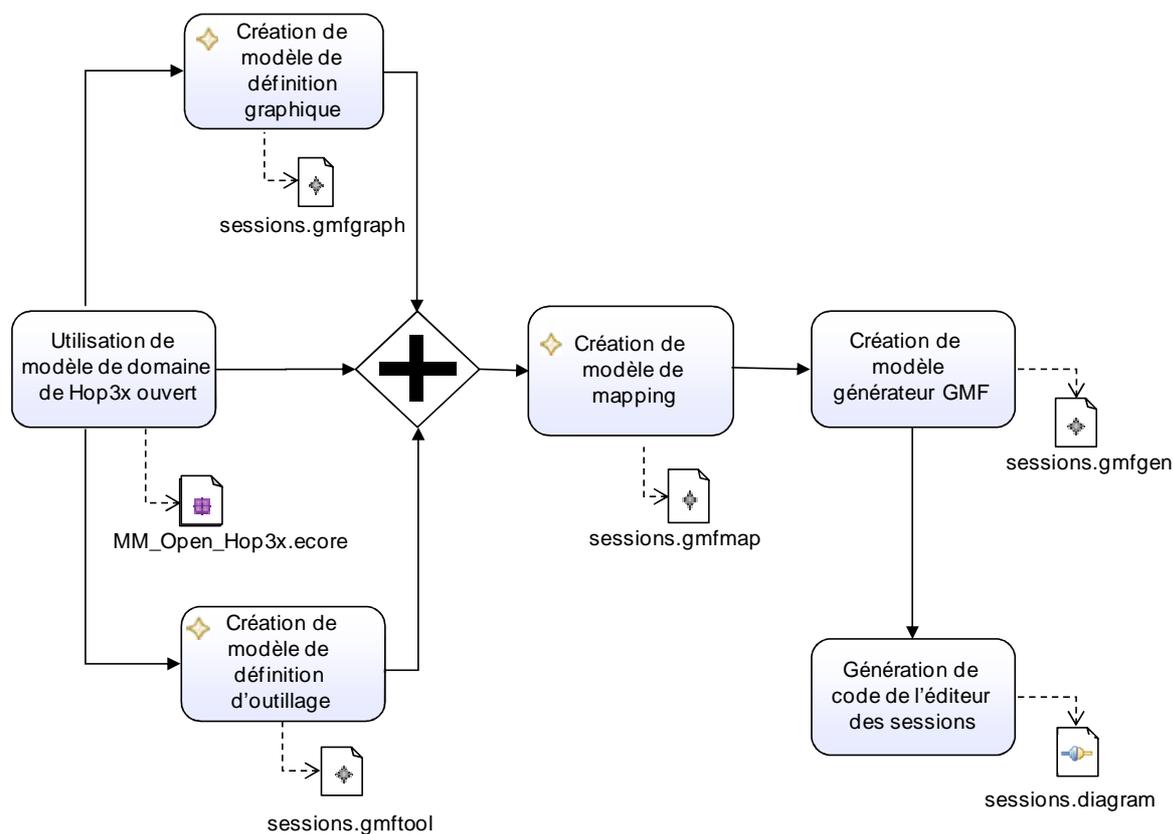


Figure 60 : Processus GMF de développement de l'éditeur des sessions.

7.2.4.1.1 Création de modèle de définition graphique : « *sessions.gmfgraph* »

Nous avons spécifié un ensemble de notations graphiques pour représenter concrètement les concepts de la syntaxe abstraite d'ADSGEML (métamodèle) qui concernent en particulier le contenu des sessions. Chacune de ces notations graphiques a été spécifiée dans le modèle « *sessions.gmfgraph* », à l'aide d'images importées ou juste en utilisant des éléments de base proposés par le Framework GMF. Ce modèle concerne la partie graphique de l'éditeur, dans la zone d'édition. Chaque objet qui apparaît dans la zone d'édition de l'éditeur : un rectangle, un cercle, un texte ou une icône ; doit être spécifié.

La figure 61 montre l'ensemble des éléments du modèle « *sessions.gmfgraph* », spécifiés pour définir la représentation graphique de chacun des concepts métier qui seront visibles dans l'espace de modélisation de l'éditeur des sessions. On distingue quatre types d'éléments : *Node* (des nœuds), *FigureDescriptor* (des figures comme ellipse, rectangle, etc., ou autre figures personnalisées), *Diagram Label* (des labels), et *Compartment* (des compartiments). En raison du nombre important d'éléments de descriptions graphiques des concepts du métamodèle de domaine, nous ne les décrivons tous pas tous en détail.

Un élément de type *Node* doit être créé pour chaque concept du métamodèle de domaine pour lequel on souhaite définir une représentation graphique. Nous nous intéressons ici particulièrement aux concepts nécessaires à la modélisation des sessions. Un *Node* est la racine de la définition graphique. Il est utilisé lors de la mise en association (*mapping*) d'un concept de domaine, tel que l'objectif d'apprentissage, avec ses caractéristiques graphiques et son outil de création.

Des éléments de type *Diagram Label* peuvent être définis pour ajouter un espace dans lequel sera affiché du texte et/ou une icône. Ces textes sont soit statiques, c'est-à-dire fournis au moment de la définition du modèle graphique, soit dynamiques, c'est-à-dire qu'ils prennent des valeurs fournies au moment de l'édition d'un élément de la session. Par exemple, l'objectif d'apprentissage de la session est représenté par un rectangle dont le titre « *Educational objective* » est souvent statique, alors que son contenu est modifiable.

Un élément de type *Figure Descriptor* doit être créé pour chaque *Node* en vue de définir les détails de la description d'une forme graphique, tels que des rectangles et des cercles, pour représenter chaque concept du métamodèle de domaine. Une forme graphique peut être décrite par plusieurs caractéristiques telles que : la position géométrique, la couleur, le titre, etc. Nous avons par exemple représenté une session par un rectangle dont le titre modifiable qui porte nom de cette session (cf. figure 62). En outre, un *Figure Descriptor* peut contenir les méthodes d'accès (cf. figure 61), équivalentes aux *Getter* de l'approche objet, permettant d'accéder à la valeur de chaque *label* ou *compartment* spécifié dans la forme graphique définie. La méthode « *getFigureSessionName_sessionFigure* » permet par exemple de récupérer la valeur du label qui définit le nom de la session, la méthode « *getFigureSessioncompartment* », permet, elle, de savoir des caractéristiques du compartiment de la session, défini par l'élément « *sessioncompartment* » (cf. figure 61).

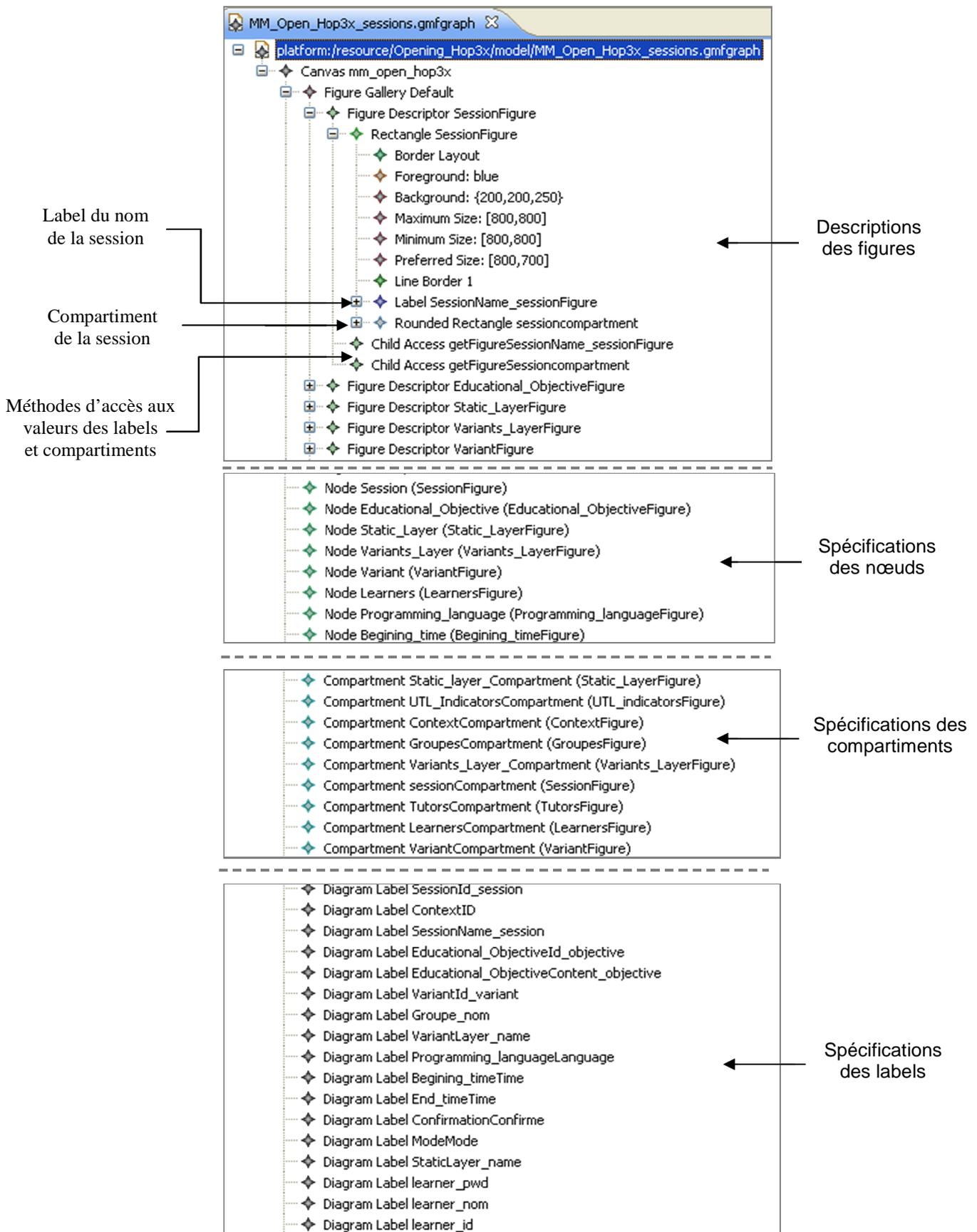


Figure 61 : Le modèle « sessions.gmfgraph » de définition graphique de l'éditeur des sessions.

Des éléments de type *compartment* consistent à définir des formes graphiques qui peuvent contenir d'autres formes des éléments qui composent la session. Le compartiment de la session est défini ici comme un rectangle avec des coins arrondis d'une couleur gris clair. Il doit pouvoir contenir les rectangles qui représentent respectivement l'objectif d'apprentissage, la couche statique, et la couche des variantes. Les rectangles de ces deux couches doivent eux aussi pouvoir contenir respectivement les formes graphiques qui représentent les concepts constituant chacune des deux couches, et ainsi de suite.

La figure 62 illustre un exemple d'une session dont le modèle de définition graphique est « *sessions.gmfgraph* ».



Figure 62 : Exemple de résultat de la définition graphique fournie par le modèle « *sessions.gmfgraph* ».

7.2.4.1.2 Création de modèle de définition de l'outillage : « *sessions.gmftool* »

La figure 63 illustre le modèle d'outillage « *sessions.gmftool* » que nous avons défini pour construire la palette d'outils de l'éditeur des sessions. Dans cette palette, nous avons catégorisé des groupes d'outils de création des éléments des sessions ouvertes, en fonction de différentes dimensions distinguées dans le métamodèle rationnel des SPO. Ceci en vue de permettre à l'enseignant concepteur de trouver aisément ces outils, et de lui assister en lui fournissant des outils de création les différents éléments constituant chaque partie de la

session ouverte en modélisation. Les outils et les groupes d'outils sont organisés selon le niveau d'imbrication. Nous avons donc créé au premier niveau, un outil pour la création des nouvelles sessions et quatre autres groupes d'outils pour créer des éléments composants les sessions. Le premier groupe rassemble les outils de création des trois dimensions principales de la session, l'objectif pédagogique, la couche statique, et la couche des variantes. Le deuxième groupe comporte les outils de création des éléments de la couche statique, le troisième, comporte les outils de création des variantes et de leurs contextes pour constituer la couche des variantes. Le dernier groupe est celui des outils de création des différents éléments qui caractérisent le contexte d'exécution de la session. Ces éléments concernent les apprenants et les groupes d'apprenants, ainsi que les indicateurs pédagogiques.

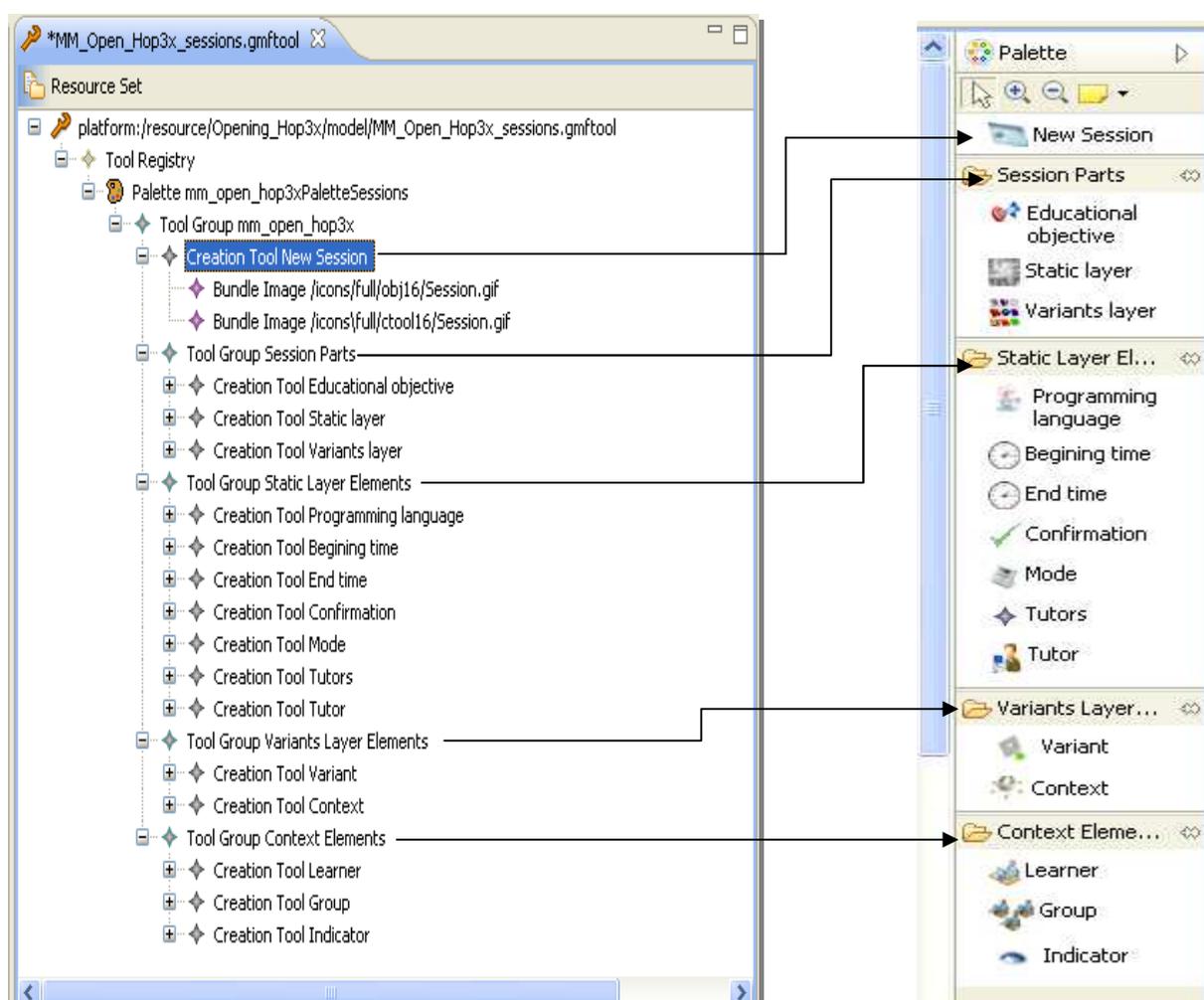


Figure 63 : Le modèle « *sessions.gmftool* » défini pour la construction de la barre d'outils de l'éditeur des sessions ouvertes de Hop3x.

7.2.4.1.3 Création de modèle de mise en association (mapping) : « sessions.gmfmap »

Bien que les éléments du modèle de définition graphique et ceux du modèle de définition d'outillage sont définis sur la base des concepts du modèle de domaine, aucune liaison n'a été créée jusque là entre les éléments de ces trois modèles. Cette étape du processus GMF consiste à créer un modèle de *mapping* pour mettre en association les concepts métiers des sessions avec leurs propres représentations graphiques et les outils de la palette définis pour leur création. Nous avons donc utilisé l'assistant de GMF pour générer une première version du modèle « sessions.gmfmap » à partir des trois autres modèles, « sessions.gmfgraph », « sessions.gmftool », et « MM_Open_Hop3x.ecore ». Ensuite, nous avons corrigé et complété le modèle généré en y ajoutant tous les éléments nécessaires.

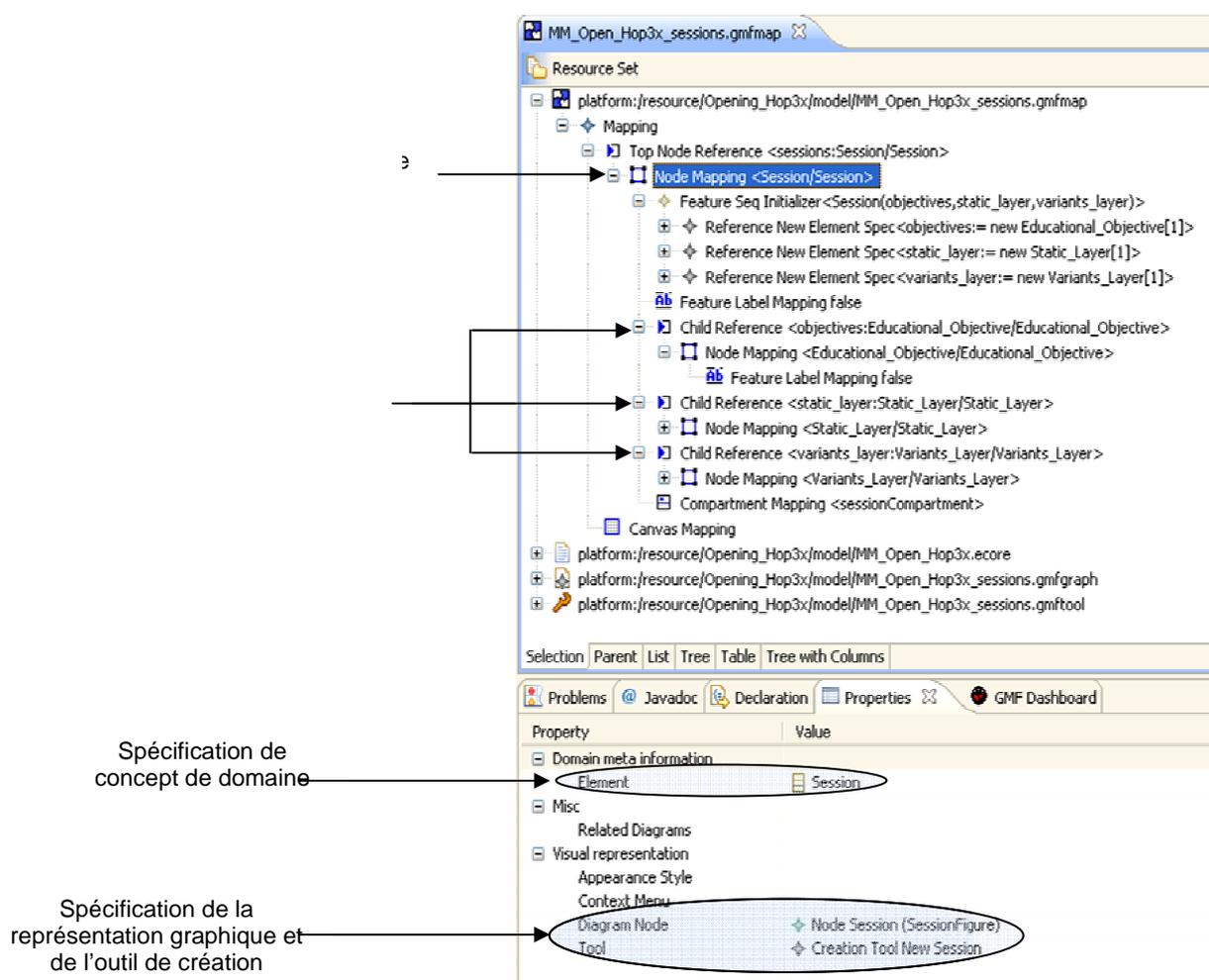


Figure 64 : Le modèle « sessions.gmfmap ».

La figure 64 présente un extrait du modèle « *sessions.gmfmap* ». Dans la vue des propriétés, chaque concept métier doit être spécifié dans la valeur de la propriété « *Element* ». Sa représentation graphique et son outil de création doivent être spécifiés respectivement dans les valeurs des propriétés « *Diagram Node* » et « *Tool* ». La vue des propriétés montre un exemple de la liaison du concept métier « *session* » avec sa propre représentation graphique (*Node Session (Session Figure)*) et son outil de création (*Creation Tool New Session*).

7.2.4.1.4 Création de modèle générateur de code : « *sessions.gmfgen* »

L'assistant de GMF permet de créer automatiquement le modèle générateur de code de plug-in d'un éditeur en se basant sur les modèles définis précédemment (*.ecore, *.gmfgraph, *.gmftool, et *.gmfmap). Nous avons utilisé cet assistant pour créer une première version du modèle « *sessions.gmfgen* », qui contient les paramètres de configuration des différents composants de l'interface graphique de l'éditeur des sessions. Ce modèle permet de générer le code Java de l'éditeur des sessions sous forme d'un plug-in Eclipse, que nous avons nommé « *Session.diagram* ». Ce plug-in se base sur les autres plug-ins générés par EMF (« *Opening_hop3x.edit* » et « *Opening_hop3x.editor* ») dans le but d'assurer la conformité des sessions (modèles) au métamodèle de domaine.

7.2.4.1.5 Génération de code du plug-in de l'éditeur graphique des sessions

L'éditeur des sessions a été construit après l'exécution du code de son plug-in (*Opening_Hop3x_Session.diagram*) généré à l'aide du modèle générateur « *sessions.gmfgen* ». Nous avons modifié certaines parties de ce code pour ajouter des menus liés à la capitalisation et le déploiement des sessions. Cet éditeur permet donc à l'enseignant concepteur de modéliser le contenu des différentes parties d'une session Hop3x ouverte en glissant les outils de la palette dans l'espace d'édition. La figure 65 est une capture d'écran de l'éditeur développé pour la modélisation graphique des sessions ouvertes de Hop3x.

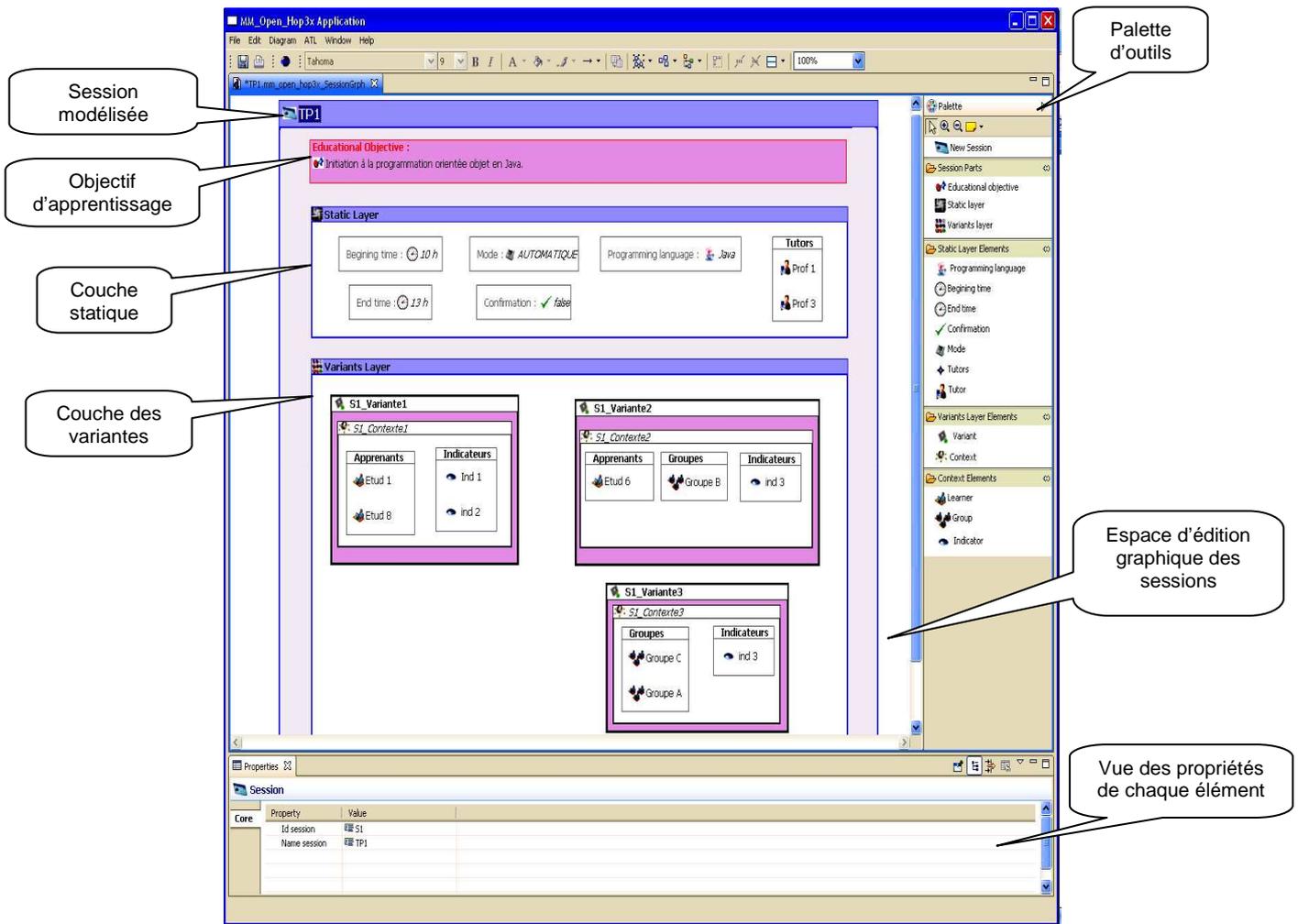


Figure 65 : Capture d'écran de l'éditeur graphique des sessions ouvertes de Hop3x.

En revanche, certains éléments de la session ne sont pas modélisés en détail dans cet éditeur, pour des raisons que nous avons déjà évoquées dans la section 7.2.4. Ces éléments concernent les informations des profils de chaque apprenant, des groupes d'apprenants, des tuteurs, ainsi que le contenu de chaque variante. D'autres éditeurs ont été développés pour définir respectivement ces éléments en détail. Leurs captures d'écran sont présentées dans la section 7.2.6.

7.2.5 Réingénierie du système Hop3x

Le système Hop3x dans sa version existante avait besoin d'un certain nombre de modifications. Lors de notre expérimentation, les entretiens ont révélé que les enseignants tuteurs souhaitaient avoir dans le système un certain nombre de nouvelles fonctionnalités qui se fondaient sur un certain nombre de nécessités¹⁹. Avec la mise en œuvre de notre environnement d'édition graphique des sessions d'apprentissage ouvertes, d'autres exigences de réingénierie se sont apparues. Afin d'y répondre, nous avons effectué un certain nombre de modifications sur le code source des différentes parties du système.

- ❖ *Au niveau de l'interface-Apprenant* : le code de cette partie du système a été amélioré pour répondre à deux besoins :
 - développer des fenêtres de notification pour prévenir l'apprenant en temps réel que sa variante a été adaptée (cf. figure 66) ;
 - déterminer les actions de l'apprenant qui provoquent le déclenchement des mises à jour de la variante en exécution. Ces actions sont définies dans l'objectif d'éviter la perturbation de l'apprenant au moment où il répond à une question. Les actions définies concernent particulièrement la consultation des consignes et le moment du passage d'une question à une autre, c'est-à-dire quand l'apprenant clique sur le bouton « suivante » ou « précédente » dans la zone d'affichage des questions.

- ❖ *Au niveau de l'interface-Tuteur* : le code de cette partie a été amélioré également pour ajouter des fonctionnalités permettant à l'enseignant tuteur de :
 - consulter l'énoncé de chaque variante et les apprenants impliqués, (cf. figure 67) ;
 - basculer d'une variante à une autre, pour pouvoir suivre simultanément tous les apprenants impliqués dans la session, bien que chacun suive sa propre variante. (cf. figure 68)

¹⁹ La majorité de ces fonctionnalités ont été intégrées dans le système existant par les concepteurs initiaux de Hop3x et d'autres chercheurs du LIUM.

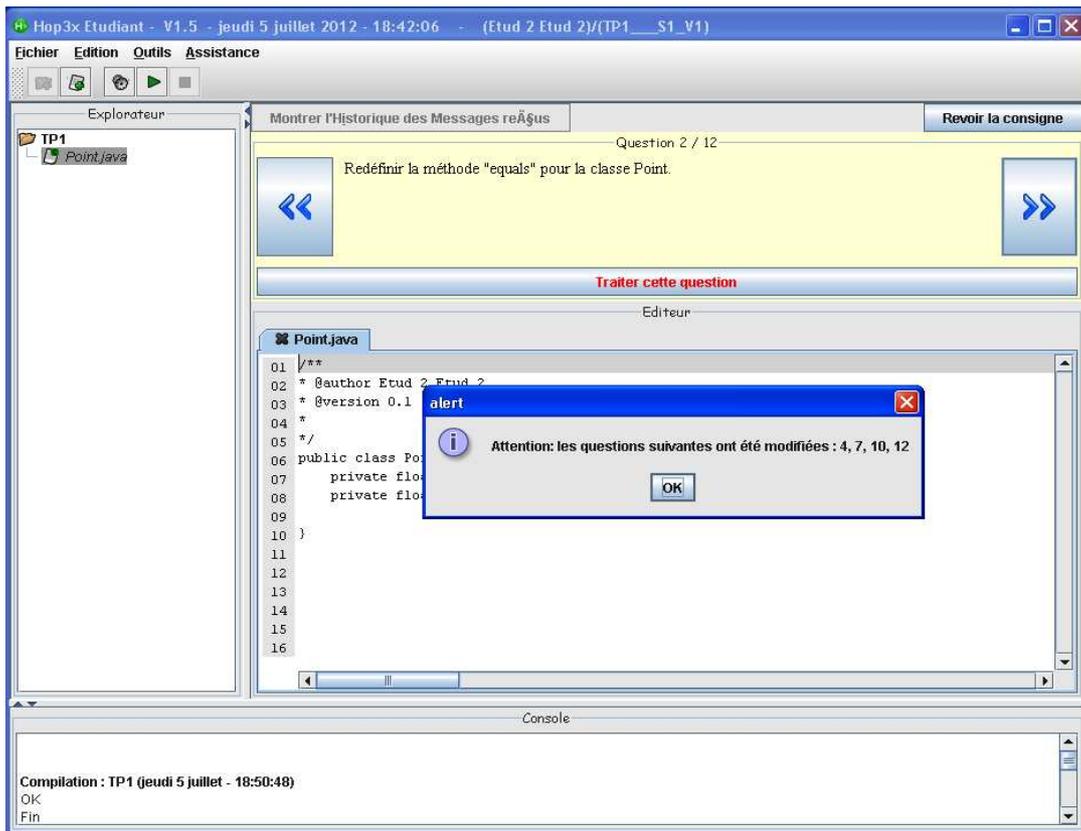


Figure 66 : Capture d'écran d'interface-apprenant illustrant la notification d'adaptation de la variante en cours d'exécution.

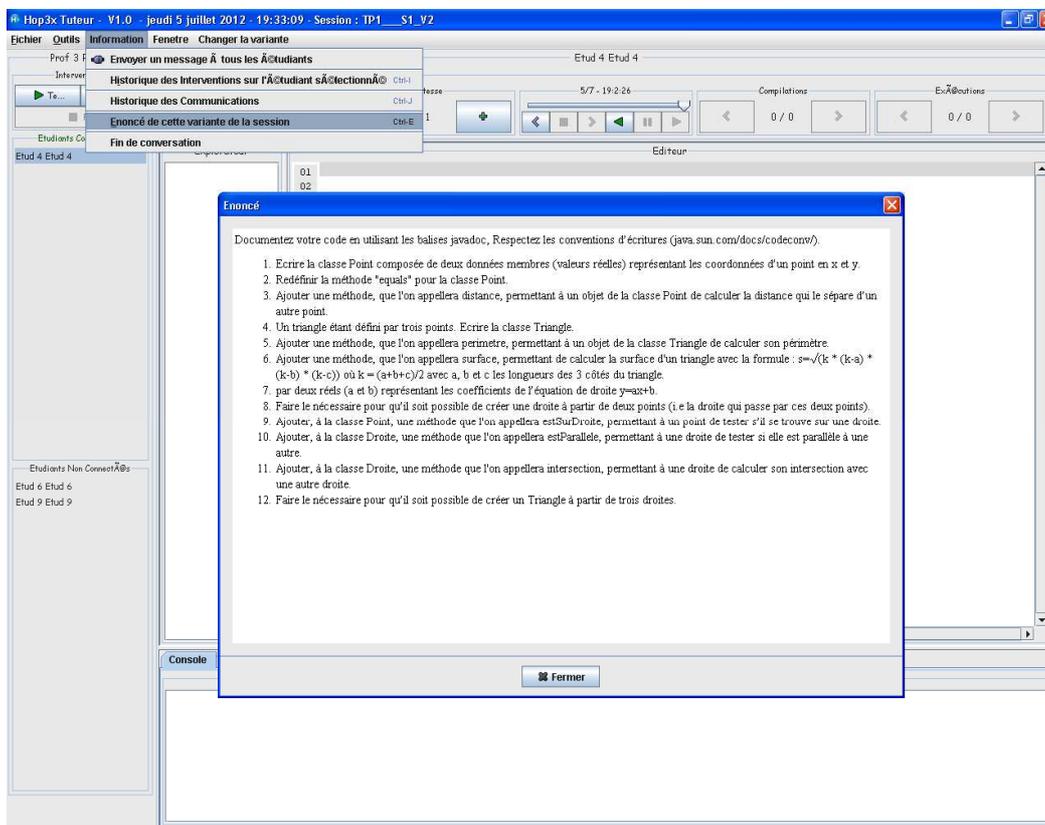


Figure 67 : Capture d'écran d'interface-tuteur montrant l'énoncé d'une variante.

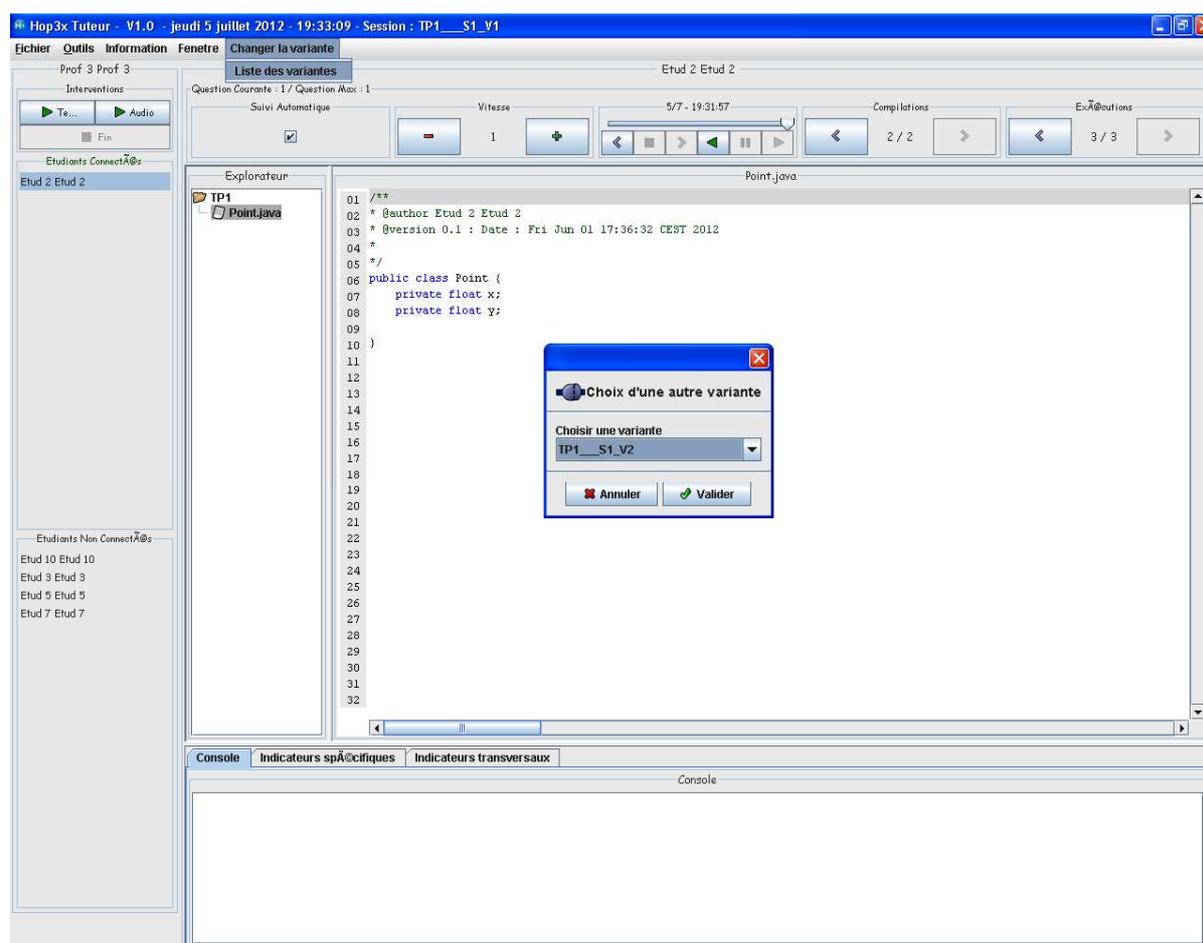


Figure 68 : Capture d'écran d'interface-tuteur montrant le basculement entre des différentes variantes.

- ❖ *Au niveau du serveur*, il nous a fallu ajouter du code pour mettre en place des mécanismes permettant :
 - le déploiement des sessions ouvertes par l'interprétation des fichiers produits en format XMI²⁰ par l'environnement d'édition graphique. Pour ce faire, nous avons défini un ensemble de règles XSLT (Annexe B) pour assurer la transformation de chacun de ces fichiers sans perte sémantique, puisque l'ADSGEML comporte déjà les concepts métier utilisés par le serveur de Hop3x ;
 - le chargement, pour chaque apprenant (ou groupe), du contenu de sa variante appropriée ;

²⁰ XMI : *XML Metadata Interchange*, est un standard créé par l'OMG pour l'échange d'informations basé sur XML.

- le chargement pour chaque tuteur les contenu des variantes de la session dans laquelle il est impliqué ;
- d'assurer l'opérationnalisation en temps réel d'adaptation des variantes que l'enseignant a réalisée dans l'environnement d'édition graphique.

La figure 69 illustre la nouvelle architecture de Hop3x, où l'enseignant dispose de l'environnement d'édition graphique pour définir les différentes parties de la session. Il a la possibilité de définir plusieurs variantes de la même session et de les adapter dynamiquement. Il peut toujours utiliser l'interface-tuteur, qui a été améliorée pour permettre la supervision des apprenants de chaque variante, et de basculer entre les différentes variantes pour pouvoir suivre tous les apprenants impliqués dans la session.

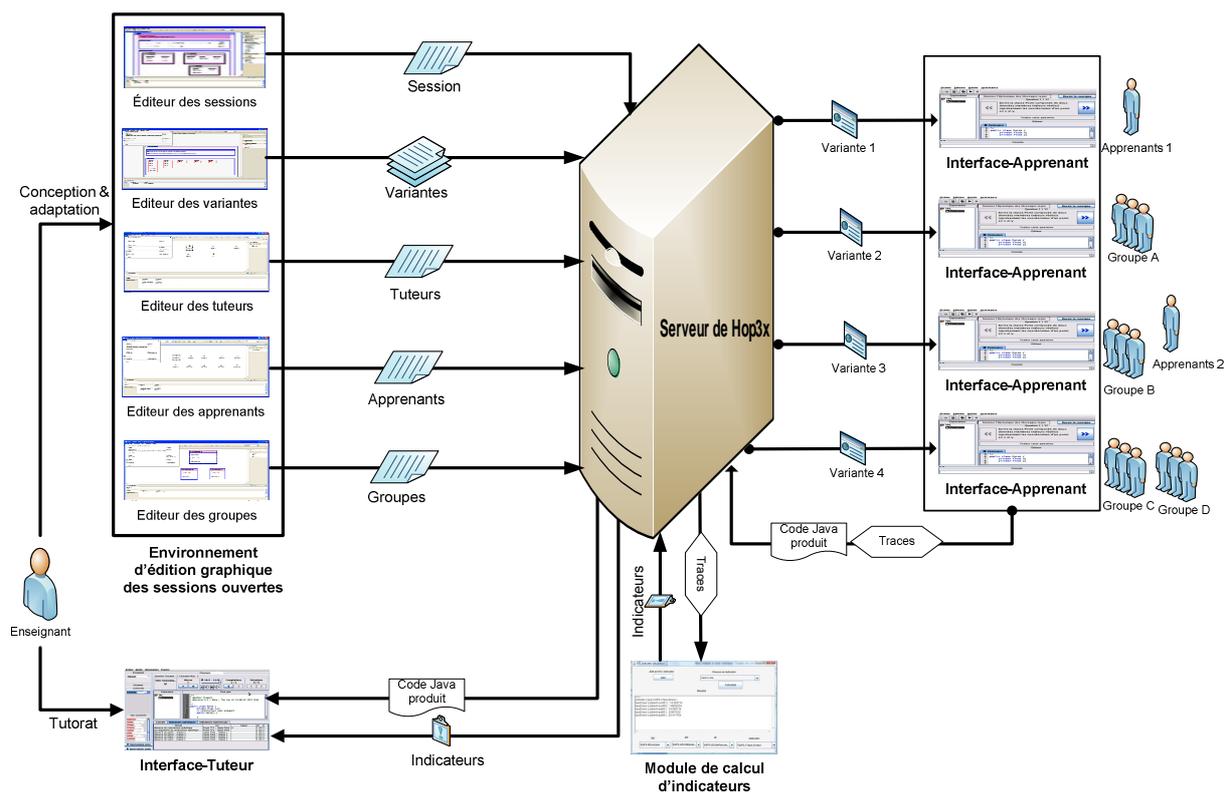


Figure 69 : Architecture de Hop3x ouvert.

7.2.6 Exemple d'utilisation de Hop3X ouvert

Nous présentons dans cette section un exemple simple d'utilisation de Hop3x ouvert. Dans sa nouvelle version, Hop3x permet désormais de continuer la conception des sessions d'apprentissage via l'adaptation pendant l'exécution. L'environnement d'édition graphique est donc mis à disposition des enseignants, pour leur permettre de concevoir et d'adapter des sessions ouvertes à un niveau abstrait.

Dans cet exemple, nous considérons une session intitulée « TP1 », qui a pour objectif de permettre aux étudiants d'apprendre des principes de base de la programmation orientée objet en Java. Cette session se compose de 12 questions (Annexe A). Elle commence à 10h et dure 3 heures, et elle implique deux tuteurs (Prof1, Prof3), dix apprenants (Etud1,...Etud10), et trois groupes (A, B, C) composés de certains de ces apprenants.

Lors de la phase de conception initiale, l'enseignant commence en premier lieu par la définition des informations de tous les acteurs. Pour ce faire, il utilise :

- l'éditeur des tuteurs, pour définir la liste des tuteurs (cf. figure 70) ;
- l'éditeur des apprenants, pour définir la liste des apprenants (cf. figure 71) ;
- et l'éditeur des groupes, pour définir la liste des groupes et les apprenants qui composent chaque groupe (cf. figure 72).

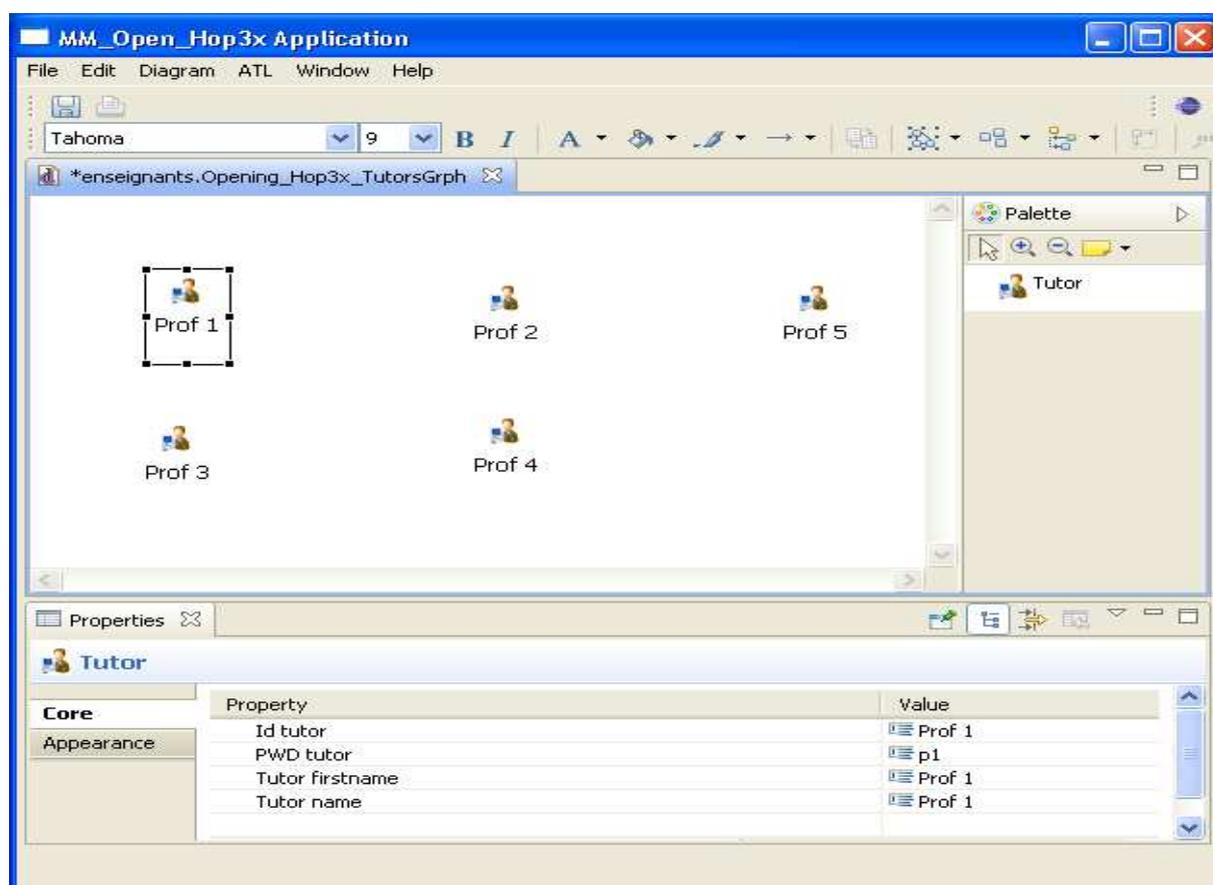


Figure 70 : Capture d'écran illustrant l'éditeur de la liste des tuteurs.

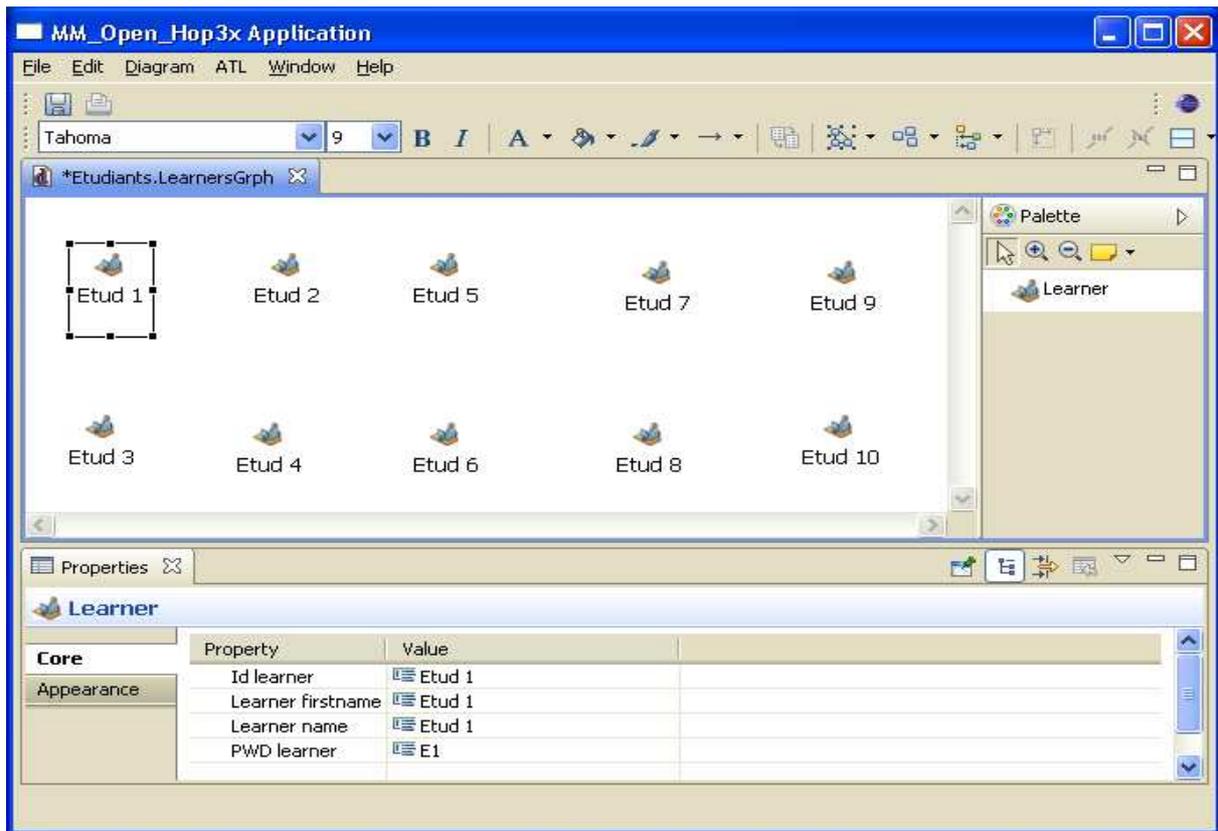


Figure 71 : Capture d'écran illustrant l'éditeur de la liste des apprenants.

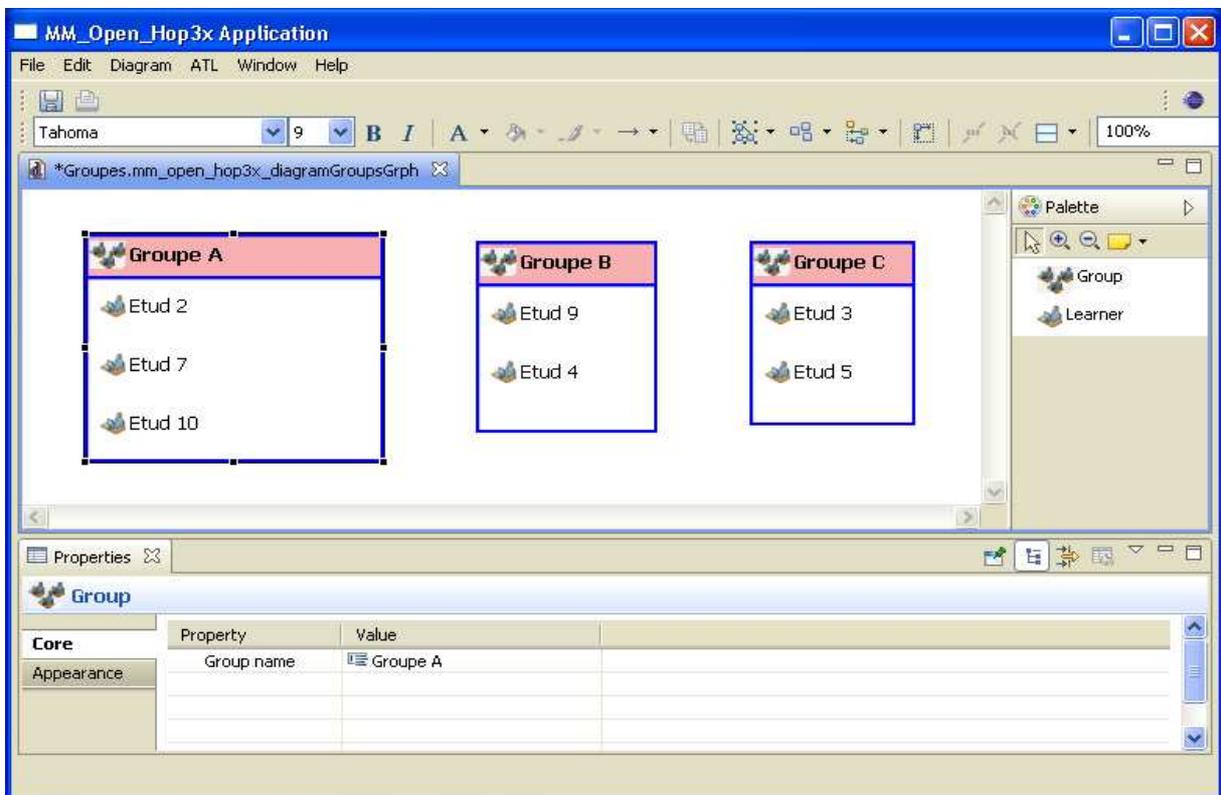


Figure 72 : Capture d'écran illustrant l'éditeur de la liste des groupes des apprenants.

Ensuite, l'enseignant utilise l'éditeur des variantes pour spécifier les différentes combinaisons des questions en fonction de différents contextes. On suppose que pendant la conception initiale, deux variantes de déroulement de cette session peuvent être anticipées suivant deux contextes prévus, qui sont définis notamment par les apprenants et/ou les groupes impliqués. Le premier contexte concerne les apprenants du niveau débutant, alors que le second concerne ceux qui ont un niveau avancé. Pour amener les apprenants à atteindre l'objectif d'apprentissage visé par cette session, la variante destinée aux débutants est assez détaillée. Elle comporte 12 questions afin d'accompagner les apprenants étape par étape, depuis la création de la première classe « *Point* » jusqu'à la création de la classe finale « *Triangle* ». Quant à la seconde variante, destinée aux apprenants du niveau avancé, elle ne propose que 3 questions principales. La figure 73 illustre le contenu de la première variante.

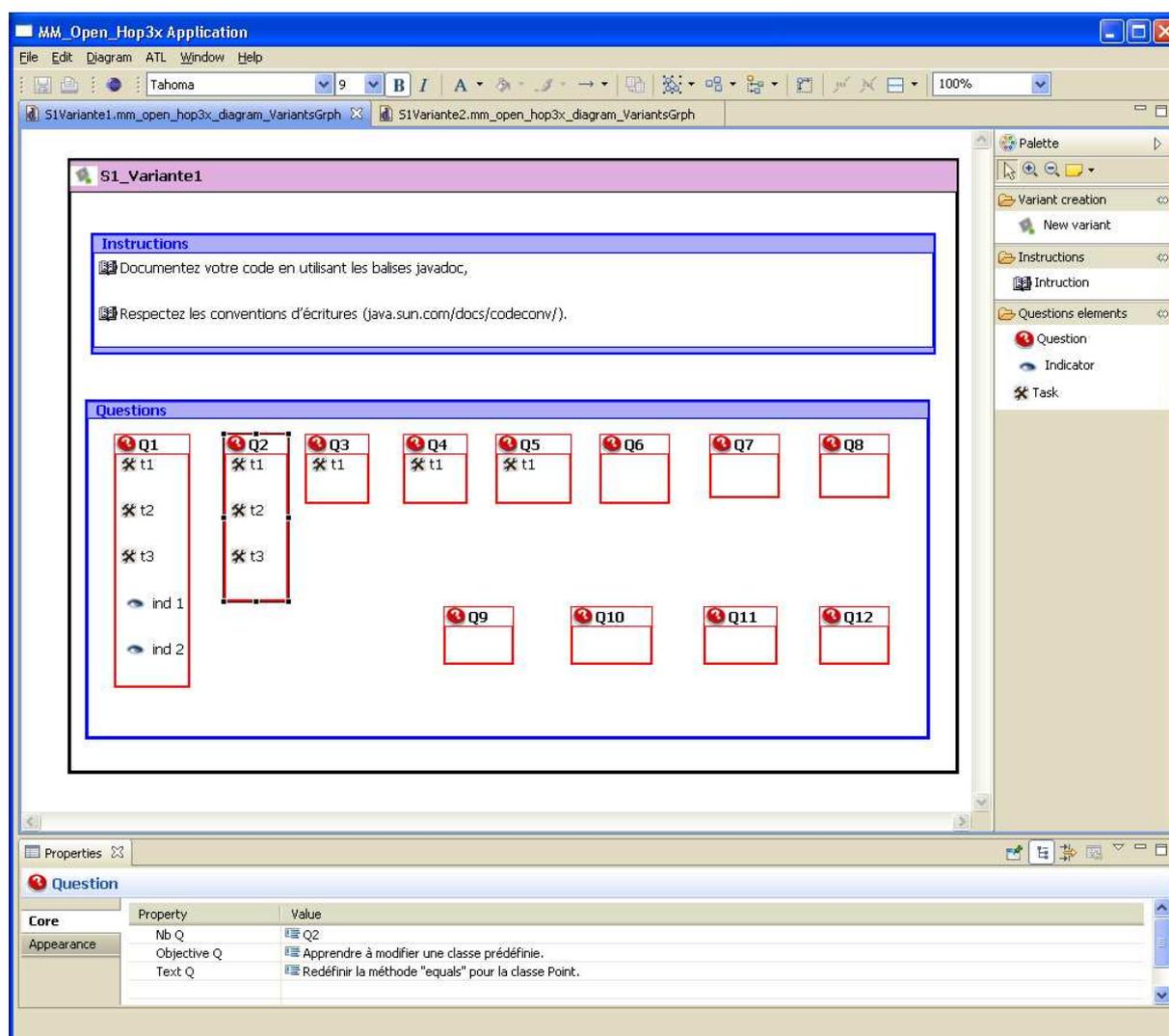


Figure 73 : Capture d'écran illustrant l'éditeur de la première variante de la session « TP1 ».

Quant à l'éditeur principal des sessions ouvertes (cf. figure 74), il permet à l'enseignant d'avoir une vision globale sur l'ensemble des éléments constituant la session. Pendant la conception initiale, cet éditeur permet de spécifier des éléments déjà connus des différentes dimensions de la session ouverte, en l'occurrence : l'objectif d'apprentissage ciblé, les éléments de la couche statique qui sont des propriétés de la session (nom, temps de début et fin, mode, etc.) et les noms des tuteurs impliqués. Dans la couche des variantes, le nom de chacune des variantes doit être identifié avec son contexte de déroulement qui est défini essentiellement par le public cible de la variante (apprenants et/ou groupes).

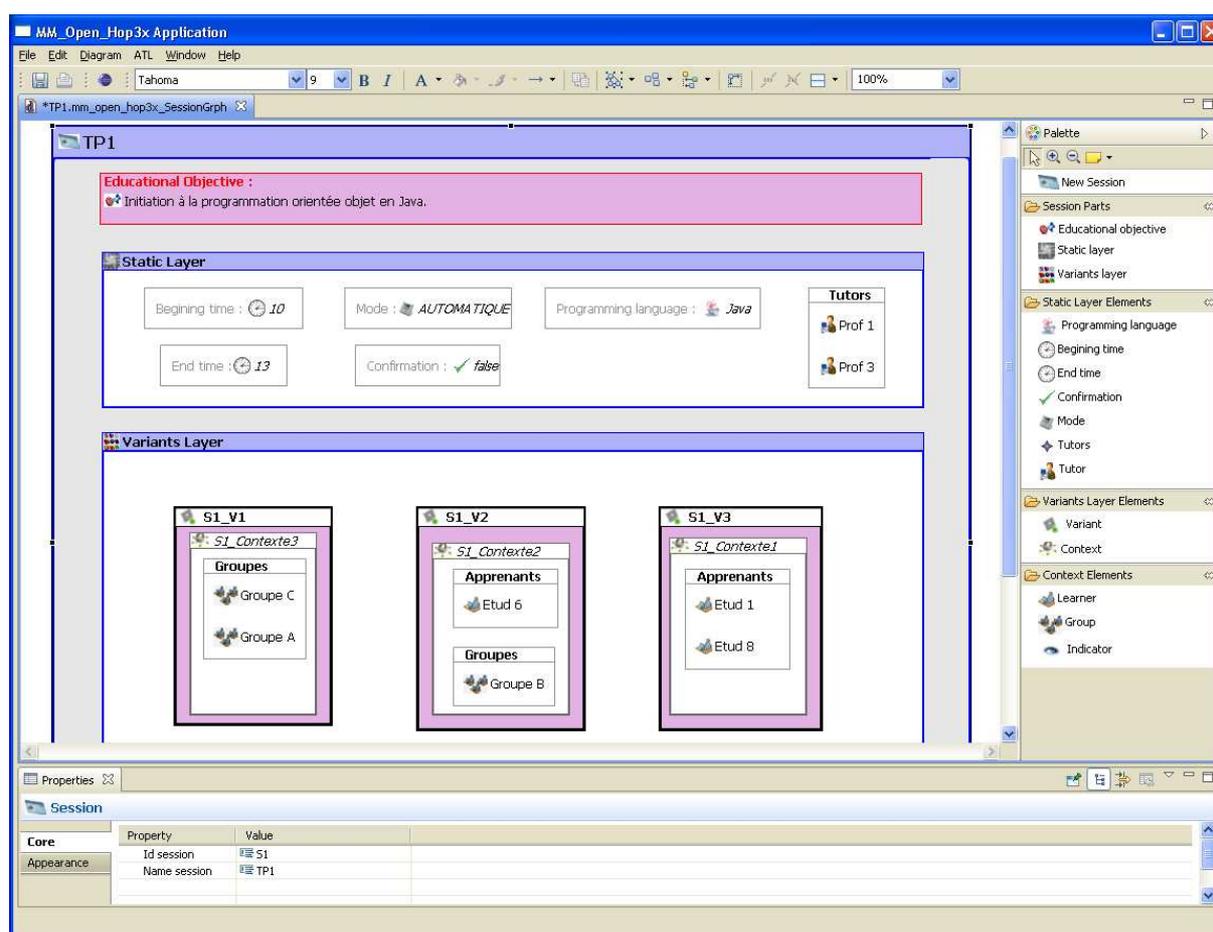


Figure 74 : Capture d'écran illustrant l'éditeur principal de la session « TP1 ».

Pendant le déploiement, la session est individualisée selon les différents contextes de déroulement caractérisés dans notre exemple par les apprenants et/ou les groupes concernés. Une fois que le serveur Hop3x a reçu la session produite par l'environnement d'édition, il identifie les différents contextes prédéfinis et attribue à chacun sa variante. Autrement dit, il charge pour les apprenants (ou les groupes) les variantes dans laquelle ils sont impliqués.

Pendant l'exécution, l'enseignant peut utiliser l'interface-tuteur, l'éditeur des sessions et l'éditeur des variantes. L'interface-tuteur permet de suivre la progression de chacun des apprenants, d'afficher l'énoncé de la variante qu'il observe (cf. figure 67) et aussi de basculer d'une variante à une autre (cf. figure 68). Cette interface permet à l'enseignant d'effectuer des adaptations temporaires (approche par déviation) à travers les interventions textuelles ou audio.

Afin d'illustrer les possibilités d'adaptation de notre session, nous supposons qu'un troisième contexte imprévu peut être identifié lors de l'exécution, celui des apprenants du niveau moyen, c'est-à-dire ceux qui peuvent répondre aux questions de la seconde variante si on détaille davantage certaines questions. L'utilisation de l'éditeur des variantes est nécessaire ici pour créer le contenu d'une nouvelle variante. Il est également possible d'adapter les contenus des deux variantes existantes. Ces adaptations peuvent être effectuées de différentes manières, par exemple : en modifiant l'énoncé des questions, en ajoutant de nouvelles questions, en fragmentant des questions existantes en plusieurs plus précises. Une fois que l'enseignant a effectué ces adaptations et les a opérationnalisés, les apprenants concernés les reçoivent en temps réel (cf. figure 66). L'enseignant peut également utiliser l'éditeur principal des sessions (cf. figure 74), pour mettre à jour les associations variantes-contextes, c'est-à-dire adapter la structure de la session. Pour attribuer les questions de la nouvelle variante, définie en temps réel, aux apprenants (ou groupes) du niveau moyen, l'éditeur des sessions permet à l'enseignant tuteur de les déplacer vers le nouveau contexte défini pour cette variante émergente.

Dans un acte de réingénierie, l'enseignant concepteur peut conserver dans le serveur de Hop3x les modifications qu'il a effectuées que ce soit dans l'éditeur principal des sessions ou dans celui des variantes. Il peut capitaliser la variante émergente et/ou les adaptations des contenus des deux variantes prescrites. Pour ce faire, il utilise le sous-menu « *Capitalize variant* » du menu « *File* » de l'éditeur des variantes. De la même façon, l'enseignant peut utiliser le sous-menu « *Capitalize session* » du menu « *File* » de l'éditeur des sessions pour capitaliser les mises à jour effectuées sur la structure globale de la session.

7.3 Apports de notre approche pour Hop3x

L'application de notre approche a permis à Hop3x de proposer une ingénierie de sessions ouvertes. Avec la nouvelle architecture (cf. figure 69), l'enseignant peut toujours utiliser l'interface-tuteur pour superviser tous les apprenants impliqués dans la session en basculant entre les différentes variantes. L'environnement d'édition graphique, associé au langage dédié au domaine de Hop3x, a favorisé :

- la prise en main grâce à l'implication des enseignants praticiens à travers les étapes de notre méthode d'instrumentation.
- Le travail à un niveau élevé d'abstraction en manipulant les concepts métiers représentés par les notations graphiques.
- La conception de sessions adaptables par la création de plusieurs variantes. La continuation de la conception des sessions pendant l'exécution grâce aux adaptations dynamiques des variantes prescrites et/ou à la création de nouvelles variantes.
- L'opérationnalisation en temps réel des adaptations effectuées dans l'environnement d'édition.
- La réingénierie des sessions par la capitalisation des adaptations dans les variantes émergentes, pour les réutiliser dans les processus ultérieurs de conception.
- L'amélioration de la version initiale du système par l'ajout de nouvelles fonctionnalités garantissant l'opérationnalisation des sessions ouvertes.

Pour distinguer les quatre approches d'adaptation des sessions de Hop3x, nous nous sommes inspirés des travaux sur la flexibilité des flux de travail [Schonenberg et al. 2007]. Les adaptations peuvent être effectuées sur les éléments des variantes, en particulier sur des questions posées aux apprenants :

- ❖ *Pendant la conception (adaptation proactive) : par la définition en amont d'ensembles différents de questions pour définir les variantes prescrites*

❖ *Pendant l'exécution (adaptation réactive) :*

- *Par sous-spécification :* en complétant la définition d'une variante en cours d'exécution de deux manières possibles : soit *par la sélection tardive* parmi un ensemble de questions prédéfinies (par exemple dans d'autres variantes), soit *par modélisation tardive* en créant à la volée des nouvelles questions pour compléter la définition de la variante.
- *par changement :* par la mise à jour de la structure et/ou des instances des éléments d'une variante en cours d'exécution.
- *par déviation :* à travers les interventions temporaires du tuteur qui peut parfois demander aux apprenants de répondre à d'autres questions non définies dans la variante déployée. Ces nouvelles questions sont éphémères, elles ne seront pas gardées après l'exécution.

8

Conclusions et perspectives

Sommaire

- 8.1 Synthèse de nos travaux
- 8.2 Analyse critique de nos travaux
- 8.3 Perspectives

8.1 Synthèse de nos travaux

Dans cette thèse, nous nous sommes intéressés à l'ouverture des EIAH, pour répondre à leur faible déploiement dans les établissements de formation, en facilitant leur appropriation par des usagers. L'objectif de notre travail était de définir un cadre d'ingénierie pour rendre un EIAH ouvert, permettant aux enseignants de poursuivre la conception d'une activité d'apprentissage dans l'usage. Nous avons donc proposé une approche d'ingénierie et de réingénierie basée sur la modélisation des scénarios pédagogiques que nous qualifions d'ouverts, pour permettre de continuer la conception via l'adaptation en fonction des évolutions du contexte d'exécution.

Notre première proposition scientifique consistait à clarifier d'abord la notion de scénario pédagogique ouvert (SPO) pour en définir ainsi un modèle générique qui formalise ce type de scénarios sans les rendre rigides et leur donne une structure simple à manipuler par des praticiens. Nous qualifions ce modèle de rationnel puisqu'il s'appuie principalement sur l'approche du Design Rationale. Il structure les SPO en deux couches : couche des éléments statiques et couche des variantes. Ce modèle permet à l'enseignant d'avoir des scénarios qui s'adaptent à la particularité de chaque situation d'apprentissage (des variantes spécifiques aux contextes d'apprentissages), et de conserver le caractère ouvert pour qu'ils puissent intégrer d'éventuelles nouvelles variantes qui émergent en temps réel, suite aux adaptations.

Après avoir rendu possible la représentation des scénarios conformément au modèle rationnel des SPO, il a fallu définir les étapes de cycle de vie d'un SPO en vue de guider l'enseignant dans la conception et l'adaptation. Nous avons donc proposé un processus itératif et incrémental d'ingénierie et de réingénierie des SPO. Ce processus se constitue de quatre phases : la conception initiale pour définir les variantes qui peuvent être anticipées selon les contextes prévus ; le déploiement pour mettre en œuvre la variante appropriée au contexte courant ; l'exécution pour dérouler la variante dans le contexte réel d'apprentissage où de nouvelles variantes imprévues peuvent émerger suite aux adaptations dynamiques ; et la réingénierie pour incorporer les variantes émergentes pour améliorer le SPO.

Afin de permettre la spécification des SPO conformément au modèle rationnel et respectant le processus itératif et incrémental d'ingénierie, notre troisième proposition consistait à fournir des instruments dédiés et conviviaux aux enseignants, pour leur faciliter la conception, l'adaptation et la gestion des SPO à un haut niveau d'abstraction. Nous avons donc proposé une méthode dirigée par les modèles et spécifique au domaine métier pour instrumenter le processus d'ingénierie et de réingénierie des SPO d'un EIAH à ouvrir.

L'objectif était de permettre aux enseignants de surmonter les difficultés sémantiques et techniques posées par des EML génériques et leurs outils. Cette méthode d'instrumentation, reposant sur l'IDM et le DSM, implique les enseignants utilisateurs de l'EIAH dans plusieurs étapes qui amènent progressivement à définir un langage dédié d'expression graphique des SPO (ADSGEML) et un éditeur associé, permettant la conception et l'adaptation des SPO dans l'univers métier de l'EIAH.

Ainsi avons nous vérifié et raffiné nos propositions théoriques en les appliquant sur l'EIAH «Hop3x», préalablement conçu au LIUM, pour pratiquer la programmation orientée objet. Notre objectif était de rendre Hop3x ouvert aux enseignants qui veulent l'utiliser, grâce à l'instrumentation du processus d'ingénierie de ses sessions d'apprentissage ouvertes. Pour ce faire, nous avons donc fait appel aux enseignants concepteurs de Hop3x dans le but d'extraire et de cristalliser les connaissances métier et d'étudier les possibilités d'ouverture et d'adaptation des sessions pour pouvoir les structurer selon le modèle rationnel. Nous avons donc élaboré un ADSGEML et un environnement d'édition graphique pour permettre aux enseignants de concevoir et d'adapter dynamiquement des sessions ouvertes de Hop3x, à un niveau élevé d'abstraction.

8.2 Analyse critique de nos travaux

8.2.1 Apports de nos propositions

L'approche proposée dans cette thèse vise l'ouverture des EIAH, en donnant la possibilité aux enseignants d'avoir la main sur le dispositif d'enseignement/apprentissage, pour pouvoir continuer la conception des SPO pendant l'usage.

La structuration des SPO selon le modèle rationnel permet d'assurer leur consistance sémantique et syntaxique, notamment au moment de l'adaptation. Ce modèle aide les enseignants à formaliser comment et pourquoi un SPO opérationnel doit être adapté. Ceci se fait grâce à l'identification des possibilités de variantes de ce SPO selon l'évolution de son contexte d'exécution. Afin de faciliter son appropriation, ce modèle propose des dimensions simples pour modéliser les aspects de la variabilité pour aider l'enseignant à expliquer et justifier les choix pédagogiques qu'il effectue. Il permet également d'établir des conditions pour la réutilisation, en conservant toutes les alternatives explorées durant le processus de conception et des décisions prises.

Le processus itératif et incrémental d'ingénierie des SPO que nous avons proposé permet d'introduire les actions d'adaptation en dynamique, lors des différentes phases du cycle de vie : conception initiale, déploiement, exécution et réingénierie. De plus, ce processus prend en compte les différentes approches d'adaptation : proactive par la conception des variantes anticipées ; réactive en temps réel, par la déviation éphémère d'instance de la variante déployée, par le changement de sa définition abstraite et/ou de son instance, par la complétude de sa définition via la liaison tardive des éléments prescrits ou via la modélisation tardive de nouveaux éléments. Ce processus d'ingénierie des SPO à base de variantes spécifiques aux contextes d'exécution permet de conserver une image de l'historique du processus continu de conception et des raisons des choix effectués, ainsi que de donner un aperçu exhaustif du SPO produit au final. Cela peut réduire les efforts pour les enseignants concepteurs, améliorer la communication de l'expertise et favoriser la réutilisation systématique des pratiques communes ou éprouvées dans un domaine particulier d'enseignement.

Quant à notre méthode d'instrumentation dirigée par les modèles et spécifique au domaine métier, elle consiste à construire des instruments dédiés au domaine éducatif des enseignants, en vue de les aider à assumer leur rôle de concepteurs et de tuteurs en continuant la conception des SPO par des adaptations dynamiques. Cette méthode prend en compte le modèle rationnel et le processus d'ingénierie des SPO. Ainsi, la construction des instruments s'effectue progressivement, de façon partagée et négociée, sans négliger la compréhension du changement des habitudes et des modalités de travail des enseignants. On peut dire que même si elle nécessite plus de temps, l'un de ses avantages est que les enseignants doivent développer une analyse réflexive sur leurs pratiques.

Par notre approche, nous voulions en effet dépasser les lacunes des travaux que nous avons identifiées dans la littérature. Nous pouvons dénombrer un ensemble d'apports dus au fait qu'elle s'appuie sur l'IDM et le DSM comme cadre théorique et technique :

- ❖ Elle permet aux enseignants de spécifier, comprendre, adapter et valider des SPO par eux-mêmes à un niveau abstrait, avec une représentation spécifique au domaine métier. De cette façon, les enseignants peuvent surmonter les barrières conceptuelles et techniques posées par les EMLs génériques et leurs outils complexes, qui sont basés sur des métaphores difficilement compréhensibles et ne prennent pas en compte des particularismes des domaines éducatifs particuliers.

- ❖ Elle assure l'opérationnalisation des SPO sans perte sémantique, alors que les approches basées sur la transformation et la fusion des modèles en souffrent ; et aussi l'adaptation en temps réel que les approches basées sur DSM n'abordaient pas jusqu'à maintenant.
- ❖ Elle consiste à spécifier la logique d'adaptation à un niveau « méta », indépendamment de l'architecture interne du système d'apprentissage, car les dimensions de variabilité définies dans le modèle rationnel des SPO sont incorporées dans le métamodèle d'ADSGEML.
- ❖ Contrairement aux éditeurs génériques qui sont inaccessibles pour les praticiens et ne répondent pas à leurs besoins spécifiques, les éditeurs dédiés permettent de travailler dans l'univers métier, ce qui favorise leur appropriation et facilite aux enseignants la continuation de la conception des SPO durant l'exécution.
- ❖ Elle propose l'ADSGEML et les SPO comme moyens intermédiaires de communication entre les développeurs et les enseignants qui sont considérés comme experts du domaine.
- ❖ Elle permet la capitalisation des connaissances (abstractions) métier des enseignants usagers de l'EIAH dans le métamodèle qui décrit l'ADSGEML.
- ❖ Elle permet d'avoir un bon niveau de productivité, de fiabilité, de maintenabilité, et de portabilité des SPO à travers l'ensemble des phases de leurs processus d'ingénierie. Il est possible de capitaliser, partager et réutiliser des décisions de conception et d'adaptation incorporées dans les variantes.

Pour conclure, l'ouverture d'un EIAH est nécessaire pour faciliter sa manipulation dans des contextes réels d'usage. Le principe est de donner la possibilité aux utilisateurs de continuer eux-mêmes la conception pendant l'usage. Notre approche d'ouverture des EIAH s'appuie sur la réingénierie, qui vise leur capacité d'être manipulés par l'utilisateur, y compris dans ce qu'il a de plus enfoui ou en tout cas de moins accessible traditionnellement. En outre, cette approche consiste à promouvoir en quelque sorte à ce que l'utilisateur soit plus autonome et ait prise sur l'EIAH, en ayant recours, au fur et à mesure, de moins en moins à l'informaticien concepteur.

8.2.2 Limites de notre travail

L'application de notre approche sur Hop3x nous a permis de mettre en œuvre nos propositions théoriques et de les vérifier dans un contexte réel, qui reste toutefois réduit. Les sessions d'apprentissage demeurent assez simples et ne permettent pas réellement de pouvoir vérifier la capacité réelle d'expressivité pédagogique du modèle rationnel proposé. Elles ne comportent pas les concepts classiques des scénarios pédagogiques, tels que : activités d'apprentissage, séquences d'activités, ressources pédagogiques, etc. La critique principale que l'on pourra formuler à l'encontre de notre travail est de ne pas valider nos propositions scientifiques à une échelle plus large et diversifiée, c'est-à-dire de les appliquer sur des EIAH de domaines pédagogiques différents.

En effet, nous avons proposé par ce travail une approche nouvelle, mais qu'il faudrait qu'elle soit largement essayée pour vraiment démontrer sa valeur. Nous avons démontré la faisabilité de la mettre en œuvre, nous avons des éléments pour dire que l'acceptabilité est meilleure, mais nous devons encore faire des tests à grande échelle pour évaluer l'utilisabilité et l'efficacité.

8.3 Perspectives

L'une de nos perspectives à court terme est d'appliquer notre approche sur d'autres EIAH dans des domaines de formation variés. La mise en œuvre de nos propositions théoriques dans d'autres contextes pourra permettre de vérifier et valider la capacité du modèle rationnel à représenter les scénarios ouverts de différents domaines métier, et aussi de vérifier l'acceptabilité de ce modèle par des praticiens et l'utilisabilité des instruments proposés. Nous pensons qu'une mise en œuvre de notre approche sur un cas des « *Serious Games* » serait pertinente, étant donné le caractère ouvert et fluctuant des scénarios des jeux.

Une seconde perspective concerne l'amélioration de notre approche pour assurer un maintien continu d'ouverture de l'EIAH, afin de prendre en compte de nouveaux besoins et exigences qui apparaissent, dans le temps, suite aux changements de pratiques des enseignants. L'EIAH doit être capable d'accompagner ces changements par l'évolution de l'ADSGEML, par l'intégration des nouveaux concepts métier dans le métamodèle, ainsi que la construction d'une nouvelle version d'éditeur dédié qui contient les nouveaux concepts, cela afin d'assurer en permanence une meilleure expressivité pédagogique.

Nous envisageons dans le futur de creuser la piste d'explicitation et de formalisation des stratégies d'adaptation utilisées implicitement par l'enseignant. Le but de cette formalisation est de pouvoir capitaliser ces stratégies et les réutiliser à un niveau métier compréhensible. Une fois répertoriées avec les critères pertinents pour leur application, le système peut les déclencher automatiquement pour adapter une variante si leurs critères sont accomplis. Cela permet en effet d'alléger le travail de l'enseignant en partageant les tâches d'adaptation entre lui (adaptabilité) et le système (adaptivité).

Enfin, on peut envisager également une proposition pour optimiser le niveau de granularité de la variabilité. Notre approche propose en effet une solution de macro-variabilité, définissant un ensemble de variantes pour dérouler le même scénario pédagogique de différentes manières. Une autre approche plus fine est possible, consistant à accentuer le niveau de détails en termes de variabilité, en permettant la micro-variabilité par la définition des variantes pour chacun des éléments du scénario, par exemple : spécifier plusieurs alternatives d'une même activité d'apprentissage ou d'une ressource à utiliser, etc. Une redéfinition du modèle des SPO est donc nécessaire pour prendre en compte les particularités de cette approche par micro-variabilité. La solution envisageable est de spécifier dans le métamodèle d'ADSGEML les points de variation qui doivent être précisés par l'expert de domaine.

Table des figures

| | |
|--|----|
| Figure 1 : Les approches interprétative et constructive de la conception pédagogique [El-Kechaï 2008]. | 42 |
| Figure 2 : Approche interprétative de la conception d'un scénario pédagogique [Choquet 2007]. | 43 |
| Figure 3 : Approche constructive de la conception d'un scénario pédagogique [Choquet 2007]. | 44 |
| Figure 4 : Le spectre de la flexibilité d'un processus [Schonenberg et al. 2008]. | 48 |
| Figure 5 : Solutions envisagées pour un problème donné [Lacaze 2005]. | 53 |
| Figure 6 : La notation du modèle QOC (<i>Questions, Options, Criteria</i>). | 58 |
| Figure 7 : Modèle conceptuel d'IMS-LD [Ferraris et Lejeune 2007]. | 59 |
| Figure 8 : Liaison d'adaptation en conception et l'adaptation en exécution [Van Rosmalen 2008]. | 62 |
| Figure 9 : Phases de conception itérative d'une unité d'apprentissage [Zarraonandia et al. 2007]. | 68 |
| Figure 10 : La structure de LD Player [Zarraonandia et al. 2007]. | 69 |
| Figure 11 : Le métamodèle du langage LDL [Martel et al. 2007]. | 70 |
| Figure 12 : Élément de scénario pédagogique et ses principaux composants [Caeiro-Rodriguez et al. 2007]. | 71 |
| Figure 13 : Architecture à quatre niveaux [Bezivin et Blanc 2002]. | 74 |
| Figure 14 : Cycle de transformations et EIAH [Nodenot et al. 2005]. | 76 |
| Figure 15 : Le DSM élève le niveau d'abstraction par rapport aux autres approches. | 77 |
| Figure 16 : Exemple de modélisation des produits financiers et d'assurance pour une application web J2EE avec un langage DSM basé sur des concepts d'experts de domaine [Tolvanen 2006]. | 78 |
| Figure 17 : Exemples des syntaxes concrètes : (a) par UML, (b) par arborescence et (c) textuelle (XML) [De Moura Filho 2007]. | 80 |
| Figure 18 : Vue d'Ecore simplifiée en quatre classes [Vanworkmhoudt 2004]. | 82 |
| Figure 19 : Processus d'EMF pour développer un éditeur à partir d'un métamodèle du domaine. | 83 |
| Figure 20 : Processus GMF pour le développement des éditeurs graphiques. | 85 |
| Figure 21 : Paquetage du métamodèle CPM [Laforcade 2004]. | 86 |
| Figure 22 : Profil CPM (extraits) [Laforcade, 2004]. | 87 |
| Figure 23 : Approche de conception du projet Bricoles [Caron 2007b]. | 90 |

| | |
|---|-----|
| Figure 24 : Transformation du scénario PPC [Abdallah 2009]. | 91 |
| Figure 25 : Vue de modélisation de transformation de l'environnement MDLD [Dodero et al. 2008]. | 94 |
| Figure 26 : Construction de scénarios à base de patrons pédagogiques [De Moura Filho 2007]. | 96 |
| Figure 27 : Modélisation sur la base du contenu de l'OI [El-Kechaï 2008]. | 97 |
| Figure 28 : Notre méthodologie de recherche. | 114 |
| Figure 29 : Modèle rationnel des SPO [Ouraiba et al. 2011b]. | 127 |
| Figure 30 : Exemple du résultat de conception d'un SPO. | 129 |
| Figure 31 : Le métamodèle qui implémente le modèle rationnel des SPO. | 131 |
| Figure 32 : Processus itératif et incrémental d'ingénierie d'un SPO. | 137 |
| Figure 33 : Diagramme de paquetages UML représentant le processus itératif et incrémental d'ingénierie d'un SPO. | 141 |
| Figure 34 : Vue en niveaux d'OMG de l'ingénierie des scénarios pédagogiques ouverts [Ouraiba et al. 2011b]. | 148 |
| Figure 35 : Collaboration entre l'enseignant concepteur et l'informaticien. | 151 |
| Figure 36 : Méthode d'instrumentation pour l'ouverture des scénarios pédagogiques d'un EIAH. | 153 |
| Figure 37 : Méthodes pour collecter les données du domaine métier. | 155 |
| Figure 38 : Définition de métamodèle des scénarios pédagogiques existants. | 156 |
| Figure 39 : Processus de développement d'éditeur à partir de métamodèle métier. | 156 |
| Figure 40 : Projection des concepts de DSEML, scénario pédagogique et contexte sur les niveaux d'OMG. | 157 |
| Figure 41 : Evolution de métamodèle initial après négociation de l'informaticien avec les enseignants. | 158 |
| Figure 42 : Projection des concepts d'ADSGEML, SPO, variante et contexte sur les niveaux d'OMG. | 159 |
| Figure 43 : Processus dirigé par les modèles pour le développement d'un éditeur graphique des SPO. | 160 |
| Figure 44 : Architecture du système Hop3x. | 168 |
| Figure 45 : L'interface-Apprenant de Hop3x. | 169 |
| Figure 46 : L'interface-Tuteur de Hop3x. | 170 |
| Figure 47 : Relation entre les différents fichiers XML constituant une session Hop3x. | 172 |
| Figure 48 : Extrait du fichier XML des sessions dans Hop3x. | 173 |
| Figure 49 : Extrait du fichier XML des groupes dans Hop3x. | 173 |
| Figure 50 : Extrait du fichier XML des apprenants dans Hop3x. | 174 |

| | |
|---|-----|
| Figure 51 : Extrait du fichier XML des tuteurs dans Hop3x..... | 175 |
| Figure 52 : Extrait du fichier XML d'un scénario de TP dans Hop3x. | 175 |
| Figure 53 : Processus d'élaboration de DSEML de Hop3x et d'un éditeur spécifique..... | 183 |
| Figure 54 : Métamodèle des sessions Hop3x non-ouvertes..... | 183 |
| Figure 55 : Editeur basique généré par l'outillage d'EMF..... | 184 |
| Figure 56 : Processus d'évolution du métamodèle initial en métamodèle des sessions ouvertes..... | 185 |
| Figure 57 : Le métamodèle évolué qui décrit l'ADSGEML de Hop3x. | 187 |
| Figure 58 : Les processus dirigés par les modèles pour le développement d'un environnement d'édition graphique dédié à Hop3x. | 190 |
| Figure 59 : Capture d'écran illustrant l'ensemble des modèles GMF définis pour le développement des différents éditeurs de Hop3x..... | 192 |
| Figure 60 : Processus GMF de développement de l'éditeur des sessions. | 193 |
| Figure 61 : Le modèle « <i>sessions.gmfgraph</i> » de définition graphique de l'éditeur des sessions..... | 195 |
| Figure 62 : Exemple de résultat de la définition graphique fournie par le modèle « <i>sessions.gmfgraph</i> ». | 196 |
| Figure 63 : Le modèle « <i>sessions.gmftool</i> » défini pour la construction de la barre d'outils de l'éditeur des sessions ouvertes de Hop3x. | 197 |
| Figure 64 : Le modèle « <i>sessions.gmfmap</i> ». | 198 |
| Figure 65 : Capture d'écran de l'éditeur graphique des sessions ouvertes de Hop3x. | 200 |
| Figure 66 : Capture d'écran d'interface-apprenant illustrant la notification d'adaptation de la variante en cours d'exécution. | 202 |
| Figure 67 : Capture d'écran d'interface-tuteur montrant l'énoncé d'une variante..... | 202 |
| Figure 68 : Capture d'écran d'interface-tuteur montrant le basculement entre des différentes variantes..... | 203 |
| Figure 69 : Architecture de Hop3x ouvert..... | 204 |
| Figure 70 : Capture d'écran illustrant l'éditeur de la liste des tuteurs..... | 205 |
| Figure 71 : Capture d'écran illustrant l'éditeur de la liste des apprenants. | 206 |
| Figure 72 : Capture d'écran illustrant l'éditeur de la liste des groupes des apprenants..... | 206 |
| Figure 73 : Capture d'écran illustrant l'éditeur de la première variante de la session « TP1 ». | 207 |
| Figure 74 : Capture d'écran illustrant l'éditeur principal de la session « TP1 ». | 208 |

Références bibliographiques

[Abdallah 2009] Abdallah, F. (2009) Méta-modélisation pour décrire et instrumenter une situation d'apprentissage de Pédagogie par Projet Collectif. Doctorat informatique de l'université du Maine.

[Ayersman et Minden 1995] Ayersman, D., et von Minden, A. (1995). Individual differences, computers, and instruction. *Computers in Human Behavior*, 11(3-4), 371-390.

[Ahmad et al. 2004] Ahmad, A., Basir, O., et Hassanein, K. (2004). Adaptive user interfaces for Intelligent e-Learning: issues and trends. *Proceedings of the Fourth International Conference on Electronic Business, ICEB2004, Beijing*.

[Alfanet-Tool 2005] Site de « The Alfanet Authoring Tool », accessible à : <http://sourceforge.net/projects/alfanetat/>, consulté le 10/06/2009.

[Alonso 2005] Alonso, M. (2005). Modélisation, adaptation et opérationnalisation de scénarios d'apprentissage. Rapport de Master 2eme année, Université Joseph Fourier.

[AI-Watch 1995] AI-Watch (1995). Special issue on the EuroKnowledge project, volume 4(8).

[Ainsworth et al. 2003] Ainsworth, S., Major, N., Grimshaw, S., Hayes, M., Underwood, J., Williams, B. et Wood, D. (2003). REDEEM: Simple Intelligent Tutoring Systems from Usable Tools. Chapter 8 in Murray, T., Blessing, S. & Ainsworth, S. (Eds.). *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publishers, Dordrecht.

[Bannon et Bodker 1991] Bannon, L.J., Bodker, S. (1991). Beyond the interface: encountering artifacts in use, in *Designing Interaction. Psychology of Human Computer Interface*, Carroll J. M. ed. Cambridge University Press.

[Bourguin et Derycke 2005] Bourguin, G. et Derycke, A. (2005). Systèmes Interactifs en Co-évolution Réflexions sur les apports de la Théorie de l'Activité au support des Pratiques Collectives Distribuées. *Revue d'Interaction Homme-Machine* Vol 6 N°1.

[Bowker, Star et Turner 1997] Bowkers, G., Leigh Star, S., Turner, W., Gasser, L. (1997). Social science, technical systems and cooperative work: beyond the great divide. Lawrence Erlbaum Associates, "Computer, cognition and work" series, 470 p.

[Bourmaud 2007] Bourmaud, G. (2007). L'organisation systémique des instruments : Méthodes d'analyse, propriétés et perspectives de conception ouvertes. Acta-Cognitica, ARCo'07 – Cognition, Complexité, Collectif. pp. 61-75

[Bonnardel et Rech 1998] Bonnardel, N., et Rech, M. (1998). Les objets – sources d'inspiration dans les activités de conception, chapitre in : Les objets dans la conception, edited by trousse and Zreik édition.

[Bérubé 1996] Bérubé B. (1996). L'intégration des TIC dans les pratiques pédagogiques au collégial. Site web : <http://www.usherb.ca/performa/tic>

[Boticario et Santos 2007] Boticario, J. G., et Santos, O. C. (2007). An open IMS-based user modelling approach for developing adaptive learning management systems. Journal of Interactive Media in Education (Adaptation and IMS Learning Design. Special Issue, ed. Daniel Burgos), 2007/2, from <http://jime.open.ac.uk/> .

[Burgos et Ruiz-Mezcua 2003] Burgos, D., and Ruiz-Mezcua, B. (2003). Building an interactive training methodology to develop multimedia elearning software. Proceedings of International Conference On Education, Information Systems, Technologies and Applications EISTA 2003, Orlando, Florida, USA

[Burgos 2008] Burgos, D. (2008). Extension of the IMS Learning Design Specification based on Adaptation and Integration of Units of Learning. Doctoral Thesis, Carlos III University of Madrid, Leganés, Spain.

[Burgos, Tattersall et Koper 2006] Burgos D., Tattersall C., and Koper R. (2006). Representing adaptive eLearning strategies in IMS Learning Design. Proceeding TEN Competence Conference. Sofia, Bulgaria, March 31st 2006.

[Brusilovsky 1996] Brusilovsky, P. (1996), Methods and techniques of adaptive hypermedia. User Models and User Adapted Interaction.

[Brusilovsky 2001] Brusilovsky, P. (2001). Adaptive Hypermedia. User Modelling and User-Adapted Interaction, Kluwer Academic Publishers, the Netherlands, 11, pp 87-110.

[Brusilovsky et Miller 2001] Brusilovsky, P., and Miller, P. (2001): Course Delivery Systems for the Virtual University. In Tschang F.T. & T. Della Senta (Eds.), Access to Knowledge: New Information Technologies and the Emergence of the Virtual University (pp. 167-206). Amsterdam: Elsevier Science and International Association of Universities.

[Brusilovsky et Paylo 2003] Brusilovsky, P. and Paylo, C. (2003): Adaptive and Intelligent Web-based Educational Systems. *International Journal of Artificial Intelligence in Education*, 13, 156-169, 2003.

[Butz et al. 2003] Butz, M. V., Olivier, S., et Gérard, P. (2003). Anticipatory Behavior in Adaptive Learning Systems: Foundations, Theories, and Systems. In Berlin: Springer Verlag.

[Bézivin et al. 2005] Bézivin J., Hillairet G., Jouault F., Kurtev I. Et Piers W. (2005b). Bridging the ms/dsl tools and the eclipse modeling framework. In *Proceedings of the International Workshop on Software Factories at OOPSLA*.

[Berlanga et Garcia 2005a] Berlanga, A.J., García Peñalvo, F.J. (2005). Using IMS LD for Characterizing Techniques and Rules in Adaptive Educational Hypermedia Systems. In R. Koper, C. Tattersall, D. Burgos (Eds), *Proceedings of Current research on IMS Learning Design UNFOLD/Prolearn Workshop* (Valkenburg, The Netherlands, September, 22-23).

[Berlanga et Garcia, 2005b] Berlanga, A. and Garcia, F. (2005). IMS LD reusable elements for adaptive learning designs. *Journal of Interactive Media in Educatio* 2005(11). ISSN: 1365-893X.

[Berlanga 2006] Berlanga, A. (2006). Adaptive Learning Designs: Personalized and Reusable Instruction in Learning Environments. Ph.D. thesis in Computer Science Department of Computer Science, University of Salamanca, Spain.

[Berlanga 2007] Berlanga, A.J., García Peñalvo, F.J. (2008). Learning Design in Adaptive Educational Hypermedia Systems. In *Journal of Universal Computer Science*, vol. 14, no. 22, p3627-3647.

[Baker 2007] Baker, M. (2007). Rôle et usage des modèles en Environnements Informatiques pour l'Apprentissage Humain. 5ème école thématique du CNRS sur les EIAH - Personnalisation des EIAH , 7-12 juillet 2007 - Saint Quentin sur Isère, France

[Burge et Brown 1998] Burge J. and Brown D.C. (1998). Design Rationale: Types and Tools, Technical Report, Worcester Polytechnic Institute, Computer Science Dept.

[Bezivin et Blanc 2002] Bezivin, J. et Blanc, X. (2002). MDA : Vers un important changement de paradigme en genie logiciel. Développeur référence, en ligne : <http://www.devreference.net/>

[Bhanot et al. 2005] Bhanot V., Paniscotti D., Roman A., Trask B. (2005). Using Domain-Specific Modeling to Develop Software Defined Radio Components and Applications, In: the 5th OOPSLA Workshop on Domain-Specific Modeling (DSM'05), San Diego, California, USA, accessible à: <http://www.dsmforum.org/events/DSM05/Papers.html>, dernière consultation Juin 2012.

[Béguin 2005] Béguin P. (2005). Concevoir pour les genèses professionnelles. In : P. Rabardel et P. Pastré (ed.) Modèles du sujet pour la conception. Dialectiques activités développement, Octarès, Toulouse, 31-52, 2005.

[Cottier et Choquet 2005] Cottier, P. et Choquet, C. (2005). De l'usager construit à l'usager participant. Actes EIAH 2005, Montpellier, INRP, Paris, p.449-454, 25-27 mai 2005.

[Cottier 2008] Cottier P. (2008), LÉA : Expérimentation de la conception d'un EIAH ouvert. Bilan et résultats, Université du Maine, Compte rendu de fin de recherche. Novembre 2008. Aide n°04 L 402. (460 pages).

[Cottier et al. 2008] Cottier P., Choquet C. et Tchounikine P. (2008). Repenser l'ingénierie des EIAH pour des enseignants concepteurs. In: Usages, usagers et compétences informationnelles au XXIème siècle, Jérôme Dinet (ed.), p. 159-193, Édité par Hermes Lavoisier, ISBN 978-2-7462-2193-2.

[Cottier et El-Kechaï 2009] Cottier, P. et EL-Kachai, H. (2009). L'usager concepteur en situation: Conception collective d'un livret électronique d'apprentissage (LÉA).In: Ingénierie des systèmes d'information, ISSN1633-1311.vol.14, no3.p162

[Cottier 2010] Cottier, P. (2010). L'expérience LÉA : vers une autre relation entre conception et pratiques. Dans les actes de la journée scientifique « Conception des EIAH à l'ère du Web 2.0 et à l'aube du 3.0 », 2 Juillet 2010, Amiens

[Choquet 2007] Choquet, C. (2007). Ingénierie et réingénierie des EIAH – L'approche REDiM. Habilitation à diriger les recherches en informatique, Université du Maine.

[Chikofsky et Cross 1990] Chikofsky J.-E. and Cross J.-H. (1990). Reverse engineering and design recovery: taxonomy. Software IEEE, 7: 13–17.

[Corbière 2006] Corbière A. (2006). Analyses des apports du méta-standard, ODP-RM à la communauté EIAH, Instances sur un système de formation. Doctorat, Université du Maine.

[Choquet et Iksal 2007] Choquet C. et Iksal S. (2007). Modélisation et construction de traces d'utilisation d'une activité d'apprentissage : une approche langage pour la réingénierie d'un EIAH. *Revue STICEF*, Vol. 14.

[Cristea 2003] Cristea, A. (2003). Adaptive Patterns in Authoring of Educational Adaptive Hypermedia. *Educational Technology & Society*, 6 (4), 1-5.

[Chen et Magoulas 2005] Chen, S. Y., et Magoulas, G. D. (2005). *Adaptable and Adaptive Hypermedia Systems*. Hershey, PA: IRM Press.

[Chin 2001] Chin D. (2001). Empirical Evaluation of User Models and User-Adapted Systems. *User Modeling and User-Adapted Interaction*, 11, 181-194.

[Crozat 2002] Crozat S. Trigano P. (2002). Structuration et scénarisation de documents pédagogiques numériques dans une logique de massification, *STE (Sciences et Techniques Educatives)*, vol.9, N°3, Ed°Hermès.

[Charlier et Daele 2002] Charlier B. et Daele A. (2002). *Recre@sup: Réseau des centres de ressources pour l'enseignement supérieur*. Rapport final.

[Conklin et Yakemovic 1991] Conklin, J.; Yakemovic, K. (1991). A Process-Oriented Approach to Design Rationale. *Human-Computer Interaction* 6 (3 & 4): 357–391.

[Conklin et al. 1988] Conklin, J., Begeman, M.L. (1988). GIBIS: A Hypertext Tool for Exploratory Policy Discussion. *ACM Transactions on Office Information Systems*, 6(4).

[Caeiro-Rodriguez et al. 2007] Caeiro-Rodriguez, M., Marcelino, M.J., Llamas-Nistal, M., Anido-Rifon, L., Mendes, A.J. (2007). Supporting the Modeling of Flexible Educational Units PoEML: A Separation of Concerns Approach. *Journal of Universal Computer Science*, vol. 13, no. 7, pp 980-990.

[Caron 2007a] Caron P.A. (2007). *Ingénierie dirigée par les modèles pour la construction de dispositifs pédagogiques sur des plateformes de formation*. Doctorat de l'université des sciences et technologiques de Lille.

[Caron 2007b] Caron P.A. (2007). *Bricoles: une approche dispositive des applications Web 2.0 utilisables pour enseigner*. In: *Actes de la conférence EIAH 2007, 27-29 Juin 2007, Lausanne (Suisse)*, p. 137-142.

[Czarnecki et Eisenecker 2000Czarnecki] K. and Eisenecker U.W. (2000). Generative Programming: Methods, Tools, and Applications. Addison Wesley.

[Clark et al. 2004] Clark T., Evans A., Sammut P. and Willans J. (2004). Applied Metamodelling – A Foundation for Language Driven Development. version 0.1.

[Carroll 2000] Carroll, J.M. (2000). Five reasons for scenario-based design. *Interacting with Computers*, 13 (1), pp. 43-60.

[Como et Snow 1986] Como, L., and Snow, E. R. (1986). Adapting teaching to individual differences among learners. In M. C. Wittrock (Ed.), *Handbook of research on teaching* (3rd ed.). New York: Macmillan

[Chen 2000] Chen, G. and Kotz, D. (2000). A survey of context -aware mobile computing research. Dartmouth Computer Science Technical Report TR2000-381.

[Gueroui 2002] Guerouil S. (2002). L'Engineering pédagogique en 5 questionnements. Dans la Journées Pédagogiques de la Faculté de Médecine de Annaba, les 1 et 2 juillet 2002. Disponible à : <http://www.ppt2txt.com/r/867d463f/>, dernière consultation le 10/07/12.

[Dessus 2005] Dessus, P. (2005). Quels sont les soubassements cognitifs de l'activité d'enseignement ? *Dossiers Sci.Educ.*, 14, 111-122.

[Dessus et K.Schneider 2006] Dessus, P. et K.Schneider D. (2006). Scénarisation de l'enseignement et contraintes de la situation. Actes du colloque organisé dans le cadre de la 8ème Biennale de l'Education, scénariser l'enseignement et l'apprentissage : une nouvelle compétence pour le praticien, Lyon, 14 Avril 2006.

[Dodero et Díez 2006] Dodero, J. M. and Díez, D. (2006). Model-Driven Instructional Engineering to Generate Adaptable Learning Materials. *Proceedings of ICALT2006*. July, Kerkrade, The Netherlands: IEEE. p1188-1189.

[Dodero et al. 2008] Dodero,J.M., Tattersall, C., Burgos, D. and Koper, R. (2008). Transformational Techniques for Model-Driven Authoring of Learning Designs. H. Leung et al. (Eds.): *ICWL 2007, LNCS 4823*, pp. 230 – 241, Springer-Verlag Berlin Heidelberg.

[Dolog et al. 2007] Dolog P., Kravcik M., Cristea A., Burgos D., De Bra P., Ceri S., Devedzic V., Houben G-J., Libbrecht P., Matera M., Melis E., Nejd W., Specht M., Stewart C., Smits

D., Stash N., and Tat C.(2007). Specification, authoring and prototyping of personalised workplace learning solutions. *International Journal of Learning Technology* V 3(3), pp 286-308

[De Bra et Calvi 1999] De Bra, P. and L. Calvi (1999). AHA! An open Adaptive Hypermedia Architecture. *New Review of Hypermedia and Multimedia*.

[De Bra et al. 2004] De Bra, P., Aroyo, L., and Cristea, A. (2004). Adaptive Web-based Educational Hypermedia. In M. Levene & A. Poulouvassilis (Eds.), *Web Dynamics, Adaptive to Change in Content, Size, Topology and Use* (pp. 387-410): Springer.

[Deursen et al. 2000] Deursen, A.V., Klint, P. et Visser, J. (2000). Domain-specific languages: An annotated bibliography. *ACM SIGPLAN Notices*, 35(6):26–36.

[De Moura Filho 2007] De Moura F.C.O., MDEduc (2007). conceiving and implementing a language-oriented approach for the design of automated learning scenario. *Doctorat informatique de l'Université des sciences et technologies de Lille*.

[De Moura Filho et Derycke 2007] De Moura Filho C.O. et Derycke A. (2007). Concevoir des Scénarios Pédagogiques Exécutables avec des Patrons de Conception Pédagogiques, In : *EIAH 2007, 27-29 juin 2007, Lausanne (Suisse)*, p.119-130.

[Dillenbourg et Tchounikine 2007] Dillenbourg, P. et Tchounikine P. (2007). Flexibility in macro-scripts for CSCL. *Journal of Computer Assisted Learning* , 23(1), 1-13.

[Dey 2001] Dey A. (2001). Understanding and using context. *Personal and ubiquitous computing*, 5(1): 4–7.

[Desprès et Jacoboni 2010] Desprès, C., Jacoboni, P. (2010). *Hop3x : Un Environnement de Suivi de TP de Programmation, Logiciel, Université du Maine*.

[Engestrom 1987] Engeström Y. (1987). *Learning by expanding: An activity-theoretical approach to developmental research*. Helsinki, Finland: Orienta-Konsultit Oy.

[El-Kechaï 2008] El-Kechaï H. (2008). *Conception collective de scénarios pédagogiques dans un contexte de réingénierie : une approche par la métamodélisation située*. Doctorat de l'université du Maine, 2008.

[Ehrig et al. 2005] Ehrig, K. and Ermel, C. and H"ansgen, S. and Taentzer, G. (2005). *Generation of Visual Editors as Eclipse-Plugins*. *Automated Software Engineering' 05*, IEEE Computer Society,

[EMP 2008] Eclipse Modeling Projects (2008). <http://www.eclipse.org/modeling/>. Dernière consultation Avril 2012.

[Eastman 1969] Eastman C. (1969). Cognitive processes and ill-defined problems: a case study from design. Proceeding of the First Joint International Conference on I.A, Washington, D.C, pages 669–690.

[Ferraris et al. 2005] Ferraris C., Lejeune A., Vignollet L. et David J.-P. (2005). Modélisation de scénarios pédagogiques collaboratifs. Actes de la conférence EIAH 2005, Montpellier (France), pages 285–296.

[Ferraris et al. 2007] Ferraris C., Martel C., Vignollet L. (2007). LDL for Collaborative Activities, In: Handbook of Visual Languages in Instructional Design: Theories and Practices, Botturi L., Stubbs T. (Eds), Hershey, PA (USA): Idea Group, p. 226-254, ISBN : 978-159904729-4.

[Ferraris et Lejeune 2007] Ferraris C. et Lejeune A. (2007). Le couplage "scénario - traces". exemple du langage LDL. Cinquième école thématique du CNRS sur les EIAH "Personnalisation des EIAH".

[Fredericksen, Pickett et al. 2000] Fredericksen, E., Pickett, A., Shea, P., and Pelz, W. (2000). Student satisfaction and perceived learning with on-line courses: Principles and examples from the SUNY Learning Network. Journal of Asynchronous Learning Networks, 4(2).

[Frank 1965] Frank, H. (1965). Lehrautomaten für die Einzel- und Gruppenschulung. Lehrmaschinen in kybernetischer und pädagogischer Sicht, Nürnberg, Klett.

[Fischer et al. 1995] Fischer, G., Lemke, A., McCall, R. and Morch, A. (1995). Making Argumentation Serve Design, in Design Rationale Concepts, Techniques, and Use, T. Moran and J. Carroll, (eds), Lawrence Erlbaum Associates, pp. 267-294.

[Favre et al. 2006] Favre J.-M., Estublier J., Blay-Fornarino M. (2006). L'ingénierie dirigée par les modèles – Au delà du MDA, Traité IC2, série Informatique et Systèmes d'Information, Hermès, Lavoisier, Paris

[Feilkas, 2006] Feilkas M. (2006). How to represent Models, Languages and Transformations?<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.104.7999>, dernière visite Juin 2012

[Farail et al. 2006] Farail P., Gaufillet P., Canals A., Le Camus C., Sciamma D., Michel P., Crégut X, Pantel M. (2006). The TOPCASED project: a Toolkit in Open source for Critical Aeronautic SystEms Design. Dans : European Congress on Embedded Real-Time Software (ERTS 2006), Toulouse, Société des Ingénieurs de l'Automobile.

[Griffiths et al. 2005] Griffiths, D., Blat, J., Garcia, R., Vogten, H., & Kwong, KL. (2005). Learning Design Tools. In R. Koper & C. Tattersall (Eds.), Learning Design, A Handbook on Modelling. Springer-Verlag Berlin. p. 114-116.

[Grandbastien 2006] Grandbastien M., Labat J.-M. (ed.). (2006). Traité IC2 Information Commande Communication, Hermès, Paris.

[Goodyear 1997] Goodyear P. (1997). instructional Design Environments: Methods and Tools for the Design of Complex Instructional Systems. In : instructional Design : International Perspectives. Volume 2: Solving Instructional Design Problems. Edited by : S. Dijkstra, N.Seel, F.Schott,R.D. Tennyson.

[Garcia et al. 1993] Garcia, A., Howard, H., Stefik, M. (1993), Active Design Documents: A New Approach for Supporting Documentation in Preliminary Routine Design, Tech. Report 82, Stanford Univ. Center for Integrated Facility Engineering, Stanford, Calif.

[Gruber 1990] Gruber, T. (1990). Model-based Explanation of Design Rationale, in Proceedings of the AAAI-90 Explanation Workshop, Boston, July 30.

[Ganeshan et al. 1994] Ganeshan R., Garrett J., Finger, S. (1994), A framework for representing design intent, Design Studies Journal, V15 No. 1, January, pp. 59-84.

[Gruber et Russell 1990] Gruber, T. R. and Russell, D. M. (1990). Design Knowledge and Design Rationale: A Framework for Representation, Capture, and Use. Technical Report, Knowledge Systems Laboratory, Stanford University, July 27.

[Henri 2007] Henri, F. (2007). Modèles du sujet pour la conception. Journal of distance education revue de l'éducation à distance, vol. 22, no. 1, 101-106. Disponible sur : <http://www.jofde.ca/index.php/jde/article/viewPDFInterstitial/57/469>.

[Henri et al. 2007] Henri, F., Compte, C. et Charlier, B. (2007). La scénarisation pédagogique dans tous ses débats...Revue internationale des technologies en pédagogie universitaire, 4(2), <http://www.ritpu.org/>

[Henri et Maina 2007] Henri F. et Maina, M. (2007). Pour une meilleure intégration de l'utilisateur dans le processus de conception et d'évaluation des outils TELOS. In Environnements cognitifs centrés utilisateur : de la théorie à la pratique; Quatrième conférence annuelle du réseau de chercheurs LORNET - I2LOR.

[He, Kinshuk et al. 2002] He, S., Kinshuk, Hong H., and Patel A. (2002). Granular Approach to Adaptivity in Problem-based Learning Environment. Proceedings of ICALT (pp. 3-7)

[Hummel et al. 2004] Hummel H., Manderveld J., Tattersall C. and Koper R. (2004). Educational modelling language and learning design: new opportunities for instructional reusability and personalised learning. International Journal on Learning Technology, Vol 1, No 1, 111-126.

[Huberman 1986] Huberman, A.M. (1986). Répertoires, recettes et vie de classe : comment les enseignants utilisent l'information. In M.Crahay & D.Lafontaine (Eds.). L'art et la science de l'enseignement, p.151-183, Bruxelles, Labor.

[Hooey et Foyle 2007] Hooey B. L., Foyle D. C. (2007). Requirements for a Design Rationale Capture Tool to Support NASA's Complex Systems. International Workshop on Managing Knowledge for Space Missions. Pasadena, CA. July 17–19 2007

[Horner et Atwood 2006] Horner, J.; Atwood, M.E. (2006), "Effective Design Rationale: Understanding the Barriers", in Dutoit, A.H.; McCall, R.; Mistrík, I. et al., Rationale Management in Software Engineering, Springer Berlin Heidelberg, pp. 73-90

[Hernández-Leo et al. 2006] Hernández-Leo, D., Villasclaras-Fernández, E.D., Asensio-Perez, J.I., Dimitriadis, Y., Jarrín-Abellán, I.M., Ruiz-Requies, I. et Rubia-Avi, B. (2006). COLLAGE: A collaborative Learning Design editor based on patterns. Educational Technology and Society, 9 (1), 58-71.

[Halff et al. 2003] Halff, H, Hsieh, P., Wenzel, B., Chudanov, T., Dirnberger, M., Gibson, E. et Redfield, C. (2003). Requiem for a Development System: Reflections on Knowledge-Based, Generative Instruction, Chapter 2 in Murray, T., Blessing, S. & Ainsworth, S. (Eds.). Authoring Tools for Advanced Technology Learning Environments. Kluwer Academic Publishers, Dordrecht.

[Hutchinson et al. 2011] J. Hutchinson, M. Rouncefield, and J. Whittle. (2011). Model-driven engineering practices in industry. In Proceedings of the 33rd International Conference on Software Engineering (ICSE '11). ACM, New York, NY, USA, pp 633-642

[IMS-LD 2003] IMS-LD (2003). IMS Learning Design v1.0 Final Specification. IMS Global Learning Consortium, Accessible in <http://www.imsglobal.org/learningdesign/index.html>

[Iksal et al. 2004] Iksal S., Barré V., Choquet C., Corbière A. (2004). Comparing prescribed and observed for the re-engineering of e-learning systems, In: IEEE Sixth International Symposium on Multimedia Software Engineering (MSE), 13-15 décembre 2004, Miami (USA), p. 106-109

[Jameson 2003] Jameson, A. (2003). Systems that adapt to their users: An integrative Overview. in Tutorial presented at 9th International Conference on User Modelling, Johnstown, PA, USA

[Jarczyk et al. 1992] Jarczyk A. P.J., Löffler P. , and Shipman III F. M. (1992). Design Rationale for Software Engineering: A Survey. 25th Hawaii International Conference on System Sciences, 2, pp. 577-586

[Jézéquel et al. 2012] Jézéquel J-M, Combemale B., Vojtisek D. (2012). Ingénierie Dirigée par les Modèles : des concepts à la pratique, Ellipses.

[Jouault et Bézivin 2006] Jouault F. and Bézivin J. (2006), KM3: A DSL for Metamodel Specification. ;In Proceedings of FMOODS,171-185.

[Jouault et al. 2006] Jouault F, Bézivin J, Consel C, Kurtev I, and Latry F. (2006). Building DSLs with AMMA/ATL, a Case Study on SPL and CPL Telephony Languages. In: Proceedings of the 1st ECOOP Workshop DSPD, July 3rd, Nantes, France.

[Jarke et al. 1998] Jarke, M., Bui, T., and Carroll, J.M. (1998) Scenario Management: An Interdisciplinary Approach. Requirements Engineering, 3 pp. 155-173

[Karsenty 2001] Karsenty L. (2001). Capitaliser le contexte des décisions en conception. Livre Management des connaissances : modèles d'entreprise et applications, p 49-69.

[Karampiperis et Sampson 2004] Karampiperis, P., et Sampson, D. (2004). Adaptive learning object selection in intelligent learning systems. Journal of Interactive Learning Research, 15(4), 389-408.

[Koper 2001] Koper R. (2001). Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML, Open University of the Netherlands.

[Koper et Burgos 2005] Koper, R., and Burgos, D. (2005). Developing advanced units of learning using IMS Learning Design level B. *International Journal on Advanced Technology for Learning*, 2(3).

[Koper et Tattersall 2005] Koper, R., et Tattersall, C. (Eds.) (2005). *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*. Berlin Heidelberg: Springer.

[Kelly et al. 1996] Kelly S., Lyytinen K., Rossi M. (1996). MetaEdit+: a fully configurable multi-user and multi-tool CASE environment, in: *Proceedings of CAISE'96, Lecture Notes in Computer Science 1080*, Springer-Verlag, pp.1–21

[Kelly 2004] Kelly S. (2004). Comparison of Eclipse EMF/GEF and MetaEdit+ for DSM. Dans *19th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, Workshop on Best Practices for Model Driven Software Development*

[Kelly et Tolvanen 2008] Kelly S. et Tolvanen J. (2008). *Domain-specific modeling: enabling full code generation*. Wiley-IEEE Computer Society Pr.

[Kunz et Rittel 1970] Kunz, W. and Rittel, H. (1970). *Issues As Elements of Information Systems*. Berkeley : University of California.

[Kravčik et Gašević 2006] Kravcik, M. et Gasevic, D. (2006). *Sharing Knowledge in Adaptive Learning Systems*. *Proceedings of ICALT2006*. Kerkrade, The Netherlands: IEEE. Retrieved July 30th, 2006, from <http://dSPACE.learningnetworks.org>

[Lacaze 2004] Lacaze, X. (2004). *La conception rationalisée pour les systèmes interactifs*. Dans : *Rencontres Jeunes Chercheurs en Interaction Homme Machine, RJCiHM' 04*, Lacanau, France.

[Lacaze 2005] Lacaze X., (2005). *Conception rationalisée pour les systèmes interactifs Une notation semi formelle et un environnement d'édition pour une modélisation des alternatives de conception*. Doctorat soutenue le 20 juin 2005

[Laforcade et al. 2003] Laforcade P., Barbier F., Nodenot T. Et Sallaberry C. (2003). *Profiling co-operative problem-based learning situations*. *Proceedings of the 2nd IEEE International Conference on Cognitive Informatics (ICCI'2003)*, London, UK

[Laforcade 2004] Laforcade P.(2004). Modélisation et méta-modélisation UML pour la Conception et la mise en œuvre de situations problèmes coopératives. Doctorat de l'Université de Pau et des Pays de l'Adour

[Laforcade 2005] LaforcadeP. (2005). Approche par transformation de modèles pour la Conception d'EIAH-illustration entre les langages CPM et IMS-LD. Actes de la conférence EIAH 2005, Montpellier (France), pages 213–224

[Laforcade 2007a] Laforcade, P., (2007). Ingénierie Dirigée par les Modèles et Environnements Informatiques pour l'Apprentissage Humain. Cours de Master EIAH, Université du Maine, 2007.

[Laforcade 2007b] Laforcade P., Barré V., Zendagui B. (2007). 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: Actes d'EIAH'07, Lausanne (Suisse), 27-29 juin 2007.

[Laforcade 2010] Laforcade P. (2010). A Domain-Specific Modeling approach for supporting the specification of Visual Instructional Design Languages and the building of dedicated editors. *Journal of Visual Languages and Computing*, V 21 (6), December 2010, pp 347-358.

[Lalonde et Pinette 1999] Lalonde, J. et Pinette, A. (1999). Portail cantic. Centre d'Apprentissage des Nouvelles Technologies de l'Information et des Communications.

[Langlois et al. 2007] Langlois B., Consuela-Elena J. et Jouenne E. (2007). DSL Classification. *Proceedings of the 7thOOPSLA Workshop on Domain-Specific Modeling*.

[Leutner 1992] Leutner, D. (1992). *Adaptive Lehrsysteme; Instruktionspsychologische Grundlagen und experimentelle Analysen*. Weinheim, Beltz.

[Lee 1991] Lee, J. (1991). Extending the potts and bruns model for recording design rationale. In: *13 th International Conference on Software Engineering*, pp.114–125.

[Lee 1997] Lee J. (1997). *Design Rationale Systems: Understanding the Issues*". *IEEE Expert* 12 (3), pp 78–85

[Le Gac 2007] E. Le Gac. (2007). Point du Vue sur les DSM : Augmenter la productivité de vos développements J2EE grâce à l'approche orientée modèles DSM. Intervention au XIIème Symposium de l'Architecture, 28 Novembre 2007.

[Ledeczi et al. 2001] Ledeczi A., Maroti M., Bakay A., Karsai G., Garrett J., Thomason IV C., Nordstrom G., Sprinkle J., Volgyesi P. (2001). The Generic Modeling Environment. In Proceeding of the IEEE Workshop on Intelligent Signal Processing, Budapest, Hungary,

[Lekira 2010] Lekira A. (2010). Les indicateurs à la base de l'aide à l'auto-régulation tutorale. In Actes des Rencontres Jeunes Chercheurs en EIAH (RJC2010), Lyon, France, 2010.

[Lekira et al. 2011] Lekira A., Després C., Jacoboni P., Choquet C., Iksal S., Py D., et Pham-Thi D. (2011). Using indicators during synchronous tutoring of practical work. In The 11th IEEE International Conference on Advanced Learning Technologies (ICALT2011), Athens - Gerogia(USA).

[McCall 1987] McCall, R.(1987). PHIBIS: procedural hierarchical issue-based information systems. In: Proceedings of the International Congress on Planning and Design Theory.

[MacLean et al. 1991] Maclean A., Young R.M., Bellotti V.M.E., and Moran T.P. (1991). Questions, Options, and Criteria: Elements of Design Space Analysis. Human- Computer Interaction, Vol.6.

[McKerlie et MacLean 1993] McKerlie D., and MacLean A. (1993). Experience with QOC Design Rationale. INTERCHI'93, Amsterdam : ACM,

[MacLean, Bellotti et Shum 1993] MacLean A., Bellotti V.and Shum S. (1993). Developing the Design Space with Design Space Analysis. Published in: Byerley, P.F., Barnard, P.J., & May, J. (eds) Computers, Communication and Usability: Design issues, research and methods for integrated services, pp. 197-219. Elsevier: Amsterdam.

[Merceron et Yacef 2003] Merceron, A. et Yacef, K. (2003). A Web-based tutoring tool with mining facilities to improve learning and teaching. AI-Ed'2003, IOS Press.

[Murray et al. 2003] Murray, T., Blessing, S. et Ainsworth, S. (Eds.). (2003). Authoring Tools for Advanced Technology Learning Environments. Kluwer Academic Publishers, Dordrecht.

[Murray 2004] Murray, T. (2004). Design Tradeoffs in Usability and Power for Advanced Educational Software Authoring Tools. Educational Technology, 44(5), 10-16.

[Martel 1998] Martel C. (1998). La modélisation des activités conjointes. Rôles, places et positions des participants, these de l'Université de Savoie, Septembre 1998.

[Martel et al. 2007] Martel C., Vignollet L., Ferraris C. (2007). Une Ingénierie des Environnements Informatiques pour l'Apprentissage Humain basée sur un modèle de l'activité – de la conception à l'opérationnalisation et l'exécution des scénarios d'apprentissage, In: Actes du colloque IDM'07 – 3èmes journées « Ingénierie Dirigée par les Modèles », Toulouse (France), p. 183-198.

[Martel 2005] V. Martel. (2005). Emergence d'une communauté d'apprentissage en réseau à l'ordre primaire : L'activité de transformation d'un environnement d'apprentissage par la direction, les enseignants et les élèves (Étude de cas). Maîtrise en psychopédagogie présenté à la Faculté des études supérieures de l'Université Laval. Québec, Août 2005.

[Miao et al. 2008] Miao, Y., Sodhi, T., Brouns, F., Sloep, P., et Koper, R. (2008). Bridging the Gap between Practitioners and E-learning Standards: A Domain-specific Modeling Approach. Educational Technology Expertise Center, Open University of the Netherlands.

[MS DSLT 2010] Microsoft Domain-Specific Language Tools: <http://msdn.microsoft.com/vstudio/DSLTools/>, trouvé en Mars 2010

[Miloud, 2010] Site Web : <http://miloud.webs.com/>, dernière consultation Juin 2012

[Munro et al. 1997] Munro, A., Johnson, M.C., Pizzini, Q.A., Surmon, D.S., Towne, D.M, et Wogulis, J.L. (1997). Authoring simulation-centered tutors with RIDES. International J. of Artificial Intelligence in Education. Vol. 8 , No. 3-4, pp. 284-316.

[Merrill 2003] Merrill, M.D. (2003). Using Knowledge Objects to Design Instructional Learning Environments. Chapter 7 in Murray, T., Blessing, S. & Ainsworth, S. (Eds.). Authoring Tools for Advanced Technology Learning Environments. Kluwer Academic Publishers, Dordrecht.

[Nagase et al. 2004] Nagase Y., Hashimoto D., Sato M. (2004). Applying Model-Driven Development to Business Systems using RM-ODP en EDOC, Workshop on ODP for Enterprise Computing, EDOC2004, p. 36-42.

[Nodenot et al. 2003] Nodenot T., Laforcade P., Marquesuzaà C. et Sallaberry C.(2003). Knowledge modeling o fco-operative learning situations: Towards a UML profile. Proceedings of the 11th International Conference on Artificial Intelligence in Education (AIED'2003), Sydney, Australia.

[Nodenot 2005] Nodenot T. (2005). Contribution à l'Ingénierie dirigée par les modèles en EIAH: le cas des situations-problèmes coopératives. Habilitation à diriger des recherches en informatique.

[Nodenot et al. 2005] Nodenot T., LePallec X. et Marino O. (2005). La manipulation des modèles: modélisation, méta-modélisation, transformation. Troisième école thématique du CNRS sur les EIAH "MODÈLES, ARCHITECTURES LOGICIELLES ET NORMES", organisée par le CNRS-Département STIC-RTP39.

[Nichols 2003] Nichols, M. (2003): A theory for eLearning. *Educational Technology & Society*, 6(2), 1-10.

[Newman et Marshall 1991] Newman S.E. and Marshall, C.C. (1991). Pushing Toulmin Too Far: Learning From an Argument Representation Scheme. Xerox PARC Technical Report No. SSL-92-45.

[OMG 2004] OMG (2004). Enterprise collaboration architecture (ECA) specification, version 1.0. Technical report.

[OMG 2006] OMG (2006). MDA specifications. <http://www.omg.org/mda/specs.htm>, consulté Juin 2012.

[OMG 2010] OMG (2010), Object Constraint Language, Version 2.0, accessible à <http://www.omg.org/spec/OCL/2.0/PDF/>, consulté en Juin 2012

[Ouraiba et al. 2008a] Ouraiba, E.A., Chikh, M. A. et Chikh, A. (2008). Filtrage Neuronal des Objets d'Apprentissage selon les Profils des Apprenants. In: 10th Maghrebian Conference on Information Technologies (MCSEAI'08), Oran, Algeria, 28 - 30 April 2008.

[Ouraiba et al. 2008b] Ouraiba E.A., Chikh A., Taleb-ahmed A. (2008). Recommandation automatique des objets d'apprentissage aux apprenants en utilisant SVM. 1st International Conference on Web and Information Technologies (ICWIT'08), Sidi Bel Abbes (Algeria), p.7-13, 29-30 Juin 2008.

[Ouraiba et al. 2009] Ouraiba E.A., Chikh A., Taleb-ahmed A., El Yebdri Z. (2009). Automatic personalization of learning scenarios using SVM. 9th IEEE International Conference on Advanced Learning Technologies (ICALT'09), Riga (Latvia), p.183-185, July 14-18 2009.

[Ouraiba 2010] Ouraiba E.A. (2010). Ingénierie des scénarios pédagogiques ouverts. Troisièmes Rencontres Jeunes Chercheurs en EIAH (RJC-EIAH'10), Lyon, p.83-88.

[Ouraiba et al. 2010] Ouraiba E.A., Choquet C., Cottier P., Després C., and Jacoboni P. (2010). Engineering of open learning scenarios: the case of Hop3x learning scenarios. Proceedings of IEEE ICALT'10, Tunisia, , pp. 264-268

[Ouraiba et al. 2011a] Ouraiba E.A., Choquet C., Cottier P. (2011). Instructional Design Rationale with QOC: a Model Driven Engineering and Domain Specific Modeling Approach. The 2011 International Conference of Information Engineering, The World Congress on Engineering organized by the International Association of Engineers (IAENG), London, U.K, July 2011.

[Ouraiba et al. 2011b] Ouraiba EA, Choquet C, Cottier P. (2011). Domain Specific Supports for Design Rationale of Open Pedagogical Scenarios. In IAENG International Journal of Computer Science (V38 Issue4), edited by International Association of Engineers, p.368-377, 2011.

[Ouraiba et al. 2011c] Ouraiba E.A., Choquet C., Cottier P. (2011). Opening TEL systems for teachers: A Domain-Specific Modeling & Model-Driven Engineering Approach. The 3rd International Conference on Computer Supported Education (CSEDU'11), Noordwijkerhout, Netherlands, May 2011.

[Ouraiba et al. 2011d] Ouraiba E.A., Choquet C., Cottier P. (2011). Domain-Specific Modeling Approach To Support Instructional Design Rationale". Proceedings of IEEE ICALT'11, Athens, Georgia, USA, 6-8 July 2011, pp. 205-206.

[Park, Perez et Seidel 1987] Park, O., Pérez, R. S., et Seidel, R. J. (1987). Intelligent CAI: Old wine in new bottles or a new vintage? In G. Kearsley (Ed.), *Artificial intelligence and instruction: Applications and methods*. Boston, MA: Addison–Wesley.

[Park et Seidel 1989] Park, O., et Seidel, R. J. (1989). A multidisciplinary model for development of intelligent computer-assisted instruction. *Educational Technology Research and Development*, 37, 72–80.

[Park et Lee 2001] Park O., Lee J. (2001). Adaptive Instructional Systems. In: *Handbook of Research for Educational Communications and Technology* Edited by David H. Jonassen, accessible à: <http://www.aect.org/edtech/ed1/25.pdf>, dernière consultation Juin 2012.

[Puustinen et al. 2006] Puustinen, M., Baker, M. et Lund, Gestalt, K. (2006). A framework for redesign of educational software. *Journal of Computer Assisted Learning* 22, pages 1- 13.

[Paquette 2002] Paquette, G. (2002). L'ingénierie du télé-apprentissage, pour construire l'apprentissage en réseaux, Presses de l'Université du Québec, 490 pages, ISBN 2-7605-1162-6, D-1162

[Paquette 2004] Paquette, G. (2004). Instructional engineering for learning objects repositories networks, 2nd International Conference on Computer Aided Learning in Engineering Education, pp 25-36, Grenoble, France.

[Pask 1964] Pask, G. (1964). The adaptive teaching system. Teaching machines and programming. K. Austwick. Oxford, London, Pergamon Press.

[Pasco 1998] Pasco, J. (1998). Adding generic contextual capabilities to wearable computers. In Proceedings of the Second IEEE International Symposium on Wearable Computers, pages 92–100.

[Pernin 2003] Pernin, J.P. (2003). Objets pédagogiques : unités d'apprentissage, activités ou ressources ?, Revue "Sciences et Techniques Educatives", Hors série 2003 " Ressources numériques, XML et éducation", pp 179-210, éditions Hermès.

[Pernin et Lejeune 2004a] Pernin J.-P., Lejeune A. (2004). Nouveaux dispositifs instrumentés et mutations du métier de l'enseignant, 7ème biennale de l'Education, Lyon, 2004, <http://www.inrp.fr/Acces/Biennale/7biennale/Contrib/longue/7134.pdf>, dernière consultation, Juin 2012

[Pernin et Lejeune 2004b] Pernin J.-P. et Lejeune A. (2004). Dispositifs d'apprentissage instrumentés par les technologies : vers une ingénierie centrée sur les scénarios. In: actes du colloque TICE, Compiègne (France), p.407-414.

[Pernin 2007] Pernin, J.-P. (2007). Mieux articuler activités pour l'apprentissage, artefacts logiciels et connaissances : vers un modèle d'ingénierie centré sur le concept de scénario. In M. Baron, D. Guin & L. Trouche (Eds.), Environnements informatisés et ressources numériques pour l'apprentissage : conception et usages, regards croisés, p 161-190, Hermès, Paris.

[Pena-Mora et al. 1995] Pena-Mora, F., Sriram, D., Logcher, R. (1995), Design Rationale for Computer-Supported Conflict Mitigation, ASCE Journal of Computing in Civil Engineering, pp. 57-72.

[Pham Thi Ngoc 2010] Pham-Thi D., Iksal S. et Choquet C. (2010). Learning tracks' analysis with DCL4UTL: An Instantiation with the Division of Labor Indicator. In AACE, editor, Global Learn Asia Pacific 2010, pages 3334–3343, Penang, Malaysia.

[Pham Thi Ngoc 2011] Pham-Thi D.(2011). Spécification et conception de services d'analyse de l'utilisation d'un environnement informatique pour l'apprentissage humain. Doctorat de l'université du Maine, 2011.

[Previt 2003] Prévér, D. (2003). Différentes modélisations pour prendre en compte les usages dans la conception d'un EIAH. Environnements Informatiques pour l'Apprentissage Humain, Montpellier.

[Rabardel 1995] Rabardel P. (1995). Les hommes et les technologies. Approche cognitive des instruments contemporains, Paris, A. Colin.

[Randriamalaka et Iksal 2006] Randriamalaka N. et Iksal S. (2006). Patterns approach in the re-engineering process of learning scenario. In The 6th IEEE International Conference on Advanced Learning Technologies (ICALT2006), Kerkrade, The Netherlands.

[RAP 1999] Réseau National des Technologies du Logiciel, Rapport de synthèse du groupe B2 « Systèmes interactifs et produits multimédia », accessible à : <http://www.industrie.gouv.fr/observat/innov/rntl/groupeb2.htm>, 1999.

[Rawlings et al. 2002] Rawlings A., Rosmalen P., Kopera R., Rodríguez-Artacho M. et Lefrere P. (2002). Survey of educational modeling languages. CEN/ISSWS/LT Learning Technologies Workshop.

[Reload 2004] The Reload Project, The University of Bolton, The University of Strathclyde and JISC, 2004. <http://www.reload.ac.uk/>, dernière consultation Mai 2012

[Rumetshofer et Wöß 2003] Rumetshofer, H and Wöß, W. (2003): XML-based Adaptation Framework for Psychological-driven E-learning Systems. Educational Technology & Society, 6 (4), 18-29.

[Roditi 2003] Roditi, É. (2003). Régularité et variabilité des pratiques ordinaires d'enseignement. Le cas de la multiplication des nombres décimaux en sixième, Recherches en Didactique des Mathématiques. 23 (2).

[Rogalski 2003] Rogalski, J. (2003). Y a-t-il un pilote dans la classe ? Une analyse de l'activité de l'enseignant comme gestion d'un environnement dynamique ouvert. *Recherches en Didactique des Mathématiques*. 23(3)342-348.

[Sampson et al. 2006] Sampson, D.G., Karampiperis, P. et Zervas, P. (2005). ASK-LDT: A Web-based learning scenarios authoring environment based on IMS Learning Design. *Advanced Technology for Learning* 2(4).

[Santos, Boticario, Campo et Saneiro 2007] Santos, O.C., Boticario, J.G., Campo, E. et Saneiro, M. (2007). IMS-LD as a workflow to provide personalized support for disabled students in higher education institutions. *Workshop Towards User Modelling and Adaptive Systems for All. International Conference on User Modelling*.

[Santos et al. 2007] Santos, O.C., Boticario, J.G., Raffenne, E. et Pastor, R. (2007). Why using dotLRN? UNED use cases. *FLOSS (Free/Libre/Open Source Systems) International Conference*. Accessed online on 1 March 2007 at: http://dotlrn.org/file-storage/view/papers/floss-dotlrn-ocsjgberpp_final.pdf

[Santos et Boticario 2007] Boticario, J.G. et Santos, O.C. (2007). An open IMS-based user modelling approach for developing adaptive learning management systems. *Journal of Interactive Media in Education (Adaptation and IMS Learning Design. Special Issue, ed. Daniel Burgos)*, 2007/02. ISSN:1365-893X. Accessed online on 8 June 2009 at: jime.open.ac.uk/2007/02.

[Salomon 1975] Salomon, G. (1975). Heuristische Modelle für die Gewinnung von Interaktionshypothesen. *Adaptiver Unterricht*. R. Schwarzer. München, Kösel: 127-145.

[Schonenberg et al. 2007] M.H. Schonenberg, R.S. Mans, N.C. Russell, N.A. Mulyar, and W.M.P. van der Aalst. (2007). *Towards a Taxonomy of Process Flexibility (Extended Version)*. BPM Center Report BPM-07-11, BPMcenter.org.

[Schonenberg et al. 2008] Schonenberg, H., Mans, R., Russell, N., Mulyar, N. et Aalst, W. (2008). Process flexibility: A survey of contemporary approaches, *Advances in Enterprise Engineering I*, p.16–30.

[Schneider 2006] Schneider D.K. (2006). La norme Learning Design, supports du cours Technologie Internet et Education, TECFA, accessible à: <http://tecfa.unige.ch/guides/tie/pdf/files/pedago-ld.pdf>, dernière consultation Juin 2012.

[Schilit 1994] Schilit, B. Adams N and Want, R. (1994). Context-aware computing applications. Proceedings of the IEEE workshop on mobile computing systems and applications, Santa Cruz IEEE Press, Piscataway, p 85–90

[Simondon 1958] Simondon, G. (1958). Du mode d'existence des objets techniques, Aubier, Paris.

[Sim et Duffy 1994] Sim, S., Duffy, A. (1994), A New Perspective to Design Intent and design Rationale, in Artificial Intelligence in Design Workshop Notes for Representing and Using Design Rationale, 15-18 August, pp. 4-12.

[Simon 1973] Simon H. (1973). The structure of Ill structured problems. Artificial Intelligence, n°4, pages 181 - 201.

[Stiegler 2007] Stiegler, B. (2007). Le réseau numérique à l'origine d'un nouveau modèle industriel. Conférence plénière en L'INRIA a quarante ans, Institut national de recherche en informatique et en automatique. Lille, les 10 et 11 décembre 2007.

[Specht 1998] Specht, M. (1998). Adaptive Methoden in computerbasierten Lehr/Lernsystemen. Trier, University of Trier.

[Specht et Burgos 2006] Specht, M. and D. Burgos (2006). Implementing Adaptive Educational Methods with IMS Learning Design. Adaptive Hypermedia 2006, Dublin, Ireland.

[Specht et Burgos 2007] Specht, M. et Burgos, D. (2007). Modeling Adaptive Educational Methods with IMS Learning Design. Journal of Interactive Media in Education (Adaptation and IMS Learning Design. Special Issue, ed. Daniel Burgos), ISSN:1365-893X. Accessed online on 8 June 2009 at : jime.open.ac.uk/2007/08

[Seidel, le parc et Perez, 1989] Seidel, R. J., Park, O., and Perez, R. (1989). Expertise of CAI: Development requirements. Computers in Human Behaviors, 4, 235–256

[Steinberg et al. 2008] D. Steinberg, F. Budinsky, M. Paternostro. (2008). EMF: Eclipse Modeling Framework, Second Edition. Publisher: Addison Wesley Professional.

[Stash 2007] Stash, N. (2007). Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System". PhD doctoral Thesis, Technische Universiteit Eindhoven.

[Sodhi et al. 2007] Sodhi, T., Miao, Y., Brouns, F., et Koper, R. (2007). Supporting the non-expert in the authoring of personalized learning using IMS LD. Proceedings of the ePortfolio

2007 Conference. Maastricht, The Netherlands, October, 18-19, 2007. Accessed online on 8 June 2009 at : <http://dspace.ou.nl/handle/1820/974>

[Tattersall et al. 2005] Tattersall, C., Vogten, H., Brouns, F., Koper, R., van Rosmalen, P., Sloep, P., et van Bruggen, J. (2005). How to create flexible runtime delivery of distance learning courses. *Educational Technology & Society*, 8 (3), 226-236.

[Tang et al. 2006] A. Tang, M.A. Babar, I. Gorton, and J. Han.(2006). A survey of architecture design rationale. *Journal of Systems and Software*. 79, 12, December 2006, pp 1792-1804.

[Tchounikine 2002] Tchounikine. P. (2002). Pour une ingénierie des EIAH. *Revue I3 Information-Interaction-Intelligence*, Volume 2, n°1 : 59–95.

[Tchounikine et al. 2004] Tchounikine P., Baker M., Balacheff N., Baron M., Derycke A., Guin D., Nicaud J.-F., et Rabardel P. (2004). Platon-1 : quelques dimensions pour l'analyse des travaux de recherche en conception d'EIAH. Technical report, D'épartement STIC du CNRS.

[Tchounikine 2006] Tchounikine, P. (2006). Introduction à l'ingénierie des EIAH. In: *Environnements informatiques pour l'apprentissage humain*, Grandbastien M., Labat J.M. (ed.), *Traité IC2 Information Commande Communication*, p. 141-160, Édité par Hermes, ISBN 2-7462-1171-8.

[Tchounikine 2008] Tchounikine, P. (2008). Operationalizing macro-scripts in CSCL technological settings. *International Journal of Computer Supported Collaborative Learning (IJCSCL)*, 2008.

[Tchounikine et Tricot 2008] Tchounikine P. et Tricot A. (2008). Environnements informatiques et apprentissages humains. À paraître dans C. GARBAY, D. KAYSER, « *Informatique et Sciences Cognitives : influences ou confluences ?* », collection « *cogniprisme* », Ophrys/MSH.

[Tennyson et Christensen 1988] Tennyson, R.D. and Christensen D.L. (1988). MAIS: An intelligent learning System, in *Instructional Designs for microcomputer courseware*, D.H. Jonassen, Editor. N.J.:Erlbaum: Hillsdale.

[Towle et Halm 2005] Towle, B., and Halm, M. (2005). Design Adaptive Learning Environments with Learning Design. In R. Koper and C. Tattersall (Eds.), *Learning Design. A*

Handbook on Modelling and Delivering Networked Education and Training. The Netherlands: Springer, 215-226.

[Toulmin 1958] Stephen Toulmin (1958). *The Uses of Argument*. Cambridge: Cambridge University Press.

[Tolvanen 2006] J-P Tolvanen (2006). *Domain-Specific Modeling: How to Start Defining Your Own Language*. Accessible à : <http://www.devx.com/enterprise/Article/30550/1763/page/2>, dernière consultation Juin 2012

[TIGER 2012] The Tiger Project (Transformation-based generation of modeling environments) <http://user.cs.tu-berlin.de/~tigerprj/>, dernière consultation Juin 2012

[Tetchueng 2008] Tetchueng, J-L. (2008). *Une approche de conception d'EIAH adaptatifs fondée sur la co-conception de scénario*. Thèse soutenue le 8 janvier 2009 à Brest (Telecom Bretagne).

[Van Rosmalen et al. 2004] an Rosmalen, P., Boticario, J.G., et Santos, O. (2004). *The Full Life Cycle of Adaptation in aLFanet eLearning Environment*. In *Learning Technology Newsletter Vol. 6, Issue 4*.

[Van Rosmalen et Boticario 2005] Van Rosmalen, P., et Boticario, J. (2005). *Using Learning Design to support design- and runtime adaptation*. In R. Koper & C. Tattersall (Eds.), *Learning Design: A Handbook on Modeling and Delivering Networked Education and Training*. Heidelberg, Germany: Springer Verlag.

[Van Rosmalen et al. 2006] Van Rosmalen, P., Vogten, H., Van Es, R., Passier, H., Poelmans, P., et Koper, R. (2006). *Authoring a full life cycle model in standards-based, adaptive e-learning*. *Educational Technology & Society*, 9 (1), 72-83.

[Van Rosmalen, Vogten et al. 2006] Van Rosmalen, P., Vogten, H., Van Es, R., Van, P., H., Poelmans, P., et Koper, R. (2006). *Authoring a full life cycle model in standards-based, adaptive e-learning*. *Educational Technology & Society*, 9(1), 72-83.

[Van Rosmalen 2008] Van Rosmalen, P. (2008). *Supporting the tutor in the design and support of adaptive e-learning*. Thesis , Open Universiteit Nederland , 18 april 2008.

[Vantroys 2003] Vantroy, T. (2003). *Du langage métier au langage technique, une plateforme flexible d'exécution de scénarios pédagogiques*. Thèse de doctorat en informatique. Université des Sciences et Technologies de Lille.

[Vantroys et Peter 2005] Vantroys, T. et Peter, Y. (2005). COW, un service de support d'exécution de scénarios pédagogiques. *Revue STICEF*, Volume 12, ISSN : 1764-7223.

[Vanworkmhoudt 2004] Vanworkmhoudt G. (2004). Vérification de modèles avec EMFet OCL. Disponible à : <http://www.enic.fr/people/Vanwormhoudt/siteEMFOCL/documents/EMFOCLpresentation.pdf>

[Vogten et Martens 2005] Vogten, H., Martens, H. (2005) CopperCore 2.2.2. Heerlen: Open University of The Netherlands. Retrieved at: www.coppercore.org . Retrieved on August 9th, 2005

[Vygotsky 1934] Vygotski, L. S. (1934). *Myslenie i rec'*. Traduction française : *Pensée et langage*. F. Sève. Paris : Messidor/Éditions Sociales (1985).

[Van Joolingen et de Jong 1996] van Joolingen, W., et de Jong, T. (1996). Design and Implementation of Simulation Based Discovery Environments: The SMILSE Solution. *Jl. of Artificial Intelligence in Education* 7(3/4) p 253-276.

[Villiot-Leclercq 2007] Villiot-leclercq E. (2007). La méthode des Pléiades : un formalisme pour favoriser la transférabilité et l'instrumentation des scénarios pédagogiques, *Article de recherche*, Volume 14.

[Wang et Lindvall 1984] Wang, M., and Lindvall, C. M. (1984). Individual differences and school learning environments. *Review of Research in Education*, 11, pp 161–225.

[Wasson 1997] Wasson, B. (1997). Advanced educational technologies: The learning environment, *Computers in Human Behavior*, 13(4): 571-594.

[Weber et Specht 1997] Weber, G. and Specht M. (1997). User modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems. in *User Modeling*. Chia Laguna: Springer.

[Weber et al. 2007] Weber B., Rinderle S.B., and Reichert M.U. (2007). Change Support in Process-Aware Information Systems - A Pattern-Based Analysis. Technical Report Technical Report TR-CTIT-07-76, ISSN 1381-3625, Centre for Telematics and Information Technology, University of Twente, Enschede, <http://eprints.eemcs.utwente.nl/11331/>

[Weibelzahl 2005] Weibelzahl, S. (2005). Problems and pitfalls in the evaluation of adaptive systems. In S. Chen & G. Magoulas (Eds.). *Adaptable and Adaptive Hypermedia Systems*. Hershey, PA: IRM Press, 285-299.

[Wenger 1987] Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*, CA, USA: Morgan Kaufman.

[Wild 2006] Wild P-J. (2006). *Scenarios: a review, and case for their use in KIM*. University of Bath, August 2006.

[Yongwu 2005] Yongwu, M. (2005). *Enabling learning designers to model dynamic learning processes*. Fifth IEEE International Conference on Advanced Learning Technologies 'ICALT'2005). Volume, Issue ,Page(s): 399 – 401, 5-8 July 2005.

[Zarraonandia et al. 2006a] Zarraonandia, T., Fernández, C. et Doderó, J. M. (2006). *A late modelling approach for the definition of computer-supported learning process*. In *ADALE Workshop on Adaptive Learning and Learning Design*, Dublin, Ireland.

[Zarraonandia et al. 2006b] Zarraonandia, T., Doderó, J.M., et Fernandez C. (2006). *Crosscutting runtime adaptations of LD execution*. *Educational Technology & Society*, 9, (1), 123-137.

[Zarraonandia et al. 2007] Zarraonandia, T., Doderó, J. M., Fernández, C., Aedo, I. et Díaz, P.(2007). *Iterative Design of Learning Processes*. Chapter 15 in *Computers and Education: E-learning - from theory to practice* , Springer.

[Zarraonandia 2007] Zarraonandia, T. (2007) *Adaptaciones de Unidades de Aprendizaje en Tiempo de Ejecución*. Tesis Doctoral, Departamento de Informática, Escuela Politécnica Superior, Universidad Carlos III de Madrid, Leganés, Madrid.

[Zendagui 2010] Zendagui B.(2010). *Support 'a la spécification des besoins d'observation dans un contexte de ré-ingénierie des scénarios pédagogiques : une approche dirigée par les modèles*. Doctorat, Université du Maine, 2010.

Annexe A La session Hop3x utilisée

Cette annexe explicite la liste des tâches et des indicateurs spécifiques et transversaux de la session TP1 (Point, Droite, Triangle)²¹, menée pour initier des étudiants à la programmation orientée objet en Java. Cette session a été utilisée dans l'expérimentation de collecte de données sur le domaine de Hop3x (cf. section 7.2.1.2), et dans l'exemple illustrant l'utilisation de Hop3X ouvert (cf. section 7.2.6).

❖ Consignes à respecter

Pour ce TP vous veillerez à :

- documenter votre code en utilisant les balises *javadoc*
- respecter les conventions d'écritures (java.sun.com/docs/codeconv/)

❖ Les indicateurs transversaux de la session

1. La fréquence de compilation manuelle (par minute).
2. La fréquence d'exécution (par minute).
3. Le taux de correction des erreurs à la compilation (Est-ce qu'une même erreur revient souvent ou est-ce que l'étudiant a assimilé qu'il s'agissait d'une erreur ?).
4. Le pourcentage de noms de variable significatifs (On se basera sur la longueur des noms).
5. Le pourcentage de noms de méthode significatifs (On se basera sur la longueur des noms).
6. Le pourcentage de variables d'instances privées.
7. La détection de l'utilisation d'un membre d'instance (variable ou méthode) dans une méthode d'instance (s'il n'y a pas de membre d'instance, la méthode devrait être une méthode de classe).
8. Quel est le temps moyen passé par question par étudiant.
9. Le nombre de compilation par question par étudiant.

²¹ Le contenu de cette session a été défini par A. Lekira

❖ Tâches et Indicateurs spécifiques de chaque question**Question 1 :**

Enoncé : Ecrire la classe Point composée de deux données membres (valeurs réelles) représentant les coordonnées d'un point en x et y.

Tâches obligatoires : Créer une classe publique Point avec des champs privés x et y réels

Tâches souhaitées :

- Un constructeur spécifique avec paramètres
- Des méthodes d'accès pour x et y

Indicateurs :

1. Une classe s'appelant « Point » doit être créée
2. La visibilité doit être « public »
3. La classe doit avoir 2 champs
4. Les champs doivent être réels (float ou double)
5. La visibilité des champs doit être « private »
6. La classe doit avoir un constructeur spécifique à 2 arguments
7. Les arguments de ce constructeur spécifique doivent être réels (double ou float)
8. Les champs doivent avoir un getter
9. Les champs doivent avoir un setter

Question 2 :

Enoncé : Redéfinir la méthode equals pour la classe Point.

Doit faire : Redéfinir la méthode equals : *public boolean equals(Point)*

Tâches interdites :

- Un message binaire : equals(Point, Point)
- Avoir un paramètre de type Point
- Mettre une majuscule à equals
- Omettre le type de retour
- Utiliser un type de retour qui ne soit pas booléen

Indicateurs :

1. Une méthode appelée equals doit exister dans la classe Point
2. La classe Point doit donc exister
3. La méthode doit s'appeler equals et non Equals
4. La méthode equals avoir une visibilité « public »
5. La méthode equals doit avoir un type de retour booléen
6. La méthode equals doit avoir 1 paramètre
7. Le paramètre de la méthode equals doit être de type java.lang.Object

Question 3 :

Enoncé : Ajouter une méthode, que l'on appellera distance, permettant à un objet de la classe Point de calculer la distance qui le sépare d'un autre point.

Tâches obligatoires : Ecrire une méthode permettant de calculer la distance entre 2 points : *public float/double distance(Point)*

Tâches interdites :

- un message binaire : distance(Point, Point)
- Omettre le type de retour

Indicateurs :

1. Une méthode appelée distance doit exister dans la classe Point
2. La classe Point doit donc exister
3. La méthode doit s'appeler distance et non Distance
4. La méthode distance avoir une visibilité « public »
5. La méthode distance doit avoir un type de retour réel (float ou double)
6. La méthode distance doit avoir 1 paramètre
7. Le paramètre de la méthode distance doit être de type Point

Question 4 :

Enoncé : Un triangle étant défini par trois points. Ecrire la classe Triangle.

Tâches obligatoires : Créer une classe publique Triangle contenant trois champs privés de type Point

Tâches souhaitées :

- Un constructeur spécifique avec paramètres
- des méthodes d'accès aux champs

Tâches interdites :

- Ecrire la classe Triangle dans le même fichier que la classe Point
- La classe Triangle extends Point

Indicateurs :

1. Une classe s'appelant «Triangle» doit être créée
2. Sa visibilité doit être « public »
3. La classe doit avoir 3 champs
4. Les champs doivent être de type Point
5. La visibilité des champs doit être « private »
6. La classe doit avoir un constructeur spécifique à 3 arguments
7. Les arguments de ce constructeur spécifique doivent être de type Point
8. Les champs doivent avoir un getter
9. Les champs doivent avoir un setter
10. La classe Triangle ne doit pas hériter de la classe Point

Question 5 :

Enoncé : Ajouter une méthode, que l'on appellera perimetre, permettant à un objet de la classe Triangle de calculer son périmètre.

Tâches obligatoires : Ecrire une méthode permettant de calculer le perimetre d'un Triangle :
public float/double perimetre()

Tâches souhaitées :

- Utiliser la méthode distance écrite à la question 3

Tâches interdites :

- Ecrire la méthode avec un argument de type Triangle

Indicateurs :

1. Une méthode appelée perimetre doit exister dans la classe Triangle
2. La classe Triangle doit donc exister
3. La méthode doit s'appeler perimetre et non Perimetre
4. La méthode perimetre avoir une visibilité « public »
5. La méthode perimetre doit avoir un type de retour réel (float ou double)
6. La méthode perimetre doit avoir 0 paramètre
7. La méthode perimetre doit utiliser la méthode distance

Question 6 :

Enoncé : Ajouter une méthode, que l'on appellera surface, permettant de calculer la surface d'un triangle avec la formule $s = \sqrt{k * (k-a) * (k-b) * (k-c)}$ où $k = (a+b+c)/2$ avec a, b et c les longueurs des 3 côtés du triangle.

Tâches obligatoires : Ecrire une méthode permettant de calculer le surface d'un Triangle :
public float/double surface()

Tâches souhaitées :

- Utiliser la méthode distance écrite à la question 3

Tâches interdites :

- Ecrire la méthode avec un argument de type Triangle

Indicateurs :

1. Une méthode appelée surface doit exister dans la classe Triangle
2. La classe Triangle doit donc exister
3. La méthode doit s'appeler surface et non Surface
4. La méthode surface avoir une visibilité « public »
5. La méthode surface doit avoir un type de retour réel (float ou double)
6. La méthode surface doit avoir 0 paramètre
7. La méthode surface doit utiliser la méthode distance

Question 7 :

Enoncé : Ecrire la classe Droite, une droite étant définie par deux réels (a et b) représentant les coefficients de l'équation de droite $y=ax+b$.

Tâches obligatoires : Créer une classe publique Droite contenant deux champs privés a et b réels

Tâches souhaitées :

- Un constructeur spécifique avec paramètres
- Des méthodes d'accès à a et b

Indicateurs :

1. Un classe s'appelant «Droite» doit être créée
2. Sa visibilité doit être « *public* »
3. La classe doit avoir 2 champs
4. Les champs doivent être réels
5. La visibilité des champs doit être « *private* »
6. La classe doit avoir un constructeur spécifique à 2 arguments
7. Les arguments de ce constructeur spécifique doivent être réels
8. Les champs doivent avoir un getter
9. Les champs doivent avoir un setter

Question 8 :

Enoncé : Faire le nécessaire pour qu'il soit possible de créer une droite à partir de deux points (i.e la droite qui passe par ces deux points).

Tâches obligatoires : Ecrire un constructeur pour la classe Droite prenant deux objets de type Point en argument

Tâches interdites :

- Créer une méthode *creerDroite*

Indicateurs :

1. Un constructeur avec deux arguments de type Point doit exister dans la classe Droite
2. La classe Droite doit donc exister

Question 9 :

Enoncé : Ajouter, à la classe Point, une méthode que l'on appellera estSurDroite, permettant à un point de tester s'il se trouve sur une droite.

Tâches obligatoires : Ecrire une méthode permettant à un objet Point de tester s'il se trouve sur une droite :

```
public boolean estSurDroite(Droite)
```

Tâches interdites :

- Ecrire la méthode prenant en paramètre deux objets : un de type Point et un autre de type Droite

Indicateurs :

1. Une méthode appelée estSurDroite doit exister dans la classe Point
2. La classe Point doit donc exister
3. La méthode doit s'appeler estSurDroite et non EstSurDroite
4. La méthode estSurDroite avoir une visibilité « public »
5. La méthode estSurDroite doit avoir un type de retour booléen
6. La méthode estSurDroite doit avoir 1 paramètre
7. Le paramètre de la méthode estSurDroite doit être de type Droite.

Question 10 :

Enoncé : Ajouter, à la classe Droite, une méthode que l'on appellera estParallele, permettant à une droite de tester si elle est parallèle à une autre.

Tâches obligatoires : Ecrire une méthode permettant à une Droite de tester si elle est parallèle à une autre

```
public boolean estParallele(Droite)
```

Tâches interdites :

- Ecrire la méthode prenant en paramètre deux objets de type Droite

Indicateurs :

1. Une méthode appelée estParallele doit exister dans la classe Droite
2. La classe Droite doit donc exister
3. La méthode doit s'appeler estParallele et non EstParallele
4. La méthode estParallele avoir une visibilité « public »
5. La méthode estParallele doit avoir un type de retour booléen
6. La méthode estParallele doit avoir 1 paramètre
7. Le paramètre de la méthode estParallele doit être de type Droite

Question 11 :

Énoncé : Ajouter, à la classe Droite, une méthode que l'on appellera intersection, permettant à une droite de calculer son intersection avec une autre droite.

Tâches obligatoires : Créer une méthode permettant à une Droite de calculer son intersection (i.e. un Point) avec une autre droite : *public Point intersection(Droite)*

Tâches souhaitées :

- Gérer le cas des droites parallèles

Tâches interdites :

- Ecrire la méthode prenant en paramètre deux objets de type Droite
- Gérer le problème des droites parallèles avec un affichage

Indicateurs :

1. Une méthode appelée intersection doit exister dans la classe Droite
2. La classe Droite doit donc exister
3. La méthode doit s'appeler intersection et non Intersection
4. La méthode intersection avoir une visibilité « public »
5. La méthode intersection doit avoir un type de retour Point
6. La méthode intersection doit avoir 1 paramètre
7. Le paramètre de la méthode intersection doit être de type Droite
8. La méthode intersection doit faire appel à la méthode estParallele (écrite à la question 10)

Question 12 :

Énoncé : Faire le nécessaire pour qu'il soit possible de créer un Triangle à partir de trois droites.

Tâches obligatoires : Ecrire un constructeur pour la classe Triangle prenant trois objets de type Droite en argument

Tâches souhaitées :

- Gérer le cas des droites parallèles

Indicateurs :

1. Un constructeur avec trois arguments de type Droite doit exister dans la classe Triangle
2. La classe Triangle doit donc exister

Ce constructeur doit utiliser la méthode *estParallele* (écrite à la question 10)

Annexe B Règles de transformation

Cette partie présente les règles XSLT définies pour la transformation des fichiers XML générés par l'environnement d'édition graphique, afin de déployer les sessions d'apprentissage ouvertes sur le système Hop3x.

Les règles XSLT pour transformer le fichier des sessions :

```

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:mm_open_hop3x="http://lium.fr"
  xmlns:xsi="http://lium.fr" >

  <xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="no" indent="no"/>
  <xsl:template match="/">
    <xsl:apply-templates/>
  </xsl:template >

  <xsl:template match="mm_open_hop3x:Hop3x_sessions">
    <xsl:apply-templates/>
  </xsl:template >

  <xsl:template match="sessions">
    <SESSIONS>
      <xsl:apply-templates/>
    </SESSIONS>
  </xsl:template >
  <xsl:template match="variants_layer">
    <xsl:apply-templates/>
  </xsl:template >
  <xsl:template match="variants">
    <SESSION>
<xsl:attribute name="LANGAGE">
<xsl:value-of
select="ancestor::sessions/static_layer/constituet_elements[@langage]/@langage"/>
</xsl:attribute>
<xsl:attribute name="SCENARIO">
<xsl:value-of select="@id_variant"/><xsl:text>.xml</xsl:text>
</xsl:attribute>
<xsl:attribute name="NOM">
<xsl:value-of select="ancestor::sessions/@name_session"/><xsl:text>___</xsl:text>
<xsl:value-of select="@id_variant"/>
</xsl:attribute>
<xsl:attribute name="MODE">
<xsl:value-of
select="ancestor::sessions/static_layer/constituet_elements[@mode]/@mode"/>
</xsl:attribute>

```

```

<xsl:attribute name="DEBUT">
<xsl:value-of
select="ancestor::sessions/static_layer/constituet_elements[@debut]/@debut"/>
</xsl:attribute>
<xsl:attribute name="FIN">
<xsl:value-of select="ancestor::sessions/static_layer/constituet_elements[@fin]/@fin"/>
</xsl:attribute>
<xsl:attribute name="CONFIRMER">
<xsl:choose>
<xsl:when test="ancestor::sessions/static_layer/constituet_elements[@confirmer='true']">
<xsl:text>>true</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:text>>false</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:attribute>

<!-- Création des groupes -->
<xsl:apply-templates select="according_to/context_elements/groups"/>

<!-- Création des étudiants -->
<xsl:apply-templates select="according_to/context_elements/learners"/>

<!-- Création des enseignants -->
<xsl:apply-templates select="ancestor::sessions/static_layer/constituet_elements/tutors"
mode="TUTOR"/>

<HOPHUI>HopHui-<xsl:text><xsl:value-of select="@id_variant"/></xsl:text> </HOPHUI>

</SESSION>

</xsl:template>
<xsl:template match="according_to/context_elements/groups">
<GROUPE>
<xsl:value-of select="@group_name"/>
</GROUPE>
</xsl:template>

<xsl:template match="according_to/context_elements/learners">
<ETUDIANT>
<xsl:value-of select="@id_learner"/>
</ETUDIANT>
</xsl:template>

<xsl:template match="static_layer/constituet_elements/tutors" mode="TUTOR">
<ENSEIGNANT>
<xsl:value-of select="@id_tutor"/>
</ENSEIGNANT>
</xsl:template>

</xsl:stylesheet>

```

Les règles XSLT pour transformer le fichier des variantes :

```

<?xml version="1.0" encoding="iso-8859-1" ?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:mm_open_hop3x="http://lium.fr" >

<xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="no"
indent="no"/>
  <xsl:template match="/">
    <xsl:apply-templates/>
  </xsl:template >
  <xsl:template match="mm_open_hop3x:Variants_Layer">
    <xsl:apply-templates/>
  </xsl:template >

  <xsl:template match="variants">
    <SUJET>
      <!-- Création des consignes -->
      <xsl:apply-templates
select="variant_elements/instructions_to_respect"/>

      <!-- Création des questions -->
      <xsl:apply-templates
select="variant_elements/questions"/>

    </SUJET>
  </xsl:template>

  <xsl:template
match="variant_elements/instructions_to_respect">
    <CONSIGNES>
      <xsl:value-of select="@content"/>
    </CONSIGNES>
  </xsl:template>

  <xsl:template match="variant_elements/questions">
    <QUESTION>
      <xsl:attribute name="NUMBER">
        <xsl:value-of select="@nb_Q"/>
      </xsl:attribute>
      <ENONCE>
        <xsl:value-of select="@text_Q"/>
      </ENONCE>
    </QUESTION>

  </xsl:template>

</xsl:stylesheet>

```

Les règles XSLT pour transformer le fichier des apprenants :

```
<xsl:stylesheet version="1.0"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >

<xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="no"
indent="no"/>
  <xsl:template match="/">
    <xsl:apply-templates/>
  </xsl:template >

  <xsl:template match="mm_open_hop3x">
    <xsl:apply-templates/>
  </xsl:template >
  <xsl:template match="Learners">
    <ETUDIANTS>
      <!-- Création des Etudiants -->
      <xsl:apply-templates select="learners"/>
    </ETUDIANTS>
  </xsl:template>

  <xsl:template match="learners">
    <ETUDIANT>
      <!-- Création des attributs noms et prénoms des
étudiants -->

      <xsl:attribute name="NOM">
        <xsl:value-of select="@learner_name"/>
      </xsl:attribute>

      <xsl:attribute name="PRENOM">
        <xsl:value-of select="@learner_firstname"/>
      </xsl:attribute>

      <!-- Création des logins et mots de passe -->

      <UTILISATEUR>
        <xsl:value-of select="@id_learner"/>
      </UTILISATEUR>
      <MDP>
        <xsl:value-of select="@pWD_learner"/>
      </MDP>
    </ETUDIANT>

  </xsl:template>

</xsl:stylesheet>
```

Les règles XSLT pour transformer le fichier des groupes :

```
<xsl:stylesheet version="1.0"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                >
<xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="no"
indent="no"/>
  <xsl:template match="/">
    <xsl:apply-templates/>
  </xsl:template >

  <xsl:template match="mm_open_hop3x">
    <xsl:apply-templates/>
  </xsl:template >

  <xsl:template match="Groups">
    <GROUPEES>
      <!-- Création des groupes -->
      <xsl:apply-templates select="groups"/>
    </GROUPEES>
  </xsl:template>

  <xsl:template match="groups">
    <GROUPE>
      <xsl:attribute name="NOM">
        <xsl:value-of select="@group_name"/>
      </xsl:attribute>

      <!-- Création des utilisateurs -->
      <xsl:apply-templates
select="composed_of_learners"/>
    </GROUPE>

  </xsl:template>

  <xsl:template match="composed_of_learners">
    <UTILISATEUR>
      <xsl:value-of select="@id_learner"/>
    </UTILISATEUR>

  </xsl:template>

</xsl:stylesheet>
```

Les règles XSLT pour transformer le fichier des tuteurs :

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >

<xsl:output method="xml" encoding="UTF-8" omit-xml-declaration="no"
indent="no"/>
  <xsl:template match="/">
    <xsl:apply-templates/>
  </xsl:template >

  <xsl:template match="mm_open_hop3x">
    <xsl:apply-templates/>
  </xsl:template >

  <xsl:template match="Tuteurs">
    <ENSEIGNANTS>
      <!-- Création des Enseignants -->
      <xsl:apply-templates select="tuteurs"/>
    </ENSEIGNANTS>
  </xsl:template>

  <xsl:template match="tuteurs">
    <ENSEIGNANT>
      <!-- Création des attributs noms et prénoms des
enseignants -->

      <xsl:attribute name="NOM">
        <xsl:value-of select="@tutor_name"/>
      </xsl:attribute>
      <xsl:attribute name="PRENOM">
        <xsl:value-of select="@tutor_firstname"/>
      </xsl:attribute>

      <!-- Création des logins et mots de passe -->

      <UTILISATEUR>
        <xsl:value-of select="@id_tutor"/>
      </UTILISATEUR>

      <MDP>
        <xsl:value-of select="@pWD_tutor"/>
      </MDP>

    </ENSEIGNANT>

  </xsl:template>
</xsl:stylesheet>
```