



HAL
open science

Approche Domain-Specific Modeling pour l'opérationnalisation des scénarios pédagogiques sur les plateformes de formation à distance

Aymen Abedmouleh

► **To cite this version:**

Aymen Abedmouleh. Approche Domain-Specific Modeling pour l'opérationnalisation des scénarios pédagogiques sur les plateformes de formation à distance. Autre [cs.OH]. Université du Maine, 2013. Français. NNT : 2013LEMA1028 . tel-01019917

HAL Id: tel-01019917

<https://theses.hal.science/tel-01019917v1>

Submitted on 7 Jul 2014

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

**Approche *Domain-Specific Modeling* pour
l'opérationnalisation des scénarios pédagogiques
sur les plateformes de formation à distance**

Par

Aymen ABEDMOULEH

Soutenu le 11 juillet 2013 devant le jury composé de

Rapporteurs :

Mme Monique GRANDBASTIEN
M. Henri BASSON

Professeur des Universités, Université de Lorraine
Professeur des Universités, Université de Lille nord

Examineurs :

M. Sébastien GEORGE
Mme Dalila TAMZALIT

Maître de conférences HDR, INSA de Lyon
Maître de Conférences, Université de Nantes

Directeur :

M. Christophe CHOQUET

Professeur des Universités, Université du Maine

Co-Encadrant :

M. Lahcen OUBAHSSI
M. Pierre LAFORCADE

Maître de Conférences, Université du Maine
Maître de Conférences, Université du Maine

Année 2013

Numéro d'Ordre :

Remerciements

Je tiens à remercier Mme Monique GRANDBASTIEN, Professeur des universités à l'université de Lorraine et M. Henri BASSON, Professeur des universités à l'université de Lille nord, de m'avoir fait l'honneur d'être rapporteurs de la thèse.

Je tiens à remercier M. Sébastien GEORGE, Maître de conférences Habilité à Diriger des Recherches à l'INSA Lyon, de m'avoir fait l'honneur d'être examinateur de la thèse, et Mme Dalila TAMZALIT Maître de conférences à l'université de Nantes, de m'avoir fait l'honneur d'être examinatrice de la 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. Lahcen OUBAHSSI, 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 l'encouragement qu'il m'a fourni pendant toute la thèse.

Je tiens à remercier aussi M. Pierre LAFORCADE, Maître de conférences à l'université du Maine qui a assuré également 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 son encouragement pendant toute la thèse.

J'adresse également tous mes remerciements à tous les membres de l'équipe IEIAH 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 : les enseignants de l'IUT de Laval, en particulier les 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 chers amis de Laval (Nouri, Redha, Riadh, etc.), mes chers amis en Tunisie (Maher, Mohamed, Karim, etc.) 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 toute ma famille et mes amis.

Aymen

Résumé

Ce travail de recherche s'inscrit dans le cadre des travaux de l'équipe IEIAH (Ingénierie des Environnements Informatique pour l'apprentissage Humain) dans une perspective d'ingénierie et de réingénierie des plateformes de formation à distance ou LMS (*Learning Management System*).

La problématique étudiée au sein de cette thèse concerne la spécification et l'opérationnalisation des scénarios pédagogiques sur les dispositifs d'apprentissage. Contrairement aux approches traditionnelles de conception indépendant des plateformes, notre approche est centrée plateformes/LMSs. Nous posons l'hypothèse que les plateformes disposent de leur propre paradigme de conception. Pour cela, nous proposons aux développeurs et à la communauté de développement des plateformes un processus spécifique d'identification et d'explicitation du langage de conception pédagogique des plateformes de formation. L'application de ce processus permet la formalisation de ce langage par le biais d'un métamodèle.

Dans le cadre de nos travaux, nous exploitons l'approche outillée *Domain-Specific Modeling* (DSM) comme cadre pratique pour le développement de langages et d'outils de conception graphiques basés sur le métamodèle du langage de conception pédagogique. Afin de répondre à des besoins récurrents d'opérationnalisation de scénarios pédagogiques, nous utilisons également ce métamodèle pour la spécification des modules d'import/export que nous intégrons au sein des plateformes. Ces modules sont capables d'opérationnaliser les scénarios conformes au langage identifié. Ils permettent également d'exporter les scénarios existants sur les plateformes afin de permettre leur adaptation par les outils développés. Nous avons expérimenté nos propositions théoriques sur deux plateformes de formation Moodle et Ganesha qui sont fréquemment utilisés au sein des établissements éducatifs. Enfin, nous avons vérifié la cohérence de notre approche par la mise à l'essai des outils développés. Par le biais de notre approche, nous garantissons une opérationnalisation de ces scénarios sans pertes d'information ni de sémantique.

Mots clés : Scénario pédagogique, Opérationnalisation, Plateformes de formation, langage de conception pédagogique, Domain Specific Modeling.

Sommaire

1. Introduction	13
1.1. Contexte initial	14
1.2. Positionnement des travaux dans le contexte de l'équipe IEIAH	16
1.3. Aperçu de la problématique	18
1.4. Aperçu de nos contributions.....	19
1.5. Organisation du document	21
I. Première partie : positionnement scientifique et problématique de recherche	23
2. Etat de l'art et positionnement scientifique	25
2.1. La conception et la scénarisation pédagogique	26
2.1.1. La conception pédagogique.....	26
2.1.2. La scénarisation et les scénarios pédagogiques.....	27
2.1.3. Processus de scénarisation ou de conception d'un scénario pédagogique	28
2.1.4. Les deux principales approches de conception d'un scénario pédagogique	30
2.1.5. Les langages de modélisation pédagogique	31
2.1.6. Conclusion.....	34
2.2. L'Ingénierie Dirigée par les Modèles.....	34
2.2.1. L'architecture dirigée par les modèles (MDA)	35
2.2.2. Les niveaux d'abstraction dans l'IDM	35
2.2.3. Les types de modèles.....	36
2.2.4. Les transformations entre les modèles	36
2.2.5. La modélisation spécifique au domaine (DSM).....	37
2.3. Travaux menés sur la conception de scénarios pédagogiques.....	38
2.3.1. Les approches standards.....	39
2.3.2. Approches centrées sur les langages de modélisation pédagogique.....	41
2.3.3. Approches par l'Ingénierie Dirigée par les Modèles.....	46

2.4.	Les plateformes de formations à distance.....	50
2.4.1.	Définitions des plateformes de formation à distance.....	50
2.4.2.	Exemples de plateformes de formation	51
2.4.3.	La conception pédagogique sur les plateformes de formation	52
2.4.4.	Difficultés de la conception pédagogique sur les plateformes	53
2.5.	Travaux menés sur l'opérationnalisation des scénarios pédagogiques.....	53
2.5.1.	Introduction et définitions	54
2.5.2.	Approches des chaînes éditoriales	55
2.5.3.	Approches propriétaires par les EMLs	56
2.5.4.	Approches propriétaires par les plateformes	57
2.5.5.	Approches propriétaires par l'IDM	59
2.5.6.	Discussion globale.....	61
3.	Problématique scientifique et approche	63
3.1.	Synthèse des travaux existants et problématique scientifique.....	64
3.2.	Approche de conception centrée plateformes.....	67
3.2.1.	Identification et formalisation du langage de conception pédagogique.....	68
3.2.2.	Exposer le langage des plateformes pour la spécification de VIDLs et d'éditeurs graphiques.....	69
3.2.3.	Opérationnaliser les scénarios pédagogiques par des modules d'import/export	70
3.2.4.	Conception itérative des scénarios.....	71
3.3.	Méthodologie de recherche	72
II.	Deuxième Partie : contributions	75
4.	Processus d'analyse, d'identification et de formalisation du métier.....	77
4.1.	Vue générale du processus	78
4.1.1.	Vue générale de l'analyse centrée IHM.....	79
4.1.2.	Vue générale de l'analyse technique	81
4.1.3.	Vue générale de la confrontation des modèles et de la formalisation du modèle final..	81
4.2.	Analyse centrée IHM.....	81
4.2.1.	Analyse IHM-macro	81

4.2.2.	Analyse fonctionnelle.....	85
4.2.3.	Analyse IHM-micro	90
4.2.4.	Composition du modèle IHM.....	96
4.3.	Analyse centrée technique.....	99
4.3.1.	Parcours des tables	101
4.3.2.	Identification des tables/champs relevant de la conception pédagogique	101
4.3.3.	Spécification du modèle logique de données réduit de la base de données	102
4.3.4.	Spécification du modèle MCD réduit.....	102
4.4.	Confrontation et formalisation du langage de conception pédagogique final	103
4.4.1.	Intérêt de la confrontation et de la formalisation.....	104
4.4.2.	Formalisme du modèle final.....	105
4.4.3.	Description du processus de confrontation.....	106
4.4.4.	Processus de vérifications	108
4.5.	Conclusion.....	114
5.	Instrumentation du processus sur les plateformes d'étude	115
5.1.	Introduction.....	116
5.2.	Cas de la plateforme Moodle	116
5.2.1.	Introduction	116
5.2.2.	Application de l'analyse centrée IHM sur Moodle	116
5.2.3.	Application de l'analyse technique sur Moodle	122
5.2.4.	Confrontation et formalisation	123
5.3.	Cas de la plateforme Ganesha	125
5.3.1.	Introduction	125
5.3.2.	Application de l'analyse centrée IHM sur Ganesha	125
5.3.3.	Application de l'analyse technique sur Ganesha.....	129
5.3.4.	Confrontation et formalisation	130
5.4.	Conclusion.....	131
6.	Développement des outils et mise à l'essai	133
6.1.	Développement du service d'opérationnalisation	134

6.1.1.	Introduction	134
6.1.2.	Cas de la plateforme Moodle	134
6.1.3.	Cas de la plateforme Ganesha	141
6.1.4.	Conclusion	144
6.2.	Développement des outils de conception	144
6.2.1.	Présentation de l'outillage : Eclipse Modeling Project	144
6.2.2.	Mooditor : éditeur graphique pour Moodle	149
6.3.	Mise à l'essai	159
6.3.1.	Description du scénario pédagogique	159
6.3.2.	Mise à l'essai : cas de la plateforme Moodle	160
6.3.3.	Mise à l'essai : cas de la plateforme Ganesha	165
6.4.	Conclusion	169
III. Troisième partie : conclusion générale		171
7. Conclusions et perspectives		173
7.1.	Bilan des travaux réalisés	174
7.2.	Analyse critique des travaux réalisés	175
7.2.1.	Apports de nos propositions	175
7.2.2.	Limites de notre travail	176
7.3.	Perspectives	178
7.3.1.	Exploitation du métier identifié pour des VIDLs centrés LMSs et praticiens et pour l'interopérabilité entre les plateformes	178
7.3.2.	Extension du métier identifié par le processus	180
Table des figures		182
Références bibliographiques		185
Annexe A : Application du processus sur la plateforme Moodle		203
Annexe B : Application du processus sur la plateforme Ganesha		261
Annexe C : Procédure d'import/export de Moodle et Ganesha		291

1

Introduction

Sommaire

- 1.1. Contexte initial
- 1.2. Positionnement des travaux dans le contexte de l'équipe IEIAH
- 1.3. Aperçu de la problématique
- 1.4. Aperçu de nos contributions
- 1.5. Organisation du document

Ce chapitre expose les principaux éléments nécessaires à la compréhension de ce travail. Il s'agit ici de donner un aperçu global du contenu de la thèse. On y trouvera une présentation du contexte de recherche de cette thèse (Ingénierie des EIAH), une première problématisation de notre étude ainsi que nos contributions, nos propositions et les travaux réalisés à la fin de ce chapitre. Certaines sections de ce chapitre seront reprises en détail dans la suite de ce document.

1.1. Contexte initial

Cette thèse s'inscrit dans le domaine de l'ingénierie et de la réingénierie des Environnements Informatiques pour l'Apprentissage Humain (EIAH). Un EIAH est la combinaison d'un système informatique et d'un système d'apprentissage. Selon [Tchounikine et al. 2004], un EIAH représente un environnement informatique proposant une intention didactique et/ou pédagogique et dont la finalité est de susciter ou d'accompagner un apprentissage. En tant qu'environnement informatique, un EIAH offre des services spécifiques dédiés au domaine de l'éducation pour ses multiples clients et usagers (des universités, des académiques, des enseignants, des apprenants, des gestionnaires, etc.). Les services offerts incluent généralement un contrôle d'accès, des outils de communication (synchrones et/ou asynchrones), l'administration des groupes d'utilisateurs, etc. En tant qu'environnement d'apprentissage, un EIAH est proposé pour toute personne impliquée dans un processus d'apprentissage ou dans un parcours pédagogique. Il embarque une intention didactique/pédagogique et des éléments métiers relevant de la conception pédagogique. Durant les dernières décades, plusieurs systèmes informatiques spécifiques aux EIAH sont apparus tels que les plateformes d'apprentissage en ligne, les plateformes e-learning (ou de la FOAD), les systèmes de gestion de l'apprentissage, les environnements numérique d'apprentissage, etc.

Actuellement, les plateformes de Formation Ouverte et A Distance (FOAD) sont largement utilisées et ne sont pas limitées à l'apprentissage distant. La plupart des organisations académiques comme les établissements scolaires et universitaires fournissent à leurs enseignants des plateformes dont l'objectif est d'améliorer ou de compléter leurs cours présentiels par des activités supplémentaires comme le simple accès à des ressources ou à des évaluations en ligne. Dans notre travail, nous nous focalisons particulièrement sur les plateformes de formation à distance, encore appelées systèmes LMS (*Learning Management System*). Ces dispositifs ont fait l'objet de plusieurs travaux de recherche et de développement. Plusieurs définitions ont été attribuées aux plateformes de formation à distance. Par exemple, [Oubahssi 2005] définit une plateforme de formation à distance comme étant "*un logiciel de création et de gestion de contenus pédagogiques et de supports aux acteurs destiné à trois types d'utilisateurs : l'enseignant, l'apprenant et l'administrateur. Elle regroupe les outils nécessaires aux trois types d'intervenants permettant d'incorporer des ressources pédagogiques multimédias, de participer à des activités et d'effectuer le suivi pédagogique et administratif des apprenants*".

Les plateformes représentent un environnement d'apprentissage permettant le déploiement et l'exécution des scénarios pédagogiques. Les scénarios pédagogiques sont mis en œuvre par les enseignants concepteurs avant d'être exécutés par les usagers finaux qui sont généralement les apprenants. Le scénario pédagogique est un élément central de la conception pédagogique (ou le design pédagogique - *Learning Design*) qui représente une activité pour la conception des unités, des activités ou des environnements d'apprentissage. [Paquette 2002] définit la conception pédagogique (*Instructional Design*) comme "*l'ensemble des théories et des modèles et des modèles permettant de comprendre, d'améliorer et d'appliquer des méthodes d'enseignement favorisant l'apprentissage*". [Berger et Kam 1996] la définissent comme suit : "*Instructional Design is the systematic development of instructional specifications using learning and instructional theory to ensure the quality of instruction. It is the entire process of analysis of learning needs and goals and the development of a delivery system to meet*

those needs. It includes development of instructional materials and activities; and tryout and evaluation of all instruction and learner activities”.

Plusieurs approches ont été définies pour supporter la conception pédagogique telles que les approches centrées sur les connaissances, ou sur les ressources (appelée aussi l’approche documentaire), ou sur les interactions, ou sur les activités (appelée aussi l’approche EML), etc. Les EMLs (*Educational Modeling Language*) ont défini les éléments caractéristiques pour la spécification d’une situation d’apprentissage. Ce courant est apparu avec les travaux réalisés à l’*Open University of the Netherland* sur la conception et le développement d’un langage de description adapté à l’éducation et nommé EML. D’après [Rawlings et al. 2002] *"An EML is a semantic information model and binding, describing the content and process within a 'unit of learning from a pedagogical perspective in order to support reuse and interoperability"*. L’objectif principal des EMLs est de guider la spécification de scénarios pédagogiques.

Notre travail se situe dans le domaine de la scénarisation pédagogique. Plusieurs travaux de recherche se sont intéressés à la définition des scénarios pédagogiques en tant que produit résultant de l’activité de scénarisation ou de conception pédagogique. [Charlier et Daele 2002] définissent un scénario pédagogique comme *"le résultat du processus de conception d’une activité d’apprentissage, processus s’inscrivant dans un temps donné et aboutissant à la mise en œuvre du scénario. Donc un scénario comprend des objectifs, une planification des activités d’apprentissage, un horaire, une description des activités des étudiants, des modalités d’évaluation qui sont définis, agencés et organisés au cours d’un processus de design"*. Cette définition considère le scénario pédagogique comme un agencement d’activités et d’évaluations prévu pour des étudiants ayant un objectif didactique/pédagogique déterminé et un horaire bien défini. Ainsi, elle met l’accent sur les concepts caractéristiques d’un scénario pédagogique.

Depuis quelques années, certains travaux s’intéressent à la mise en œuvre des scénarios sur les plateformes de formations à distance. Cette mise en œuvre, appelée aussi opérationnalisation, peut être décrite par la projection des concepts de scénarios selon les services caractéristiques de la plateforme. Elle est souvent traduite par la configuration manuelle des interfaces de la plateforme en suivant le scénario *a priori* décrit par un langage de modélisation ou un standard. Ce point a été et reste encore le centre de plusieurs travaux de recherche.

La recherche scientifique et plus spécifiquement la recherche en EIAHs stimule toujours des travaux d’améliorations et d’évolution pour les approches existantes ainsi que pour les outils utilisés. Ces besoins sont dus à la non satisfaction des besoins initiaux des usagers, au manque d’interprétation/implémentation des services attendus par les usagers finaux et/ou au besoin des évolutions suite à l’avancement de la recherche et des technologies. Ces travaux interviennent dans un contexte de réingénierie. [Chikowsky et Cross 1990] 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"*. [Choquet 2007] définit la réingénierie pédagogique d’un EIAH comme étant l’examen d’une situation pédagogique médiatisée et la modification du dispositif d’apprentissage afin de reconstituer ce dernier sous une nouvelle forme et mettre ainsi en place une nouvelle situation pédagogique, prenant mieux en compte les usages.

Dans le cadre de nos travaux de recherche, nous nous sommes intéressés aux activités de scénarisation et d'opérationnalisation. Cette thèse s'inscrit dans le contexte d'une approche *Domain Specific Modeling* pour l'opérationnalisation des scénarios pédagogiques sur les plateformes de formation à distance. Notre objectif est d'instrumenter la conception et la mise en œuvre des scénarios pédagogiques spécifiés par les enseignants-concepteurs.

1.2. Positionnement des travaux dans le contexte de l'équipe IEIAH

Le projet IEIAH (Ingénierie des Environnements Informatiques pour l'Apprentissage Humain) est mené au sein du laboratoire LIUM (Laboratoire Informatique de l'Université du Maine). Il se situe dans un contexte d'ingénierie et de réingénierie des EIAHs en adoptant l'approche d'Ingénierie Dirigée par les Modèles (IDM) [Jézéquel et al. 2012]. Les modèles manipulés dans ce projet sont les scénarios pédagogiques. Le projet IEIAH s'intéresse à toutes les activités de conception pédagogique qui couvrent le cycle de vie du scénario pédagogique (conception, opérationnalisation, exécution, observation et analyse, et réingénierie du scénario). Les notions de scénarios prédictifs et scénarios descriptifs, proposés par [Pernin et Lejeune 2004], sont adoptées. Le scénario prédictif représente la description formelle de la situation d'apprentissage établie *a priori* par l'enseignant-concepteur. Le scénario descriptif décrit *a posteriori* le déroulement de la situation d'apprentissage afin d'apporter des améliorations au scénario prédictif.

L'approche IDM est basée sur l'utilisation des métamodèles et des transformations de modèles. Un scénario est généralement conforme à un métamodèle qui décrit un langage de conception spécifique (métamodèle d'un EML particulier, ou d'une plateforme, etc.).

Dans un contexte d'environnement informatique pour l'apprentissage humain, nous distinguons deux catégories de réingénierie.

- La première catégorie concerne la réingénierie des EIAH en tant que systèmes d'apprentissage. Elle s'intéresse particulièrement aux plateformes de formation à distance, les systèmes de type LMS (*Learning Management System*), etc. en termes d'usage, de fonctionnalités, d'ergonomie, etc.
- La deuxième catégorie concerne un élément central de la conception pédagogique : c'est le scénario pédagogique. La réingénierie du scénario pédagogique considère la conception des situations d'apprentissage à partir de plusieurs points de vue : qualitatif, concepteurs (enseignants, praticiens, etc.), langage de modélisation, observation, support de déploiement, etc. Elle s'intéresse à amener des améliorations éventuelles au scénario initial (prédictif) en utilisant les nouvelles informations issues du scénario descriptif.

Le projet REDiM (Réingénierie des EIAH Dirigée par les Modèles), à l'origine du nouveau projet IEIAH, a porté principalement sur la deuxième catégorie de la réingénierie. Le projet actuel IEIAH s'adresse aux deux catégories de réingénierie. Différents travaux ont été réalisés au sein du projet REDiM pour aborder différents axes de recherche. Nous donnons un historique de ces travaux au sein de cette équipe.

Les premiers travaux de l'équipe REDiM ont porté sur la définition du processus de réingénierie d'un EIAH. Les problèmes traités concernent la modélisation du cadre de travail des acteurs de la conception et du développement d'un EIAH en termes de processus, d'interactions, de rôles et de tâches. [Corbière et Choquet 2005], [Corbière 2006] et [Choquet et Corbière 2006] ont proposé le modèle métier de la réingénierie d'un EIAH. Ensuite, les travaux de l'équipe se sont intéressés à des parties particulières du processus de réingénierie. Trois travaux de recherche ont été démarrés en parallèle.

1. [El-Kechaï 2008] et [Choquet 2007] se sont focalisés sur l'activité de modélisation d'une situation d'apprentissage. Au cours de leurs travaux, Choquet et El-Kechaï ont favorisé la modélisation constructive des scénarios pédagogiques parmi les approches possibles (constructive et interprétative). Cette approche permet aux enseignants-concepteurs de définir collectivement les modèles métiers qui serviront à la modélisation d'une situation d'apprentissage.

2. [Abdallah 2009] s'est intéressé à la spécification de situations d'apprentissage dans un contexte de Pédagogie par Projet Collectif (PPC) en milieu universitaire. La PPC est une méthode de la pédagogie active "regroupant" les deux méthodes actives : Pédagogie par Projet (PP) et Apprentissage Collectif (AC). Abdallah a proposé un processus de conception et de médiatisation de situations d'apprentissage dans un contexte de pédagogie par projet collectif en mettant l'enseignant au cœur du travail à toutes les étapes du cycle de vie de développement du logiciel.

3. Les travaux de [Oubahssi et al. 2010] et [Laforcade 2010] ont porté sur la réingénierie du Livret Electronique d'Apprentissage (LÉA) dans le but de faciliter son utilisation. Le projet LÉA¹ [Cottier et al. 2008] vise à effectuer le suivi de l'avancement des apprenants pendant leurs formations en entreprise et au CFA (Centre de Formation des Apprentis). La réingénierie du système LÉA a porté sur l'élaboration du service de gestion de multi-rôle ainsi que la spécification d'un éditeur externe pour configurer graphiquement le livret au niveau institutionnel. Le développement est guidé par les modèles au sens de l'IDM, en appliquant une approche outillée du DSM (*Domain-Specific Modeling*). La communication entre l'éditeur externe et le système LÉA est assurée par un module d'import/export des configurations (fichiers XML), ajouté au système existant.

4. Les travaux de [Randriamalaka 2008] ont porté sur une approche de réingénierie pour une assistance à la modélisation et à l'analyse des traces d'usage dans un EIAH. L'objectif de ce travail était de proposer la modélisation et l'instrumentation d'un processus d'élicitation des indicateurs dans le contexte de la réingénierie des scénarios pédagogiques. Ses propositions scientifiques ont été basées sur l'utilisation d'UTL (*Usage Tracking Language*) [Iksal et al. 2010] qui permet à un enseignant d'exprimer ses besoins d'observation sous une forme textuelle.

Ensuite, les travaux de l'équipe se sont centrés plus spécifiquement sur des problématiques plus fines.

5. [Zendagui 2011] a proposé une approche dirigée par les modèles pour la spécification des besoins d'observation dans un contexte de réingénierie. Les observations sont effectuées dans le double but de

¹ Le projet LEA est un projet de recherche ANR, résultat d'un partenariat qui rassemble le LIUM (Laboratoire d'Informatique de l'Université du Maine) et le CFAVM (Centre de Formation des Apprentis des trois Villes de la Mayenne), subventionné par le ministère de la recherche, direction de la technologie.

vérifier la cohérence de l'utilisation effective avec celle qui était prévue et de pouvoir détecter des utilisations émergentes afin d'améliorer la qualité du système.

6. Les travaux de [Pham Thi Ngoc 2011] ont porté sur la spécification et la conception de services d'analyse de l'utilisation d'un EIAH. L'objectif de ce travail est de modéliser le calcul d'indicateurs et de proposer un langage de spécification outillé autour du langage UTL. Pham Thi Ngoc a proposé ainsi le langage DCL4UTL et son outil support comme instruments pour le calcul des indicateurs pédagogiques durant et/ou après les sessions d'apprentissage.

7. [Ouraiba 2012] a proposé une approche dirigée par les modèles spécifique au domaine métier pour la scénarisation pédagogique au sein des EIAH ouverts. Cette approche vise à rendre un EIAH ouvert aux enseignants afin de leur permettre de poursuivre le processus de conception pendant l'usage.

Dans le contexte des travaux de l'équipe IEIAH, notre thèse ne concerne pas autant de concept, elle n'en retient qu'une partie. Elle se focalise sur les deux thématiques de réingénierie évoquées par le projet IEIAH. Nous considérons la conception des scénarios pédagogiques sur les EIAHs, en particulier les plateformes de formation à distance et les systèmes LMSs. Nous abordons la problématique d'opérationnalisation des scénarios pédagogiques sur les plateformes. Ce travail de thèse évoque une suite aux travaux antécédents de l'équipe ([El-Kechai 2008], [Abdallah 2009], [Oubahssi et al. 2010] et [Ouraiba 2012]) en proposant une nouvelle approche de mise en œuvre des situations d'apprentissage ou de scénarios pédagogiques. Notre objectif est de faciliter leur mise en œuvre lorsqu'ils sont spécifiés en dehors des plateformes. Nous posons l'hypothèse que les plateformes de formation embarquent un langage de conception qui leur est propre. La conception des scénarios pédagogiques peut être plus aisée aux enseignants-concepteurs en leur proposant des outils de conception proches de leurs pratiques et conformes au langage de conception de la plateforme utilisée.

1.3. Aperçu de la problématique

La scénarisation pédagogique est un domaine qui trouve ses sources dans le design pédagogique dont l'origine remonte à John Dewey [Dewey 1900]. Au cours d'un peu plus d'un siècle, plusieurs théories d'apprentissage et des pratiques éducatives sont apparues. Egalement, plusieurs langages de modélisation pédagogique (EMLs) ont été proposés, ils ont fait l'objet de plusieurs travaux de recherche. L'arrivée des nouvelles technologies a favorisé l'apprentissage à distance par le moyen de nouveaux systèmes d'apprentissage de type LMS ou via des plateformes de formation à distance.

Nous montrons dans le chapitre 3 qu'il n'existe pas encore une forte corrélation entre ces langages de modélisation et les systèmes d'apprentissage. Ceci implique que la conception des scénarios pédagogiques par les EMLs est indépendante des systèmes d'exécution.

Le premier verrou principal de ce travail de recherche est le suivant. En tenant compte qu'un système d'apprentissage est un système informatique avec des finalités didactiques, quelles données/informations sont utiles pour la conception pédagogique d'un scénario ou d'un cours ? Malgré la distinction de ces informations via des modules destinés pour des catégories d'utilisateurs différentes, le module d'enseignant créateur de cours par exemple, embarque des données techniques de bas niveaux.

Ces données sont difficilement interprétées par les enseignants et parfois non compréhensibles. Elles compliquent la spécification des scénarios sur ces systèmes. La question que nous posons est :

Comment faciliter la spécification des scénarios pédagogiques sur les plateformes de formation à distance ?

Le second verrou principal de ce travail de recherche est le suivant. Plusieurs scénarios pédagogiques sont spécifiés en adoptant une approche de scénarisation ou bien un langage de modélisation pédagogique. Ces travaux font l'objet du chapitre 2. A titre d'exemple, [Caron 2007b] souligne que les scénarios spécifiés en dehors des LMS sont difficilement implémentés. Il propose d'utiliser l'Ingénierie Dirigée par les Modèles pour appliquer des transformations des modèles. Ces transformations produisent un modèle informatiquement opérationnel mais elles causent plusieurs pertes d'informations. Les deux questions que nous posons sont :

- Comment faciliter l'opérationnalisation des scénarios pédagogiques sur les plateformes de formation à distance ?
- Comment garantir l'opérationnalisation des scénarios pédagogiques sur les plateformes de formation sans pertes d'informations ni de sémantique ?

Une première problématisation de nos recherches peut donc se résumer comme suit : dans un premier temps, nous souhaitons faciliter la spécification des scénarios pédagogiques sur les plateformes de formation. Nous proposons de tenir compte du métier ou du langage de conception pédagogique des plateformes afin de préserver la sémantique du scénario. Une partie de ce langage est utile pour la spécification ou bien pour la mise en œuvre de ces scénarios. Cette partie doit être identifiée pour spécifier des scénarios en conformité avec elle. Dans un second temps, nous souhaitons proposer des moyens (1) pour faciliter la mise en œuvre des scénarios pédagogiques sur les plateformes de formation et (2) pour garantir leur opérationnalisation sans pertes d'informations ni de sémantique.

1.4. Aperçu de nos contributions

Les contributions de notre travail de recherche reposent sur une nouvelle approche de conception basée sur le langage de conception pédagogique embarqué dans les plateformes de formation. Cette approche s'appuie sur un processus spécifique pour l'analyse, l'identification et la formalisation de ce langage. Le langage identifié sera utilisé comme base pour le développement de nouveaux outils de conception pédagogique. Egalement, il représente un format de communication entre ces outils et les plateformes concernées. Nous avons développé un éditeur graphique externe pour la spécification de scénarios en dehors des plateformes. Nous avons ajouté de nouvelles fonctionnalités aux plateformes afin de les rendre capable d'importer et d'exporter les scénarios. Nos principales contributions sont les suivantes :

➤ **Un processus d'analyse, d'identification et de formalisation du métier pédagogique d'une plateforme de formation à distance**

Le processus d'analyse, d'identification et de formalisation du métier de conception pédagogique d'une plateforme a été élaboré suite à plusieurs expérimentations et analyses menées sur deux plateformes principales (Moodle et Ganesha). Ces travaux ont été réalisés dans le but de proposer un processus commun aux plateformes. Pendant sa spécification, nous avons pris en compte deux points de vue différents et complémentaires : le point de vue IHM et le point de vue technique.

Le processus a pour but de spécifier le modèle métier de conception pédagogique d'une plateforme de formation. Ce métamodèle servira ensuite de base de spécification pour les modules d'import/export des LMS, ainsi que la spécification et le développement de langages/éditeurs de conceptions pédagogiques externes. Le processus est composé de trois parties principales :

- **L'analyse centrée IHM** : la première partie est conduite par 3 analyses séquentielles (macro-IHM, fonctionnelle et micro-IHM) selon une démarche descendante. Chaque analyse possède ses propres caractéristiques et produit son (ses) modèle(s) conforme(s) au formalisme proposé. Ensuite, une étape de composition basée sur les modèles résultants des analyses précédentes consiste à formaliser le modèle métier final par la dérivation de celui issu de l'analyse centrée IHM.
- **L'analyse centrée technique** : la seconde partie concerne l'analyse centrée technique. Plusieurs méthodes techniques peuvent être adoptées (base de données, code source, les sauvegardes de cours, etc.) mais la principale source d'informations sera l'analyse de la base de données.
- **La confrontation et formalisation** : la troisième partie porte sur la confrontation entre le modèle IHM et le modèle technique puis sur la formalisation du modèle métier final (métamodèle) de conception pédagogique.

➤ **Une approche DSM pour la spécification de nouveaux DSMLs**

Afin de surmonter les difficultés de spécification des scénarios pédagogiques sur les plateformes de formation, nous proposons l'utilisation des langages spécifiques à des domaines particuliers, appelés des DSMLs (*Domain Specific Modeling Languages*). Le domaine qui nous intéresse est la conception pédagogique. Notre approche repose sur le langage de conception pédagogique de la plateforme, identifié par un processus spécifique. Ce langage spécifié dans un format de méta-modélisation servira comme base (domaine métier) pour le développement de nouveaux DSMLs. Par conséquent, les DSMLs sont des prototypes d'utilisation du langage identifié via l'approche DSM (*Domain Specific Modeling*). Nos travaux représentent un nouveau terrain d'expérimentation de l'approche DSM. Cette approche est capable d'apporter une valeur ajoutée à la spécification de nouveaux DSMLs.

➤ **Des outils de conception plus adaptés aux pratiques des enseignants**

Nous avons proposé aux enseignants-concepteurs des outils de conception graphiques reposant sur le même langage des plateformes utilisées tout en excluant les détails techniques. Des outils tels qu'un VIDL (*Visual Instructional Design Language*) associé avec son éditeur permettent aux enseignants de

dépasser les difficultés d'utilisation directe des interfaces des plateformes pour la spécification des situations d'apprentissage.

Nous avons choisi des plateformes de formation utilisées au sein de l'université du Maine et également au sein d'autres universités nationales et internationales. Ces plateformes sont open-source et modulaires. Elles représentent l'environnement de nos expérimentations et d'instrumentation de nos propositions.

1.5. Organisation du document

Ce mémoire est structuré en 3 parties. La première partie est consacrée à la présentation du cadre scientifique de notre travail. Elle est composée de deux chapitres :

- Le **chapitre 2** présente l'état de l'art et notre positionnement scientifique. Dans un premier temps, nous présentons la conception pédagogique avec ses différentes activités. Nous nous intéressons spécifiquement à la scénarisation pédagogique dont la notion du scénario est l'élément central. La deuxième section représente l'Ingénierie Dirigée par les Modèles (IDM). Dans la troisième section, nous présentons les travaux existants portant sur la conception des scénarios pédagogiques. La quatrième section présente les plateformes de formation à distance. Nous nous intéressons aux travaux portant sur la conception pédagogique sur ces plateformes. La dernière section présente et discute les travaux sur l'opérationnalisation des scénarios pédagogiques. L'activité de scénarisation sur les plateformes de formation ainsi que l'activité d'opérationnalisation représentent le centre de nos travaux.
- Le **chapitre 3** détaille, après une première problématisation présentée dans l'introduction du document, notre problématique scientifique en exposant nos questions de recherche. En s'appuyant sur le positionnement scientifique, nous argumentons notre problématique. Ensuite, nous présentons notre approche et la méthodologie de recherche que nous avons suivie pour la réalisation de nos études et nos propositions.

La seconde partie de ce document détaille nos propositions qui portent sur (1) la présentation d'un processus d'analyse et d'externalisation du langage de conception pédagogique d'une plateforme de formation, (2) l'instrumentation du processus sur deux plateformes différentes pour valider leur mise en pratique, (3) l'extension des plateformes par des nouvelles fonctionnalités d'import/export, et (4) le développement de deux outils de conception dédiés aux plateformes étudiées et basés sur le langage identifié.

- Le **chapitre 4** décrit le processus que nous proposons pour l'analyse, l'identification et la formalisation du langage de conception pédagogique embarqué dans les plateformes de formation.
- Le **chapitre 5** présente les expérimentations que nous avons menées sur deux plateformes de formation. Nous avons instrumenté le processus, présenté dans le chapitre précédent, pour identifier et formaliser leur langage de conception pédagogique.

- Le **chapitre 6** présente le développement et la mise à l'essai des outils proposés pour les deux plateformes d'expérimentations (Moodle et Ganesha). Dans un premier temps, nous présentons les modules de communication (import/export) intégrés au sein de leur espace de cours. Dans un deuxième temps, nous présentons les outils de conception graphiques développés en se basant sur l'approche DSM via des frameworks de développement spécifiques : EMF (*Eclipse Modeling Framework*) et GMF (*Graphical Modeling Framework*). A la fin de ce chapitre, nous présentons la mise en essai des outils développés (éditeurs et modules de communication).

Nous enchaînons sur la dernière partie de ce document, le **chapitre 7**, qui présente un bilan de notre travail, une analyse critique et les limites de nos propositions. Elle ouvre ainsi des voies de poursuite de nos travaux que nous présentons en perspectives.

I

Positionnement scientifique et problématique de recherche

2

Etat de l'art et positionnement Scientifique

Sommaire

- 2.1. La conception et la scénarisation pédagogique
- 2.2. L'ingénierie dirigée par les modèles
- 2.3. Travaux menés sur la conception des scénarios pédagogiques
- 2.4. Les plateformes de formation à distance
- 2.5. Travaux menés sur l'opérationnalisation des scénarios pédagogiques

Ce chapitre expose l'état de l'art et le positionnement scientifique de notre travail. Nous positionnons notre approche dans le cadre des travaux menés en conception pédagogique (section 2.1). Plusieurs d'entre eux considèrent l'Ingénierie Dirigée par les Modèles (IDM) comme un cadre théorique et pratique. La section 2.2 représente ainsi les fondamentaux de l'IDM. Dans un premier temps, nous nous intéressons aux approches de conception de scénarios pédagogiques dans la communauté EIAH (section 2.3). Ensuite, nous présentons les environnements d'exécution des scénarios pédagogiques, plus spécifiquement les plateformes de formation à distance. La dernière section (section 2.5) est consacrée aux travaux d'opérationnalisation des scénarios sur ces plateformes.

2.1. La conception et la scénarisation pédagogique

2.1.1. La conception pédagogique

La conception pédagogique (*Learning Design* ou *Instructional Design*) représente l'activité de conception des unités, des situations ou des environnements d'apprentissage. [Paquette 2002] la définit comme "l'ensemble des théories et des modèles permettant de comprendre, d'améliorer et d'appliquer des méthodes d'enseignement favorisant l'apprentissage". [Berger et Kam 1996] la définissent comme suit : "*Instructional Design is the systematic development of instructional specifications using learning and instructional theory to ensure the quality of instruction. It is the entire process of analysis of learning needs and goals and the development of a delivery system to meet those needs. It includes development of instructional materials and activities and tryout and evaluation of all instruction and learner activities*".

Plusieurs approches ont été définies pour instrumenter la conception pédagogique telles que les approches de conception centrées sur les connaissances, ou sur les ressources (appelée aussi l'approche documentaire), ou sur les interactions, ou sur les activités (appelée aussi l'approche EML (*Educational Modeling Language*)), etc. Nous présentons ces approches dans les sections suivantes.

La conception pédagogique comporte plusieurs activités centrées sur la spécification des scénarios pédagogiques. En tenant compte des premiers besoins pédagogiques et didactiques, ce processus consiste dans un premier temps à spécifier un scénario, à le mettre à disposition des usagers et à favoriser son exécution. Comme tout produit logiciel, plusieurs observations ou besoins d'observations se révèlent après leur utilisation. Cela entraîne des analyses des traces d'utilisation, soit pour une intervention en situation d'apprentissage, soit pour amener des améliorations à cette situation. Nous illustrons par la figure 1 les cinq activités principales de la conception pédagogique :

- La scénarisation : cette phase concerne la spécification des situations d'apprentissage (scénarios pédagogiques, cours, exercices, etc.). Plusieurs travaux dans la communauté EIAH ont abordé cette activité.
- La mise en œuvre : cette phase a pour objectif de déployer un scénario sur un dispositif d'apprentissage (plateforme de formation, système de type LMS (*Learning Management System*), etc.) pour qu'il puisse être utilisé par ses usagers.
- L'exécution : la situation d'apprentissage est exécutée par ses usagers (apprenants, tuteurs, etc.).
- L'observation : cette phase vise à recueillir des données sur le déroulement de la situation d'apprentissage. L'enseignant ou le formateur exprime souvent des besoins d'observation pour suivre les activités réalisées par les différents utilisateurs de l'EIAH.
- L'analyse des traces : cette phase consiste à analyser et à interpréter les données collectées par la phase précédente dont l'objectif est d'identifier des motifs récurrents (usage particulier) ou d'extraire des informations sur l'activité ou le comportement des utilisateurs.

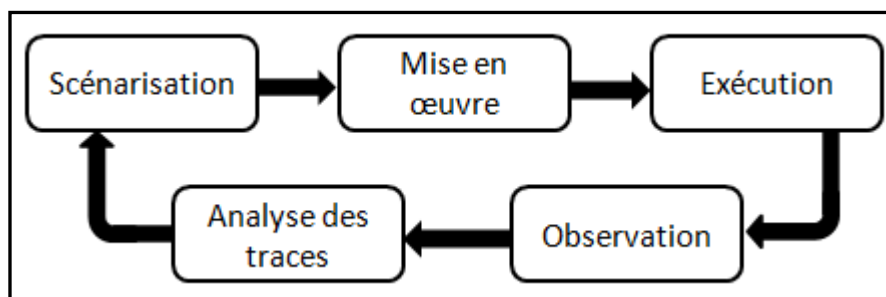


Figure 1 : Les principales activités de la conception pédagogique

2.1.2. La scénarisation et les scénarios pédagogiques

La scénarisation pédagogique se situe au premier niveau dans le cycle de conception pédagogique. Elle consiste à spécifier des situations d'apprentissage et des scénarios pédagogiques. Selon [El-Kechai 2008], "*l'activité menant à la spécification d'un scénario pédagogique est aussi importante que le scénario lui-même*". En effet, les enseignants-concepteurs sont invités à positionner l'activité d'apprentissage qu'ils souhaitent mettre en œuvre en s'interrogeant sur les intentions, les objectifs, ainsi que le contexte dans lequel l'activité d'apprentissage prend place [Brassard et Daele 2003]. Le scénario pédagogique vise ainsi à prendre en compte différentes dimensions liées à la mise en œuvre d'environnements d'apprentissage.

[Pernin 2004] définit les trois niveaux de scénarisation suivants (le niveau de description de la situation d'apprentissage) en fonction de la granularité :

- Scénariser la structuration pédagogique : organiser les unités de structuration pédagogique telles que les cours, les modules, les leçons, les cursus, les crédits, etc.
- Scénariser l'enchaînement d'activités : organiser les activités d'un apprenant ou d'un groupe d'apprenants au sein d'une unité d'apprentissage tels qu'une séquence pédagogique, une section, une séance, etc.
- Scénariser la réalisation de tâche : décrire le déroulement d'une tâche précise (comme par exemple parcourir un hyper-document, résoudre un exercice, répondre à des questions, manipuler une simulation).

La définition du scénario pédagogique dépasse la simple description en tant que 'préparation de leçon par un enseignant'. Plusieurs travaux de recherche ont recensé la définition du scénario pédagogique représentant le centre de l'activité de scénarisation pédagogique.

[Charlier et Daele 2002] définissent un scénario pédagogique comme "*le résultat du processus de conception d'une activité d'apprentissage, processus s'inscrivant dans un temps donné et aboutissant à la mise en œuvre du scénario. Donc un scénario comprend des objectifs, une planification des activités d'apprentissage, un horaire, une description des activités des étudiants, des modalités d'évaluation qui sont définis, agencés et organisés au cours d'un processus de design*".

[Paquette et al. 1997] définissent le scénario pédagogique comme la "*conjonction d'un scénario d'apprentissage et d'un scénario de formation qui lui est associé avec l'expression des liens qui les unissent*".

- Un scénario d'apprentissage est "*l'ensemble des activités destinées aux apprenants et organisées en un tout cohérent*".
- Un scénario de formation est "*l'ensemble des activités destinées au formateur et organisées en un tout cohérent*".

[Schneider 2003] définit le scénario pédagogique comme étant "*une séquence orchestrée de phases dans lesquelles les apprenants ont des tâches à effectuer et des rôles spécifiques à jouer*". Selon [Pernin 2004], le scénario pédagogique est la "*description du déroulement d'une situation d'apprentissage en termes de rôles, d'activités et d'environnement nécessaire à sa mise en œuvre, mais aussi en termes de connaissances manipulées*".

[Pernin et Lejeune 2004] font la différence entre deux scénarios :

- Le scénario prédictif : défini comme établi *a priori* par un concepteur en vue de mettre en place 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.

[Koper et Tattersall 2005] définissent un scénario pédagogique comme "*la concrétisation du savoir didactique des enseignants lors de la réalisation d'un acte pédagogique, un cours, une séquence, l'ensemble d'un parcours*".

La notion de scénario pédagogique n'est pas standardisée. De nombreuses définitions ont été publiées, mais elles se chevauchent. La définition que nous adoptons est la suivante : "un scénario pédagogique est une structuration d'activités et de ressources pédagogiques dans une séquence représentant un cours ou une section de cours".

2.1.3. Processus de scénarisation ou de conception d'un scénario pédagogique

En s'inspirant du processus de développement en génie logiciel, plus précisément du processus unifié de développement RUP (*Rational Unified Process*), [Vantroys 2003] propose un processus de conception d'un scénario pédagogique composé de six étapes. Ce cycle a été repris dans plusieurs travaux de recherche en EIAH tels que [Laforcade 2004], [Choquet 2007], [El-Kechai 2008], [Abdallah 2009] et [Zendagui 2010]. Une représentation du cycle est donnée dans la figure 2. Les six étapes sont définies de manière successive dans un cycle itératif :

1. Expression initiale des besoins : cette phase est indépendante de toute approche de conception. Elle a pour objectif d'exprimer les besoins initiaux du produit envisagé (situation d'apprentissage, cours, scénario). Le concepteur décrit de sa propre manière (sur un papier, un espace de dessin, etc.) le contenu de la situation d'apprentissage. Il spécifie ses différents éléments (les activités, les ressources, les objectifs, le temps, etc.).

2. Analyse et conception : cette phase consiste à formaliser les étapes préliminaires de développement de la situation d'apprentissage en tenant en compte les besoins énoncés à l'étape précédente. La phase d'analyse est basée sur un énoncé informel (le besoin exprimé à l'étape précédente et complété par des recherches d'informations auprès des experts du domaine pédagogique, tels les enseignants)

ainsi que de l'analyse de l'existant éventuel (c'est-à-dire la manière dont les processus à traiter par le système d'apprentissage se déroule chez les futurs utilisateurs). Elle permet de lister les résultats attendus. La phase de conception permet de décrire de manière non ambiguë la situation d'apprentissage. Cette description peut être menée par un langage de modélisation pédagogique. Souvent, cette phase nécessite l'intervention d'un expert de domaine ou d'un ingénieur pédagogique en collaboration avec l'enseignant. Elle amène à produire un scénario pédagogique.

3. Implémentation ou implantation : cette phase consiste à transformer le scénario pédagogique issu de la phase précédente (par exemple exprimé à l'aide d'un langage de modélisation pédagogique) en langage interprétable par la machine (en XML par exemple) en vue de le rendre opérationnel.

4. Déploiement : cette phase consiste à la mise en œuvre du scénario formalisé en format interprétable par la machine sur un artefact informatique comme une plateforme de formation ou un système de type LMS (*Learning Management System*). Le déploiement des scénarios peut être réalisé de façon manuelle (configuration manuelle des éléments composant le cours) ou bien de façon automatisée (via des services web, des transformations, des API d'import, etc.). En fin de cette phase, le scénario conçu peut être exécuté par ses usagers (apprenants, etc.).

5. Tests : cette phase consiste à tester la validité des spécifications de la conception dans la réalité d'apprentissage concrète. Le scénario est réalisé par les acteurs impliqués dans la situation d'apprentissage. Ensuite, un expert ou un ingénieur pédagogique contrôle via des tests le bon déroulement pédagogique et fonctionnel de la situation. Plus les tests sont concluants et plus ils sont répétés et affinés, plus les spécifications de conception sont jugées stables.

6. Évaluation : Cette phase consiste à vérifier que le modèle prévu initialement correspond bien aux besoins des différents acteurs. Les tests de la phase précédente sont analysés afin de déduire les améliorations pouvant être intégrées au système de formation. Ceci peut amener à revoir les phases d'expression des besoins et/ou d'analyse et de conception.

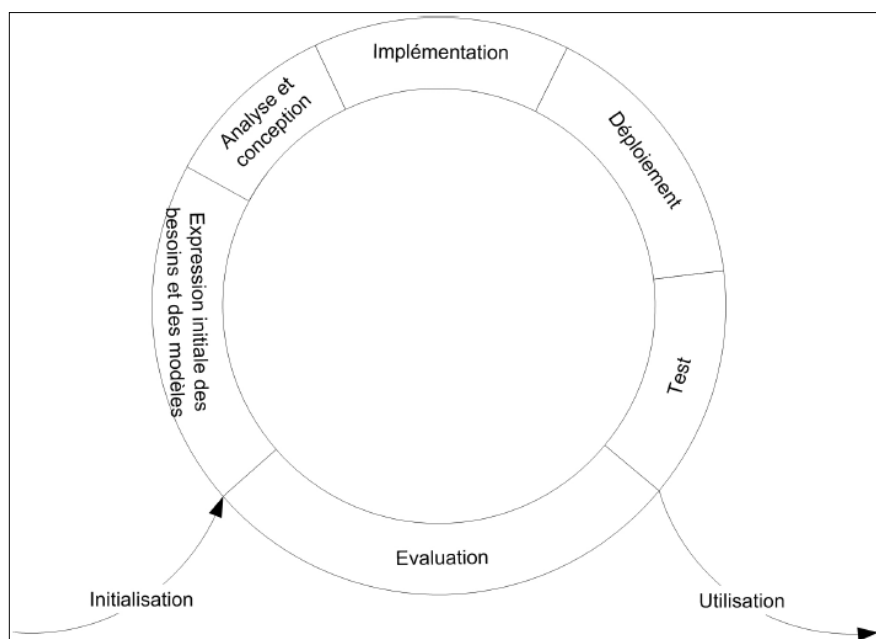


Figure 2 : Processus itératif de conception d'un scénario pédagogique [Vantroys 2003]

Le cycle proposé par [Vantroys 2003] détaille l'ensemble des étapes portantes sur la spécification d'un scénario pédagogique. De notre point de vue, certaines phases ne sont pas essentielles et d'autres peuvent être unies. Nous envisageons alors un cycle composé de trois phases : La conception, le déploiement et l'évaluation. Nous illustrons ce cycle par la figure 3.

- **La conception** : les deux premières phases (expression initiale des besoins et analyse et conception) ont le même objectif : modéliser la situation d'apprentissage. Elles sont ainsi unies dans une seule phase. Cette phase porte sur la spécification d'un scénario *a priori* avant le déroulement de la situation d'apprentissage. Le design pédagogique peut être établi de plusieurs manières : formelle et informelle.
- **La mise en œuvre** : les phases d'implémentation et de déploiement peuvent être traitées en même temps. Leur objectif est d'opérationnaliser le scénario conçu sur un artefact informatique pour l'apprentissage humain. À la fin de cette phase, le scénario conçu *a priori* peut être exécuté par ses usagers.
- **L'évaluation** : les deux dernières phases (test et évaluation) ont le même objectif. Elles consistent à vérifier le bon déroulement du scénario pédagogique afin de détecter les ajustements et les améliorations qui peuvent être menés sur le scénario initial. Pour cela, nous les regroupons dans une seule phase. Les modifications seront réalisées dans la phase de conception.

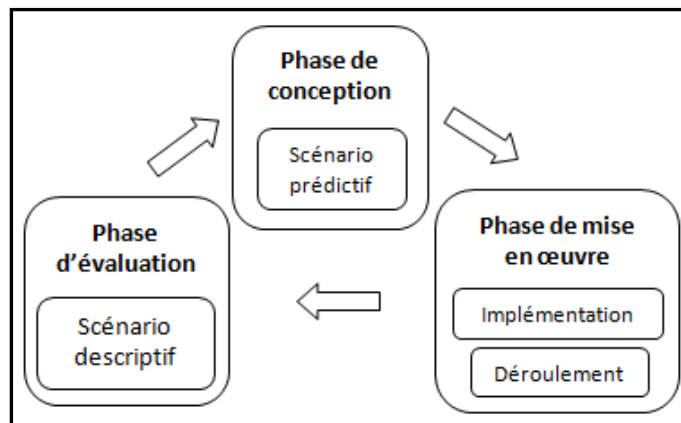


Figure 3 : Cycle de vie simplifié d'un scénario pédagogique

2.1.4. Les deux principales approches de conception d'un scénario pédagogique

Selon [El-Kechaï 2008] et [El-Kechaï et Choquet 2005], deux approches sont possibles pour mener le processus de scénarisation : l'approche interprétative et l'approche constructive.

- **L'approche interprétative** repose sur l'exploitation d'un modèle (métamodèle) prédéfini. Ce modèle, souvent générique, est développé par des experts de scénarisation sans impliquer les concepteurs de scénarios dans l'implémentation du modèle pédagogique. Cette approche consiste à sélectionner un modèle avant de l'instancier pour représenter le scénario pédagogique. Ce modèle peut être "*réutilisable pour décrire des scénarios dans différents domaines et/ou pour différentes approches pédagogiques*" [El-Kechaï 2008].

L'avantage de cette approche est qu'elle est réutilisable pour décrire des scénarios dans différents domaines et/ou pour différentes approches pédagogiques. Le travail de la modélisation n'est pas alourdi parce que la scénarisation est guidée par un métamodèle². Cependant, cette approche reste limitée pour décrire les spécificités d'une situation d'apprentissage cible. Toute la modélisation dépend directement du métamodèle utilisé. L'utilisation des langages de modélisation pédagogiques (EML) représentent un exemple de cette approche. Nous les présentons plus loin dans ce document.

- **L'approche constructive** repose sur la construction du scénario et du métamodèle par abstraction depuis une source. Les enseignants sont impliqués dans la conception du métamodèle et du modèle de scénario pédagogique. Ainsi, en se situant dans un contexte d'apprentissage bien précis, l'approche constructive consiste à identifier et à cristalliser les connaissances (objets d'apprentissage visés) pour construire un métamodèle d'expression pédagogique. Ce dernier est utilisé pour exprimer les futurs scénarios. Cette approche ouvre l'opportunité de produire un métamodèle flexible, adapté et adaptable pour la description des situations d'apprentissage. L'approche de conception des patrons pédagogiques est un exemple de ce type d'approche. Nous représentons, dans la section 2.3.3.2, le projet MDEduc, basé sur cette approche, intégrant les enseignants dans le processus de modélisation et de méta-modélisation.

L'approche constructive accorde plus d'importance aux enseignants concepteurs des situations d'apprentissage qui se sont impliqués dans le processus de conception et de validation de la modélisation. Elle favorise la prise en compte des connaissances et des spécificités d'un domaine pédagogique cible. Toutefois, il est difficile de juger la pertinence de la réutilisation de cette approche dans d'autres contextes d'apprentissage [El-Kechaï 2008]. Son principal inconvénient est qu'elle nécessite (1) une analyse du domaine pédagogique plus conséquente que l'approche interprétative et (2) un travail d'abstraction qui alourdit le travail de modélisation.

Nos travaux s'inscrivent dans une approche interprétative. Nous n'avons pas l'objectif d'impliquer les enseignants dans le processus de spécification du métamodèle. Nous voulons fournir un métamodèle spécifique pour la conception des situations d'apprentissage. Ce métamodèle décrit le langage de conception pédagogique de la plateforme de formation utilisée. Il sera obtenu par abstraction des mécanismes de scénarisation pédagogique proposés par la plateforme. Ce métamodèle sera le résultat de l'application d'un processus spécifique d'identification et de formalisation de ce langage que nous présentons au sein du quatrième chapitre. Ensuite, nous utilisons le métamodèle comme une base pour le développement des outils de conception et des modules d'import/export que nous proposons aux enseignants-concepteurs. Nous détaillons cette partie au sein du sixième chapitre. Notre objectif est de faciliter l'opérationnalisation des scénarios pédagogiques qui seront modélisés sur des outils graphiques dans un langage conforme à celui de la plateforme cible.

2.1.5. Les langages de modélisation pédagogique

Les travaux de l'*Open University of the Netherlands* (OUN) dus à Rob Koper portent sur la description des situations d'apprentissage à l'aide d'un Langage de Modélisation Pédagogique (EML : *Educational Modelling Language*). [Rawlings et al. 2002] définissent un EML comme suit : "An EML is a semantic

² Un métamodèle est un modèle d'un langage de modélisation.

information model and binding, describing the content and process within a 'unit of learning' from a pedagogical perspective in order to support reuse and interoperability". Le modèle EML (figure 4) repose au plus haut niveau sur la structuration des situations d'apprentissage en "unités d'étude" (*units of study*). Typiquement, une unité d'étude peut être un cours, une leçon, une étude de cas, un travail pratique, etc. Ces unités d'études font ultérieurement l'objet des "unités d'apprentissage" (*units of learning*) dans la spécification IMS-LD. Les unités d'études sont composées de trois éléments principaux : « rôle », « activité » et « environnement ». Les activités, placées au centre du modèle, sont réalisées par un ou plusieurs acteur(s) tenant chacun un rôle. Elles sont effectuées dans un environnement donné fournissant un ensemble de ressources (ou objets pédagogiques). Outre la diversité des types d'activités (activités d'apprentissage, activités d'accompagnement, activités d'instrumentation, etc.), elles doivent répondre à des objectifs pédagogiques précis.

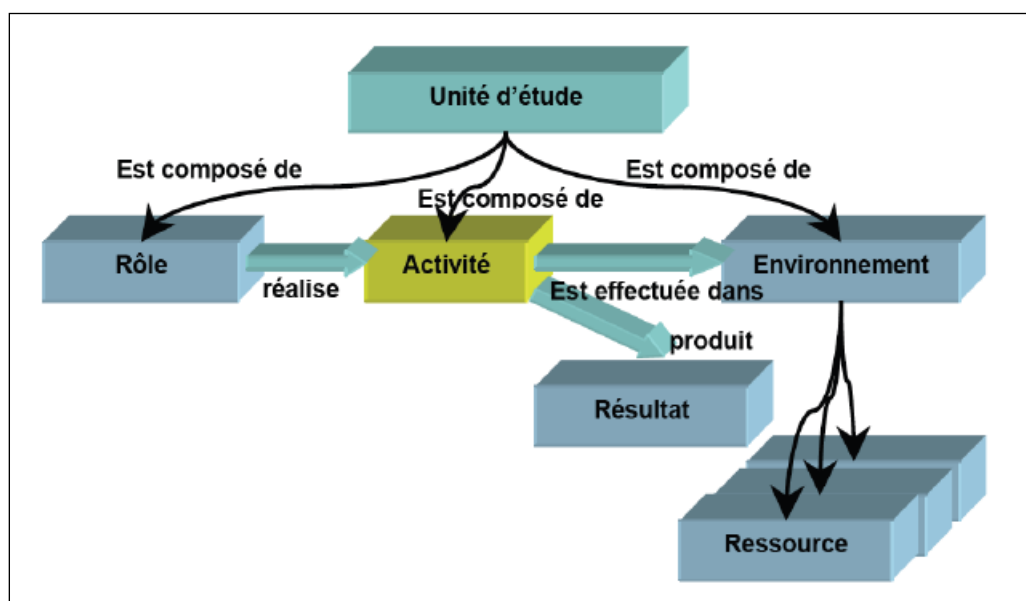


Figure 4 : Vue simplifiée du modèle EML [Pernin 2003]

Les EMLs proposent un ensemble de concepts et de relations pour la description des situations d'apprentissage (scénario pédagogique, etc.). Ces langages ont permis de passer d'une vision statique centrée sur les ressources numériques vers une vision plus dynamique centrée sur l'activité. Leur but est de permettre la gestion des ressources et les activités en tenant compte des notions complexes comme celles des acteurs, des rôles, des structures d'activités, etc. [Pernin 2004] a identifié cinq apports majeurs des EMLs sur la réflexion pédagogique :

- La conception centrée sur l'activité d'apprentissage ;
- L'adoption de plusieurs stratégies pédagogiques lors de la conception des activités ;
- La flexibilité d'orienter le choix de l'enseignant ;
- La possibilité de concevoir des activités réutilisables dans des contextes différents avec d'autres ressources et un autre contenu ;
- La pérennité et l'indépendance de la conception pédagogique par rapport aux plateformes.

Dans la section 2.3, nous présentons quelques langages de modélisation pédagogique tels qu'IMS-LD, LDL et quelques EMLs graphiques (CADMOS et LAMS).

2.1.5.1. Les langages de modélisation graphiques

Les VIDLs (*Visual Instructional Design Language*) sont des langages de modélisation pédagogique (EML) proposant une notation graphique. Ces langages incluent une partie dédiée à l'interprétation et la visualisation centrée utilisateur et une partie à l'interprétation centrée machine. Dans cette section, nous allons présenter deux classifications de VIDLs. Ensuite, dans les sections 2.3.2.2 et 2.3.2.3, nous nous intéressons davantage aux VIDLs centrés plateformes qui partagent nos propositions et nous présentons deux exemples de ces VIDLs : CADMOS et LAMS.

La première classification des VIDLs est proposée par [Laforcade 2010]. Elle présente trois catégories de VIDLs :

- Les VIDLs centrés praticiens : le vocabulaire de ces VIDLs est partagé par une équipe pluridisciplinaire de conception ou est commun à une théorie de conception pédagogique. Leur objectif est de faciliter la spécification des scénarios pédagogiques, guider la conception pédagogique et supporter la communication et la réflexion entre les praticiens. Les VIDLs dédiés à l'approche de conception basée sur les patrons pédagogiques sont des exemples de VIDLs centrés praticiens. CPM [Laforcade et al. 2005] [Nodenot 2005], CoUML, E2ML et PoEML [Botturi et Stubbs 2007] sont des exemples concrets de ces VIDLs.
- Les VIDLs abstraits : le vocabulaire de ces VIDLs est généralement indépendant des artefacts d'exécution (plateformes de formation, systèmes LMS, etc.). Leur objectif est de faciliter la réutilisation des scénarios pédagogiques, l'interopérabilité des dispositifs et l'abstraction des EML ou des approches spécifiques à un domaine. MOT+LD [Paquette et al. 2008], Collage [Hernández-Leo et al. 2006] et UML4LD [Laforcade 2007] sont des exemples de VIDLs abstraits.
- Les VIDLs centrés artefacts (plateformes ou LMS) : le vocabulaire de ces VIDLs est conforme à celui d'un artefact particulier. L'objectif de ces VIDLs est de servir comme guide pour la configuration manuelle, semi-automatique ou complètement automatique de l'artefact utilisé. Par exemple, LAMS [Dalziel 2010] [Dodero et al. 2007a] est un VIDL centré artefacts.

Toutefois, certains VIDLs proposent une notation d'interprétation basée sur d'autres VIDLs tels que Collage qui est basé sur la notation du langage IMS-LD. La figure 5 illustre un ensemble de VIDLs répartis sur les trois catégories proposées selon la notation qu'ils proposent.

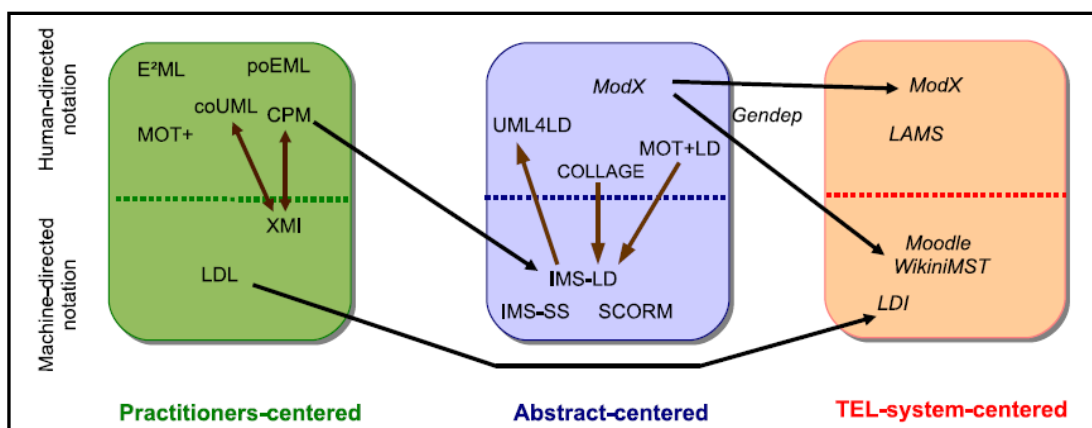


Figure 5 : Classification des exemples de VIDLs [Laforcade 2010]

La deuxième classification est proposée par [Katsamani et Retalis 2011]. Elle catégorise quatre VIDLs (E2ML, coUML, PCeL, PoEML) en se basant sur des critères proposés par [Botturi et al. 2006] qui sont : les compétences des utilisateurs (*user skills*), la formalisation, la stratification, l'élaboration, la perspective et la notation. Le Tableau 1 illustre une comparaison entre des exemples de VIDLs selon les six critères cités. Ces VIDLs sont plutôt centrés praticiens, ou des sur-couches graphiques ou des services vers des EMLs de traduction. Peu de ces VIDLs sont centrés plateformes ; ils sont soit des add-ons aux plateformes ou bien des services partiels de transformation.

CRITERIA	E2ML	PCeL	coUML	POEML
User Skills	Novice	Medium	Medium	Expertise
Formalization	Semi-Formal	Semi-Formal	Semi-Formal	Formal
Stratification	Flat	Layered	Layered	Layered
Elaboration	Conceptual	Conceptual	Conceptual	Implementation
Perspective	Multiple	Single	Multiple	Multiple
Notation	Visual	Visual	Visual	Visual

Tableau 1 : Classification des exemples de VIDL [Katsamani et Retalis 2011]

2.1.6. Conclusion

Les approches de conception pédagogique, ainsi que les langages de modélisation pédagogique ont permis d'encadrer cette activité et de fournir les instructions nécessaires pour la spécification de situation d'apprentissage. La complexité engendrée par leur utilisation a poussé les chercheurs à proposer de nouveaux EMLs mieux adaptés aux pratiques des enseignants tels que les VIDLs qui ont fourni un environnement de conception supportant la communication et la réflexion entre les praticiens.

Notre travail s'inscrit dans une approche centrée sur les langages de modélisation (EMLs). Plus spécifiquement, nous nous situons dans les VIDLs centrés plateformes sur la base de la classification proposée par [Laforcade 2010]. Nous voulons identifier le langage de conception pédagogique embarqué dans les plateformes de formation et le formaliser par un VIDL ainsi que des outils de conception graphique dédiés. Vis-à-vis de la classification proposée par [Katsamani et Retalis 2011] et [Botturi et al. 2006], nous nous situons principalement dans une perspective de proposer des EMLs avec une notation visuelle et formelle à des enseignants-concepteurs de moyennes compétences (*user skills: medium*).

L'avantage d'adoption d'un langage conforme à celui de la plateforme est la préservation de la sémantique des scénarios pédagogiques. Les scénarios peuvent être ainsi implémentés sans perte d'informations ou de données.

2.2. L'Ingénierie Dirigée par les Modèles

L'ingénierie dirigée par les modèles (IDM), en anglais *Model Driven Engineering* (MDE), est une approche spécifique du génie logiciel ouvrant de nouvelles voies d'investigation pour faire face à la complexité et à l'évolution croissante des applications [Bézivin et Blanc 2002]. L'originalité de

l'approche IDM vis-à-vis des méthodes de modélisations traditionnelles (Merise, UML, etc.) réside dans sa préoccupation à rendre les modèles productifs plutôt que contemplatifs. L'objectif de l'IDM est de définir un cadre théorique pour produire des modèles productifs (interprétables et manipulables par une machine) [Bézivin et al. 2004].

L'IDM est une forme d'ingénierie générative mettant à disposition des outils, concepts et langages pour créer et transformer des modèles ainsi que pour générer le code (entier ou partiel) d'une application informatique en utilisant des transformations successives de modèles. Ce courant s'est développé suite à la définition de l'approche MDA (*Model Driven Architecture*) en fin de l'année 2000 par l'OMG (*Object Management Group*) [OMG 2003].

Dans le domaine de la conception pédagogique, de nombreuses approches utilisent l'Ingénierie Dirigée par les Modèles comme cadre théorique et pratique. Ce cadre a été adopté dans plusieurs travaux de recherche de modélisation des EIAHs [Corbière et Choquet 2005] [Nodenot 2005] et de conception des scénarios pédagogiques, en particulier [Laforcade 2004], [De Moura Filho 2007], [Caron 2007c], [El-Kechai 2008], [Abdallah 2009], [Ferraris et al. 2009] [Zendagui 2010], [Ouraiba 2012], etc.

2.2.1. L'architecture dirigée par les modèles (MDA)

Le MDA [OMG 2003] est une architecture basée sur les modèles proposant de concevoir des applications en séparant la logique métier de l'entreprise, de toute plateforme technique afin d'augmenter la réutilisation du code déjà développé, réduire le temps de développement, et faciliter l'intégration des nouvelles technologies [Boulet et al 2003]. Cette architecture favorise la conception des modèles indépendants des plateformes et de l'environnement d'implémentation [Thi-Lan-anh et al. 2005] pour garantir par conséquent leur interopérabilité. Pour cela, elle offre le pouvoir d'abstraction, de raffinement et de vues différentes sur les modèles.

Pour situer l'IDM et le MDA, nous pouvons dire que le MDA est un processus de type IDM [Laforcade 2007]. L'IDM consiste à appliquer les mêmes principes du MDA à tout espace technologique et à les généraliser ; le MDA concerne par contre seulement les technologies OMG.

2.2.2. Les niveaux d'abstraction dans l'IDM

L'architecture des modèles en IDM, initialement proposée dans le cadre du MDA, est basée sur quatre niveaux d'abstraction [Bezivin et Blanc 2002] :

- Le niveau M3 : est le niveau supérieur le plus abstrait. Il représente les méta-méta-modèles qui décrivent les notions de bases pour l'expression des modèles du niveau inférieur (M2 et M1) ainsi que les règles de conformité.
- Le niveau M2 : représente les métamodèles. Il définit le langage de modélisation et la grammaire de représentation des modèles du niveau M1 en utilisant le vocabulaire et la grammaire spécifiés au niveau 3 [Thi-Lan-anh et al. 2005]. A ce niveau sont également définies les règles de transformations, de fusion, de composition ou de tissage entre les métamodèles [Caron 2007c].
- Le niveau M1 : définit les modèles. Il représente la description de l'application du monde réel, conforme à son métamodèle (niveau M2) selon des règles de conformité définies au niveau M3.
- Le niveau M0 : représente les éléments du monde réel. Ce niveau est une instance du niveau M1, représentant le code généré pour une application spécifique.

2.2.3. Les types de modèles

L'architecture MDA propose la modélisation des applications indépendamment de leur implémentation. Un des objectifs du MDA est de permettre l'opérationnalisation, la réutilisation et les transformations des modèles [Parrend 2006] [Caron et al. 2007]. Elle définit trois niveaux de modèles.

- CIM (*Computation Indépendant Model*) : représente le modèle métier ou le modèle du domaine dans lequel sont définis les besoins des usagers. Ce modèle est indépendant de tout formalisme ou architecture informatique [Caron 2007c].
- PIM (*Platform Independent Model*) : correspond à la spécification de la partie métier d'une application ou d'un modèle de conception indépendamment de toute technologie d'implémentation ou de déploiement.
- PSM (*Platform Specific Model*) : décrit l'application spécifique à une plateforme d'exécution. Il est obtenu par transformation du modèle PIM et contient des informations en termes de code de programme, de types pour l'implémentation, de programmes liés, etc. [OMG 2003].

2.2.4. Les transformations entre les modèles

La transformation de modèles se situe au centre de l'approche MDA [Bezivin 2001]. Elle permet d'aboutir à des modèles informatiquement opérationnalisables [OMG 2003]. Elle nécessite la définition des règles de transformation entre les métamodèles au niveau M2. Ces règles décrivent la correspondance entre les concepts des métamodèles sources et cibles. Elles doivent être écrites dans un langage bien défini afin d'être compilées et exécutées. L'application de ces règles sur un modèle source (exemple d'un scénario écrit par un EML particulier) permet la génération d'un modèle cible (exemple un scénario opérationnalisable sur la plateforme de formation Moodle) [OMG 2003] [Bézivin 2001] [Czarnecki et Helsen 2003]. La transformation se fait grâce à un "*mapping*", entre un modèle initial et un modèle cible, défini comme une traduction entre le métamodèle initial et le métamodèle cible [Parrend 2006]. La génération automatique des modèles cibles sous-entend que ces derniers doivent être écrits dans un langage compréhensible par les machines [Abd-ali et El-Guemhioui 2005].

Dans le cadre de la scénarisation pédagogique, certains travaux de la communauté EIAH se sont basés sur l'approche MDA proposant un cycle de transformation entre les modèles/niveaux. La figure 6 illustre ce cycle.

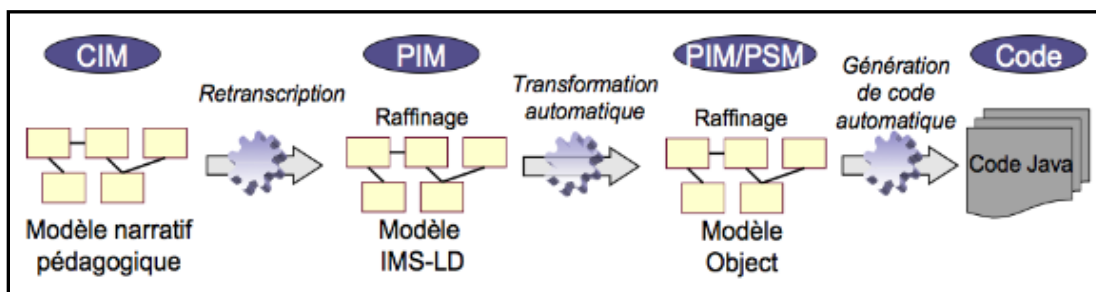


Figure 6 : Cycle de transformations appliqué en EIAH [Nodenot et al. 2005]

2.2.5. La modélisation spécifique au domaine (DSM)

Le DSM (*Domain-Specific Modeling*) est une méthodologie d'ingénierie logicielle pour la conception et le développement des systèmes, basée sur l'application des principes et techniques de l'Ingénierie Dirigée par les Modèles. Il implique l'utilisation systématique de langages graphiques de modélisation spécifiques au domaine (DSMLs : *Domain-Specific Modeling Language*) pour représenter les différentes facettes d'un système [Kelly et Tolvanen 2008] [Laforcade et al. 2008]. Le DSM permet d'augmenter le niveau d'abstraction au-delà de la programmation [DSM 2012]. Les notations des modèles sont spécifiques à un domaine et situées dans un niveau d'abstraction plus élevé que ceux des autres approches traditionnelles de développement logiciel (figure 7) [Rath et al. 2005].

L'approche DSM est basée sur un modèle du domaine (*Domain model*) pour le développement des applications dédiées. En particulier, ce modèle décrit les éléments spécifiques du domaine. Le DSM permet, sur la base de ce modèle, la génération automatique de la quasi-totalité du code final et facilite ainsi le développement ultérieur d'applications centrées sur ce domaine métier. Contrairement à l'approche DSM, les autres approches procèdent par une modélisation standard indépendante du domaine (des modèles UML, Merise, etc.). Ensuite, des travaux de mapping pour la génération du code sont nécessaires pour aboutir au produit final. Ces travaux nécessitent des efforts de retranscription des concepts du modèle et ceux du domaine.

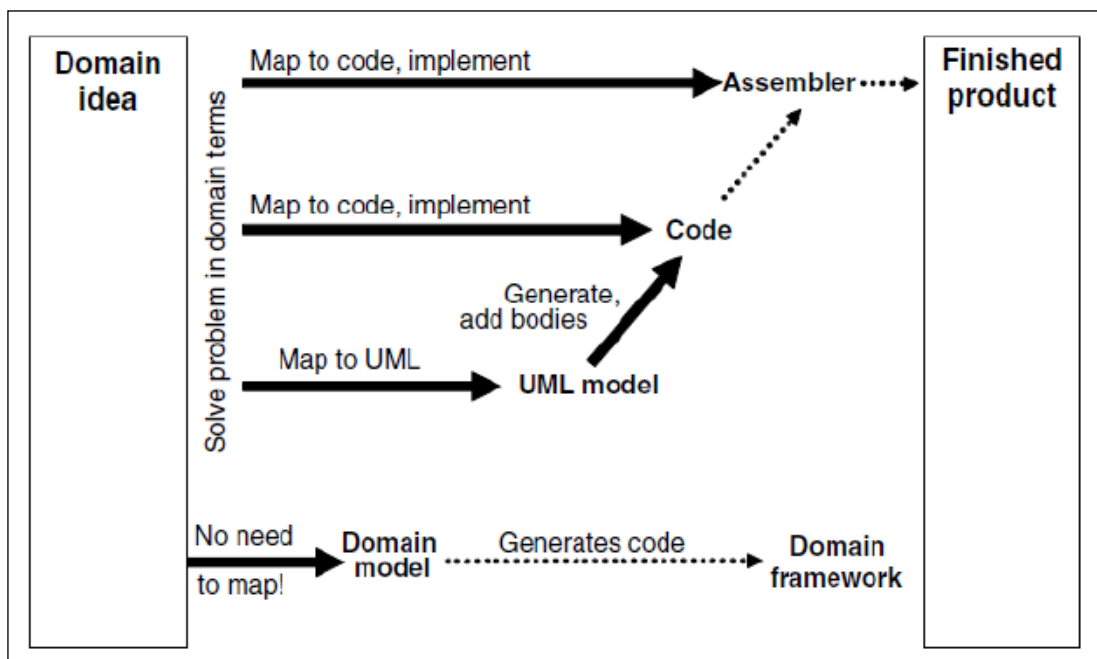


Figure 7 : Implémentation d'un logiciel à partir des termes de son domaine [Kelly et Tolvanen 2008]

Les DSMLs sont des langages spécifiques à un domaine, représentant l'ensemble de concepts et de relations d'un champ de problèmes particulier. Le formalisme graphique des DSL (*Domain-Specific Language*) permet de donner une représentation plus intuitive aux concepts du domaine où les praticiens sont généralement familiers à leur sémantique. Actuellement, il existe plusieurs approches et outils associés pour supporter le DSM :

- les approches des grands éditeurs telles celles de Microsoft basée sur Visual Studio et les Software Factories [MicrosoftDsl 2013] ou celle de MetaEdit+ [Kelly et al. 1996]
- et les projets tant académiques qu'industriels tels ATL [Jouault et al. 2006], GMF [GMF 2013], TIGER [Taenzer 2006], etc.

La définition d'un DSL est généralement basée sur les étapes de méta-modélisation suivantes : (1) la définition des concepts, de leurs propriétés et des relations entre eux, (2) la définition des règles et des contraintes associées aux concepts dans le but est de préserver au maximum la sémantique du domaine et (3) la définition des symboles de la notation concrète [Kelly et Tolvanen 2008]. Ensuite, les outils de DSM se chargent de la production automatique des éditeurs de modèles centrés-diagrammes, des navigateurs de modélisation, des générateurs de modèles et de code, etc.

Le *Domain Specific Modelling* appliqué à la scénarisation pédagogique

L'application concrète de l'approche DSM au domaine de la conception pédagogique permet d'aider la communauté des praticiens et les enseignants-concepteurs à définir leur propre langage de scénarisation. Il permet également de formaliser le langage pédagogique des dispositifs d'apprentissage. Le DSM représente une nouvelle approche pour la modélisation et la spécification formelle des scénarios pédagogiques [Laforcade et al. 2007]. Il est également capable d'apporter une valeur ajoutée (1) pour l'opérationnalisation de scénarios pédagogiques et de langages de modélisation pédagogique, et (2) pour la définition des VIDLs et des outils graphiques centrés sur le métier des praticiens ou des standards ou des plateformes, etc. [Laforcade et al. 2007].

Certains travaux menés au sein de notre laboratoire (LIUM) tels que [Ouraiba 2012], [Zendagui 2010], [Oubahssi et al. 2010], etc. se sont centrés sur l'approche outillée DSM pour la représentation graphique des scénarios pédagogiques et les transformations entre les différents langages de modélisation pédagogiques (EML) [Laforcade et al. 2008]. Dans notre contexte, le DSM convient autant aux scénarios spécifiques à une plateforme de formation. Il est appliqué pour la spécification de VIDLs et d'outils graphiques dédiés à une plateforme particulière. Les enseignants-concepteurs ainsi que les praticiens des plateformes seront les usagers de ces VIDLs et outils.

2.3. Travaux menés sur la conception de scénarios pédagogiques

Après avoir présenté, dans la section précédente, l'activité de conception pédagogique ainsi que ses spécificités, nous présentons ici les travaux menés sur la scénarisation pédagogique. Le point commun de ces travaux porte sur l'expression et la formalisation des scénarios pédagogiques.

Tout d'abord, nous présentons les travaux portant sur les approches centrées sur les langages de modélisation pédagogique standards. En particulier, nous présentons le standard SCORM [Gonzalez-Barbonea et Anido-Rifonb 2010] et la spécification IMS-LD [IMS-LD 2012]. Ensuite, nous présentons les travaux centrés sur les EMLs et les VIDLs, en particulier, les langages LDL [Ferraris et al. 2005], CADMOS [Katsamani et Retalis 2011] et LAMS [Dalziel 2003]. Enfin, nous présentons les approches par l'IDM. Nous donnons trois exemples des travaux les plus représentatifs de l'IDM appliqué en scénarisation pédagogique : le langage CPM [Laforcade 2004] et les deux projets Bricoles [Caron 2007c] et MDEduc [De Moura Filho 2007].

2.3.1. Les approches standards

Dans l'objectif de proposer un langage standard de design pédagogique, il existe actuellement quelques propositions de normalisation dans le monde des technologies éducatives. Nous présentons dans cette section deux standards : SCORM et IMS-LD.

2.3.1.1. SCORM

SCORM (*Sharable Content Object Reference Model*) est un standard proposé en 2000 par le consortium ADL (*Advanced Distributed Learning*) [Gonzalez-Barbonea et Anido-Rifonb 2010] [Pernin 2004]. Ce standard vise à assurer la qualité des contenus en termes de réutilisabilité, accessibilité, pérennité et l'interopérabilité entre les objets pédagogiques et les différents environnements de formation en ligne. Il propose une méthode de structuration des unités d'apprentissage basée sur le contenu dans un parcours de formation (*Content Agregation Model*). Il représente également un guide pour l'exécution des contenus, la communication et le suivi des activités dans un environnement web [Pernin 2004]. La représentation SCORM d'une activité de formation ou d'un cours est formalisée en langage XML. De nombreux dispositifs (éditeurs, LMS, chaînes éditoriales, etc.) utilise actuellement le standard SCORM le plus souvent en proposant d'inclure des contenus SCORM dans les unités d'apprentissage. Par exemple, les plateformes Moodle et Ganesha proposent des activités de type SCORM en intégrant un package spécifique.

2.3.1.2. IMS-LD

La spécification IMS Learning Design [IMS-LD 2012] est issue des travaux de l'Université Ouverte des Pays-Bas portant sur les langages de modélisation de l'apprentissage (les EML (cf. section 2.1.5)). Le but de cette spécification est d'intégrer les préoccupations de conception éducative (*Instructional Design*) dans le mouvement des standards internationaux [IMS-LD 2012].

IMS-LD est considérée comme le standard *de facto* du domaine *Learning Design*. Ce langage formel pour la spécification des situations d'apprentissage considère une grande variété d'approches pédagogiques (situations-problèmes, apprentissage par projet, etc.) [De-Vries et al. 2006]. Ces situations sont sauvegardées de manière formelle en XML.

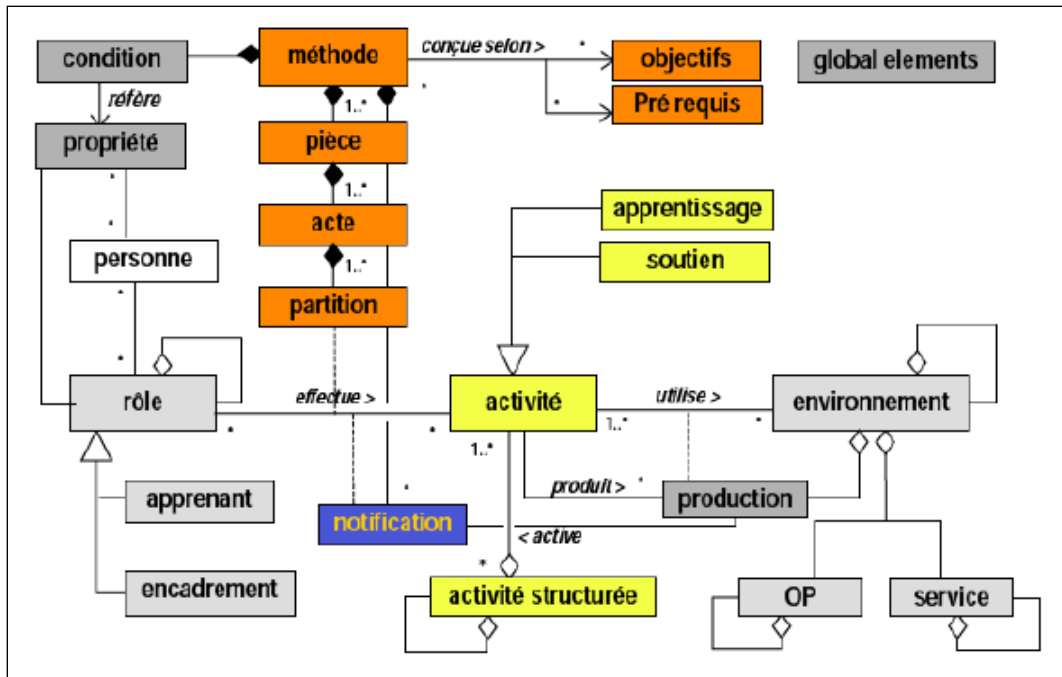


Figure 8 : Modèle conceptuel du langage IMS-LD [Ferraris et Lejeune 2007]

La figure 8 illustre le métamodèle spécifique à IMS-LD. Ce métamodèle introduit les concepts pédagogiques permettant de modéliser les unités d'apprentissage [IMS-LD 2012]. Une activité d'apprentissage est modélisée grâce à la description des différents rôles, activités, environnements, méthodes [IMS-LD 2012], propriétés, conditions et notifications. Le langage IMS-LD est basé sur une métaphore théâtrale [Burgos et al. 2005] sous forme de plusieurs pièces (*Play*) indépendantes les unes des autres. Une pièce est divisée en un ou plusieurs acte(s) (*Act*) et conduites par plusieurs acteurs réalisant un certain nombre d'activités et assumant différents rôles (*role_part*) à des moments. IMS-LD intègre une collection de ressources dans une méthode (*Method*) spécifiant les aspects dynamiques de l'apprentissage.

2.3.1.3. Discussion

Les deux standards IMS-LD et SCORM proposent leurs propres métamodèles et offrent un cadre générique pour la spécification des situations pédagogiques en favorisant l'interopérabilité et la réutilisation. Cependant, ils ont tendance à éloigner encore un peu plus les enseignants et les formateurs du rôle de concepteur en supposant implicitement que la conception soit assurée par un spécialiste ou un ingénieur pédagogique [Choquet 2007].

Dans un premier temps, nous partageons notre point de vue avec plusieurs travaux [Griffiths et al. 2005] [Dodero et al. 2007b] [Ortiz et al. 2008] [Gutiérrez-Santos et al. 2008] sur le fait que la conception pédagogique centrée sur les standards reste une tâche complexe en raison de sa terminologie et de sa notation XML. Il existe peu d'éditeurs compatibles avec les trois niveaux proposés par le standard IMS-LD. "Malgré de nombreux travaux réalisés pour simplifier l'appréhension du standard, sa complexité le rend difficile à maîtriser" [Caron 2007c]. Les enseignants éprouvent beaucoup de difficultés à manipuler les standards pour concevoir les scénarios pédagogiques. "La tâche de transposition d'un

modèle pédagogique demeure complexe pour l'enseignant qui n'est pas familiarisé avec les concepts IMS-LD" [Burgos et al. 2005].

Dans un deuxième temps, nous avons constaté que les standards sont utilisés sans prévoir une relation avec les plateformes d'exécution. Il existe quelques tentatives d'intégration de ces standards au sein des LMS mais elles n'ont pas dépassé la limite d'un prototype ou d'une approche théorique [Burgos et al. 2005]. Ceci souligne que malgré la capacité d'IMS-LD à exprimer toute situation d'apprentissage, il ne permet pas toujours de répondre aux besoins de certains concepteurs et d'être en conformité avec le langage de conception d'un LMS.

Dans le cadre de nos travaux, nous n'avons pas l'objectif de nous centrer sur les approches standards ni de faciliter leur utilisation. Nous nous situons dans une approche centrée sur le langage de conception pédagogique embarqué dans les plateformes de formation à distance. L'utilisation des standards sur les plateformes est souvent réduite à l'importation d'un package spécifique en tant qu'activité pédagogique.

Dans les sections suivantes, nous présentons des approches de conception pédagogique propriétaires par les langages de modélisation pédagogique et des approches propriétaires exploitant l'Ingénierie dirigée par les modèles pour leur mise en œuvre concrète.

2.3.2. Approches centrées sur les langages de modélisation pédagogique

Les langages de modélisation pédagogique (EML) représentent une évolution des spécifications d'apprentissage centrées sur les contenus. Vis-à-vis de la complexité de la notation textuelle, les VIDLs ont proposé une notation graphique et un focus spécifique aux besoins de conception des enseignants-concepteurs. Dans cette section, nous présentons le langage de modélisation pédagogique LDL et les deux langages graphiques (VIDL) : CADMOS et LAMS.

2.3.2.1. LDL (Learning Design Language)

Le langage LDL (*Learning Design Language*) [Ferraris et al. 2005] [Ferraris et al. 2007] [Durand 2006] a été initié en 2003 dans le cadre du projet "ScéClasse" par SysCom³ (Université de Savoie), ARCADE⁴ du CLIPS-IMAG (Université de Grenoble) et la société Pentila.

Contrairement au standard IMS-LD initié pour industrialiser les activités pédagogiques indépendamment de toute approche pédagogique, LDL est un EML qui a été élaboré dans une approche centrée sur l'enseignant dont le but est de décrire des scénarios pédagogiques collaboratifs sous une forme réutilisable et partageable, mais aussi opérationnalisable au sein d'un ENT (*Environnement Numérique de Travail*).

LDL s'appuie une approche de description des scénarios dirigée par les modèles. La figure 9 illustre le métamodèle simplifié de LDL. Ce langage est composé de sept concepts principaux : structure d'activité, interaction, enceinte, rôle, règles, position et observables. La structuration de modélisation de LDL permet de scénariser les activités collaboratives comme un enchaînement d'interactions entre participants. Les scénarios spécifiés en LDL sont implémentés sur des plateformes particulières

³ Systèmes Communicants.

⁴ Ateliers de Réalisation et de Conception d'Applications Destinées à l'Éducation.

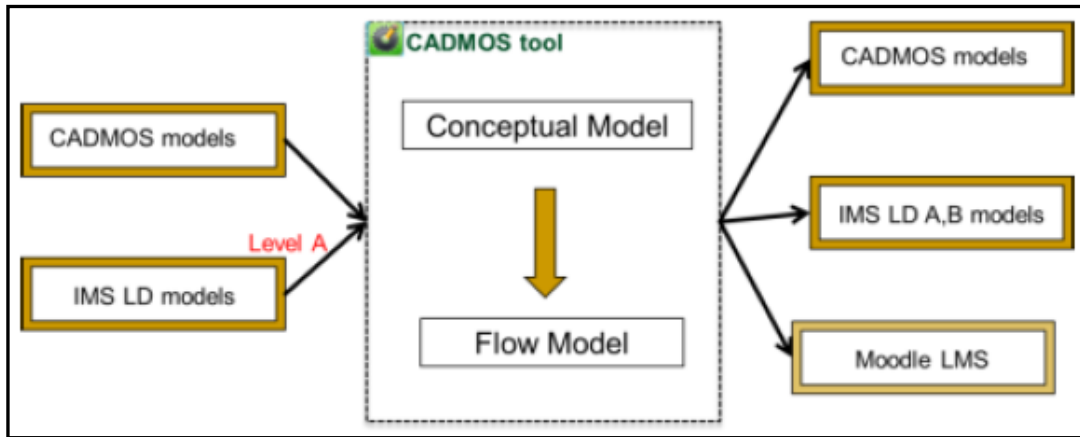


Figure 10 : Processus de conception pédagogique de CADMOS [Boloudakis et al. 2012]

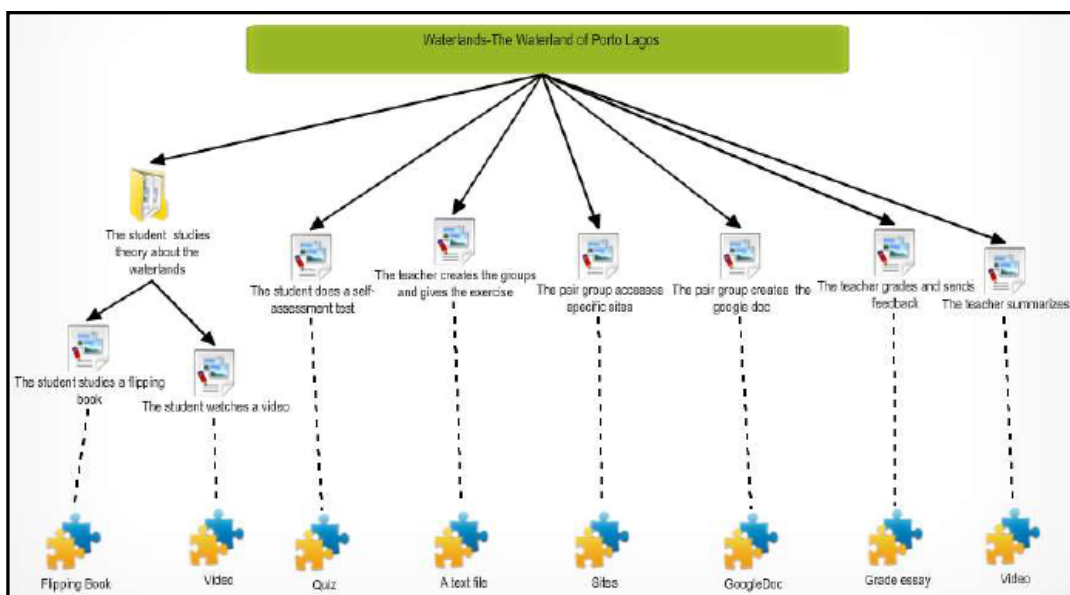


Figure 11 : Exemple de modèle conceptuel de CADMOS [Boloudakis et al. 2012]

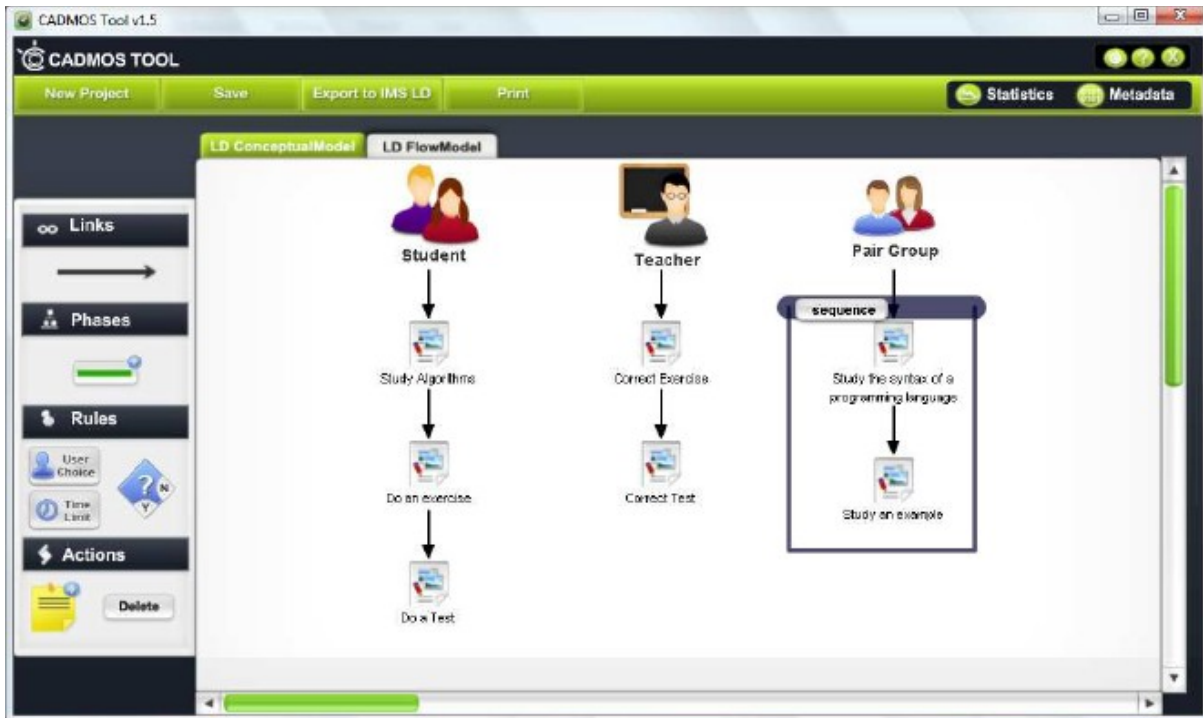


Figure 12 : Exemple de modèle de flux de CADMOS [Katsamani et Retalis 2011]

2.3.2.3. LAMS (Learning Activity Management System)

LAMS (*Learning Activity Management System*) [Dalziel 2003] est un environnement de conception et d'implémentation des situations d'apprentissage collaborative où les apprenants réalisent leurs activités dans un environnement interactif. Il propose, en se basant sur son propre modèle source, un outil graphique auteur (*graphical authoring tools*) d'édition et de monitorat des séquences d'activités. Cet outil permet de créer facilement un scénario pédagogique et un moteur d'exécution pour jouer ces scénarios. Il propose trois espaces pour la conception, le monitorat et l'exécution des activités d'apprentissage.

- L'espace auteur de LAMS (figure 13) permet aux enseignants de concevoir graphiquement des situations d'apprentissage sous forme des séquences d'activités enchaînées en utilisant des concepts graphiques. Les séquences d'activités peuvent contenir différents services tels que chat, forum, etc. et différents contenus tels que QCM, URL, textes, images, etc.
- L'espace moniteur permet d'inscrire les participants aux activités/contenus du scénario. Il permet au moniteur, enseignant par exemple, de lancer et de suivre, en temps réel, le déroulement du scénario et le niveau d'avancement de chaque apprenant ainsi que ses contributions.
- L'espace apprenant permet aux apprenants de jouer/exécuter le scénario prévu par l'enseignant-concepteur et de réaliser leurs activités de façon séquentielle, individuelle ou collective.

LAMS peut être utilisé comme un outil de conception indépendamment de toute plateforme d'exécution. Toutefois, il peut être utilisé de façon intégrale avec certaines plateformes libres [Lengyel et Herdon, 2009]. Par exemple, un scénario pédagogique peut être spécifié en utilisant l'outil LAMS avant d'être importé sur la plateforme ou associé à un cours particulier en tant qu'une activité de type LAMS (ensemble d'activités enchaînées pour les apprenants). L'intégration entre LAMS et une plate-

forme particulière est assurée par l'intermédiaire d'un outil spécifique permettant de synchroniser les deux systèmes. Par exemple, le module (*MOODLE Tool Adapter*) assure la communication entre LAMS et la plateforme Moodle par le biais des services Web. Ce module est capable d'interroger la base de données de Moodle et d'établir les manipulations nécessaires.

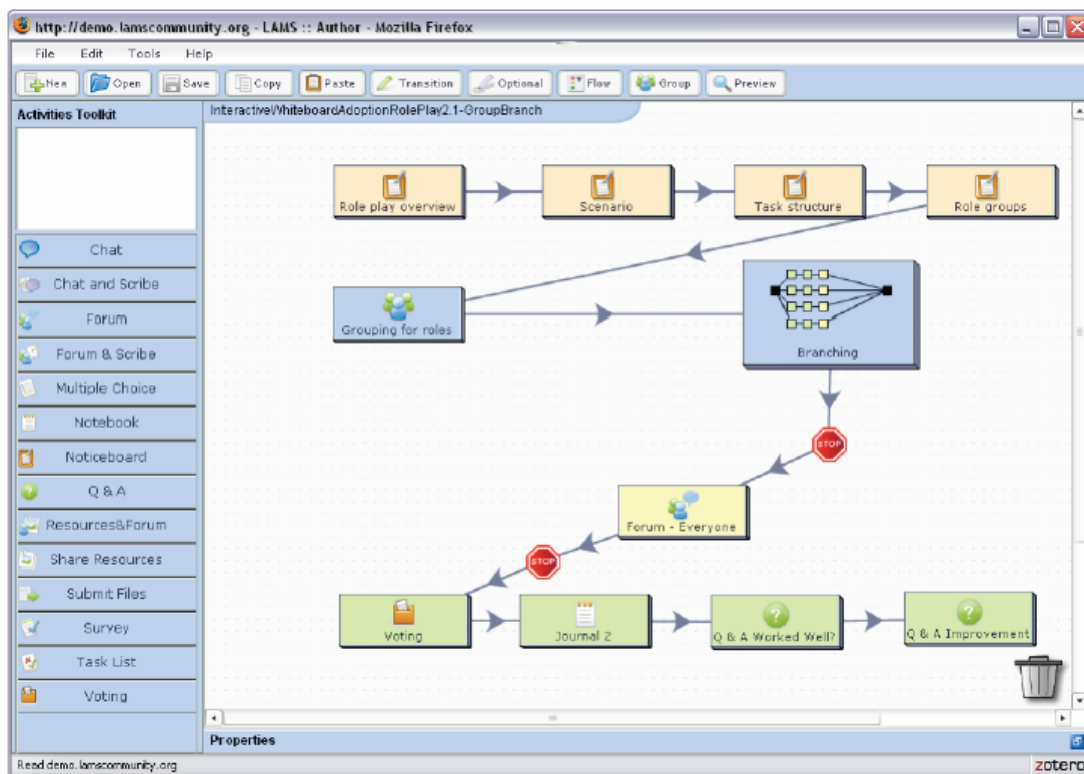


Figure 13 : Ecran de l'environnement auteur LAMS [Dalziel 2010]

2.3.2.4. Analyse des approches centrées sur les langages de modélisation pédagogique

Les approches centrées sur les langages de modélisation pédagogique propriétaires sont apparues suite à la complexité de la conception basée sur les standards. Ces approches cherchent à se rapprocher des pratiques des enseignants et de leurs besoins. Aujourd'hui, il existe un nombre conséquent de langages de modélisation pédagogique (EML), dont certains d'entre eux proposent une notation graphique (VIDL).

LDL propose un métamodèle de conception plus flexible qu'IMS-LD. Cependant, la métaphore du théâtre implicite du LDL impose une structure de modélisation spécifique. Cela nécessite l'appropriation de ce langage et également de sa syntaxe abstraite qui peut être étrangère à des enseignants-concepteurs. LDL s'intéresse à la conception pédagogique indépendamment des plateformes d'exécution mais il propose une infrastructure d'opérationnalisation que nous présentons plus loin dans ce document.

CADMOS a proposé une conception spécifique de scénarios pédagogiques. Cette conception est basée sur deux modèles pour la description et l'orchestration des activités et les ressources d'apprentissage. L'approche CADMOS permet par l'outil dédié la spécification des modèles conformes à des langages ou spécifiques à des plateformes. Cependant, CADMOS ne propose pas une méthode

d'explicitation de ces langages afin de les utiliser pour d'autres finalités (comparaison de plateformes, des éditeurs graphiques spécifiques, etc.). La description des éléments du modèle en termes d'attributs et de propriétés est différente de celle de la plateforme (Moodle par exemple).

LAMS propose son propre environnement pour concevoir des situations d'apprentissage. La convivialité de l'interface graphique de LAMS facilite la tâche des enseignants-concepteurs. L'utilisation d'un outil graphique de conception externe de la plateforme nous semble intéressante. Cependant, le langage de LAMS n'est pas complètement compatible avec le langage des plateformes utilisées. À l'exception de certains concepts (forum, chat, etc.), LAMS ne propose pas tous les éléments de conception pédagogique d'une plateforme parce qu'il est créé pour la conception des situations d'apprentissage collaboratives sans relation avec une plateforme d'exécution. Malgré les efforts d'intégration de LAMS au sein de certaines plateformes, cette incompatibilité requiert des transcriptions des scénarios/cours pour leur opérationnalisation.

Nous avons constaté qu'un nombre important des approches EMLs et VIDLs s'intéressent à la conception pédagogique à la fois externe et indépendante de toutes les technologies d'exécution (plateformes de formation, systèmes LMS, etc.). Elles ne prennent pas en compte le langage de conception pédagogique implicite des LMS.

Notre hypothèse est que chaque dispositif d'apprentissage dispose d'un langage de conception pédagogique propre. Ce langage représente les règles et les éléments de conception des situations d'apprentissage sur ces dispositifs ainsi que leurs attributs et leurs propriétés. Il décrit la structuration des éléments des situations, les activités et les ressources pédagogiques, les intervenants en termes d'apprenants, de tuteurs, de groupes d'apprenants, etc.

Notre travail s'inscrit dans une perspective similaire à LDL en proposant un métamodèle de conception des situations d'apprentissages et également à LAMS et CADMOS en proposant un VIDL et un environnement de conception pédagogique graphique externe des plateformes utilisées. Cependant, nous situons notre approche, le VIDL et l'environnement de conception dans un domaine centré plateforme. Plus spécifiquement, nous nous basons sur le langage de conception pédagogique de la plateforme utilisée après sa formalisation dans un format explicite. Notre objectif est de proposer un VIDL et un environnement de conception pédagogique conforme à ce langage. Cet environnement peut permettre de dépasser les difficultés de conception sur les plateformes de formation ainsi que les contraintes techniques qu'elle impose.

2.3.3. Approches par l'Ingénierie Dirigée par les Modèles

Après avoir présenté dans section 2.2 les fondamentaux de l'Ingénierie Dirigée par les Modèles, cette section est consacrée à la présentation de trois initiatives françaises d'approches de scénarisation centrées IDM. Nous allons présenter le langage CPM et les deux projets MDEduc et Bricoles du point de vue de la scénarisation pédagogique. Ensuite, au sein de la section 2.5.4, nous les présentons du point de vue de l'opérationnalisation.

2.3.3.1. CPM (Cooperative Problem-based learning Metamodel)

Le langage de modélisation CPM (*Cooperative Problem-based learning Metamodel*) est issu des travaux du laboratoire LIUPPA (*Laboratoire Informatique de l'Université de Pau et des Pays de l'Adour*) qui proposent l'exploitation de la méta-modélisation UML pour la conception et la mise en œuvre des situations-problèmes coopératives [Laforcade 2004] [Laforcade et al. 2005] [Nodenot 2005]. Le langage CPM permet d'exprimer les scénarios pédagogiques, en particulier des situations d'apprentissage de type PBL (*Problem Based Learning*) sous la forme d'un profil UML. Il est composé de deux syntaxes. La syntaxe abstraite est décrite par le métamodèle CPM [Nodenot et al. 2003] qui décrit les concepts et leurs relations. La partie gauche de la figure 14 illustre les paquetages composant le métamodèle CPM qui repose sur plusieurs paquetages dont les deux principaux sont le paquetage de fondation, représentant un sous-ensemble du métamodèle UML, et le paquetage des extensions regroupant tous les concepts, relations et contraintes du langage. La syntaxe concrète est décrite par le profil CPM (partie droite de la figure 14) [Laforcade et al. 2003] qui définit le système de notation des concepts et des relations. Il est composé d'éléments tels que le stéréotype, la définition de valeur marquée, la notation, etc. [Nodenot 2005] [Laforcade 2004].

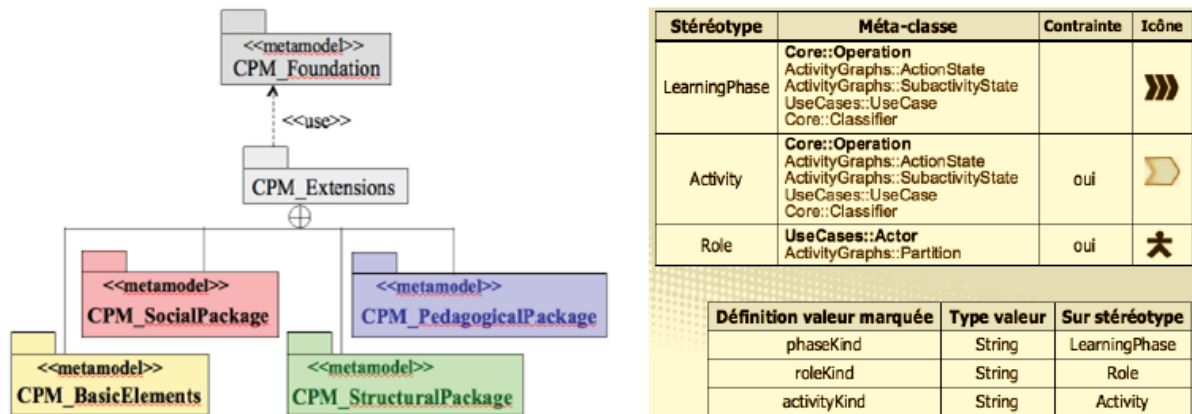


Figure 14 : Paquetage du métamodèle CPM et extraits du profil CPM [Laforcade 2004]

2.3.3.2. Le projet MDEduc

Le projet MDEduc a été mené par l'équipe de recherche NOCE (Nouveaux Outils pour la Coopération et l'Éducation) du laboratoire LIFL de l'université Lille 1, dans le cadre des travaux de thèse de C.O de Moura Filho [De Moura Filho 2007]. Ce projet a consisté à impliquer les enseignants 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. L'approche MDEduc a consisté à permettre aux enseignants de spécifier des scénarios pédagogiques basés sur des patrons pédagogiques écrits en langage naturel [De Moura Filho et Derycke 2007].

Cette approche a proposé un éditeur PPEditor (*Pedagogical Patterns Editor*) (cf. figure 15) pour faciliter l'édition des scénarios pédagogiques en respectant la syntaxe imposée par les patrons pédagogiques. MDEduc s'est inscrit dans une Ingénierie Dirigée par les Modèles. Les patrons sont utilisés comme base d'édition du modèle narratif d'un scénario pédagogique (le niveau CIM en MDA). Le métamodèle formel permettant l'expression d'un scénario au niveau PIM est construit après certaines

interactions (des éditions, des sélections, etc.) menées sur le patron. Il est utilisé pour la définition du modèle PSM par transformation et ainsi la génération du dispositif de formation correspondant.

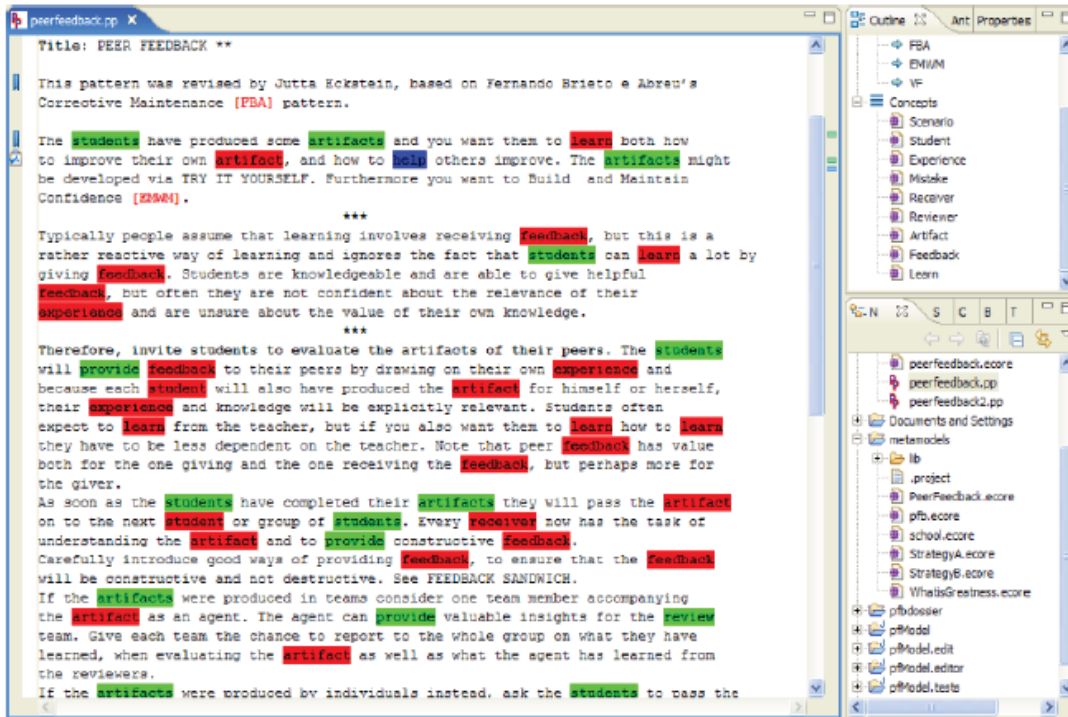


Figure 15 : Editeur de scénarios à base de patrons pédagogiques [De Moura Filho 2007]

2.3.3.3. Le projet Bricoles

Le projet Bricoles [Caron 2007b], mené également par l'équipe de recherche NOCE, a proposé une approche de scénarisation dite "dispositive". Cette approche, basée sur l'Ingénierie Dirigée par les Modèles, s'est adressée à des enseignants dont leur pratique en scénarisation est qualifiée de "bricolage" [Caron 2007a] [Caron et al. 2007]. L'approche Bricoles, comme le montre la figure 16, a consisté à définir une infrastructure qui permet de construire une application Web 2.0 simple via des services. La conception des situations d'apprentissage est guidée par le métamodèle de la plateforme qui décrit les concepts pédagogiques et fonctionnels de l'environnement d'usage. Ce métamodèle est le résultat d'application d'une série d'actions et de pratiques. Pour cela, [Caron 2007b] a évoqué deux méthodes basées sur des transformations semi automatique et la fusion des modèles. Selon [Caron 2007b], l'élaboration de ce métamodèle requiert non seulement des spécialistes de la plateforme mais également des spécialistes des usages de cette plateforme.

L'approche Bricoles est composée de cinq étapes (figure 16) :

- La conception : il s'agit de définir le métamodèle décrivant les concepts pédagogiques et fonctionnels de l'environnement d'usage.
- La définition des outils spécifiques : il s'agit de définir des outils d'édition de modèles de dispositifs pédagogiques (le cas du modeleur *ModX*) et de contextualisation de dispositifs (le cas du constructeur *GenDep*).
- La modélisation : il s'agit de modéliser le dispositif pédagogique au sein d'un modeleur.

- La contextualisation : il s'agit de compléter le modèle produit au contexte d'enseignement et par rapport aux éléments présents sur l'application.
- La construction d'un dispositif : il s'agit de créer les éléments décrits par le modèle sur la plateforme cible via des appels de services.

La mise en œuvre des situations d'apprentissage sur la plateforme cible est effectuée via leur transformation en un modèle opérationnel. Nous présentons et discutons davantage cette partie dans la section 2.5.4.1.

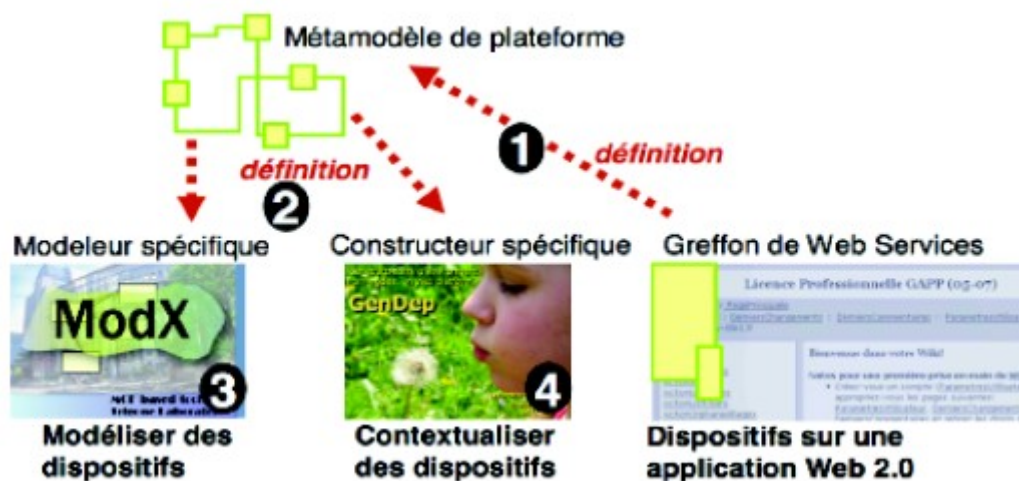


Figure 16 : Approche de conception dans le projet Bricoles [Caron 2007a]

2.3.3.4. Discussion

L'idée commune qui sous-tend les travaux présents en IDM porte sur l'utilisation d'un métamodèle prédéfini (le cas de CPM) pour concevoir des situations d'apprentissage ou sur l'implémentation d'un nouveau métamodèle respectant certaines contraintes (le cas des patrons pédagogiques dans l'approche MDEduc).

CPM exploite la richesse du langage UML afin de proposer des profils UML pour la modélisation de situations pédagogiques. La scénarisation basée sur CPM souligne la nécessité d'un ingénieur pédagogique pour mener le travail de modélisation en raison de la notation UML considérée comme trop complexe pour la faire manipuler à des enseignants. [Laforcade 2005] a affirmé que les transformations des modèles CPM vers IMS-LD, nécessitant la détermination de la méta-classe et du stéréotype équivalent à chaque concept d'IMS-LD, ne sont pas une tâche aisée.

En ce qui concerne le projet MDEduc, l'originalité des travaux de [De Moura Filho 2007] porte sur la définition des langages de modélisation adaptables aux besoins des enseignants-concepteurs. Il propose une démarche qui se veut plus accessible à des enseignants en s'appuyant sur le concept de patron de conception pédagogique. Cependant, MDEduc souligne le risque de la distance qu'il peut y avoir entre pédagogues et développeurs de logiciels [De Moura Filho et Derycke 2007]. Ce projet a proposé un prototype de personnalisation du patron pédagogique, afin d'en extraire un scénario et son métamodèle formel. Ce prototype n'a pas été testé par les enseignants-concepteurs.

Bricoles a démontré la faisabilité technique de l'IDM appliqué à la scénarisation pédagogique. L'approche « dispositive » conduit à étudier la contextualisation du scénario pédagogique par la prise en compte des spécificités de la plateforme d'exécution. Cependant, cette approche peut paraître complexe. L'usage du modéleur ModX, qui propose une personnalisation du langage de notation graphique, prouve ici que ces difficultés sont liées à la complexité du métamodèle de conception pédagogique et non pas à la syntaxe concrète du langage.

Malgré le fait que CPM, MDEduc et Bricoles soient basées sur des métamodèles, la modélisation des situations pédagogiques étaient indépendantes du dispositif d'exécution. Ceci peut offrir plus de flexibilité pour les enseignants-concepteurs mais moins de corrélations avec ces dispositifs. Notre travail s'inscrit également dans une approche centrée sur la métamodélisation. Nous nous intéressons davantage aux plateformes de formation qui représentent le support d'exécution des scénarios conformes à ces métamodèles. Nous nous situons dans un univers techno-centré sur les plateformes, en particulier centré sur son langage de conception pédagogique. Un de nos objectifs est de proposer une conception de scénarios spécifiques à des plateformes de formation et d'assurer ultérieurement leur opérationnalisation.

2.4. Les plateformes de formations à distance

La diffusion de la formation à distance et de l'apprentissage en ligne requièrent des artefacts capables de charger et d'exécuter les spécifications de situations d'apprentissage et les scénarios pédagogiques. Aujourd'hui, il existe plusieurs Environnements Informatiques pour l'Apprentissage Humain (EIAH) de type plateforme de formation à distance et ou système LMS (*Learning Management System*). Un nombre important d'universités ont adopté ces environnements comme dispositifs technologiques à travers lesquels une large gamme d'outils pédagogiques et administratifs est offerte pour l'enseignement et l'apprentissage [Coates et al. 2005]. Leur objectif est d'assurer la formation à distance et de compléter les cours en face-à-face avec quelques activités supplémentaires telles que l'accès aux ressources, les évaluations en ligne, etc.

Dans cette section, nous nous intéressons à la définition de ces environnements, la conception pédagogique sur ces environnements et les difficultés engendrées.

2.4.1. Définitions des plateformes de formation à distance

Dans la littérature, plusieurs définitions sont attribuées aux plateformes de formation à distance. [Oubahssi 2005] définit une plateforme de formation à distance comme étant *"un logiciel de création et de gestion de contenus pédagogiques et de supports aux acteurs destiné à trois types d'utilisateurs : L'enseignant, l'apprenant et l'administrateur. Elle regroupe les outils nécessaires aux trois types d'intervenants permettant d'incorporer des ressources pédagogiques multimédias, de participer à des activités et d'effectuer le suivi pédagogique et administratif des apprenants"*.

La plateforme de formation à distance est définie selon [Marchand et Depover 2002] comme *"un dispositif qui permet de distribuer et de gérer des formations à distance. Il comporte généralement des fonctionnalités en matière de télé-tutorat et de production de matériel d'apprentissage. Certaines plateformes disposent de possibilités permettant d'échanger facilement des informations avec d'autres"*

plateformes mais aussi avec les autres systèmes informatiques dont dispose l'entreprise. Les plateformes permettent notamment la création de campus virtuels consacrés aux besoins particuliers de l'entreprise".

Quant à l'Office de la Langue Française, il considère la plateforme de formation comme "un système informatique destiné à automatiser les diverses fonctions relatives à l'organisation des cours, à la gestion de leur contenu, au suivi des progrès des participants et à la supervision des personnes responsables des différentes formations" [OLF 2013]. En tenant compte qu'actuellement les activités pédagogiques sont davantage mises en avant, [George et Derycke 2005] complètent la définition du [OLF 2013] par : "une plate-forme peut être vue comme un système qui permet de gérer et de donner accès à un ensemble d'activités et de ressources pédagogiques".

Certaines dénominations ont été attribuées aux plateformes de formation dépendamment des fonctionnalités et des services proposées. Il s'agit des cinq sigles LMS (*Learning Management System*), CMS (*Content Management System*), LCMS (*Learning Content Management System*), C3MS (*Content, Community, and Collaborative Management System*) et LKMS (*Learning Knowledge Management System*). Ces sigles sont définis au sein de [Ertoran 2006] [Greenberg 2002]. Par exemple, la plateforme Moodle est à la fois un LMS et un C3MS.

Nous avons remarqué qu'un nombre important de ces définitions sont centrées sur des éléments caractéristiques des plateformes de formation. Elles décrivent leurs cas d'usages génériques tels que le support la formation à distance et la mise à disposition des supports et des activités pédagogiques, des moyens de communication, etc.

2.4.2. Exemples de plateformes de formation

Actuellement, il existe un nombre important de plateformes de formation. Dans cette section, nous présentons trois exemples de plateformes. Nous avons appliqué nos propositions et nos expérimentations sur deux d'entre elles (Moodle et Ganesha).

Moodle : la plateforme Moodle (*Modular Object Oriented Dynamic Learning Environment*) est créée en 2002 par Martin Dougiamas⁵ dans le cadre de son doctorat portant sur les apports du constructivisme social dans la pédagogie en ligne. Moodle est une plateforme libre (open-source), modulaire et extensible. Elle possède une large communauté d'utilisateurs et de développeurs et elle est utilisée dans de nombreux contextes éducatifs (lycées, universités, formations internes d'entreprises, etc.). Influencée par la pédagogie socio-constructiviste, Moodle propose des espaces de travail pour plusieurs usagers (apprenants, enseignants, créateurs de cours, administrations, invités, etc.). Elle permet la création de cours en ligne, la gestion des ressources pédagogiques et des activités d'apprentissage interactives, la communication (synchrone ou asynchrone) entre les acteurs du cours, etc.

Ganesha : Ganesha est une plateforme de formation open source (licence GPL) éditée depuis 2001 par la société ANEMA. Ganesha est développée sous la base d'une application Web (léger) personnalisable et compatible avec la plupart des navigateurs Internet et des serveurs (Windows, Mac OS, Li-

⁵ <http://dougiamas.com> (consulté le 14/03/2012)

nux). Elle intègre quelques standards d'elearning tels que SCORM 1.2, SCORM 2004, AICC, PENS et LOM. Elle supporte l'apprentissage individuel et collaboratif.

Claroline : Claroline est une plateforme de formation open source, née en 2001 à l'initiative de l'UCLouvain (Belgique), proposant un outil de soutien à l'apprentissage collaboratif en ligne [Claroline 2012]. Elle s'appuie sur un modèle pédagogique selon lequel l'information devient connaissance grâce aux activités et aux productions des apprenants. Elle dispose d'un ensemble d'outils permettant aux formateurs ou aux apprenants de mettre en place ou d'exploiter un dispositif pédagogique d'apprentissage.

2.4.3. La conception pédagogique sur les plateformes de formation

L'objectif des dispositifs d'apprentissage tels que les plateformes de formation à distance est d'assurer la formation à distance ou de compléter les cours présentiels avec la mise en place de situations d'apprentissage ou de quelques activités supplémentaires telles que l'accès aux ressources, les évaluations en ligne, etc. La conception d'une situation sur ces artefacts consiste à gérer et à s'approprier les différents écrans de configuration du cours ou d'un élément du cours. Elle porte sur le paramétrage de plusieurs interfaces basées sur des formulaires que les praticiens doivent spécifier ou saisir.

Les enseignants-concepteurs souhaitent concevoir des situations d'apprentissage de haut niveau. Ce challenge impose de paramétrer soigneusement/attentivement les fonctionnalités fournies par les plateformes. La conception pédagogique sur les plateformes de formation peut être décrite par les trois étapes de l'opérationnalisation manuelle d'un scénario pédagogique, proposées par [Martel 2006] :

- Choisir les activités ;
- Choisir les participants et les rôles correspondant ;
- Sélectionner les services et les contenus ou les ressources.

Par exemple, la spécification d'un cours d'apprentissage sur la plateforme Moodle est basée sur la configuration de nombreux écrans et d'interfaces basées sur des formulaires. La première interface de Moodle montre une liste de paramètres (nom complet, nom court, le numéro ID de cours, résumé, format de cours, nombre de semaines, les nouveautés, etc.) que les praticiens doivent ajuster. Moodle propose plusieurs formats de cours (hebdomadaire, thématique, informel). Le format thématique, à titre d'exemple, permet de définir un cours comme une liste de sections. Par défaut, Moodle crée le forum des nouvelles dans la première section (section 0). Puis, le concepteur doit définir le contenu des sections du cours. Chaque section peut contenir une ou plusieurs activités (des tests, forums, chat, sondage, wiki, etc.) et/ou ressources (des pages Web, les fichiers et répertoires). Moodle offre également des questionnaires de nombreux types (par exemple à choix multiple, numérique, vrai / faux, calculé, etc.) avec la possibilité de mener des auto-évaluations, des sondages, des tests de qualification, etc.

Ces concepts (interfaces basées sur les formulaires, activités, ressources, etc.) désignent le langage de conception pédagogique implicite de la plateforme Moodle que nous voulons identifier. Le processus d'identification de ce langage, représentant une de nos propositions, est détaillé ultérieurement au sein du quatrième chapitre.

2.4.4. Difficultés de la conception pédagogique sur les plateformes

La conception des situations pédagogiques sur les dispositifs d'apprentissage tels que les plateformes de formation ou les systèmes LMS n'est pas une tâche aisée. De nombreux enseignants-concepteurs rencontrent des difficultés pendant l'utilisation des plateformes et la conception de scénarios pédagogiques [Steel 2009]. Ils ne sont pas habitués avec le langage implicite de conception pédagogique [Ortiz et al. 2009a] et ne sont pas en mesure de mettre en œuvre les scripts requis par les plateformes [Mekpiroona et al. 2008]. Le challenge principal porte sur la définition des fonctionnalités en accordance avec leur niveau d'expertise des LMS et leurs compétences en conception pédagogique. La conception pédagogique sur les plateformes n'est pas flexible et impose un paradigme bien spécifique. Elle dépend directement de la compréhension des praticiens et de la façon de "penser et de concevoir" de ces plateformes. Par exemple, la spécification du cours sur Moodle nécessite la gestion et l'appropriation de nombreux écrans et interfaces. Ces interfaces proposent de longues listes de paramètres de configuration dont certains paramètres sont importants pour la spécification du cours, mais d'autres sont facultatifs ou concernent des détails techniques à un niveau stratégique très bas.

Les enseignants, non impliqués dans la technologie, risquent d'avoir des difficultés d'expression de leurs intentions sur les dispositifs pédagogiques. Ils peuvent choisir de ne pas utiliser ces dispositifs ou de les utiliser d'une manière incompatible avec leurs intérêts pédagogiques, parce qu'ils n'aperçoivent pas la technologie bien adaptée à leur approche pédagogique. Ces difficultés engendrent une utilisation minimale ou incohérente de la plateforme qui sera ainsi exploitée comme un système de dépôt de support de cours (des pages web, des fichiers pdf, etc.) [Steel 2009]. Un nombre important de praticiens et d'enseignants-concepteurs ne savent pas comment l'utiliser efficacement conformément à son potentiel. Souvent, ils sollicitent l'intervention d'un ingénieur pédagogique ou d'un expert de la plateforme pour effectuer les manipulations nécessaires sur la plateforme cible. Cette solution peut engendrer des inconvénients à cause du manque de communication entre le concepteur du cours et le technicien de la plateforme, du manque de disponibilité de l'expert, etc.

Les enseignants, comme ceux de l'université du Maine et de Paris Descartes, réclament des formats de cours plus simples et plus facile à configurer. Ils sollicitent des outils de conception plus facile à utiliser [Steel 2009] [Ferraris et al. 2009] [ParisMoodle 2009].

2.5. Travaux menés sur l'opérationnalisation des scénarios pédagogiques

Après avoir présenté (1) les plateformes de formation et les LMS en tant que dispositifs d'apprentissage en ligne ou à distance et (2) certains travaux de scénarisation, nous allons nous intéresser dans cette section à l'opérationnalisation des scénarios pédagogiques sur ces dispositifs. Plusieurs travaux dans la communauté EIAH ont déjà abordé l'activité d'opérationnalisation. Leur point commun porte sur l'importance de mettre à disposition des praticiens ou des enseignants-concepteurs, une solution pratique facilitant cette activité.

Dans cette section, nous présentons l'opérationnalisation en tant qu'activité de conception pédagogique. Nous donnons une classification générique des approches existantes. Ensuite, nous détaillons et nous discutons les approches d'opérationnalisation automatisées.

2.5.1. Introduction et définitions

Rares sont les travaux qui définissent l'activité d'opérationnalisation des scénarios pédagogiques sur les plateformes de formation à distance. Certains se contentent de présenter leurs principales étapes. L'opérationnalisation représente l'activité de conception pédagogique intermédiaire qui se situe entre la scénarisation et l'exécution de ces scénarios. Elle consiste (1) à configurer les éléments du scénario qui sont déjà fournis par la plateforme et (2) à retranscrire les éléments non supportés par la plateforme avant de suivre leur déroulement exécutif. Nous rappelons que [Martel 2007] récapitule l'opérationnalisation en trois étapes : (1) créer l'ensemble des activités, (2) choisir les participants et attribuer leurs rôles et (3) sélectionner les services et les contenus requis pour le scénario (tels que les ressources mises à disposition des apprenants).

Cette description est assez générique et abstraite. L'opérationnalisation des scénarios pédagogiques ne se réduit pas à un travail d'ingénieur de création de ressources, de participants au cours et de liens entre eux. Elle consiste à rendre le scénario décrit *a priori* par l'enseignant, utilisable et manipulable sur un dispositif d'apprentissage tout en préservant les intentions et les sémantiques pédagogiques initiales. Nous avons identifié deux approches pour opérationnaliser les scénarios pédagogiques : une approche 'manuelle' et une approche 'automatisée'.

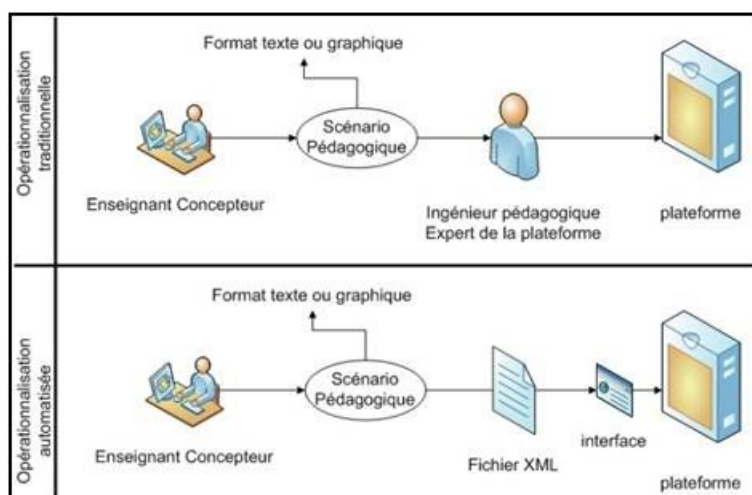


Figure 17 : Opérationnalisation manuelle versus automatisée des scénarios pédagogiques

- L'approche manuelle : la plupart des scénarios sont informatisés de manière traditionnelle "lorsque l'environnement avec les outils nécessaires est simplement mis à la disposition des acteurs, en leur laissant la responsabilité d'effectuer leurs activités conformément au scénario qui leur est transmis textuellement" [David et al. 2008]. Le concepteur se charge d'effectuer, à la main, les manipulations nécessaires sur la plateforme cible comme la création des utilisateurs, l'attribution des rôles aux acteurs, l'intégration des ressources pédagogiques, etc. Cette approche exige, souvent, l'intervention d'un ingénieur pédagogique ou bien d'un expert de la plateforme. La coopération entre l'expert de la plateforme et le concepteur est fortement recommandée pour garantir les finalités didactiques du scénario. La partie supérieure de la figure 17 illustre cette approche.
- L'approche automatisée : elle consiste à implémenter le scénario pédagogique spécifié à l'extérieur de la plateforme de manière automatisée sur un Environnement Informatique

d'Apprentissage Humain. Contrairement à l'approche manuelle, cette approche permet de réduire l'intervention des experts de la plateforme ou des ingénieurs pédagogiques. Un outil ou un service particulier peut se charger des manipulations nécessaires. Ensuite, certaines manipulations complémentaires peuvent être réalisées directement sur l'environnement. La partie basse de la figure 17 illustre cette approche.

Nous nous intéressons dans les sections suivantes aux approches qui visent à automatiser l'opérationnalisation des scénarios pédagogiques sur les plateformes de formation à distance. Ces approches peuvent être classifiées en quatre catégories : les approches propriétaires des chaînes éditoriales, les approches propriétaires par les EMLs, les approches propriétaires par les plateformes et les approches par l'ingénierie dirigée par les modèles.

2.5.2. Approches des chaînes éditoriales

Les chaînes éditoriales sont des dispositifs informatiques dont le but est d'optimiser la production et la gestion de contenus documentaires ainsi que l'industrialisation de la production des contenus pédagogiques. Le procédé des chaînes éditoriales est fondé sur la modélisation, la structuration logique, la scénarisation et la publication sur un support interactif de contenus pédagogiques [Bachimont et al. 2002].

La chaîne éditoriale SCENARI (Système de Conception d'Enseignements Numériques Adaptables, Réutilisables et Interactifs) [Bachimont et al. 2002] [Crozat et Gebers 2009] est une solution technologique et méthodologique permettant l'application d'une ingénierie documentaire à la conception de contenus pédagogiques. Elle permet d'articuler et d'instrumenter les métiers de modélisation pédagogique, de production de contenus structurés et scénarisés, et d'édition des contenus sur les supports de distribution.

L'ensemble de la chaîne éditoriale s'appuie sur la technologie XML, ce qui permet de bénéficier de l'indépendance vis-à-vis des formats propriétaires, de la maîtrise de la production par des non-informaticiens, de la souplesse et de l'adaptation aux besoins pédagogiques spécifiques, et de la transformation automatique des contenus dans de multiples formats (HTML, PDF, etc.). La chaîne éditoriale, comme le montre la figure 18, se situe donc en aval de l'ingénierie de formation, qui traduit les attentes et les besoins des utilisateurs, et en amont d'une plateforme de gestion de la formation.

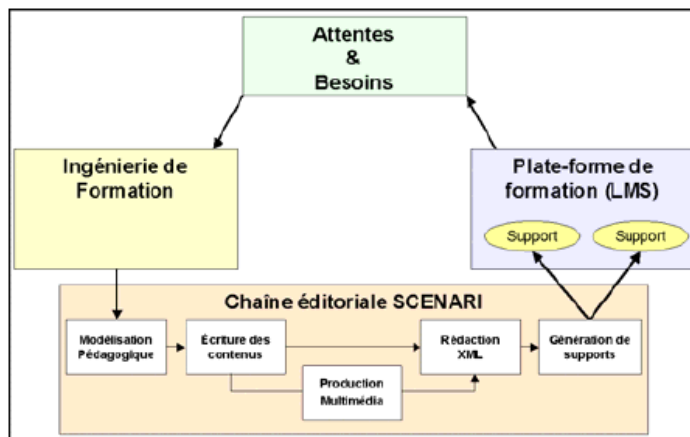


Figure 18 : Processus de conception SCENARI [Bachimont et al. 2002]

Les approches propriétaires des chaînes éditoriales représentent un exemple concret de l'opérationnalisation des scénarios pédagogiques en préservant leur sémantique et sans perte d'informations. Ces chaînes représentent des DSMLs associés à des plateformes spécifiques. Nous pensons que l'idée d'externaliser la conception des situations d'apprentissage des plateformes est utile car elle permet de dépasser les contraintes techniques imposées par les plateformes et de s'intéresser exclusivement à la conception pédagogique. Ensuite, la communication entre ces deux artefacts (la chaîne éditoriale et la plateforme) favorise la mise en place de situations spécifiées sur la chaîne éditoriale sur la plateforme concernée. Cependant, tel Scenarii, ces chaînes sont généralement payantes et dédiées à une plateforme spécifique. Elles ne présentent pas leur langage de conception pédagogique pour l'exposer comme un format de comparaison avec d'autres plateformes ou pour la spécification d'autres outils de conception.

2.5.3. Approches propriétaires par les EMLs

Les approches propriétaires par les EMLs permettent aux enseignants d'opérationnaliser, d'exécuter et de jouer les scénarios spécifiés par des EMLs particuliers dont les concepts sont généralement incompatibles avec les artefacts informatiques supports.

A titre d'exemple, CopperCore est le premier moteur d'exécution de scénarios pédagogiques écrits en IMS-LD [IMS-LD 2012] [Coppercore 2012]. L'objectif de cet outil auteur est de rendre les scénarios spécifiés en IMS-LD opérationnels et de les exécuter sur certaines plateformes capables de dialoguer avec lui via SOAP (Simple Object Access Protocol) [W3C 2000]. CopperCore se charge de la validation des scénarios spécifiés en langage IMS-LD, de leur opérationnalisation et de l'ordonnancement de leur exécution. Il charge le paquetage contenant les fichiers XML décrivant le scénario grâce à son outil publisher. Il parcourt le paquetage du scénario et le valide afin de le publier dans CopperCore. Tout d'abord, les participants et leurs rôles sont définis grâce à l'outil Clicc (*Command Line Interface CopperCore*). Ensuite, l'outil Webplayer permet aux participants de jouer le scénario.

Pour supporter l'opérationnalisation, et souvent l'exécution des modèles spécifiés via les EMLs, certaines approches propriétaires par les EML mettent à disposition des outils au service des langages et standards comme Reload [Reload 2006], Coppercore [Coppercore 2012], etc. D'autres approches consistent à intégrer aux EML des extensions plus adaptées aux situations pédagogiques rencontrées [Hernandez-Leo et al. 2004] [Gounon et al. 2005] et de les rendre capables de communiquer avec les dispositifs d'exécution (plateformes, etc.). D'autres travaux tels que [Nodenot et al. 2008] [Abdallah 2009] consistent à réaliser un mapping entre les modèles conformes à des EMLs particuliers et les artefacts d'exécution. Ce mapping engendre une perte d'informations et de sémantique parce que les concepts de ces EMLs et de ces artefacts ne sont pas définis de façon réciproque. Pour cela, nous pensons qu'il est important de tenir compte des concepts, des fonctionnalités et des services embarqués dans les dispositifs d'exécution.

2.5.4. Approches propriétaires par les plateformes

2.5.4.1. L'infrastructure LDI

Dans l'objectif d'opérationnaliser les scénarios écrits en LDL (*Learning Design Language*), les laboratoires SysCom et CLIPS-IMAG ainsi que la société Pentila ont développé une infrastructure dédiée à ce langage, intitulée LDI (*Learning Design Infrastructure*). Cette infrastructure est capable d'importer/opérationnaliser les scénarios LDL sauvegardés dans un format XML.

L'opérationnalisation d'un scénario LDL nécessite plusieurs transformations. Le « *player* » de LDI ordonnance le déroulement des activités ainsi que l'interaction entre les participants lors de l'exécution du scénario.

La figure 19 illustre le processus de conception, d'opérationnalisation et d'exécution de scénarios pédagogiques. Elle décrit également les trois étapes d'opérationnalisation suivantes :

- La réécriture de scénario : cette étape représente le passage entre la phase de conception et d'opérationnalisation. Elle consiste à transcrire le scénario, en tenant en compte des services spécifiés, afin de le formuler dans un format opérationnalisable.
- L'instanciation : cette étape consiste à réifier le scénario. Les éléments constituant le scénario tels que les activités sont créés et initialisés sur l'artefact d'exécution.
- L'initialisation : cette étape consiste à (1) créer les participants, (2) leur attribuer des rôles et (3) choisir, initialiser et configurer les services/fonctionnalités correspondants(es) aux éléments du scénario et fournies par l'artefact d'exécution.

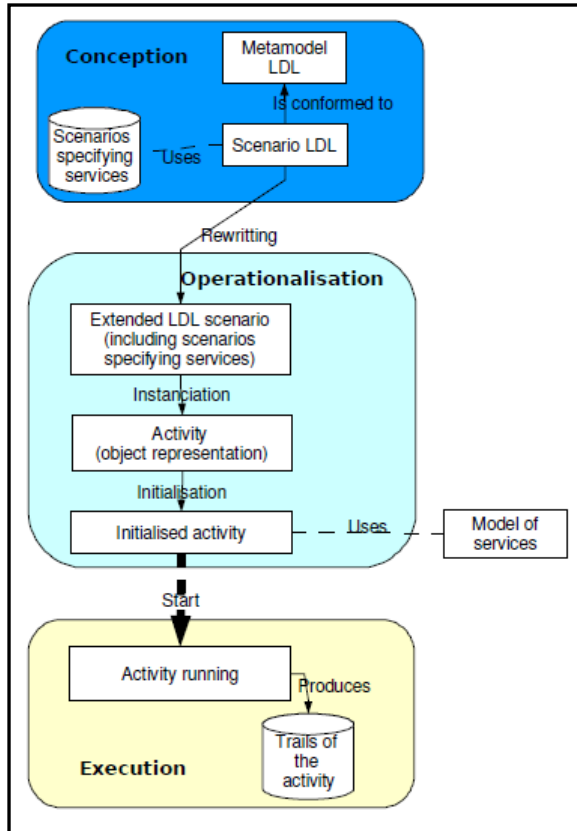


Figure 19 : Processus de conception, opérationnalisation et exécution d'un scénario LDL [Martel et Vignollet 2007]

2.5.4.2. <e-LD>

<e-LD> est une architecture (figure 20) auteur pour l'opérationnalisation des langages de modélisation pédagogique de façon modulaire et flexible [Ortiz et al. 2007]. Elle possède cinq composants :

- Les unités d'apprentissage (*units of learning*) représentent une description formelle d'un processus pédagogique particulier et d'un ensemble de ressources (des contenus, des services, etc.).
- Les outils auteurs (*authoring tools*) sont mis à disposition des concepteurs pédagogiques pour formaliser leurs scénarios en intégrant les unités d'apprentissage.
- La plateforme d'exécution (*execution platform*) représente l'environnement d'exécution du processus d'apprentissage. Elle doit impliquer les usagers du processus, fournir les contenus pédagogiques et intégrer les activités requises et les moteurs workflow qui orchestrent leur exécution.
- Les players (*players*) sont des interfaces graphiques utilisées par les enseignants et les apprenants pour interagir avec les unités d'apprentissage durant leur exécution.
- Les mappages de conception (*design mappings*) sont utilisés pour transformer un modèle pédagogique conforme à un langage (EML) spécifique dans un format exécutable.

Au sein de l'approche <e-LD>, les modèles doivent être formalisés en langage XML. Lorsqu'ils sont spécifiés dans d'autres formats, la *design mapping* permet de les transformer dans un format XML en se basant sur des langages de transformations (comme XSLT) ou bien des frameworks de programmation (comme DOM et SAX). Ensuite, <e-LD> utilise le langage de workflow BPEL4WS⁶ pour l'opérationnalisation de ces modèles via une interface de web services programmable. Les activités sont représentées comme des web services. L'opérationnalisation prend la forme de déploiement des applications dans une architecture orientée service (*SOA : Service Oriented Architecture*) [Hlaoui et al. 2012].

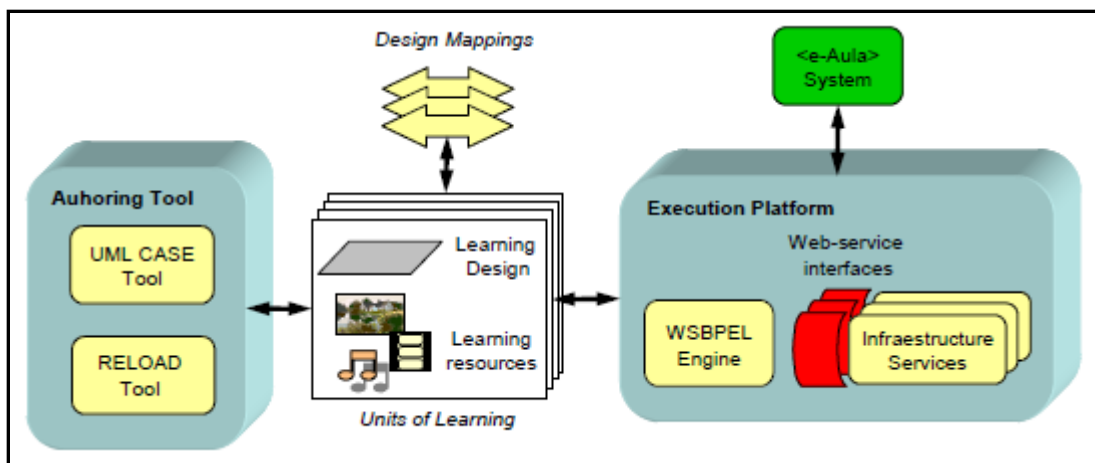


Figure 20 : Implémentation des unités d'apprentissage dans <e-LD> [Ortiz et al. 2007]

2.5.4.3. Discussion

L'objectif des approches propriétaires par les plateformes est de mettre en œuvre des situations d'apprentissage spécifiées via des langages ou des outils particuliers. Nous avons constaté que ces approches consistent à reformuler le scénario initial selon les concepts disponibles sur la plateforme

⁶ Business Process Execution Language for Web Services. Plus de détails sont donnés sur le site : <http://xml.coverpages.org/bpel4ws.html>

d'exécution. LDI définit un processus d'opérationnalisation consistant à instancier et initialiser les éléments du scénario. <e-LD> utilise le workflow des services web pour opérationnaliser les scénarios après leur transformation en scénarios supportables par les plateformes d'exécution. Cependant, ces transformations mènent à des pertes de concepts et de sémantique à cause de la différence entre le langage de conception et le langage d'opérationnalisation. Dans notre approche, nous envisageons de tenir compte du langage d'opérationnalisation pour conserver la sémantique initiale.

2.5.5. Approches propriétaires par l'IDM

2.5.5.1. Bricoles

Le projet Bricoles s'intéresse à la conception des situations d'apprentissage (cf. section 2.3.3.3) mais également à leur mise en œuvre sur un artefact d'apprentissage. Bricole propose l'outil GenDep pour déployer les dispositifs spécifiés via le modèleur MODX sur la plateforme de formation en utilisant les services web. L'interaction entre GenDep et la plateforme se déroule via le protocole SOAP (*Simple Object Access Protocol*). Le déploiement comporte une phase de contextualisation du modèle du dispositif avec les éléments déjà existants sur la plateforme. GenDep consulte les éléments existants sur la plateforme et interroge l'utilisateur qui décide de les associer à des éléments de son modèle, de les inclure comme des nouveaux éléments, ou de les ignorer.

2.5.5.2. MDEduc

Le projet MDEduc n'est pas restreint à un éditeur de spécification du métamodèle de conception et de son modèle associé. Il est "*destiné à la production de scénarios d'apprentissage dans des spécifications informelles et formelles, multi-standards pour ces dernières, et incluant la génération de codes logiciels*" [De-Moura Filho 2007]. MDEduc porte sur une approche patron basée sur l'Ingénierie Dirigée par les Modèles. Les patrons sont utilisés comme base d'édition du CIM d'un scénario. Le modèle CIM servira pour la création du métamodèle PIM après avoir effectué l'ensemble d'interactions avec le concepteur et pour la personnalisation du patron. Le métamodèle PIM est ensuite utilisé pour définir un modèle PSM par transformation, puis le dispositif de formation correspondant est généré.

L'opérationnalisation des scénarios sur des plateformes de formation est basée principalement sur les transformations entre les modèles proposées par MDA. Le modèle PSM du scénario source (MDEduc) est transformé en scénario opérationnel sur cette plateforme. Dans ses travaux, [De-Moura Filho 2007] propose, pour établir ces transformations, d'utiliser l'*Eclipse Modeling Framework* ou des outils supports à l'IDM comme ModX et GENDEP [Caron et al. 2005] développés au sein du laboratoire NOCE après avoir généré le code nécessaire à l'exécution du modèle par l'environnement MDEduc [De-Moura Filho et Derycke 2007].

2.5.5.3. LÉA

Le projet LÉA (Livret Électronique d'Apprentissage) [Cottier et al. 2008] a été mené en collaboration entre le Centre de Formation des Apprentis des Villes de la Mayenne (CFAVM) et le Laboratoire d'Informatique de l'Université du Maine (LIUM). Ce projet a visé à effectuer le suivi de l'avancement des jeunes apprentis pendant leurs formations en entreprise et au CFA. À son usage, LÉA est apparu trop rigide et difficile à utiliser ; par exemple, la mise en œuvre de LÉA nécessite plusieurs connexions sous des rôles différents (administrateur, responsable de formation, etc.). Ce constat a mené [Oubahssi

et al. 2009] [Oubahssi et al. 2010] à repenser la conception de LÉA pour améliorer et faciliter l'utilisation du système. La réingénierie du LÉA porte sur trois points principaux. Le premier point concerne l'élaboration du service de gestion multi-rôle ainsi que la mise en place d'un éditeur interne de configuration. Pour ce faire, une étude fonctionnelle du système a été menée pour définir un nouveau modèle fonctionnel tout en tenant compte des données initiales du système. Le deuxième point concerne le développement d'un éditeur externe pour favoriser la configuration graphique du livret au niveau institutionnel. Le développement a été guidé par les modèles au sens de l'IDM, en appliquant une approche outillée du DSM [Kelly et Tolvanen 2008]. Le dernier point concerne la communication entre l'éditeur externe et le système LÉA. Cette communication est assurée par un module d'import/export des configurations, à travers des fichiers XML, ajouté au système existant.

2.5.5.4. Discussion

MDEduc, Bricoles et LÉA sont trois approches proposant des solutions basées sur l'Ingénierie Dirigée par les Modèles pour l'opérationnalisation des scénarios pédagogiques sur un artefact informatique.

- Dans le cas du projet MDEduc, les travaux de [De-Moura Filho 2007] ont porté principalement sur la proposition d'une approche de conception centrée sur les praticiens. L'opérationnalisation des scénarios pédagogiques est limitée à l'utilisation de l'approche MDA pour établir les transformations entre les modèles ou l'utilisation des outils supports à l'IDM comme ModX et GEN-DEP pour produire un modèle opérationnel. Toutefois, la perte de la sémantique conséquente de cette transformation n'était pas dissimulée.
- En ce qui concerne le projet Bricoles, il vise des petites organisations d'enseignants ne disposant pas d'ingénieur pédagogique et désirant mettre en place une situation d'apprentissage [Caron 2007b]. L'approche d'opérationnalisation a été basée principalement sur les services web. Le modèle de conception, spécifié selon une telle approche ou un tel EML, est tout d'abord transformé en un modèle conforme au métamodèle d'une plateforme d'exécution. Ensuite, un constructeur spécifique, utilisant les services web, est mis à disposition de l'enseignant pour déployer le nouveau modèle. Dans le contexte des expérimentations menées au sein du projet PCDAI avec des enseignants, [Caron 2007b] a souligné que les transformations du modèle révèlent un problème de contextualisation des intentions pédagogiques avec les services de la plateforme. Ces transformations entraînent plusieurs pertes d'informations et de sémantique. Le scénario opérationnalisé n'est pas toujours conforme à celui conçu *a priori* par l'enseignant.
- Quant à LÉA, les travaux de réingénierie ont montré plus de flexibilité sur la conception des configurations institutionnelles des livrets d'apprenants en utilisant un outil graphique. Cet outil représente un nouveau format d'exploitation du métier et du langage embarqué dans LÉA. Cependant, LÉA est un ENT (Environnement Numérique de travail), plus qu'un EIAH. Ainsi, les scénarios mis en œuvre dans le LÉA ne sont pas des scénarios pédagogiques centrés activité au sens d'IMS-LD, mais plutôt des scénarios de suivi.
LÉA était une expérimentation parmi plusieurs de l'approche DSM au sein du laboratoire LIUM [Oubahssi et al. 2010] [Zendagui 2010]. Ce travail de réingénierie a été à l'origine de la proposition de cette thèse. L'idée était d'étendre ces travaux dans le domaine de la réingénierie des EIAH, spécifiquement sur les plateformes de formation à distance et les systèmes LMS.

Nous avons constaté que les travaux basés sur l'approche MDA (Bricoles, MDEduc, etc.) ont validé la possibilité d'opérationnaliser une séquence du scénario pédagogique sur une plateforme particulière. Pour créer un scénario opérationnalisable, ces travaux ont utilisé la transformation des modèles qui peut entraîner une perte d'informations et de sémantique. Bricoles, MDEduc et LÉA se sont basés sur un métamodèle de conception pédagogique. Au contraire de LÉA où la conception des scénarios est basée sur son propre langage, MDEduc et Bricoles ne tiennent pas compte du métier des plateformes lors de la conception pédagogique.

Dans le cadre de notre travail, nous pensons dans un premier temps que l'IDM présente un moyen pour réduire l'écart entre l'espace de modélisation et l'environnement d'exécution. Nous nous focalisons sur l'approche DSM, approche spécifique issue de l'IDM. Cette approche est capable d'exposer le langage de conception pédagogique sous un nouveau format (graphique).

Dans un deuxième temps, notre travail s'inscrit dans une perspective similaire aux travaux de réingénierie de LÉA. Notre hypothèse est que chaque plateforme de formation ou système LMS dispose d'un métier de conception des scénarios pédagogiques. L'explicitation et la formalisation de ce langage représente un élément principal dans notre approche qui doit garantir la sémantique et les concepts du scénario *a priori* prévu par l'enseignant pendant sa mise en œuvre.

2.5.6. Discussion globale

Un nombre significatif de travaux de recherche se sont intéressés à l'opérationnalisation des scénarios pédagogiques sur des artefacts informatiques tels que les plateformes de formation à distance. Nous avons constaté que :

- seulement quelques langages fournis avec des infrastructures d'implémentation (par exemple LDL et LDI) ou certaines approches favorisant la conception interne sur la plateforme (le cas de LAMS par exemple) permettent une opérationnalisation automatisée des scénarios pédagogiques.
- les quelques outils auteurs (*Authoring tools*) visant à faciliter la scénarisation et l'opérationnalisation sont en majorité des prototypes ou bien des chaînes commerciales dédiés à une plateforme spécifique (cas de Scénarii [Crozat et Gebers 2009]).
- malgré la multitude des autres approches (EML, standards, outils, etc.), l'opérationnalisation des scénarios s'effectue soit via leur traduction selon les concepts fournis par la plateforme cible (les approches par l'IDM), soit via l'utilisation d'un service d'import (s'il est fourni par la plateforme). Ce dernier est généralement spécifique à un standard. Par exemple, la plateforme Moodle propose l'import des paquetages SCORM ou IMS-Content. La traduction et les transformations peuvent entraîner des pertes de plusieurs informations à cause de l'incompatibilité des concepts des scénarios et des plateformes.

3

Problématique scientifique

Sommaire

- 3.1. Synthèse des travaux existants et problématique scientifique
- 3.2. Approche de conception centrée plateformes
- 3.3. Méthodologie de recherche

Sur la base des travaux existants sur la conception et l'opérationnalisation des scénarios pédagogiques, les enseignants éprouvent beaucoup de difficultés à utiliser les approches de conception, les EML génériques et leurs outils. Il existe actuellement un profond fossé conceptuel entre les instruments de conception et les dispositifs de formation disponibles tels que les plateformes de formation. La sémantique de leurs langages est difficilement compréhensible : les instruments de conception restent techniquement complexes pour des usagers praticiens et les scénarios sont difficilement opérationnalisables sur ces plateformes.

Dans ce chapitre, nous commençons par la définition de notre problématique scientifique. Nous posons les questions de recherche en nous basant sur le positionnement scientifique. Ensuite, nous présentons notre approche. Le chapitre sera conclu par la méthodologie de recherche adoptée.

3.1. Synthèse des travaux existants et problématique scientifique

Actuellement, les plateformes de Formation Ouverte et À Distance (FOAD) sont largement utilisées et ne sont pas limitées à l'apprentissage intensif et distant. La plupart des organisations académiques permettent à disposition de leurs enseignants des plateformes dont l'objectif est d'améliorer ou de compléter leurs cours présentiels par des activités complémentaires comme l'accès à des ressources ou à des évaluations en ligne. Toutefois, la gestion et l'appropriation des plateformes par les praticiens restent une tâche complexe [Al-Ajlan et Zedan 2008]. Certaines plateformes, comme par exemple Moodle, ne dissimulent pas cette complexité mais peinent, malgré des communautés souvent actives et nombreuses, à fournir une aide et des moyens adaptés. Également, chaque plateforme embarque un paradigme de conception pédagogique et une pédagogie spécifique alors que les praticiens ne sont pas familiers avec ce domaine implicite de conception pédagogique [Martinez-Ortiz et al. 09]. Ils ne sont pas capables ni de mettre en œuvre les scripts requis par ces plateformes [Mekpiroona et al. 08] ni d'ajuster les nombreux paramètres des formulaires des interfaces, qui sont parfois optionnels, non compréhensibles par les enseignants ou portent sur des détails techniques à un niveau très bas (proche du langage machine). Nous cherchons ainsi des éléments de réponses à la question suivante :

Comment faciliter la spécification de scénarios pédagogiques pour des plateformes de formation ?

Malgré les diverses approches existantes, nous avons constaté que la scénarisation sur les plateformes de formation à distance ou les systèmes LMS reste toujours une activité complexe pour les enseignants [Ferraris et al. 2009]. La spécification des scénarios pédagogiques sur ces plateformes requiert l'appropriation de ses multiples interfaces et des différentes contraintes techniques. Cependant, cette appropriation n'est pas une tâche aisée. Elle représente un obstacle pour les enseignants usagers des plateformes et qui sont généralement non informaticiens. Ainsi, **comment aider les enseignants à s'approprier les plateformes et à concevoir leurs scénarios ? Comment dépasser les contraintes techniques et d'IHM liées aux choix technologiques des plateformes et à leur propre conception ?**

Il nous semble important de proposer aux enseignants-concepteurs des outils de conception proches de leurs pratiques et des solutions techniques adaptées pour concevoir des scénarios pédagogiques spécifiques à certaines plateformes. Par exemple, l'utilisation d'un VIDL centré plateforme et associé avec son éditeur graphique peut faciliter la spécification des scénarios. L'aspect graphique facilite la conception des scénarios pédagogiques, la communication et la réflexion pour les praticiens. Ces outils (VIDLs et éditeurs) doivent ainsi répondre à l'exigence suivante : ils doivent être compatibles au langage de conception pédagogique proposé par les plateformes utilisées ou à une partie de ce langage.

Nous considérons que les plateformes de formation ne sont pas pédagogiquement neutres. Elles embarquent un langage/métier de conception pédagogique qui leur est propre. **Ainsi, quel langage ou partie du langage, relevant de la conception pédagogique, doit être identifié(e)?**

Une plateforme de formation embarque plusieurs langages et métiers pour différents cas d'usage. Elle est généralement composée de plusieurs modules qui s'adressent à plusieurs acteurs et à plusieurs objectifs. [Oubahssi 2005] a identifié trois principaux modules de cas d'usages des plateformes (les modules administration, enseignant et apprenant). Le module principal qui nous intéresse est le module enseignant. Ce module peut être composé d'autres modules (ou sous modules). Par exemple, au sein du module enseignant, [Oubahssi 2005] a identifié des modules pour la gestion pédagogique, la gestion des travaux de collaboration, la communication entre les acteurs du cours, etc. D'autres sous-modules peuvent être associés à ces modules. Le module de gestion pédagogique par exemple est composé d'un module de gestion des domaines et des disciplines de formation, un module de gestion des ressources pédagogiques, un module de gestion des plans de formation, un module de gestion des cursus des apprenants, un module de gestion des plannings et de collaboration, etc.

Le langage qui nous intéresse est situé dans le champ du module enseignant pour la spécification des situations d'apprentissages (scénarios, cours, etc.). Ce module explore les concepts mis à disposition de l'enseignant-concepteur pour spécifier son scénario. Les exercices, les tests et les questions sont des exemples d'activités proposées par certaines plateformes tandis que les supports de cours, les documents et les présentations sont des exemples de ressources pédagogiques. Le langage de conception pédagogique représente l'ensemble des activités, des ressources pouvant être intégrées au sein du scénario pédagogique. Il décrit également certains éléments et paramètres caractéristiques du scénario tels que les groupes d'apprenants. Cependant, certains éléments peuvent être exclus de ce langage (tels que la gestion du temps des activités) et font l'objet d'une conception plus fine. Ces éléments sont jugés plus techniques que pédagogiques par les enseignants-concepteurs. Ils engendrent des difficultés lors de la conception d'une situation d'apprentissage. De même, plusieurs attributs et propriétés d'un élément 'pédagogique' peuvent porter sur des détails techniques à bas niveau tels que la taille maximale du fichier à importer, les formules spécifiques de questions, etc. Ces détails peuvent être écartés dans le but de se centrer dans un premier temps sur la conception pédagogique des situations des apprentissages. Les éléments et les détails techniques peuvent être spécifiés dans un deuxième temps. Un des principaux verrous consiste à identifier ce langage de conception pédagogique et à écarter les éléments considérés techniques. Ainsi, **comment l'identifier ? Quelles sont les méthodes et les outils d'analyse permettant l'identification de ce langage ?**

Le langage de conception pédagogique d'une plateforme est implicitement situé et représenté au sein des différentes interfaces de la plateforme. Identifier un langage spécifique d'une plateforme n'est pas une tâche aisée étant donné les différents rôles qu'elle intègre, leurs différents cas d'usages, etc. Son identification requiert l'étude de plusieurs méthodes d'analyse selon plusieurs points de vue telles que les méthodes d'analyse centrée utilisation (analyse des interfaces, analyse des interactions, analyse des connaissances, etc.) et les méthodes d'analyse technique (analyse de la base de données, analyse du code source, analyse des modèles de conception, analyse du cahier des charges, etc.). Ces méthodes doivent permettre l'explicitation et la formalisation de ce langage. Ainsi, **comment expliciter et formaliser ce langage ?**

L'explicitation du langage nécessite de choisir une syntaxe pour représenter le vocabulaire et la grammaire du langage identifié. Plusieurs formats de représentation sont possibles sous forme textuelle ou graphique tels que les diagrammes, les métamodèles, les schémas XML, etc. Tout d'abord, il

est important de fixer les intérêts et les objectifs de la représentation. Ensuite, le format choisi doit répondre aux objectifs visés. Dans le cas de notre travail, le format envisagé doit répondre à des objectifs de réutilisation, d'échange/communication et de développement logiciel. Le format que nous proposons doit par exemple favoriser la communication des outils de conception externes avec les plateformes de formation étudiées.

Le langage identifié représente un langage spécifique au domaine (*Domain Specific Language*). Dans l'objectif de spécifier des scénarios pédagogiques, **comment exploiter ce langage avec une approche DSM ?** L'approche DSM, issue de l'Ingénierie Dirigée par les Modèles, a fait l'objet de plusieurs travaux de recherche tels que [Laforcade 2010], [Zendagui 2010] et [Ouraiya 2012]. Le cadre théorique général de l'Ingénierie Dirigée par les Modèles et plus précisément du *Domain Specific Modeling* permet d'envisager la formalisation de la syntaxe abstraite du langage de conception pédagogique des plateformes de formation à distance sous la forme d'un métamodèle. La spécification des scénarios pédagogiques peut être abordée selon une approche DSM centrée sur ce langage. Cette approche est capable d'apporter une valeur ajoutée pour la définition de nouveaux VIDLs centrés sur le métier des praticiens, des standards et des plateformes [Laforcade et al. 2008]. Elle est également utile pour (1) la génération de langages et outils propres aux plateformes et (2) la définition des outils de transformation entre les modèles pour envisager l'interopérabilité entre les plateformes.

Quelles sont les valeurs ajoutées et les limites de l'approche DSM pour la scénarisation centrée plateforme ? Les usagers quotidiens des plateformes de formation à distance (enseignants, créateurs de cours, concepteurs, etc.) énoncent plusieurs difficultés lors de l'implémentation des scénarios spécifiés en dehors des plateformes [Ferraris et al 2009] [Laforcade et al. 2011]. Ils déplorent la distance entre les langages de conception pédagogique (EML et VIDL) et les langages des plateformes de formation [Ferraris et al. 2009]. Un nombre important de concepts formalisant leurs scénarios n'est pas fournis par les plateformes d'exécution ou ne possède pas d'éléments équivalents. Plusieurs travaux de recherche proposent des outils pour la conception sans instrumenter l'opérationnalisation. Il en résulte de façon générale que plusieurs acteurs pédagogiques (enseignants-concepteurs, enseignants-usagers, ingénieurs pédagogiques, etc.) sollicitent des approches de conception et d'implémentation plus adéquates avec leurs pratiques et leur savoir-faire de conception. Ils préconisent des environnements pour formuler des situations d'apprentissage ou des scénarios pédagogiques à travers un langage formel de conception pédagogique (EML). L'enjeu ici concerne la mise en œuvre de ces scénarios spécifiés à l'aide de ces EML sur une plateforme d'exécution.

Comment faciliter l'opérationnalisation des scénarios pédagogiques sur les plateformes de formation ?

Plusieurs approches centrées standard, usagers, plateformes, etc. ont pour objectif de supporter l'enjeu d'opérationnalisation des scénarios pédagogiques. En analysant ces approches, nous avons constaté qu'elles sont généralement indépendantes des plateformes, à l'exception des approches de conception pédagogique internes. Elles prennent en compte le besoin d'opérationnalisation sur les LMSs. Elles proposent des techniques telles que les transformations entre les modèles, le greffon de

services web ou la modification d'architecture de la plateforme pour assurer l'opérationnalisation de la totalité ou d'une partie de ces scénarios. Elles consistent à transformer le modèle initial de la situation d'apprentissage en un nouveau modèle, (des transformations entre fichiers XML ou en adoptant l'approche MDA (*Model Driven Architecture*)) en tenant compte des éléments et des fonctionnalités fournies par la plateforme cible. Ces transformations se font 'par le code' plutôt qu'entre modèles. Elles peuvent entraîner plusieurs modifications du scénario initial et des pertes d'informations et de sémantique. **Comment conserver la sémantique des scénarios pendant leur opérationnalisation ?**

Pour garantir l'opérationnalisation de la totalité de ces scénarios (sans perte d'informations), ces derniers doivent être spécifiés dans un univers conforme à l'univers techno-centré (plateformes). Dans certains cas, l'exploitation du langage de conception pédagogique embarqué dans la plateforme pour la spécification des scénarios permet de surmonter les défauts des transformations. La conformité du langage de conception avec le langage d'opérationnalisation garantit la sémantique des scénarios initiaux pendant leur opérationnalisation. Le scénario *a priori* prévu par l'enseignant peut être opérationnalisé sans requérir de transformations ou de modifications. Il nous semble que la spécification des scénarios pédagogiques, via des outils de conception basés sur le langage "pédagogique" de la plateforme cible, peut garantir leur opérationnalisation complète mais en revanche cela peut limiter l'expressivité de ces scénarios. Toutefois, il est important que les éléments exclus du langage de conception pédagogique puissent être ajustés et spécifiés. Ainsi, **comment assurer la conception continue des scénarios pédagogiques ?**

La conception du scénario peut être continue sur la plateforme support. Plusieurs paramètres, secondaires ou techniques, non supportés par les outils de conception ou écartés du langage de conception explicite, peuvent être directement ajustés sur la plateforme. Par exemple, au cours de l'opérationnalisation, seule la définition des ressources pédagogiques pourrait être prise en compte. Le dépôt concret des ressources et des documents peut souvent être réalisé dans un second temps de conception. Egalement, la configuration de certains éléments du scénario peut être plus avancée sur la plateforme telle que la définition de la durée allouée pour chaque activité ou l'affectation des apprenants aux groupes.

La conception de scénarios pédagogiques est un processus évolutif qui doit être adapté aux circonstances de son exécution. Après le déroulement de la situation d'apprentissage (tel que l'exécution du scénario pédagogique ou du cours par les apprenants), l'enseignant-concepteur peut mener certaines modifications et adaptations à son scénario afin de surmonter certains défauts. Ces défauts peuvent être identifiés suite à plusieurs observations et analyses des traces de l'exécution du scénario initial. La re-conception ou la modification de ce scénario peut être menée de nouveau sur l'outil de conception externe, spécifique à la plateforme utilisée tout en garantissant l'opérationnalisation du nouveau scénario.

3.2. Approche de conception centrée plateformes

Dans nos travaux, nous supposons que les plateformes ne sont pas pédagogiquement neutres. Elles embarquent implicitement des paradigmes et un langage de conception pour les activités pédagogiques et les situations d'apprentissage. Ce langage peut être identifié et formalisé dans le but de l'exposer

comme langage de base pour le développement d'outils de conception pédagogique externes et des nouvelles fonctionnalités d'import/export. Notre approche se situe dans le cadre théorique général de l'Ingénierie Dirigée par les Modèles et plus précisément du *Domain Specific Modeling* [Abedmouleh 2010]. Le DSM permet d'envisager la formalisation de la syntaxe abstraite sous la forme d'un méta-modèle. Il offre un cadre théorique et pratique pour la modélisation et la spécification des scénarios pédagogiques [Laforcade 2010]. Cette approche est capable d'apporter une valeur ajoutée pour l'explicitation, la spécification et l'opérationnalisation de scénarios pédagogiques et de langages de modélisation pédagogique, également pour la définition de VIDL centrés sur le métier des praticiens, des standards et des plateformes. Nous positionnons notre approche dans le cadre des approches centrées plateformes [Abedmouleh et al. 2011b] [Abedmouleh et al. 2011c]. Nous proposons d'externaliser la conception en dehors de ces dispositifs (plateformes) afin de dépasser les contraintes imposées et les difficultés de conception. Notre hypothèse principale est que la conception pédagogique peut être facilitée lorsqu'elle est réalisée dans un univers pédagogique. Ainsi le langage de conception pédagogique doit être conforme à celui des plateformes. Ce langage ne doit pas comprendre les éléments/détails techniques susceptibles de complexifier la conception. Notre deuxième hypothèse porte ainsi sur la spécification des scénarios dans un environnement métier centré plateforme permettant de garantir leur opérationnalisation. La figure 21 récapitule les éléments caractéristiques de notre approche que nous détaillons dans les sections suivantes.

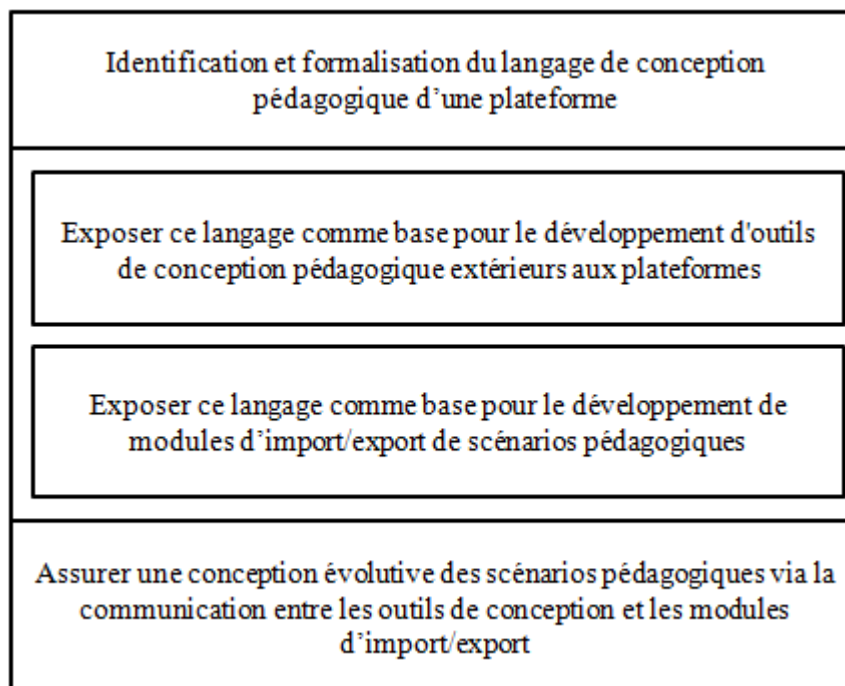


Figure 21 : Vue générale de l'approche proposée

3.2.1. Identification et formalisation du langage de conception pédagogique

Notre approche porte sur l'identification et la formalisation du langage de conception pédagogique pour les plateformes existantes [Abedmouleh et al. 2011a]. D'un point de vue DSM, celui que nous cherchons à expliciter des plateformes peut être considéré comme étant composé d'une syntaxe abstraite (les concepts et relations), d'une syntaxe concrète et d'éléments de sémantique. Nous nous inté-

ressons principalement à expliciter la syntaxe abstraite implicite des LMS que nous proposons de formaliser sous la forme d'un métamodèle (concepts, attributs, relations, types, etc.).

Le métier pédagogique concerne les éléments de conception des situations d'apprentissage sur les plateformes en termes d'activités, de ressources, de services, de paramètres, etc. Les attributs de définition de ces éléments (tels que le nom, la description, etc.) ainsi que leurs propriétés (tels que le type, les champs de valeurs, la valeur par défaut, etc.) constituent la base de ce langage. Toutefois, nous écartons les éléments non liés à la conception pédagogique ainsi que les spécifications techniques du métier final. Cependant, considérer un élément est plus technique que pédagogique peut différer d'un enseignant à un autre. Ce travail nécessite une collaboration entre l'analyste et les enseignants utilisant la plateforme. Notre objectif est de proposer un processus reproductible qui permet l'identification et la formalisation du langage de conception pédagogique [Abedmouleh 2012] [Abedmouleh et al. 2012a] [Abedmouleh et al. 2012b] [Abedmouleh et al. 2012c]. Nous détaillons ce processus dans le quatrième chapitre.

L'explicitation du métier des plateformes représente une nouvelle approche pour la spécification des VIDLs/EMLs et également une nouvelle opportunité pour l'opérationnalisation des scénarios pédagogiques. L'utilisation d'une méthode spécifique, en particulier centrée plateformes, garantit que les scénarios seront totalement opérationnalisables sur les dispositifs pédagogiques concernés.

3.2.2. Exposer le langage des plateformes pour la spécification de VIDLs et d'éditeurs graphiques

Nous proposons d'exposer le langage identifié comme base pour le développement d'outils de conception pédagogique extérieurs aux plateformes tels que les outils d'édition de scénarios [Abedmouleh et al. 2012d]. L'un des premiers objectifs de cette approche est d'offrir aux praticiens de nouveaux VIDLs et des éditeurs externes dédiés basés sur le langage métier des LMS [Laforcade et Abedmouleh 2012]. Plusieurs VIDLs centrés LMS peuvent être proposés pour le métier d'un LMS particulier. Par exemple, certains VIDLs peuvent être centrés sur le métier pédagogique des plateformes de formation. D'autres VIDLs se sont situés à un niveau d'abstraction des plateformes plus important. Ils sont de plus en plus proches de l'univers des praticiens. La figure 22 illustre des VIDLs répartis selon différents niveaux d'abstraction.

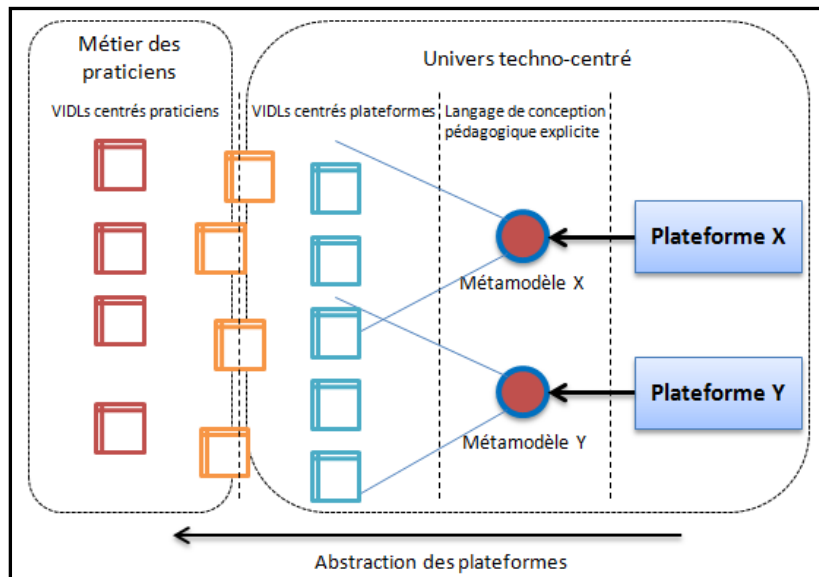


Figure 22 : Les VIDLs selon le niveau d'abstraction

Dans le cadre de notre travail, nous nous situons dans l'univers techno-centré. Plus spécifiquement, nous nous intéressons aux VIDLs centrés plateformes. Notre objectif est d'exprimer et de vérifier la valeur ajoutée d'un outillage du DSM par le biais de ces VIDLs. Afin de faciliter la réflexion et la communication, ces VIDLs doivent être composés de la même syntaxe abstraite que le langage de la plateforme (perspective sur le métamodèle du domaine), mais en proposant une notation visuelle (pour la syntaxe concrète). La syntaxe du langage décrit les multiplicités des relations, types énumérés, attributs obligatoires ou optionnels, etc. L'explicitation du langage nécessite alors de choisir une syntaxe concrète pour la représentation du vocabulaire et de la grammaire du langage dans un format graphique.

Les éditeurs dédiés à ces VIDLs doivent gérer la persistance des scénarios pédagogiques produits dans un format compréhensible par la machine et conforme au langage identifié et formalisé du LMS. Le développement des outils graphiques de conception ainsi que leur mise à l'essai sont détaillés ultérieurement au sein du sixième chapitre.

3.2.3. Opérationnaliser les scénarios pédagogiques par des modules d'import/export

Contrairement aux approches qui consistent à associer les EML avec des moteurs d'exécution dédiés (cas de CopperCore pour IMS-LD) ou avec des infrastructures spécifiques (cas de LDI pour LDL), nous proposons d'exposer le langage métier de la plateforme sous un nouveau format spécifique d'import/export [Abedmouleh 2010] [Abedmouleh et al. 2011b] [Abedmouleh et al. 2011c]. Notre objectif est de vérifier qu'il est possible d'opérationnaliser des scénarios spécifiés à l'extérieur des LMS sans pertes de sémantique ou d'information. Notre proposition consiste ainsi à étendre les LMS par des nouvelles fonctionnalités analogues à celles proposées pour le standard SCORM [Gonzalez-Barbonea et Anido-Rifonb 2010] (sans le moteur dédié), afin de les rendre capables d'importer/exporter des scénarios pédagogiques. L'opérationnalisation du scénario pédagogique d'un point de vue centré plateforme consiste alors à l'importation des scénarios formalisés conformément au métier explicite de la plateforme.

Les scénarios créés ou bien modifiés par l'outil de conception sont totalement opérationnalisables parce qu'ils sont conformes au langage pédagogique implicite de la plateforme. Les syntaxes et les concepts de la plateforme sont exploités pour la spécification des scénarios avec l'outil de conception. Par définition, il n'y a pas de perte d'information puisqu'aucune transformation de modèles n'est nécessaire. La sémantique des scénarios pédagogiques est préservée lors de l'opérationnalisation et également lors de l'exportation. Nous présentons au sein du sixième chapitre deux modules développés pour les plateformes Moodle et Ganesha. Nous présenterons également une mise à l'essai de ces outils.

3.2.4. Conception itérative des scénarios

Les scénarios pédagogiques sont considérés comme des entités qui évoluent de façon continue en fonction des contextes dans lesquels ils se déroulent. Ils doivent donc dépendre fortement des circonstances de leur déroulement. Ceci implique que l'enseignant-concepteur doit pouvoir adapter et faire évoluer les scénarios déjà implémentés sur les LMSs. Notre approche assure la conception continue des scénarios pédagogiques et facilite leur adaptation grâce à l'interaction supportée par la communication entre les outils de conception externes et les plateformes concernées. Les scénarios pédagogiques peuvent être spécifiés dans un premier temps sur les outils de conception. Ensuite, ils sont sauvegardés dans un format externe conforme au schéma XML du métier pédagogique de la plateforme cible. L'opérationnalisation de ces scénarios est assurée par les modules d'import/export ajoutés à la plateforme. Toutefois, la conception de ces scénarios, en particulier les détails techniques que nous avons écartés du métier pédagogique de la plateforme, peuvent être continue sur les interfaces concernées. Dans un deuxième temps, les scénarios opérationnels sur la plateforme peuvent être exportés dans un format compréhensible par les outils et conforme au métier pédagogique grâce à la fonctionnalité d'import/export ajoutée à la plateforme utilisée. Ces scénarios peuvent ainsi être modifiés sur les outils de conception graphique avant d'être opérationnalisés de nouveau. La figure 23 illustre le cycle proposé pour une conception itérative.

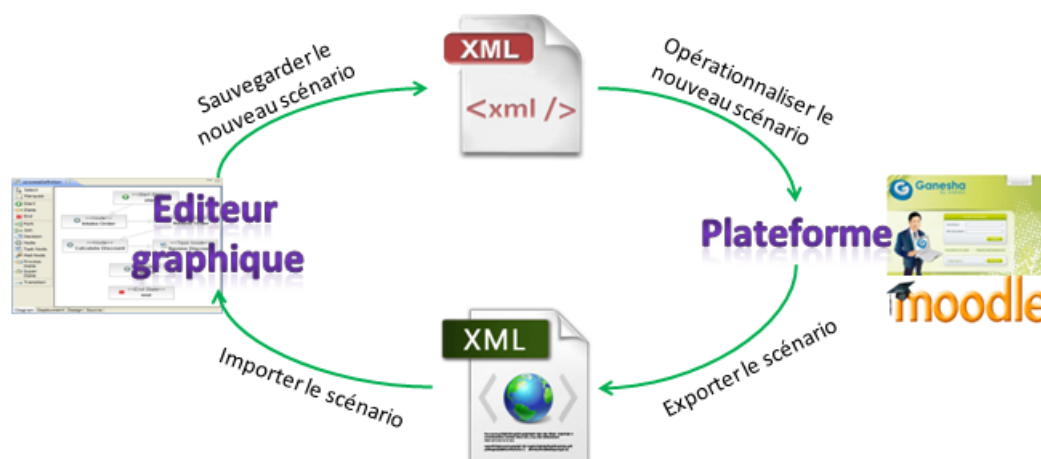


Figure 23 : Cycle de conception itérative

3.3. Méthodologie de recherche

L'objectif de notre travail est de faciliter la conception des scénarios pédagogiques et de garantir leur opérationnalisation sur les plateformes de formation à distance. Ces deux activités ont un rôle important dans le cycle de vie du scénario pédagogique. Elles ont fait l'objet de plusieurs travaux de recherche. En ce qui concerne notre travail, nous avons suivi une méthodologie de recherche qui s'appuie sur des études théoriques et pratiques. La figure 24 illustre notre méthodologie de recherche.

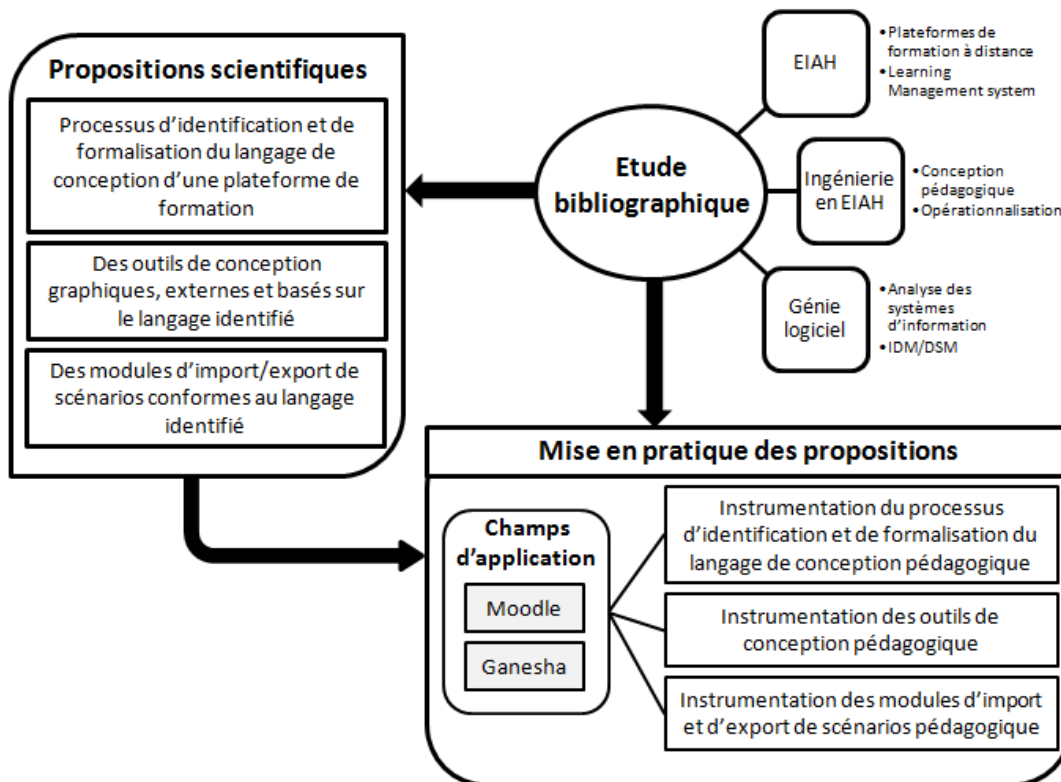


Figure 24 : La méthodologie de recherche adoptée

L'étude théorique consiste à étudier les approches et les travaux de scénarisation et d'opérationnalisation des scénarios. Nous avons donc mené un état de l'art des différentes thématiques impliquées dans notre travail : la conception pédagogique, les langages de modélisation pédagogique (EML et VIDL), l'opérationnalisation, l'ingénierie des EIAH, les plateformes de formation à distance et l'IDM/DSM. Cette étude nous a permis de nous positionner dans l'univers techno-centré. Notre approche et nos propositions sont centrées plateformes (ou LMSs). Cela nous a amené à :

- étudier et s'approprier certaines plateformes du point de vue enseignant-concepteur. Nous avons choisi deux plateformes d'étude fréquemment utilisées pour supporter l'apprentissage à distance. Ces deux plateformes sont Moodle [O'Rourke 2012] et Ganesha [Ganesha 2013].
- étudier les méthodes et les techniques pour parvenir à identifier le langage proposé par les plateformes pour concevoir des situations d'apprentissage ou des scénarios pédagogiques.

Ces études ont abouti aux contributions scientifiques qui suivent :

- Une approche de conception plus adaptée aux enseignants-concepteurs ayant des difficultés d'appropriation des détails techniques de bas niveau des plateformes.
- Un processus d'identification et de formalisation du langage de conception pédagogique d'une plateforme.
- Une spécification et une mise à l'essai des VIDLs et des outils de conception externes.
- Un développement et une mise à l'essai des modules d'import/export de scénarios pédagogiques.

Notre première proposition permet aux enseignants de dépasser les contraintes techniques imposées par les plateformes. Ils peuvent concevoir leurs scénarios pédagogiques par le biais d'un outil de conception graphique et mieux adapté à leurs pratiques. Cet outil ne requiert pas de compétences techniques. Ensuite, les scénarios sont opérationnalisés grâce à des modules ajoutés aux plateformes. L'approche proposée se situe dans un cadre théorique et pratique spécifique. Nous nous sommes appuyés sur l'approche de conception spécifique au domaine (*Domain Specific Modeling*).

Notre deuxième proposition est définie dans le but de permettre l'identification du langage de conception pédagogique des plateformes de formation. Nous avons choisi de proposer aux développeurs et à la communauté de développement des plateformes (les analystes, les ingénieurs informatiques, les ingénieurs de recherche, etc.) une méthode d'explicitation et de formalisation du langage, sous la forme d'un processus spécifique. Cependant, certains aspects critiquables du processus peuvent rendre l'exploitation du processus complexe. Toutefois, les enseignants-concepteurs ne sont pas les utilisateurs principaux de ce processus parce qu'il requiert des compétences en génie logiciel et en analyse IHM. Le résultat de cette proposition, en particulier le métamodèle du langage de conception pédagogique d'une plateforme, représente la base pour le développement des outils de conception ainsi que des modules d'import/export spécifiques aux plateformes.

Afin d'instrumenter nos propositions sur des champs d'application réels, nous avons choisi les plateformes précédemment citées (Moodle et Ganesha). Dans un premier temps, nous avons appliqué le processus pour identifier le langage de conception pédagogique de ces deux plateformes et l'explicitier sous le format d'un métamodèle. Ce métamodèle représente la couche métier pour les outils que nous avons développés. Dans le but de faciliter l'utilisation de ce langage métier, nous ajoutons une syntaxe concrète ou une notation graphique à ce langage afin de proposer un VIDL dédié à la plateforme et nous développons un éditeur graphique qui prend en compte des attentes et des besoins des enseignants-concepteurs. Cet éditeur propose la modélisation graphique de nouveaux scénarios pédagogiques et également la mise à jour des scénarios exportés de la plateforme. L'implémentation de l'éditeur s'est déroulée sous l'environnement Eclipse en utilisant deux frameworks spécifiques. Ces derniers représentent un cadre pratique outillé pour l'approche DSM.

Notre approche consiste également à intégrer un module d'import/export à la plateforme dans l'espace de cours associé à l'enseignant. Ce module assure l'opérationnalisation complète et automatisée des scénarios spécifiés par l'environnement de conception. Il permet également d'exporter les

scénarios déjà existant sur la plateforme. Le cycle d'exportation et d'opérationnalisation permet à l'enseignant d'adapter les scénarios et de les modifier en utilisant l'éditeur externe.

Enfin, nous avons mis à l'essai nos outils (éditeurs et modules) afin de vérifier la mise en œuvre technique de notre approche. Nous avons proposé un scénario pédagogique exemple sous forme textuelle. Ce scénario représente une séance de travaux pratiques de programmation web que nous réalisons souvent en classe avec les étudiants en première année en informatique. Nous avons exploité les éditeurs graphiques développés pour concevoir ce scénario. Nous avons associé à chaque partie du scénario les éléments adéquats en termes d'activités et de ressources pédagogiques proposées par les éditeurs. Le scénario est ensuite sauvegardé dans un format externe pour une éventuelle opérationnalisation. Pour cela, nous avons exploité les modules d'import/export développés. Nous avons également vérifié le mode de conception continue et adaptatif. Pour cela, nous avons exporté des scénarios existants sur les plateformes via ces modules pour les modifier sur les éditeurs graphiques et ensuite nous les avons ré-importés à nouveau.

II

Contributions

4

Processus d'analyse et d'identification du langage de conception pédagogique d'une plateforme

Sommaire

- 4.1. Vue générale du processus
- 4.2. Analyse centrée IHM
- 4.3. Analyse centrée technique
- 4.4. Confrontation et formalisation du langage de conception pédagogique final
- 4.5. Conclusion

Ce chapitre expose une partie de nos propositions. Il porte sur la présentation du processus d'analyse, d'identification et de formalisation du langage de conception pédagogique d'une plateforme de formation. Nos travaux de recherche sont basés sur la proposition d'une approche de conception centrée plateformes ou LMSs. Dans cette approche, nous avons déjà posé une première hypothèse : les plateformes de formation ou les LMSs ne sont pas pédagogiquement neutres et elles embarquent un langage de conception pédagogique implicite spécifique. Nous proposons ici un processus pour l'analyser, l'identifier et le formaliser.

Nous commençons ce chapitre par une présentation générale de ce processus. Ensuite, nous donnons une description détaillée de chaque partie.

4.1. Vue générale du processus

Dans cette section, nous présentons une vue générale du processus d'identification et de formalisation du langage de conception pédagogique d'une plateforme. Comme tout artefact informatique, les plateformes de formation sont composées de plusieurs métiers et langages (métier pédagogique, d'administration, de gestion, etc.) destinés à différents usagers (apprenants, enseignants, tuteurs, etc.). Dans le cadre de notre travail, le langage qui nous intéresse concerne la scénarisation pédagogique sur une plateforme de formation particulière. Nous proposons un processus d'analyse pour l'identification et la formalisation de ce langage [Abedmouleh 2012] [Abedmouleh et al. 2012a] [Abedmouleh et al. 2012b] [Abedmouleh et al. 2012c]. Ce processus est le résultat de plusieurs expérimentations et analyses effectuées sur plusieurs plateformes telles que Moodle et Ganesha. Après l'appropriation de ces plateformes, nous avons constaté que chacune d'entre elles embarquait un paradigme spécifique. Cependant, une même analyse pouvait être définie afin de guider le travail d'identification et de formalisation de leur métier pédagogique. Le processus vise la communauté des LMSs/plateformes de formation telle que les ingénieurs pédagogiques, les développeurs-concepteurs et les experts des plateformes ayant une compétence en informatique. Il facilitera l'explicitation du métier de conception pédagogique de leur plateforme.

Le processus est spécifié en se basant sur deux points de vue différents mais complémentaires : un point de vue centré IHM et un point de vue centré technique. Le premier point de vue porte sur l'analyse des IHMs selon deux stratégies : (1) l'analyse des situations existantes sur la plateforme (déjà spécifiées par des enseignants) et (2) l'analyse des interfaces liées à la spécification de nouvelles situations. Le deuxième point porte sur des méthodes d'analyses plus techniques s'intéressant davantage à la plateforme comme artefact informatique.

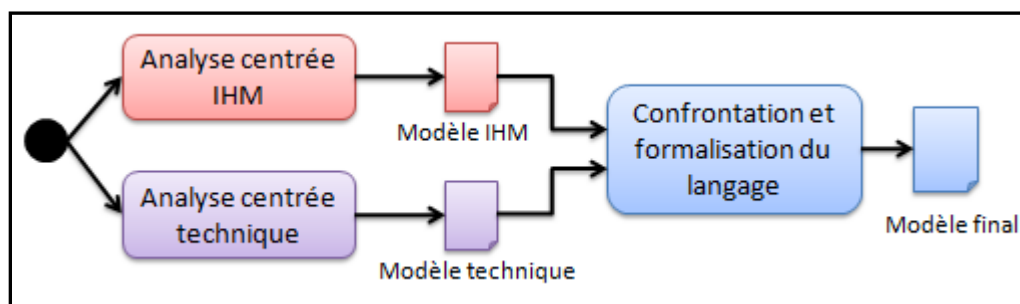


Figure 25 : Processus global d'analyse du langage de conception pédagogique d'une plateforme

La figure 25 illustre le processus que nous proposons. Il est composé de trois parties principales : l'analyse centrée IHM, l'analyse centrée technique, ainsi que la confrontation et la formalisation finale. La répartition du processus en trois étapes principales est le résultat de l'étude de plusieurs plateformes ainsi que de plusieurs réflexions, de recherches et de réunions de travail. Les deux analyses mènent à la spécification de deux modèles de langage de conception pédagogique selon deux points de vue (IHM et technique). Pour cela, nous avons envisagé de trouver un point intermédiaire entre ces deux modèles. Nous avons défini l'étape de confrontation et de formalisation du langage final de conception pédagogique de la plateforme.

L'analyse centrée IHM représente la première partie du processus. Elle consiste à analyser les IHMs de la plateforme qui concernent le profil d'un enseignant-concepteur. La plateforme est analysée selon plusieurs échelles : macro, fonctionnelle et micro. Enfin, cette analyse aboutit à la spécification du modèle IHM.

La seconde partie concerne l'analyse centrée technique. Plusieurs méthodes techniques peuvent être adoptées (analyse de la base de données, analyse du code source, analyse des sauvegardes de cours, etc.) mais la principale source d'informations sera l'analyse de la base de données. L'analyse technique aboutit à la spécification du modèle technique.

La dernière partie du processus concerne (1) la confrontation entre le modèle IHM et le modèle technique, résultant des deux premières parties et (2) la formalisation du modèle final (métamodèle) de conception pédagogique d'une plateforme. Ce métamodèle servira ensuite comme base pour la spécification des modules d'import/export des LMS et pour le développement de langages/éditeurs de conception pédagogique externes, que nous détaillons au sein du sixième chapitre.

Chaque partie du processus a un intérêt bien spécifique :

- L'analyse centrée IHM assure l'identification de la partie visible du langage de la plateforme, du point de vue utilisateur.
- L'analyse centrée technique assure que les modèles spécifiés conformément au modèle technique peuvent être opérationnalisés sur la plateforme.
- La confrontation entre les modèles assure l'affinement du modèle IHM et la détection de certains problèmes (cf. section 4.4.4).

Les sous-sections suivantes présentent un aperçu de chaque partie/sous-partie du processus. Ensuite, une description détaillée est présentée dans ce chapitre.

4.1.1. Vue générale de l'analyse centrée IHM

L'analyse centrée IHM (figure 26) représente la première partie du processus. Elle est conduite par trois analyses séquentielles (IHM-macro, fonctionnelle et IHM-micro) dont chacune possède ses propres caractéristiques et produit son (ses) modèle(s) conforme(s) au formalisme proposé. Ces trois analyses sont ensuite suivies par une étape de composition basée sur les modèles résultant. Son objectif est de formaliser le modèle du langage de conception pédagogique d'une plateforme par la dérivation de celui issu de l'analyse centrée IHM.

Dans un premier temps, nous avons choisi d'identifier les interfaces relevant de la conception pédagogique par une phase d'analyse que nous appelons IHM-macro. Dans un deuxième temps, nous avons constaté que l'analyse IHM-macro n'est pas suffisante pour identifier les spécificités du langage de conception pédagogique de la plateforme. Nous avons ainsi défini une deuxième analyse dite fonctionnelle. Cette analyse permet d'identifier ce que la plateforme est capable de proposer et également de délimiter le périmètre des fonctionnalités en relation avec la conception pédagogique. Dans un troisième temps, nous avons défini l'analyse IHM-micro qui permet d'identifier plus finement les propriétés et les relations entre les éléments de conception pédagogique. Plusieurs modèles (IHM-micro) résultent de cette analyse. Ces modèles représentent des extraits du modèle final. Ils doivent être asso-

ciés entre eux. C'est l'objectif de l'étape de composition qui permet de formaliser le modèle IHM entier. La figure 26 illustre les différentes analyses effectuées et modèles produits au sein de l'analyse centrée IHM.

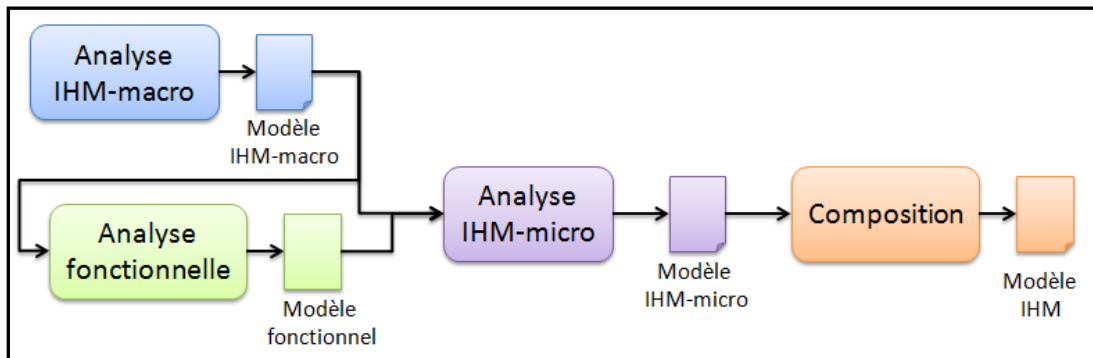


Figure 26 : Analyse centrée IHM

Vue générale de l'analyse IHM-macro

L'analyse IHM-macro consiste à identifier les IHMs de la plateforme relevant de la conception pédagogique. Ces IHMs spécifiques permettent aux enseignants de spécifier le contenu de leurs scénarios pédagogiques et de leurs situations d'apprentissage. L'objectif de cette analyse est de restreindre le champ d'analyse. Les IHMs pertinentes sont retenues et les autres sont écartées. Nous définissons une IHM comme étant l'écran complet visible à un instant particulier par le concepteur. Ces IHMs sont ensuite capitalisées au sein du modèle IHM-macro et intitulées par le concept principal apparaissant à l'écran. Les autres analyses centrées IHM (fonctionnelle et IHM-micro) seront limitées à ces IHMs. Nous avons choisi le format d'une cartographie (*mindmap*) pour ce modèle.

Vue générale de l'analyse fonctionnelle

L'analyse fonctionnelle consiste à identifier les fonctionnalités relevant de la conception pédagogique. Ces fonctionnalités sont déployées dans les différentes IHMs identifiées sein de l'analyse IHM-macro. 'Ajouter un atelier', 'modifier une leçon' et 'inscrire un groupe à un cours' sont des exemples de fonctionnalités pédagogiques. L'objectif de cette analyse est d'identifier l'aptitude de la plateforme en termes de conception pédagogique. Les fonctionnalités non pédagogiques (techniques, administrative, etc.) ne seront pas prises en compte au sein de cette analyse. Nous avons adopté la méthode d'analyse SADT (*Structured Analysis and Design Technic*) [Marca et McGowan 1987] en adoptant des modifications sur le formalisme des modèles internes que nous représentons par des diagrammes de cas d'utilisation UML (*Unified Modeling Language*) [John et Muthig 2002].

Vue générale de l'analyse IHM-micro

L'analyse IHM-micro consiste à analyser les interfaces du modèle IHM-macro ou une partie de ces interfaces à une échelle plus fine. L'objectif de cette analyse est d'identifier les caractéristiques des éléments relevant de la conception pédagogique en termes d'attributs, de champs de valeur, d'initialisations, etc. Plusieurs modèles IHM-micro résultent de cette analyse dont chacun décrit un élément spécifique. Nous avons choisi de représenter ces modèles par le format de la cartographie (*mindmap*) permettant de capturer les descriptions des éléments.

Vue générale de la composition des modèles

L'objectif de la composition est de formaliser le modèle pédagogique dérivé de l'analyse IHM par la combinaison des multiples modèles IHM-micro résultant. Ainsi, le formalisme principal du modèle IHM est une cartographie (*mindmap*) en ajoutant la notion de la multiplicité aux relations entre les modèles IHM-micro associés.

4.1.2. Vue générale de l'analyse technique

L'analyse technique représente la deuxième forme d'analyse que nous proposons en complémentarité avec l'analyse centrée IHM. Elle repose sur l'analyse de la base de données consistant à identifier les tables et les attributs relevant de la conception pédagogique. Nous désignons par tables relevant de la conception pédagogique, l'ensemble des tables stockant des éléments ou des paramètres spécifiques aux situations pédagogiques. Quant aux attributs relevant de la conception pédagogique, ils désignent un ensemble d'attributs proposés par une table relevant de la conception pédagogique, plus spécifiquement des attributs plus pédagogiques que techniques, intervenant à la définition d'un élément de conception pédagogique. Nous avons choisi le Modèle Conceptuel de Données (MCD) de la méthode Merise⁷ pour représenter le modèle technique.

4.1.3. Vue générale de la confrontation des modèles et de la formalisation du modèle final

La dernière étape du processus concerne (1) la confrontation entre le modèle centré IHM et le modèle technique et (2) la formalisation du modèle métier final (métamodèle) de conception pédagogique. L'objectif de cette étape est de valider la persistance des données du modèle IHM et du modèle technique afin de parvenir à un modèle unique, correct et représentatif. La confrontation est basée sur la comparaison entre les deux modèles IHM et technique. Certaines différences entre eux requièrent des vérifications et de nouvelles analyses (IHM et technique). D'autres méthodes d'analyses techniques (analyse du code sources, analyse des fichiers de sauvegarde, etc.) peuvent être adoptées pour éclaircir ces différences. Nous avons choisi le format du métamodèle pour représenter le modèle final de conception pédagogique d'une plateforme.

Dans les sections suivantes, nous allons détailler davantage les différentes parties/sous parties de ce processus.

4.2. Analyse centrée IHM

4.2.1. Analyse IHM-macro

Les plateformes de formation à distance sont généralement composées de plusieurs IHMs qui sont développées pour des diverses finalités et pour des catégories d'usagers différentes. Un nombre important des IHMs ne portent pas sur la conception pédagogique des situations d'apprentissage. Elles font

⁷ Merise est une méthode d'analyse, de conception et de réalisation de systèmes d'informations et de gestion de projet informatique.

partie des IHMs que nous écartons telles que les IHMs d'administration, les IHMs de gestion, etc. Pour cela, notre première hypothèse consiste à encadrer le champ d'analyse.

Cette hypothèse fait l'objet de la première activité du processus proposé (**analyse IHM-macro**) qui concerne l'analyse des IHMs de la plateforme à une échelle macro. Cette analyse consiste principalement à identifier les interfaces relevant de la conception pédagogique. Son objectif est de (1) capitaliser les concepts principaux de conception pédagogique situés dans ces interfaces et (2) de produire la cartographie de ces interfaces. Chaque interface est représentée par le concept principal qu'elle décrit. Ces concepts sont identifiés grâce à deux méthodes :

- l'analyse des titres des interfaces,
- l'analyse des chemins de navigation / des URLs des interfaces.

L'analyse IHM-macro concerne toutes les interfaces de la plateforme. Il est important de parcourir tous les liens possibles dans une interface donnée. Souvent, ces liens mènent à des nouvelles interfaces. Cependant, l'analyse IHM-macro peut se limiter aux interfaces situées dans l'espace associé aux enseignants-concepteurs. Le parcours des interfaces dépend du paradigme de la plateforme. Par exemple, la plateforme Moodle impose de parcourir ses interfaces en suivant une démarche descendante. Elle dispose d'une interface principale pour la spécification et la représentation du contenu du cours (figure 33). Les autres interfaces telles que l'interface d'ajout d'un forum (figure 28) sont accessibles à partir de l'interface principale. Cependant, le parcours des interfaces de la plateforme Ganesha s'effectue dans une démarche latérale et descendante. Son paradigme impose la création de certains éléments du module (cours) séparément. Ensuite, les relations entre eux sont spécifiées. Par exemple, les séquences d'activités sont tout d'abord spécifiées, ensuite elles sont associées aux modules concernés. Quant aux activités, elles ne sont accessibles qu'à partir de l'interface des séquences.

Nous avons modélisé l'analyse IHM-macro par un diagramme d'activité UML (*Unified Modeling Language*) qui est représenté par la figure 27. Pour commencer cette analyse, notre démarche consiste à choisir une première interface. Ensuite, l'analyse doit déterminer si cette interface dispose d'un critère pédagogique (telles que les interfaces d'ajout des activités et des ressources pédagogiques (leçon, quiz, etc.). Les interfaces dont leurs titres ou leurs chemins de navigation relèvent de la conception pédagogique sont prises en compte. Le concept principal et le plus significatif de cette interface doit être identifié et ensuite représenté sur le modèle IHM-macro. Enfin, les relations entre les concepts du modèle doivent être identifiées et définies sur le modèle.

L'identification des interfaces est un processus itératif. Lorsqu'une nouvelle interface est identifiée, l'analyste étudie les liens situés sur cette interface afin d'accéder à des nouvelles interfaces. Ensuite, si cela n'est pas possible, il vérifie si des nouvelles interfaces sont accessibles à partir des interfaces antérieurement identifiées. Les interfaces non relevant de la conception pédagogique ne sont pas analysées ni représentées dans le modèle.

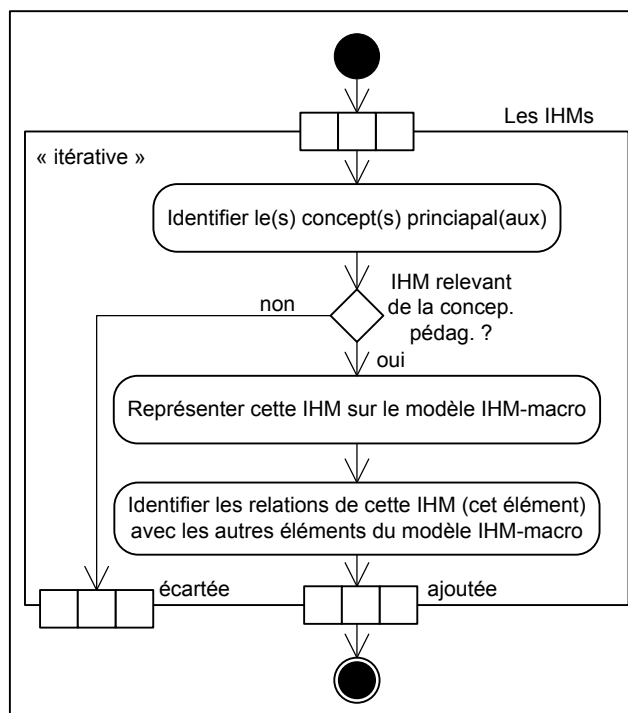


Figure 27 : Diagramme d'activité de l'analyse IHM-macro

4.2.1.1. Identification des principaux concepts pédagogiques

L'identification des concepts pédagogiques principaux dans une interface peut être réalisée par deux méthodes d'analyse : l'analyse sémantique des titres (des interfaces, des blocks, etc.) et l'analyse des chemins de navigation. Nous détaillons ci-dessous ces deux méthodes.

4.2.1.1.1. Analyse des titres

Le développement des plateformes exige une ergonomie spécifique visant à faciliter l'utilisation et la manipulation de ses interfaces et à organiser les informations des interfaces de façon simple et claire. Cette ergonomie est susceptible de mettre les titres des interfaces en relief ainsi que les titres des blocs, des menus et des parties principales d'une interface particulière. Souvent, chaque interface dispose de son propre titre. Le titre est généralement situé en haut de la page avec une mise en forme spécifique (taille plus grande, couleur différente, etc.). Il désigne le concept principal de l'interface ou bien son occurrence. Toutefois, il est possible de trouver plus qu'un titre dans une interface. Ainsi, plusieurs propositions sont possibles :

- Plusieurs éléments principaux sont représentés sur cette interface ; le concept le plus pertinent doit être choisi. Lorsque tous les concepts sont de même pertinence, un concept quelconque est choisi (les autres concepts seront identifiés pendant les autres analyses).
- Les titres décrivent la hiérarchie des éléments de l'interface. Par exemple, le premier titre représente le concept ascendant. Le deuxième représente le concept principal de l'interface tandis que les autres sont des sous-éléments du concept principal. Dans ce cas, le deuxième titre est l'élément représentatif de l'interface.

Par exemple, la figure 28 représente l'interface de spécification d'un forum sur la plateforme Moodle. Elle montre deux titres principaux dans son interface : '*course1*' et '*Ajouter Forum*' (encadrés en rouge). Nous constatons que l'interface du '*forum*' est accessible à partir de l'interface du '*course1*'. Le concept principal de cette interface est '*forum*'. '*course1*' représente son élément ascendant.

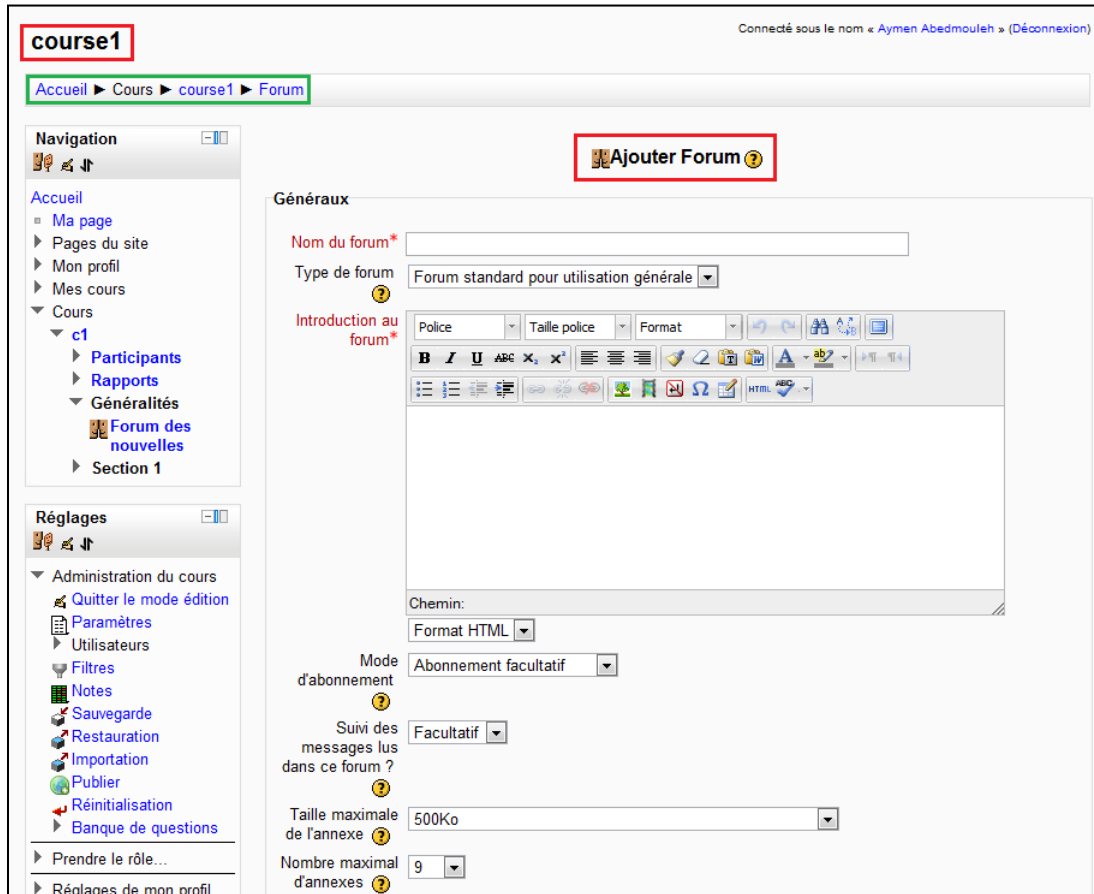


Figure 28 : Interface d'ajout d'un forum sur Moodle

4.2.1.1.2. Analyse des chemins de navigation

Plusieurs plateformes de formation ou systèmes LMS montrent le chemin de navigation sur leurs interfaces. Souvent, ces chemins sont désignés par la concaténation des concepts principaux des interfaces déjà parcourues ainsi que l'interface en cours. Leur analyse permet de déterminer l'agencement général des interfaces et d'identifier certaines relations. Outre les chemins de navigation, les URLs de plateformes de formation sont également une source d'information pour repérer le titre de l'interface et par la suite son concept principal.

Par exemple, l'analyse du chemin de navigation '*Accueil/Cours/Course1/Forum*' sur la figure 28 permet d'identifier les interfaces déjà parcourues et la relation de la hiérarchie ou d'agencement entre elles. Dans cet exemple, '*Course1*' représente un cours sur la plateforme Moodle dont '*Forum*' est un élément de leur contenu.

4.2.1.1.3. Formalisme du modèle IHM-macro

L'analyse IHM-macro consiste à recapitaliser les interfaces relevant de la conception pédagogique dans un modèle, appelé modèle IHM-macro. Nous avons choisi le format de la cartographie (mindmap

en anglais) pour le modèle IHM-macro. Ce format permet la représentation des interfaces par des nœuds ainsi les relations entre eux. La spécification du modèle IHM-macro est réalisée au fur et à mesure qu'une nouvelle interface à critère pédagogique est identifiée que nous représentons par son concept principal. Ensuite, les relations avec les autres éléments du modèle doivent être spécifiées. Ces relations sont basées principalement sur les relations entre les interfaces. Le schéma ci-dessous (figure 29) montre un extrait du modèle IHM-macro issu de l'application de l'analyse IHM-macro sur la plateforme Moodle. Le modèle IHM-macro complet est présenté dans le chapitre suivant. Toutefois, nous représentons dans l'annexe A l'application de l'analyse IHM-macro sur un ensemble cohérent d'interfaces des plateformes Moodle et Ganesha.

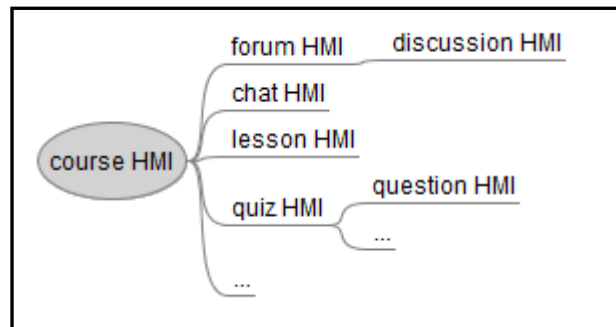


Figure 29 : Extrait du modèle IHM-macro de Moodle

L'analyse IHM-macro permet d'encadrer le champ d'analyse et d'identification du langage de conception pédagogique d'une plateforme en identifiant les IHMs liées à la conception pédagogique d'une situation d'apprentissage. Cependant, cette analyse ne permet pas d'identifier et de formaliser l'ensemble de ce langage spécifique et ses éléments caractéristiques. D'autres analyses complémentaires, telles que l'analyse fonctionnelle, sont ainsi proposées par ce processus.

4.2.2. Analyse fonctionnelle

4.2.2.1. Analyse fonctionnelle d'un nouveau système versus l'analyse fonctionnelle d'un système existant

Dans le domaine du génie logiciel, l'analyse fonctionnelle est une étape du cycle de vie d'un logiciel se situant dans la phase de spécification et d'analyse du besoin. Elle vise à exprimer le besoin en termes de services et de fonctionnalités attendus plutôt qu'en termes de solutions. Dans le cadre de notre travail, l'analyse fonctionnelle consiste à identifier les fonctions déjà existantes sur le système. Ces fonctions représentant le noyau fonctionnel du système sont généralement représentées dans des interfaces d'interaction (IHM) afin d'être facilement accessible à un utilisateur final.

4.2.2.2. Démarche de l'analyse fonctionnelle

Avant de mener une analyse fonctionnelle d'une plateforme de formation, plusieurs documents et modèles (cahier des charges, modèle fonctionnel existant, etc.) peuvent représenter une première source d'information pour l'identification de ces fonctionnalités. Dans le cadre de nos travaux, nous nous intéressons à l'identification des fonctionnalités de la plateforme à partir de ses IHMs.

Les fonctionnalités (pédagogiques et administratives) sont déployées implicitement sur les interfaces des dispositifs d'apprentissage (plateformes et LMSs) via des éléments ou des composants de

l'interface graphique (des boutons, des icônes, des figures, des liens hypertextes, etc.) qui facilitent l'interaction avec l'utilisateur. L'identification de ces fonctionnalités est basée (1) sur l'appropriation des différents éléments formant une interface et (2) sur la capacité de perception et d'interprétation de l'analyste (ingénieur pédagogique, expert de la plateforme, etc.). Souvent, il n'est pas aisé de déterminer si une fonctionnalité relève de la conception pédagogique. Pour cela, il est possible d'étudier le critère pédagogique de ses sous-fonctionnalités.

Dans un premier temps, nous proposons d'encadrer le champ d'analyse des interfaces en fixant les zones d'analyse des IHMs. Nous avons constaté que, pour certaines plateformes, plusieurs zones sont communes entre ses interfaces et pouvant être analysées une seule fois. Les zones ayant des intérêts administratifs ne sont pas concernées par l'analyse. Ensuite, en choisissant une zone particulière, les composants d'interface graphique (widgets) doivent être testés puis analysés fonctionnellement. Une fonction représentative doit être attribuée à chaque composant. Les fonctions portant sur des objectifs non pédagogiques tels que les fonctions d'affichage, de gestion, d'administration, etc. ne sont pas prises en compte ni représentées sur le modèle fonctionnel. Sur la plateforme Moodle par exemple, les fonctions '*choisir la taille maximale des fichiers*' et '*sauvegarder le cours*' sont des fonctionnalités techniques non liées à la conception pédagogique.

Les fonctions qui réfèrent à la conception pédagogique sont prises en compte et représentées sur le modèle par le biais d'un cas d'utilisation du digramme de cas d'utilisation du langage UML. Par exemple, l'analyse de l'extrait d'interface de cours sur Moodle (figure30) consiste à analyser fonctionnellement les composants de cette interface graphique tels que les éléments 1, 2, 3 et 4. Les éléments 1 et 2 désignent respectivement les fonctions '*cacher une section*' et '*modifier le résumer*' qui sont pédagogiques. Quant à l'élément 3, il permet de '*déplacer un élément de cours*'. Il porte ainsi sur une fonction plus de présentation que pédagogique. Lorsqu'une fonctionnalité est non-élémentaire, ses sous-fonctionnalités sont représentées dans un nouveau diagramme de cas d'utilisation qui sera ensuite associé au diagramme principal.



Figure 30 : Extrait d'interface de cours sur Moodle

La figure 31 illustre l'analyse fonctionnelle décrite par le biais d'un diagramme d'activité du langage UML.

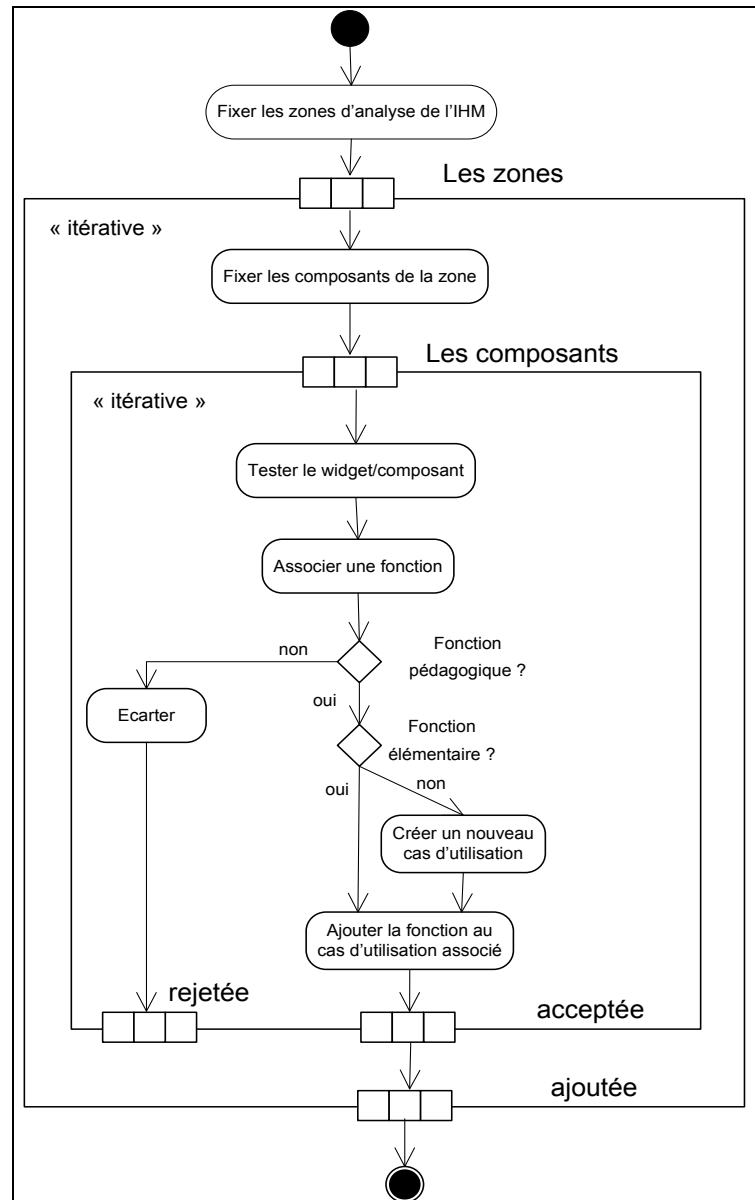


Figure 31 : Diagramme d'activité de l'analyse fonctionnelle

4.2.2.3. Formalisme du modèle fonctionnel

Nous rappelons que l'analyse fonctionnelle consiste à expliciter les fonctionnalités relevant de la conception pédagogique. Ces fonctionnalités sont représentées, dans le modèle fonctionnel, indépendamment des composants qui les réalisent.

Plusieurs représentations du modèle fonctionnel sont possibles [Wikipedia 2012] telles que :

- *Hierarchical Input Process Output* (HIPO and IPO) [Boehm 2007],
- *Structured Analysis and Design Technique* (SADT) [Marca et McGowan 1987],
- *Integration Definition for Function Modeling* (IDEF0) [Ryu et Yücesan 2007],
- *Business Process Modeling Notation* (BPMN) [Muehlen et Recker 2008],
- *Axiomatic Design* [Kulaka et al. 2010],
- *Business reference model* [Choi et al. 2005],
- Etc.

Dans notre travail, nous adoptons la méthode d'analyse SADT [Marca et McGowan 1987] mais en apportant des modifications sur le formalisme des modèles internes qui seront représentés par des diagrammes de cas d'utilisation UML (*Unified Modeling Language*) [John et Muthig 2002].

Les fonctionnalités d'une interface sont décrites par un diagramme de cas d'utilisation. Au fur et à mesure qu'une fonctionnalité est identifiée, il est nécessaire de vérifier son usage pédagogique. Seules les fonctionnalités à critère pédagogique sont représentées sur le modèle. Ensuite, les sous-fonctionnalités sont représentées dans un nouveau diagramme de cas d'utilisation. Ce dernier est émergé ultérieurement au sein du diagramme principal.

La figure 32 illustre un exemple générique du modèle fonctionnel. L'interface principale d'une plateforme est analysée. Ses fonctionnalités pédagogiques sont représentées au sein d'un premier diagramme de cas d'utilisation. Ensuite, les interfaces accessibles à partir d'une de ses fonctionnalités sont également analysées. Les fonctionnalités pédagogiques identifiées sont représentées au sein d'un nouveau diagramme de cas d'utilisation avant d'être émergé au sein du diagramme principal.

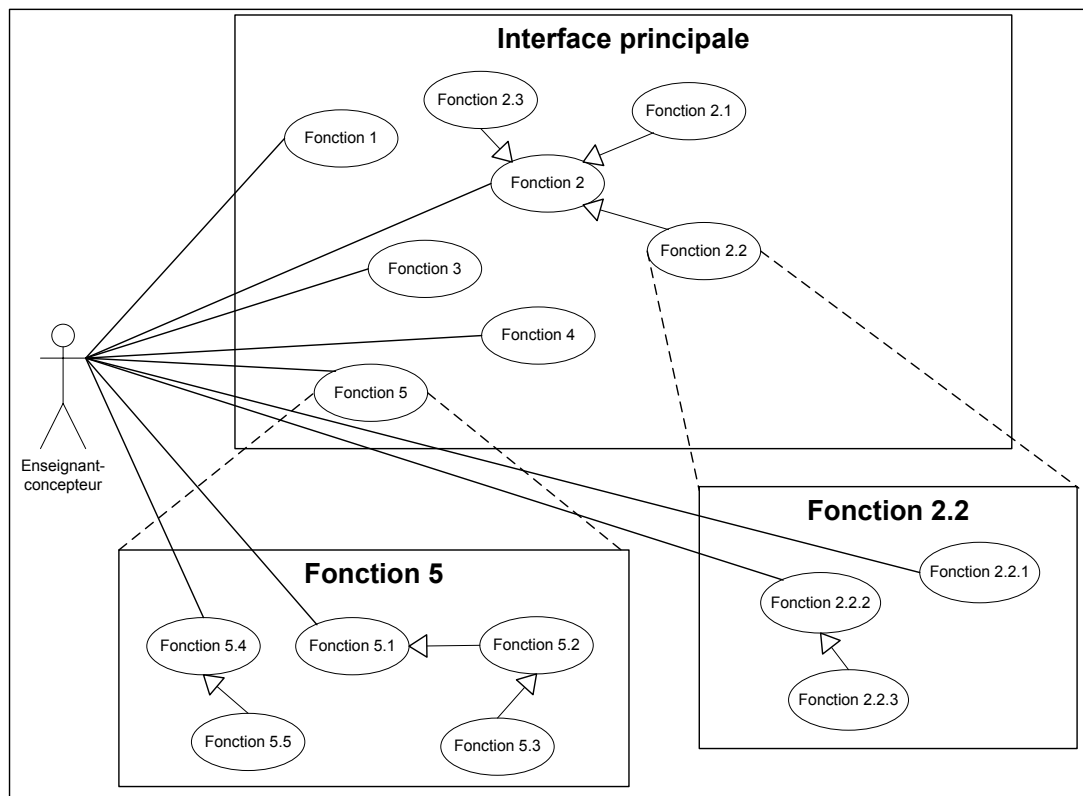


Figure 32 : Vue générale du modèle fonctionnel

4.2.2.4. Exemple d'application de l'analyse fonctionnelle sur la plateforme Moodle

Cette section présente brièvement un exemple d'application de l'analyse fonctionnelle sur l'interface principale d'un cours sur la plateforme Moodle. Nous détaillons ultérieurement, en particulier au sein de la section 5.2.2.2 et des annexes A et B, un ensemble d'exemples cohérents de l'application de l'analyse fonctionnelle sur les plateformes Moodle et Ganesha.

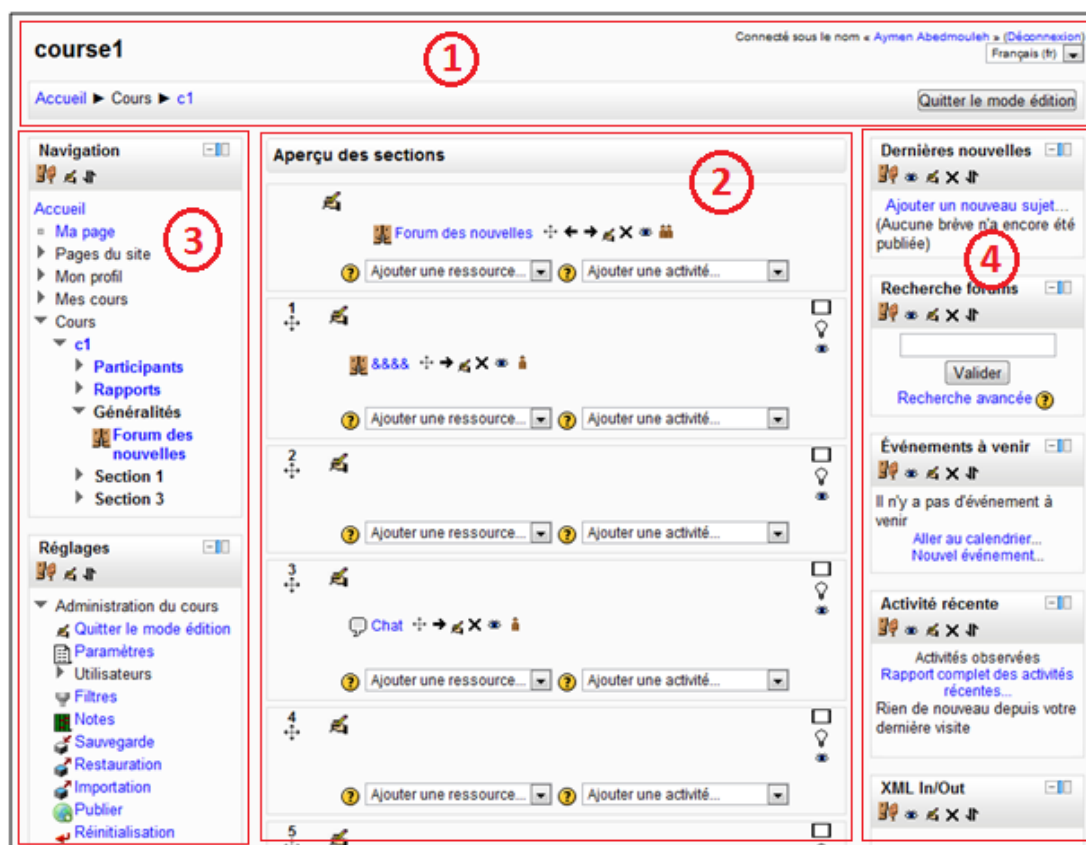


Figure 33 : Interface du cours sur Moodle

La figure 33 illustre l'interface de représentation et de spécification du contenu du cours, située dans l'espace d'enseignant-concepteur de la plateforme Moodle. Dans un premier temps, nous avons décomposé cette interface en quatre zones principales. Ensuite, nous nous intéressons à l'analyse fonctionnelle de chaque zone. Les zones 1, 3 et 4 sont communes aux interfaces et ne sont analysées qu'une seule fois. Plusieurs fonctionnalités sont accessibles par des composants de l'interface graphique tels que les multiples listes déroulantes dans la zone 2 qui disposent de plusieurs éléments et qui représentent les deux fonctionnalités 'ajouter des ressources' et 'ajouter des activités'. L'analyse fonctionnelle de ces listes a permis l'identification de nouvelles fonctionnalités en relation telles que 'ajouter une page', 'ajouter un url', 'ajouter un fichier', 'ajouter un forum', 'ajouter une leçon', etc. Nous avons également identifié d'autres fonctionnalités, représentées par d'autres composants, proposant à titre d'exemple de 'modifier le résumé d'une section', 'cacher/afficher une section', etc. La figure 34 représente un extrait du modèle fonctionnel résultant de l'analyse fonctionnelle de cette interface.

Le choix d'une fonctionnalité particulière, par exemple 'ajouter un forum', parmi les fonctionnalités identifiées nous amène à une nouvelle interface dédiée (au forum dans ce cas). L'analyse fonctionnelle de cette dernière permet l'identification des nouvelles fonctionnalités, telles que 'ajouter une discussion', qui sont déjà des sous fonctionnalités de la fonctionnalité principale ('ajouter un forum'). Elles sont représentées par un nouveau diagramme de cas d'utilisation avant d'être associées au diagramme principal.

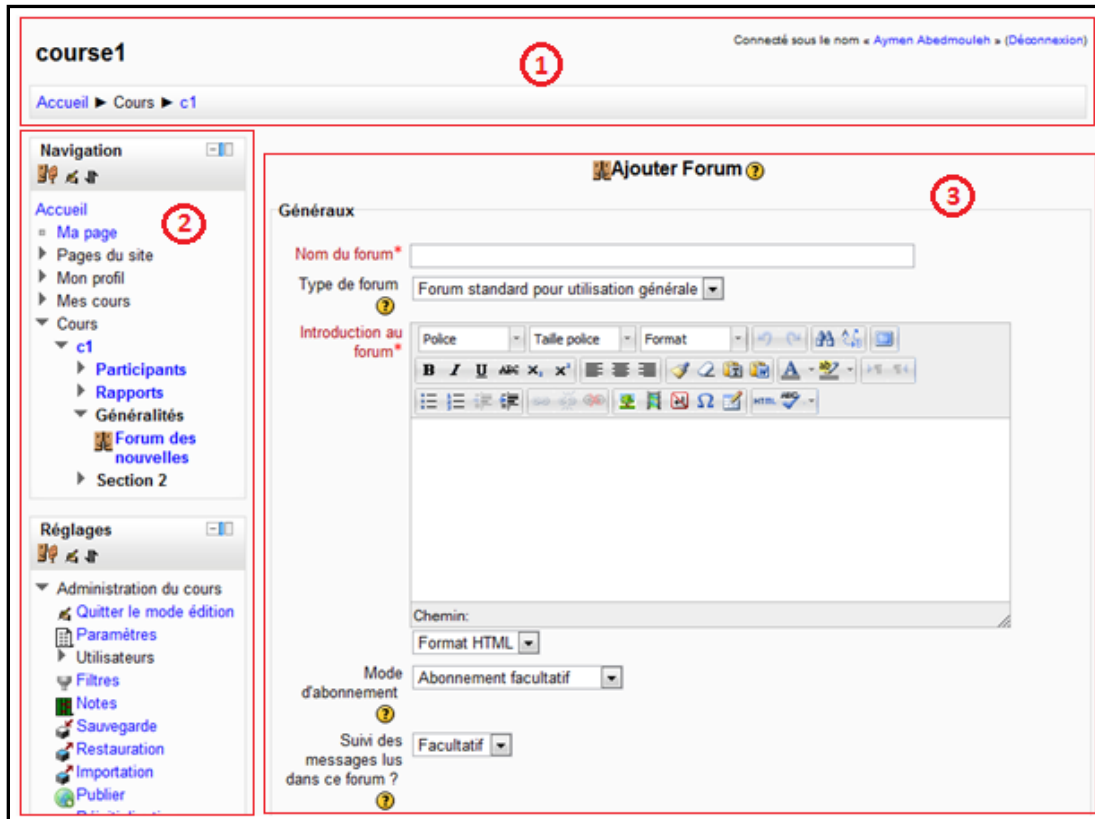


Figure 36 : Interface d'ajout d'un forum dans Moodle décomposée en trois zones

Par exemple, l'interface d'ajout d'un forum sur la plateforme Moodle (figure 36) est décomposable en trois zones principales (encadrées en rouge) :

- La première zone montre l'entête de l'interface composée du nom du cours, du chemin de navigation et de l'identité de l'enseignant-concepteur.
- La deuxième zone montre deux blocs (navigation et de réglages). Ils peuvent inclure des services, des fonctionnalités, des paramètres, etc.
- La troisième zone montre un formulaire de paramétrage de l'activité forum. Il décrit l'élément forum par plusieurs champs/attributs. Nous remarquons que les éléments requis par l'interface sont spécifiés avec une couleur particulière (colorés en rouge).

Après le découpage de l'interface en plusieurs zones, certaines zones peuvent être analysées une seule fois (lorsqu'elles sont communes pour toutes les interfaces) telles que la zone 1 et 2 ou bien elles sont écartées (parce qu'elles n'interviennent pas pour des objectifs pédagogiques). L'analyse IHM-micro est centrée uniquement sur les zones décrivant des éléments liés à la conception pédagogique (le cas de la zone 3).

4.2.3.2. Analyse fine des zones des interfaces

4.2.3.2.1. Analyse des titres

Les parties ou les zones principales des interfaces sont souvent limitées par des cadres et menées par des titres qui soulignent souvent un thème ou un concept spécifique. Ces titres sont généralement spécifiés dans un format spécifique (taille plus grande, souligné, en italique, couleur différente, etc.). Les

titres des menus, des blocs, des formulaires, etc. peuvent être également mis en relief. L'analyse sémantique des titres au sein d'une interface ou d'une zone particulière permet l'identification des éléments du langage de conception pédagogique.

Par exemple, l'interface d'ajout d'un forum (figure 37) est formée de trois zones (encadrées en rouge). Chaque zone est menée d'un ou plusieurs titre(s) spécifié(s) dans un format spécifique (encadrés en rouge et mis en gras à l'entête des cadres). L'analyse des titres a permis d'écarter certains éléments n'intervenant pas à la conception d'un cours tels que '*navigation*', etc. Cependant, l'analyse du titre '*Ajouter Forum*' de la troisième zone renseigne sur l'élément forum : élément de conception pédagogique d'un cours. Les éléments/attributs dédiés au forum sont ensuite analysés via l'analyse du formulaire proposé. Cette partie sera détaillée au sein de la section suivante.

The screenshot shows a web interface for adding a forum. It is titled 'course1' and shows a user logged in as 'teacher1 teacher1'. The interface is divided into three main sections marked with red circles and numbers 1, 2, and 3. Section 1 is the top header area. Section 2 is a left sidebar menu with 'course1' selected. Section 3 is the main content area titled 'Ajouter Forum' which contains a form with the following fields: 'Nom du forum*' (text input), 'Type de forum' (dropdown menu set to 'Forum standard pour utilisation générale'), 'Introduction au forum*' (rich text editor with a toolbar), 'Chemin:' (text input), 'Format HTML' (dropdown menu), 'Mode d'abonnement' (dropdown menu set to 'Abonnement facultatif'), and 'Suivi des messages lus' (dropdown menu set to 'Facultatif').

Figure 37 : Interface d'ajout d'un forum

4.2.3.2.2. Analyse des formulaires

Basées sur les technologies web, les interfaces des plateformes de formation disposent des formulaires qui permettent de faciliter les interactions entre les utilisateurs et la machine. Ces formulaires décrivent un élément ou un concept en termes d'attributs, des champs de valeurs, des initialisations et des valeurs par défaut, etc.

Comme le montre la figure 37, la troisième zone est composée d'un formulaire spécifique pour l'ajout de l'activité forum. Le formulaire décrit les attributs du forum tels que '*nom du forum*', '*type du forum*', '*introduction au forum*', '*suivi des messages lus dans ce forum*', etc. Nous avons constaté par le biais des expérimentations que nous avons menées sur plusieurs plateformes que les formulaires disposent généralement de deux catégories d'éléments/attributs :

- des éléments/attributs requis par les interfaces : ils sont obligatoires et primordiaux pour le fonctionnement du système. Ils sont généralement spécifiés dans un format spécifique (colorés différemment, par exemple en rouge), mis dans un style particulier, marqués avec des caractères spéciaux, de font différent (taille plus grande, souligné, italique, etc.). Ils constituent une partie du langage de conception pédagogique d'une plateforme. Par exemple sur l'interface de la figure 37, les éléments requis sont marqués en rouge et par des étoiles '*'
- des éléments/attributs optionnels ou secondaires : certains sont exploitables pour la conception pédagogique et certains représentent des détails techniques à bas niveau.

4.2.3.2.3. Identification des attributs et des propriétés des éléments

L'analyse IHM-micro consiste à identifier les caractéristiques des éléments du langage de conception pédagogique d'une plateforme. Les attributs de ces éléments ainsi que leurs propriétés telles que leurs types doivent être identifiés ; plusieurs types sont possibles tels qu'entier, booléen, chaîne de caractère, une liste de choix, etc. Certains attributs possèdent des champs de valeur spécifiques ainsi que des initialisations par défaut. Ces derniers représentent également une partie du métier de conception pédagogique et doivent être identifiés.

Basées davantage sur les technologies et les techniques favorisant les aspects graphiques, certains éléments, attributs ou propriétés sont désignés par les composants de l'interface graphique (des widgets, des icônes, des figures, etc.). Souvent, des composants graphiques sont de la même forme mais ils sont associés à des éléments différents. L'analyse IHM-micro consiste également à identifier les attributs/caractéristiques résidants sous ces composants et à les attribuer aux éléments concernés. Toutefois, certains composants peuvent être écartés parce qu'ils ne sont pas liés à la conception pédagogique ou définis pour des finalités techniques (d'affichage, d'administration, d'organisation, etc.).



Figure 38 : Widgets graphiques dans une interface

Par exemple, la figure 38 illustre l'interface de conception de cours sur la plateforme Moodle où plusieurs éléments/attributs sont représentés par des composants graphiques (widgets). Les widgets (1 et 2) et (4 et 5) ont la même forme mais ils sont associés à deux éléments différents. Le widget (1) permet d'afficher/cacher un élément du cours (tel qu'un forum ou atelier, etc.). Le widget (2) permet d'afficher/cacher une section entière du cours et également les éléments qu'elle contient. Les widgets

(4) et (5) permettent de déplacer (verticalement) respectivement une section ou un de ses éléments. Nous avons tenu compte de ces widgets parce qu'ils interviennent à la conception des éléments du cours à l'exception du widget (3) qui a une finalité d'affichage (déplacer horizontalement un élément).

4.2.3.3. Formalisme du modèle IHM-micro

Pour capitaliser les caractéristiques des éléments de la conception pédagogique au sein des modèles IHM-micro, nous avons choisi le formalisme de cartographie (*mindmap* en anglais). Ce format permet la représentation et la description des éléments de conception pédagogique en termes d'attributs, de types, de propriétés, de domaines de valeur, d'initialisations, de contraintes et de relations. De l'analyse IHM-micro résulte plusieurs modèles dont chacun représente un élément spécifique ainsi que ses attributs et ses caractéristiques. La figure 39 montre un extrait du modèle IHM-micro du concept forum sur la plateforme Moodle.

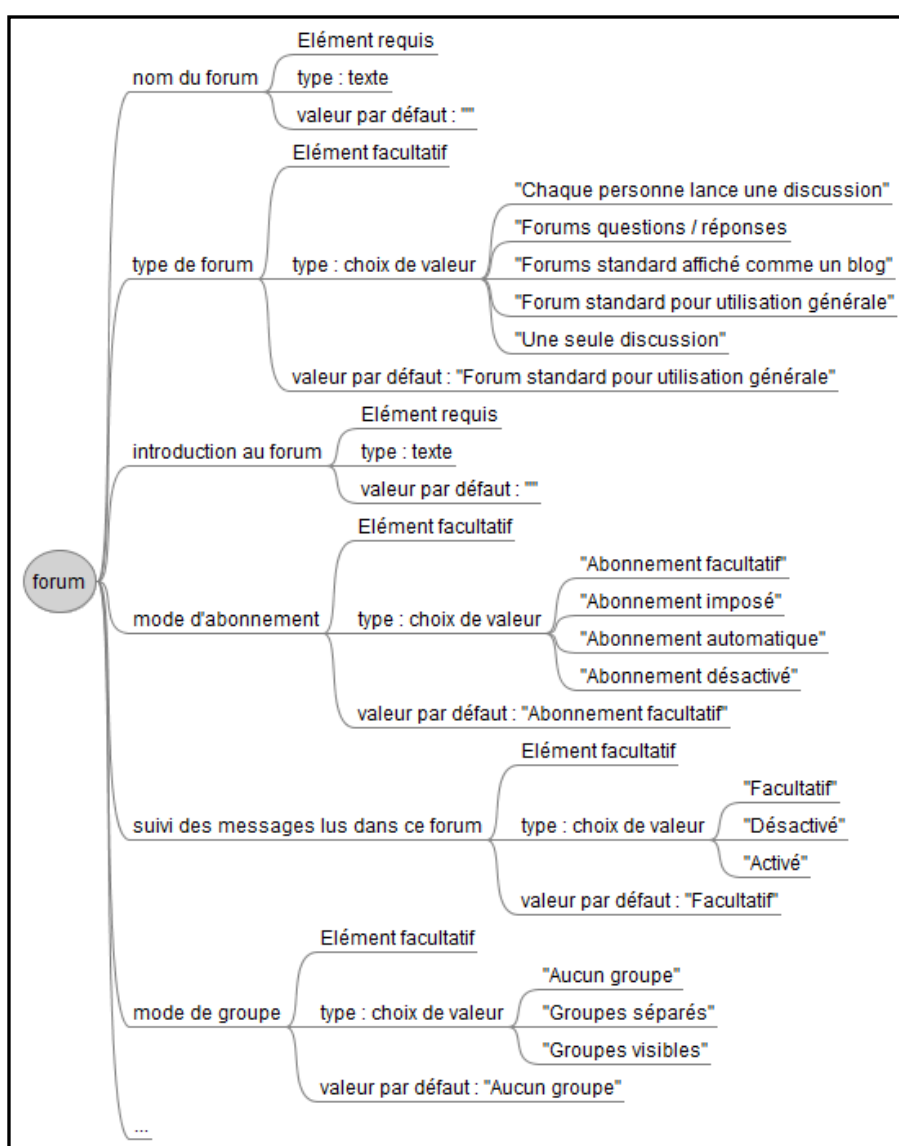


Figure 39 : Extrait du modèle IHM-micro d'un forum de Moodle

Ce modèle est le résultat de l'application de l'analyse IHM-micro sur l'interface 'ajout forum' au sein de la plateforme Moodle (figure 37). Il décrit uniquement les attributs en relation avec la conception

pédagogique du forum tels que 'nom du forum', 'type de forum', 'introduction au forum', 'mode d'abonnement', 'suivi de messages sur ce forum', etc. Pour chaque attribut, nous décrivons son critère obligatoire ou facultatif, son type, ses champs de valeurs s'ils existent, et sa valeur par défaut. Dans ce modèle, nous ne trouvons pas les attributs techniques proposés par l'interface de la plateforme parce qu'ils sont écartés pendant l'analyse.

4.2.4. Composition du modèle IHM

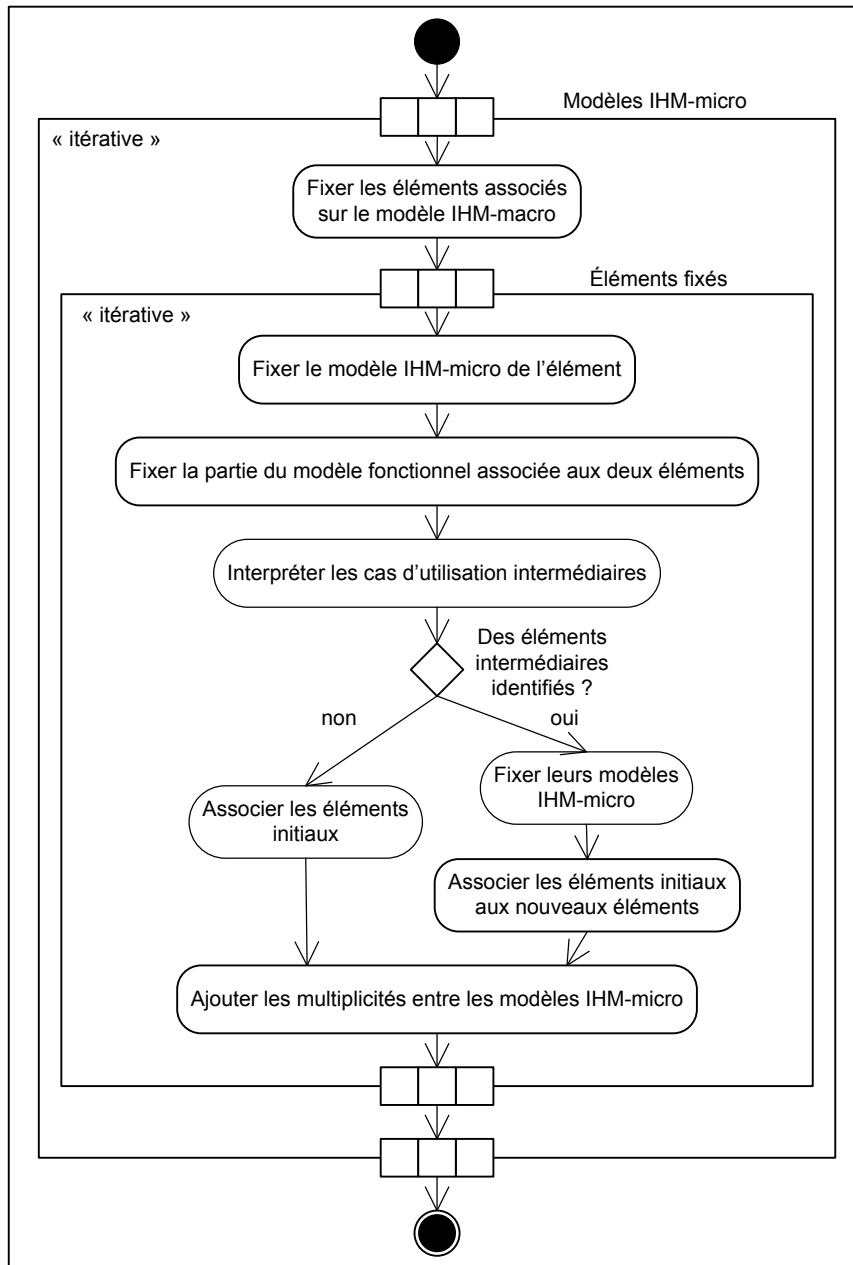


Figure 40 : Diagramme d'activité de la composition

La composition est la dernière activité au sein de l'analyse centrée IHM. Elle consiste à formaliser le langage de conception pédagogique d'une plateforme dérivé de l'analyse IHM. Le processus de composition vise à associer les modèles IHM-micro au sein d'un modèle unique (le modèle IHM). En choisissant un modèle IHM-micro d'un élément particulier, nous allons fixer les éléments qui lui sont

reliés sur le modèle IHM-macro. Après avoir choisi un de ces éléments, nous prenons en compte de son modèle IHM-micro. Sur le modèle IHM-macro, ces deux éléments sont associés. Cependant, avant de spécifier la relation entre eux sur le modèle IHM, nous menons une nouvelle vérification. Nous consultons le modèle fonctionnel pour vérifier si des nouveaux éléments intermédiaires (entre les deux éléments concernés) peuvent leur être associés. En comparaison avec l'identification sur le modèle IHM-macro, l'identification des relations à partir du modèle fonctionnel est plus complexe. Les nouveaux éléments sont identifiés à partir de la hiérarchie des diagrammes du cas d'utilisation au sein du modèle fonctionnel. Tout d'abord, nous fixons les diagrammes, et souvent leurs cas d'utilisation, dédiés aux premiers éléments. Ensuite, nous interprétons (s'ils existent) des diagrammes ou des cas d'utilisation intermédiaires et spécifiques à des nouveaux éléments. Lorsque de nouveaux éléments intermédiaires existent, leurs modèles IHM-micro sont fixés et ils sont ensuite associés aux premiers modèles IHM-micro. Lorsqu'aucun nouvel élément n'est identifié, l'association entre les deux premiers éléments est spécifiée. Une fois que les modèles IHM sont associés, les multiplicités entre ces modèles doivent être identifiées et représentées sur le modèle IHM. La figure 40 illustre l'ensemble des étapes de composition par le biais d'un diagramme d'activité du langage UML.

Par exemple, dans le cas de la plateforme Moodle, la composition du modèle IHM-micro de l'élément forum a nécessité (1) d'identifier les éléments reliés sur le modèle IHM-macro et (2) de vérifier la non-existence des éléments intermédiaires. En consultant le modèle IHM-macro, nous avons identifié l'élément discussion associé au forum. Ensuite, nous avons fixé les modèles/cas d'utilisation dédiés à ces éléments. Aucun cas d'utilisation associé à un nouvel élément n'est identifié. Cela conforme ainsi la relation entre les deux éléments forum et discussion. Leurs modèles IHM-micro sont ainsi associés. Enfin, la multiplicité de cette association est définie.

4.2.4.1. Identification des multiplicités entre les modèles

La composition consiste à définir les relations entre les modèles IHM-micro. Cependant, la description complète d'une relation nécessite la définition de la multiplicité entre ses éléments. La multiplicité souligne le nombre d'objets qui peuvent participer à une relation avec un autre objet. Elle est représentée sous forme d'intervalle d'entiers positifs ou nuls comportant une borne maximale et une borne minimale, intervalle dans lequel la multiplicité d'une entité peut prendre sa valeur :

- la borne minimale (généralement 0 ou 1) décrit le nombre minimum de fois qu'une entité ou un élément peut participer à une relation.
- la borne maximale (généralement 1 ou n) décrit le nombre maximum de fois qu'une entité ou un élément peut participer à une relation.

La détermination des multiplicités entre deux éléments consiste à interpréter les associations possibles de leurs occurrences. Deux stratégies sont possibles pour déterminer les multiplicités :

- L'interprétation des éléments concernés au sein d'une situation d'apprentissage existante : tout d'abord, les occurrences des éléments concernés doivent être identifiées. Ensuite, les associations possibles entre ces occurrences sont interprétées. Cette stratégie nécessite une banque consistante de situations d'apprentissage. Elle dépend directement de leur contenu. Certaines multiplicités ne peuvent pas être déterminées lorsque les éléments concernés ne sont pas utilisés au sein de ces situations.

- La spécification de nouveaux éléments au sein d'une situation d'apprentissage : cette stratégie consiste à étudier les multiplicités sur le plan pratique. Elle consiste (1) à créer plusieurs occurrence(s) des deux éléments puis (2) à établir les relations possibles entre elles. Deux cas sont possibles pour suivre cette stratégie :
 - Les occurrences des deux éléments sont créées séparément puis elles sont associées. Par exemple, les séquences sur la plateforme Ganesha sont créées puis elles sont associées aux modules concernées. La détermination des multiplicités dépend des associations possibles entre les occurrences des deux éléments.
 - Les occurrences du premier élément sont ajoutées à partir de l'interface du deuxième élément (le cas de Moodle). Cette méthode consiste à créer une ou plusieurs occurrences (si possible) du premier (respectivement du deuxième) élément à partir de l'interface dédiée au deuxième (respectivement au premier) élément.

Pour ces deux stratégies, lorsque plusieurs occurrences du premier élément (respectivement du deuxième) peuvent être associées à une occurrence du deuxième élément (respectivement du premier) alors la multiplicité est de type $(0,n)$ ou $(0,*)$. Dans le cas contraire, elle est de type $(0,1)$. La figure 41 illustre la démarche d'identification des multiplicités par le biais du diagramme d'activité suivant.

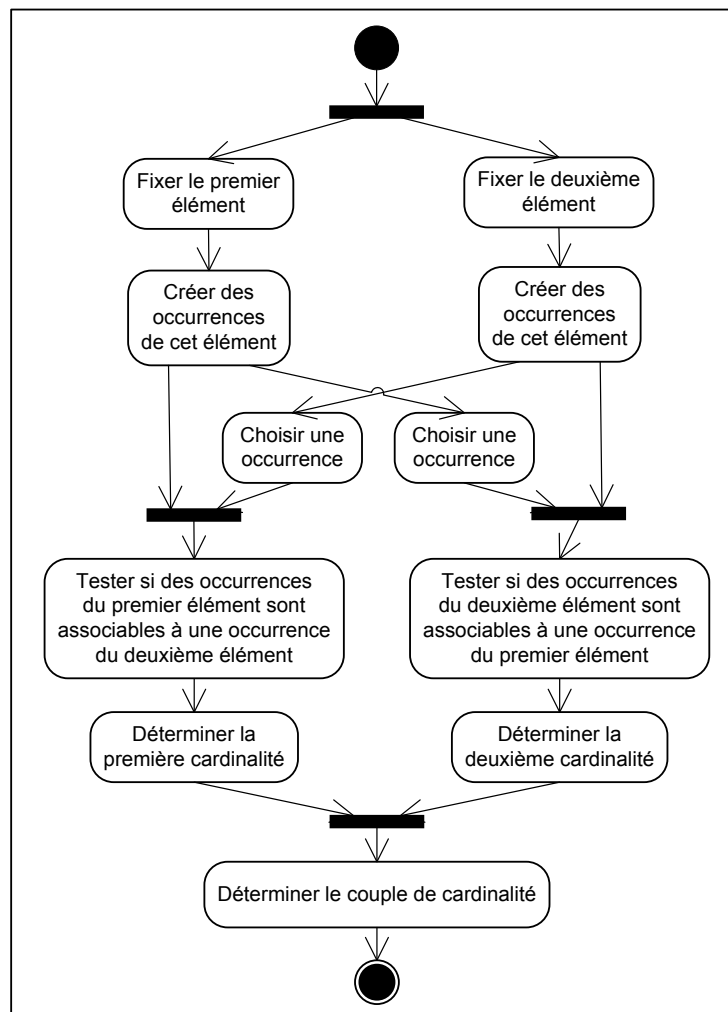


Figure 41 : Diagramme d'activité de l'identification des multiplicités

Nous avons choisi de représenter les multiplicités sur les relations en adoptant la méthode de conception *Merise*. Les multiplicités sont représentées par deux couples de type (x,x) (x,x). Les x désignent les trois possibilités suivantes : 0 ou 1 ou *.

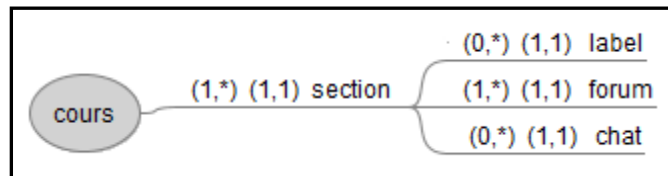


Figure 42 : Représentation des multiplicités sur le modèle IHM

La figure 42 illustre un exemple de composition d'éléments, désignant des modèles IHM-micro. Elle représente un extrait du modèle IHM de la plateforme Moodle. Chaque élément de ce modèle désigne son modèle IHM-micro. Les multiplicités sont exprimées via deux couples. Par exemple, le couple de multiplicités (1,*) (1,1) entre *Cours* et *Section* exprime qu'un cours peut être composé d'une ou plusieurs section(s) (la multiplicité (1,*)) et qu'une section est associée à un seul cours (multiplicité (1,1)). Les autres multiplicités expriment qu'une activité particulière (label, forum, chat, etc.) ne peut être associée qu'à une seule section. Par contre, une section est capable de contenir plusieurs activités (et également plusieurs ressources). Nous présentons dans le chapitre suivant et dans les annexes A et B des exemples concrets de la composition des modèles IHM-micro associés à des éléments de la plateforme Moodle et Ganesha.

4.3. Analyse centrée technique

La deuxième partie du processus concerne l'analyse centrée technique. Elle concerne plusieurs points techniques au sein des plateformes à analyser : les bases de données, le code source, la sauvegarde/restauration des cours (si elle existe), etc. Dans cette étape, la source principale d'information pour identifier le langage de conception pédagogique est la base de données de la plateforme concernée. Les autres analyses techniques sont réalisées lorsque des difficultés d'analyse de la base de données sont rencontrées et également au sein de l'étape de confrontation (troisième partie du processus d'analyse).

L'analyse de la base de données correspond en partie à la rétro-ingénierie (*reverse-engineering*) de la base de données. La rétro-ingénierie est un processus d'analyse de la version opérationnelle d'un composant logiciel qui vise à en reconstruire les spécifications techniques et fonctionnelles [Henrard et al. 1996]. Les objectifs principaux de la rétro-ingénierie sont la re-documentation, la conversion, la maintenance ou l'évolution d'anciennes applications, ainsi que la modélisation d'un système d'information existant. Cette modélisation décrit les concepts qui ont conduit à l'implémentation de ce système. En fonction du contexte du projet, diverses sources d'information disponibles peuvent être exploitées telles que les sources formelles (les bases de données, les programmes sources de l'application, le dictionnaire de données s'il existe, etc.).

Dans notre approche, l'analyse de la base de données consiste à spécifier le modèle conceptuel de données réduit. Lorsque le modèle conceptuel complet existe déjà ou bien fourni par les créateurs des plateformes, il est possible de l'étudier afin d'identifier la partie relevant de la conception pédago-

gique. Les parties administratives et techniques doivent être écartées de ce modèle. Plusieurs outils de retro-conception des bases de données peuvent être utilisés pour générer leurs modèles relationnels de données tels que *MyEclipse*, *PowerAMC*, *Win'Design*, *DBDesigner*, etc. Cependant, la majorité de ces outils ne sont pas gratuits et leurs résultats dépendent directement de la taille de la base de données.

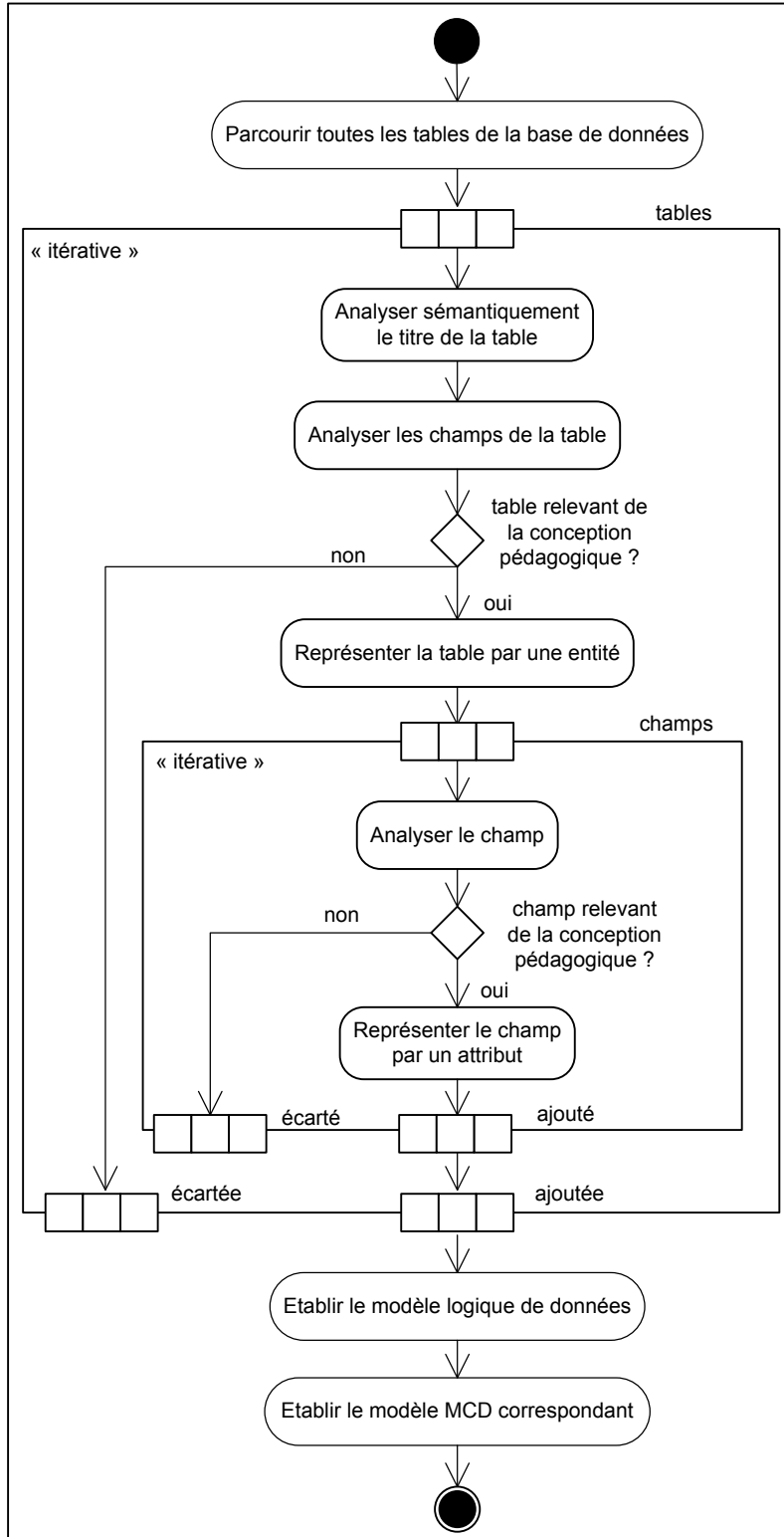


Figure 43 : Diagramme d'activité de l'analyse de la base de données

Notre approche consiste à restreindre l'analyse de la base de données aux tables/colonnes en relation avec la conception pédagogique. Ces tables sont spécifiques pour la sauvegarde des paramètres et du contenu du cours. Elles représentent aussi les attributs des éléments pédagogiques (activité, ressource, etc.). Sur la base de cette hypothèse, l'analyse a consisté dans un premier temps à la spécification du modèle logique de données réduit. Nous avons constaté que ce modèle ne permettait pas de représenter certains détails et contraintes entre les tables. Pour cela, nous avons choisi de formaliser le modèle technique par le biais du modèle conceptuel de données de la méthode Merise. L'obstacle principal concerne l'identification de ces tables spécifiques. Les informations issues de l'analyse centrée IHM semblent utiles pour ce cas.

Nous avons proposé notre propre méthodologie d'analyse pour capitaliser le langage de conception pédagogique sous la forme d'un modèle conceptuel de données (MCD). La figure 43 décrit, par le biais du diagramme d'activité UML, les étapes de la méthodologie. Tout d'abord, elle consiste à parcourir toutes les tables afin d'esquisser une première ébauche du modèle et à se focaliser sur les tables référençant des éléments de conception pédagogique. Ces tables sont identifiées via l'analyse sémantique de leurs titres ou de leurs champs d'enregistrement. D'autres tables peuvent être également identifiées selon leurs dépendances avec d'autres ou bien selon les clés étrangères proposées. Cependant, certaines plateformes définissent les relations entre certaines tables par des tables spécifiques. Ces tables doivent être identifiées afin de spécifier le modèle logique de données de ces artefacts. Notre analyse consiste à spécifier le schéma réduit de la base de données par le biais des règles de *reverse engineering* des bases de données. Le modèle conceptuel de données peut être finalement spécifié à partir de ce schéma. Ce modèle est important pour représenter le modèle métier selon un point de vue technique. Il permet par construction de cacher les détails techniques de la structure des bases de données et d'éviter les redondances et les mauvaises conceptions. Par exemple, la spécification du modèle technique par le format du modèle conceptuel de données permet d'écarter les tables dédiées à la définition des relations entre d'autres tables.

4.3.1. Parcours des tables

La première étape consiste à parcourir toutes les tables de la base de données de la plateforme. Elle a pour objectif d'avoir une vision globale sur les tables afin d'esquisser une première ébauche du modèle. Elle porte sur la lecture et l'analyse sémantique des titres des tables et de leurs champs. Malgré la diversité des plateformes et des systèmes LMS, il n'existe pas une liste exhaustive des termes et des concepts utilisés en conception pédagogique. Les nominations de certains éléments pédagogiques peuvent être différentes d'une plateforme à une autre. Par exemple, un cours dans Ganesha est composé de séquences alors qu'un cours dans Moodle est composé de sections. Pendant cette étape, des tables spécifiques à des nouveaux concepts pédagogiques, autres que les pré-requis indispensables à l'analyse, peuvent être identifiées.

4.3.2. Identification des tables/champs relevant de la conception pédagogique

La deuxième étape consiste à identifier les tables susceptibles d'avoir des données relevant de la conception pédagogique. L'analyse sémantique des titres des tables, de ses champs/colonnes ainsi que les pré-requis pédagogiques peuvent guider à l'identification de ces tables. Souvent, les tables de la base de données sont nommées par le concept principal qu'elles représentent. Par exemple, les tables de la

base de données de Moodle sont nommées soit par l'élément principal de la table (exemple *forum*, *quiz*, *workshop*, etc.), soit par la composée du nom de l'élément principal et d'une spécification de la table telles que les tables *forum_discussion*, *forum_posts*, etc.

Ces tables sont dédiées à la conception pédagogique lorsqu'une partie de leurs titres porte sur un élément de la conception pédagogique. Cette nomination peut aider l'analyste à identifier le concept pédagogique principal de la table ainsi que des relations entre ces tables. Outre les pré-requis pédagogiques que doit posséder l'analyste, d'autres moyens et méthodes permettront d'étendre l'identification des tables spécifiques à des éléments dédiés à la conception pédagogique. Par exemple, les clés étrangères et également les tables spécifiques à la définition des relations (dépendances fonctionnelles) permettent d'indiquer de nouvelles tables liées à la conception pédagogique.

L'identification des tables est plus complexe lorsqu'elles sont intitulées par des nominations non significatives (des titres ou codes spécifiques). Il est ainsi nécessaire d'analyser les enregistrements des tables et souvent d'interpréter l'ajout des éléments pédagogiques à partir des interfaces sur la table. Lors de l'analyse de la base de données, plusieurs tables sont écartées. Elles ne sont ni analysées ni représentées sur le modèle final parce qu'elles ont des finalités techniques plutôt que pédagogiques. Elles sont spécifiques à des usages de gestion, d'administration, etc. Toutefois, nous ne prenons pas en compte tous les champs des tables identifiées. Les champs considérés comme techniques plutôt que pédagogiques doivent être écartés. Par exemple, les champs représentant des détails techniques ou définis à *null* ou ayant toujours la même valeur sont écartés. Nous prenons les champs portant sur la définition des éléments tels que *name* et *introduction* et les attributs pédagogiques de description des éléments.

4.3.3. Spécification du modèle logique de données réduit de la base de données

La troisième étape consiste à établir le modèle logique réduit de la base de données à partir des tables identifiées dans l'étape précédente. La spécification de ce modèle doit répondre à certaines règles.

- Règle 1 : représenter chaque table par un modèle dont son titre est identique au celui de la table.
- Règle 2 : représenter chaque champ/colonne identifié(e) par un attribut.
- Règle 3 : déterminer les domaines de valeurs des attributs. Les champs définis à "*null*" ou ayant une valeur par défaut (0 par exemple) peuvent être ignorés.
- Règle 4 : relier les tables qui possèdent des clés étrangères aux tables concernées (celles dont leurs identifiants correspondent aux clés étrangères).
- Règle 5 : relier les tables qui possèdent des identifiants composés aux tables concernées (celles dont leurs identifiants correspondent à une partie de leur identifiant).

4.3.4. Spécification du modèle MCD réduit

La quatrième étape consiste à produire le Modèle Conceptuel de Données réduit de la base de données à partir du modèle logique de données, résultat de l'étape précédente. Cette étape est importante parce qu'elle permet de mettre en évidence les contraintes qui existent entre les entités et de cacher plusieurs

détails techniques non liés à la conception pédagogique. Le modèle logique de données de la base de données n'impose pas certaines contraintes existantes dans le monde réel (dont les plus populaires sont les dépendances fonctionnelles et les dépendances d'inclusion). Egalement, plusieurs tables au sein de ce modèle sont spécifiques seulement à la définition des relations entre d'autres tables. Ces tables, ne portant pas sur la conception pédagogique, seront disparues automatiquement pendant la génération du modèle final (le MCD réduit).

Le modèle conceptuel de données est généré grâce à l'application des règles de transformation de rétro-conception ou *reverse-engineering* sur le modèle logique de données. Ces règles correspondent aux règles de transformation du modèle logique en modèle conceptuel de données définis par la méthode Merise mais appliquées dans le sens inverse [Gruau 2006] [Panet et al. 1994] [Tardieu et al. 2000]:

- Règle 1 : chaque table dont la clé primaire ne contient pas de clés étrangères devient une entité dont l'identifiant est la clé primaire de la table et dont les attributs sont les champs de la table à l'exception des clés étrangères.
- Règle 2 : chaque table dont la clé primaire est composée exclusivement de clés étrangères qui référencent une seule clé primaire, devient une sous-entité ou une sous-association (les autres colonnes non clés étrangères de la table deviennent des attributs de cette sous-entité).
- Règle 3 : chaque table dont la clé primaire est composée exclusivement des clés étrangères qui référencent plusieurs clés primaires, devient une association autour de laquelle toutes les cardinalités maximales valent n , c'est-à-dire soit une association binaire de type (n,m) soit une association ternaire ou plus (les autres colonnes non clés étrangères de la table deviennent des attributs de l'association).
- Règle 4 : chaque table dont la clé primaire est composée partiellement de clés étrangères provient soit d'une optimisation soit d'un identifiant relatif d'une entité (auquel cas les autres colonnes non clés étrangères de la table deviennent des attributs de cette entité).
- Règle 5 : les colonnes clés étrangères restantes deviennent des associations binaires de type $(1, n)$ s'il n'y a pas de contrainte d'unicité ou de type $(1,1)$ s'il y a une contrainte d'unicité (il faut trouver un nom à cette association).
- Règle 6 : la cardinalité minimale vaut 1 pour les clés étrangères qui font partie d'une clé primaire ou qui possèdent une contrainte (non vide), sinon elle vaut 0.

4.4. Confrontation et formalisation du langage de conception pédagogique final

Les deux analyses centrées IHM et techniques mènent à la spécification de deux modèles selon deux points de vue différents et dont leurs représentations ne sont pas identiques. La dernière partie du processus concerne ainsi la confrontation entre ces deux modèles et la formalisation du langage de conception final. Son objectif est de croiser le métier de conception pédagogique manipulé par des ensei-

gnants-concepteurs au travers de l'IHM avec le métier de conception persistant stocké dans la base de données afin de parvenir à un modèle unique, correct et représentatif. La confrontation est basée principalement sur la comparaison entre les éléments des deux modèles IHM et technique. D'autres méthodes d'analyses techniques (analyse du code sources analyse du modèle des fichiers de sauvegarde, etc.) peut être adoptées lorsque des différences ou des ambiguïtés entre les données des modèles sont identifiées.

Dans cette partie, nous allons tout d'abord justifier pourquoi cette phase est importante dans ce processus en se basant sur les inconvénients de chaque méthode d'analyse. Ensuite, nous allons décrire le processus de confrontation et de formalisation. Au cours de ce processus, plusieurs opérations de vérifications peuvent être menées. Elles font l'objet de la section 4.4.4.

4.4.1. Intérêt de la confrontation et de la formalisation

Notre approche d'identification et de formalisation du langage de conception pédagogique porte sur l'exploitation de plusieurs méthodes d'analyse pour plusieurs raisons. Chaque méthode possède ses propres avantages et inconvénients. Par exemple, l'analyse IHM permet l'identification du langage de spécification des situations d'apprentissage visible de point de vue enseignant-concepteur. Cependant, elle dépend de la perception humaine et de la capacité d'interprétation. L'utilisation de plusieurs méthodes permet d'aboutir à des résultats similaires, complémentaires et parfois différents. Certaines méthodes peuvent cacher/montrer certains concepts, relations, contraintes, etc. Certains éléments sont définis par les développeurs de manière implicite. Dans cette section, nous justifions davantage notre choix consistant à (1) utiliser plusieurs méthodes d'analyse (analyses IHM et technique) et (2) à mener une étape de confrontation entre les modèles résultant afin de détecter certaines erreurs. Ce choix se justifie par le biais de l'hypothèse suivant : l'utilisation d'une seule méthode d'analyse peut relever plusieurs points négatifs. Nous avons identifié certains inconvénients pour les analyses IHM et techniques (base de données, fichiers de sauvegarde de cours, codes sources).

Inconvénients issus de l'analyse IHM

- Le langage identifié dépend directement de la compétence de l'analyste (sa capacité d'identifier les éléments, les relations, les contraintes, etc.).
- Risque de manque de concepts (non identifié pendant l'analyse).
- Le langage identifié ne garantit pas la persistance des futures modèles opérationnels.

Inconvénients issus de l'analyse technique, en particulier l'analyse de la base de données

- Plusieurs contraintes et structures de données ne sont pas déclarées explicitement lors de la création de la base de données parce qu'elles sont définies dans le code.
- Plusieurs éléments peuvent être définis différemment dans la base de données par rapport à leurs affichages dans l'interface.
- L'identification de certaines tables ou certains attributs, représentés(es) par des codes spécifiques, peut être très complexe.

- La base de données ne montre pas tous les détails du savoir-faire du système (en termes de conception pédagogique par exemple).
- Les modèles issus de l'analyse de la base de données ou bien de la retro-conception de la base de données ne sont pas capables de modéliser la sémantique de la plateforme pendant son utilisation.

Inconvénients issus de l'analyse des autres méthodes techniques

➤ **Analyse des fichiers de sauvegarde d'un cours**

- Plusieurs concepts/fonctionnalités ne sont pas toujours existants(es).
- Le format du fichier de sauvegarde est souvent non éditable, non consultable, etc.
- Les éléments de la sauvegarde sont réduits aux concepts utilisés dans le cours.
- Plusieurs détails peuvent être écartés : tout dépend du contenu et de la configuration du cours et des paramètres de la sauvegarde.

➤ **Analyse des codes sources**

- L'analyse du code source nécessite de bonnes compétences en informatique telle que la maîtrise des langages de programmation utilisés.
- La compréhension et l'interprétation des codes sources n'est pas une tâche aisée.

4.4.2. Formalisme du modèle final

Dans cette section, nous allons présenter le formalisme du modèle final avant de décrire le processus de confrontation. Parmi les formalismes possibles (schéma XML, métamodèle, cartographie (*mind-map*), modèle conceptuel de données, etc.), nous avons choisi de représenter le modèle final par le biais du format d'un métamodèle. Un métamodèle est une description formelle d'un domaine spécifique en termes d'éléments, de syntaxe et de sémantique de la notation. Notre choix a été motivé par plusieurs raisons. Ce format servira pour le développement des outils de conception pédagogique en adoptant l'approche *Domain-Specific Modeling*. Dans notre travail, nous l'utiliserons également pour le développement des modules de communication entre les plateformes de formations et les outils de conception concernés.

Le métamodèle du langage de conception pédagogique sera spécifié au fur et à mesure de l'application du processus de confrontation. Les éléments confrontés et vérifiés sont représentés par des méta-classes. Également, les attributs confrontés et vérifiés sont représentés par des méta-attributs au sein de leurs méta-classes. Enfin, les relations entre les éléments sont confrontées et vérifiées avant d'être projetées par des relations entre les méta-classes associées. Nous détaillons davantage la spécification du métamodèle au sein de la section suivante.

4.4.3. Description du processus de confrontation

La phase de confrontation consiste à comparer les éléments des modèles IHM et technique afin de spécifier le modèle métier final du système en termes de conception pédagogique, en particulier le métamodèle du langage de conception pédagogique de la plateforme concernée. Ces deux modèles (IHM et technique) sont confrontés afin d'assurer la bonne formalisation du langage de conception pédagogique et sa cohérence globale. Notre démarche consiste à mener des vérifications concernant les éléments des deux modèles et portant sur plusieurs points tels que la définition d'éléments similaires, la non-existence de certains attributs, la divergence des types d'attributs, etc.

Nous avons modélisé le processus de confrontation par le biais d'un diagramme d'activité (figure 44). Le processus considère chaque élément du modèle centré IHM. Il vérifie l'existence de cet élément sur le modèle technique. En cas de non-existence, des analyses spécifiques, centrées IHM et technique sont exigées. Lorsque le problème d'existence n'est pas toujours résolu, d'autres méthodes d'analyses techniques sont adoptées et des choix subjectifs peuvent être retenus tels que la prise en compte ou le rejet de cet élément. Les analyses techniques outre l'analyse de la base de données, telles que l'analyse du code source, peuvent être utiles pour clarifier cette ambiguïté. Par exemple, l'analyse du code révèle plusieurs détails que le développeur a choisi d'implémenter pour des raisons de portabilité et d'efficacité. Elle concerne principalement le code de la représentation des données sur les IHMs concernées et les requêtes d'interrogation de la base de données, en particulier les requêtes d'insertion/sélection des données.

Lorsqu'un élément est bien défini, il est modélisé par une méta-classe dans le métamodèle. Nous avons choisi pour la nomination de la méta-classe le nom de l'élément du modèle technique. Ce choix stratégique répond aux exigences du métamodèle utilisé pour le développement de nouveaux outils et langages. Les futures modèles spécifiés par ces outils seront opérationnalisables sur les artefacts concernés. Ensuite, la confrontation concerne les attributs des éléments du langage. Tous les attributs des éléments du modèle IHM sont confrontés avec les attributs des éléments équivalents sur le modèle technique. La vérification concerne l'existence et le type des attributs. Nous détaillons ultérieurement les processus de vérification des attributs en termes d'existence et de définition (des types, des propriétés, etc.). Les attributs vérifiés sont représentés par des méta-attributs au sein de la méta-classe de l'élément parent. En cas d'ambiguïtés ou de non existence, de nouvelles analyses IHM et techniques plus fines sont nécessaires. Les méta-attributs vérifiés sont définis avec la même nomination des attributs du modèle technique. Enfin, les relations entre les méta-classes doivent être définies en se basant sur les relations existantes dans les modèles IHM et technique. Les multiplicités sont également vérifiées sur les deux modèles. Ensuite, elles sont représentées sur le métamodèle final. La figure 44 illustre les étapes de confrontation et de formalisation du métamodèle du langage de conception pédagogique.

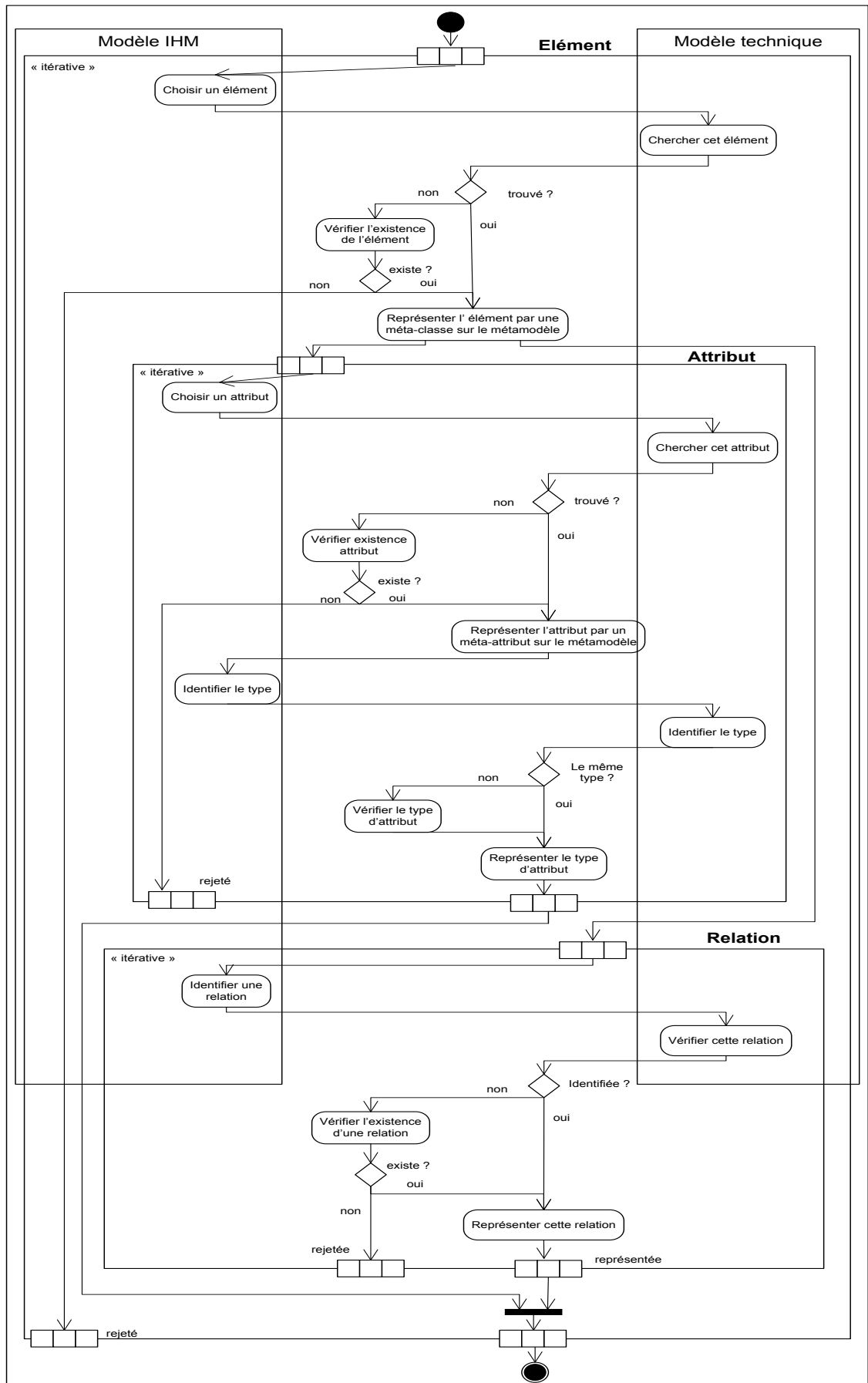


Figure 44 : Diagramme d'activité de la confrontation et formalisation

4.4.4. Processus de vérifications

L'analyse d'un système selon des méthodes différentes ne garantit pas l'obtention de modèles cohérents. Certaines parties de ces modèles peuvent être identiques, complémentaires et parfois différentes et contradictoires. Lorsqu'une différence est identifiée sur les deux modèles, elle est tout d'abord fixée avant de mener des opérations de vérification. Les vérifications exigent de nouvelles analyses spécifiques centrées IHM et technique pour les parties en question. Dans certains cas où la différence persiste malgré les analyses nouvellement effectuées (IHM et technique), d'autres méthodes d'analyse (analyse du code source, des packages de sauvegarde, etc.) doivent être menées. Plusieurs cas de différences entre les modèles peuvent être identifiés :

- La non-existence d'un ou plusieurs éléments et attributs dans un des deux modèles,
- La dissimilarité de définition d'un élément (ou d'une relation) sur les deux modèles,
- La non-existence de contraintes ou de multiplicités dans certains modèles,
- etc.

L'objectif de la phase de confrontation est de clarifier toutes les ambiguïtés. Nous avons étudié tous les cas possibles de différences entre les deux modèles IHM et technique. Les vérifications concernent ainsi les cas suivants :

- L'existence d'un élément
- Le type d'un élément
- Les attributs d'un élément (en termes d'existence et de types)
- Les multiplicités et types de relations

Les sous-sections qui suivent présentent les processus de vérifications que nous avons définis.

4.4.4.1. Vérification de l'existence d'un élément

La première étape de la confrontation entre les modèles IHM et technique concerne l'existence ou non d'un élément. En confrontant ces deux modèles, il est possible que certains éléments soient identifiés par une méthode d'analyse tandis que d'autres méthodes n'ont pas permis leurs identifications. Chaque élément du modèle IHM doit être vérifié sur le modèle technique. Nous avons spécifié le processus de vérification de l'existence d'un élément en tenant en compte de trois méthodes d'analyse : une analyse centrée IHM et deux analyses techniques (analyse de la base de données et du code source). La figure 45 illustre ce processus que nous avons modélisé par le diagramme d'activité suivant.

Le processus consiste à reprendre des analyses plus fines centrées IHM et technique. Après avoir fixé l'élément en question, l'IHM présentant cet élément doit être identifiée. Le modèle macro-IHM peut servir pour l'identification de cet élément. Lorsqu'aucune IHM n'est associée à cet élément, les interfaces en relation, présentant probablement cet élément telles que les interfaces d'un niveau supérieur (les entités parents de l'élément) doivent être identifiées et analysées. Le modèle fonctionnel peut également servir pour identifier certaines fonctionnalités dédiées à cet élément. Les relations entre les cas d'utilisation permettent d'identifier ses éléments ascendants et par la suite les interfaces dédiées. Lorsque l'IHM de cet élément a pu être identifiée, l'analyse IHM-micro est appliquée afin d'identifier les attributs et les propriétés de cet élément.

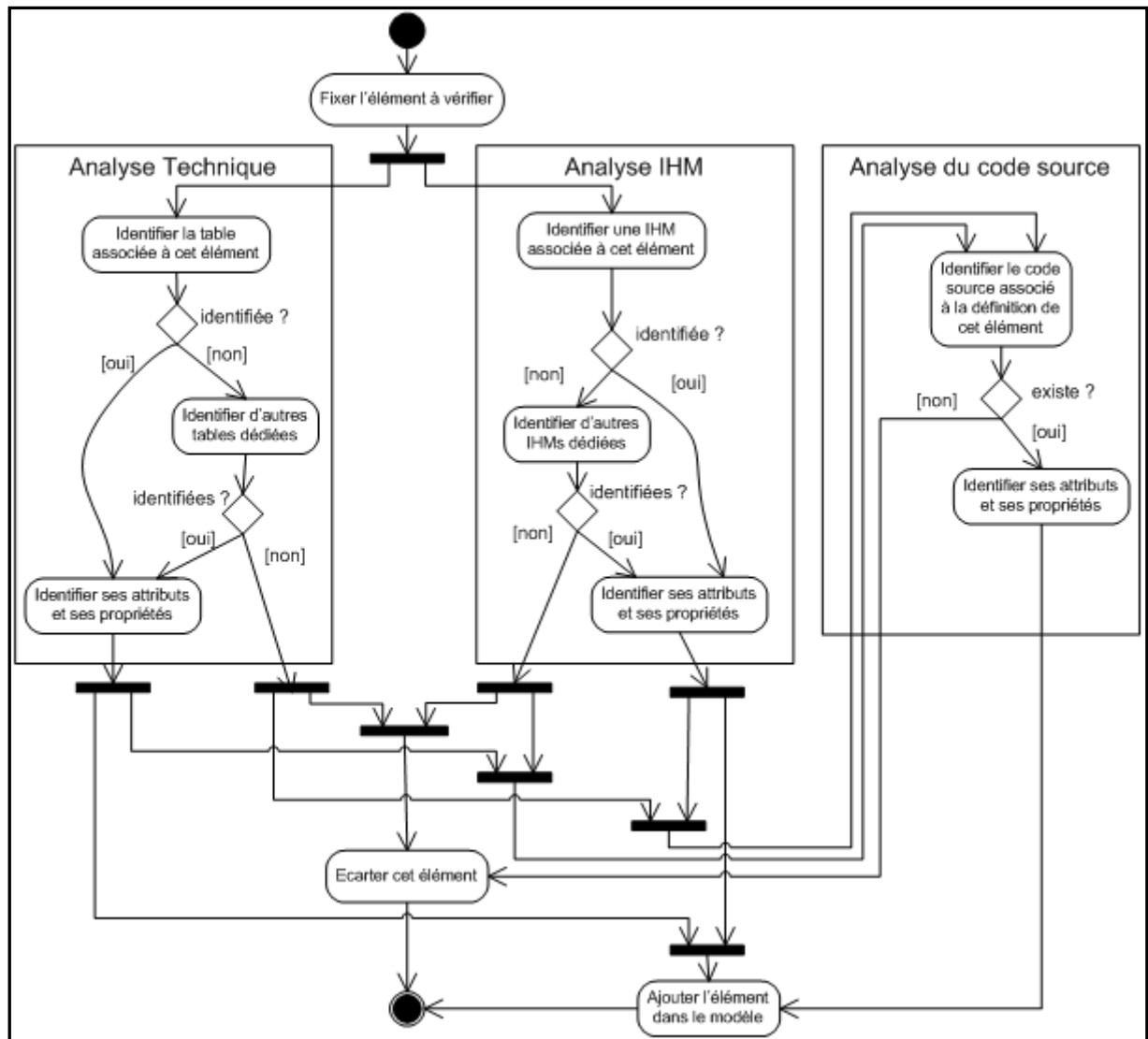


Figure 45 : Processus de vérification de l'existence d'un élément

En ce qui concerne l'analyse de la base de données, la première étape consiste à identifier la table représentant l'élément en question. Lorsqu'aucune table ne lui est associée, les tables en relation avec cet élément ou qui peuvent le contenir dans leurs champs doivent être identifiées et analysées. L'identification d'une table représentative de cet élément entraîne l'identification de ses attributs 'pédagogiques' et de ses propriétés parmi les champs de cette table.

Les nouvelles analyses des IHMs et de la base de données peuvent mener à plusieurs résultats :

- L'élément est identifié par les deux méthodes d'analyse. Il est ainsi ajouté dans le modèle en question. La non-identification précédente de cet élément est due à un manque de précision pendant l'analyse initiale.
- L'élément est identifié via une seule méthode. L'utilisation d'une nouvelle méthode d'analyse (telle que l'analyse du code source) est indispensable. La prise en compte de cet élément est basée sur le résultat de cette dernière analyse. Lorsque cette analyse a permis son identification, il est ajouté dans le modèle en question. Dans le cas contraire, il sera supprimé de ce modèle.

Toutefois, des décisions subjectives peuvent être prises. Par exemple, lorsque l'élément est toujours identifiable via l'analyse de la base de données et non par l'analyse des IHM, cet élément peut être écarté vu qu'il n'appartient pas à la partie visible du langage de la plateforme et ainsi il ne sera pas accessible aux enseignants-concepteurs dans ce cas.

- L'élément n'est pas identifié par les deux méthodes. Il est ainsi supprimé du modèle qui le définit. Ce cas est dû à une erreur pendant la première analyse.

4.4.4.2. Vérification du type d'un élément

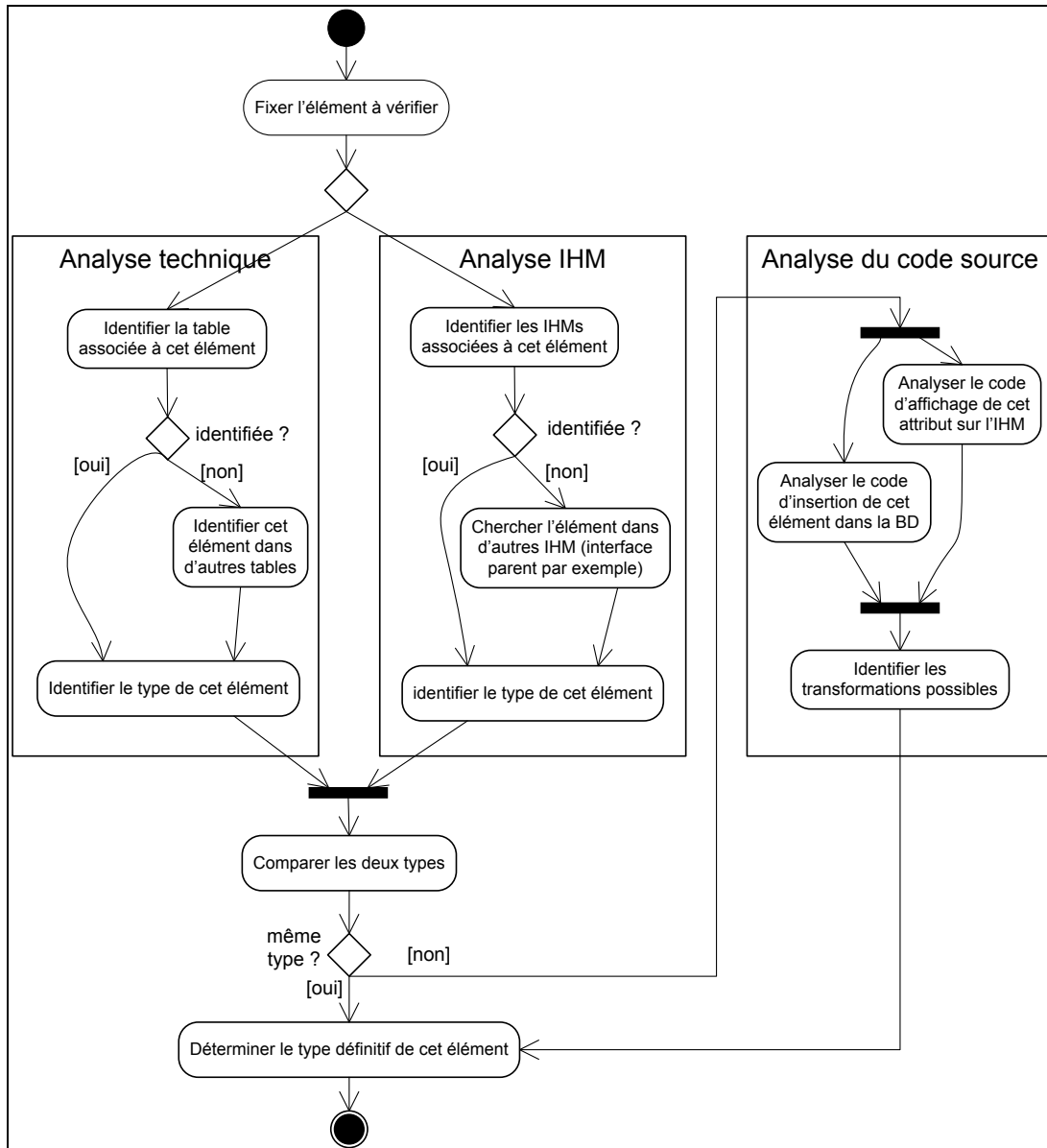


Figure 46 : Processus de vérification d'un type d'un élément

En ce qui concerne la vérification du type d'un élément, nous considérons que l'élément est déjà identifié dans les deux modèles IHM et technique. Cependant, son type est défini différemment. Les analyses, spécifiques à cet élément, centrées IHM et technique doivent être rétablies. De point de vue IHM, l'analyse consiste tout d'abord à identifier l'IHM de l'élément en question ou les IHMs en rela-

tion pouvant le représenter (par exemple les IHM d'un niveau supérieur). De point de vue base de données, l'analyse consiste à identifier la table présentant l'élément en question ou bien les tables en relation (les tables pouvant contenir des champs représentatifs de cet élément). Lorsqu'une IHM ou une table d'un élément est identifiée, son type doit être également identifié.

Les résultats des deux méthodes d'analyse sont comparés. Lorsque les deux méthodes mènent à l'identification d'un seul type, ce dernier sera pris en compte et attribué comme type définitif à cet élément. Nous pouvons déduire qu'il s'agit d'une erreur pendant la première phase d'analyse. Lorsque les deux analyses mènent toujours à deux types différents, le processus de vérification doit procéder par une autre méthode d'analyse. Nous choisissons l'analyse du code source comme méthode d'analyse complémentaire.

L'analyse du code source consiste à analyser le code de définition de l'élément sur l'IHM ainsi que le code d'interrogation des tables de la base de données dédiées à cet élément, en termes d'insertion, de mise à jour ou de sélection. Certaines transformations, codées par le développeur, peuvent être identifiées pendant (1) son enregistrement dans la base de données ou (2) son affichage sur l'interface associée. Certains choix subjectifs peuvent être pris en compte pendant l'affectation du type de l'élément. Le résultat de cette analyse est généralement conforme à un des résultats des analyses antérieures. Le type définitif de cet élément peut être ainsi déterminé. La figure 46 illustre le processus de vérification du type d'un élément en adoptant les deux méthodes d'analyse de base : l'IHM et la technique, et l'analyse du code source comme méthode complémentaire.

4.4.4.3. Vérification des attributs d'un élément

En ce qui concerne la vérification des attributs d'un élément, nous considérons que l'élément est identifié selon les deux méthodes d'analyse centrées IHM et technique. Cependant, certains de ses attributs ne sont pas identifiés via une méthode. Des nouvelles analyses IHM et technique, restreintes à cet élément, sont ainsi nécessaires.

Le processus de vérification considère dans un premier temps l'analyse IHM pour identifier l'IHM ou les IHMs associée(s) à cet élément, parmi celles du modèle IHM-macro. Ensuite, l'analyse IHM-micro est appliquée sur ces IHM afin d'identifier l'attribut en question. L'analyse de la base de données consiste à identifier la table représentative de l'élément en question. Ensuite, elle s'intéresse à l'identification de cet attribut parmi les champs de cette table. Dans un deuxième temps, les résultats des deux méthodes d'analyse sont comparés. Lorsque les deux méthodes mènent à l'identification de cet attribut, ce dernier sera ajouté au modèle en question et pris en compte au sein du métamodèle du langage de conception pédagogique. Nous pouvons déduire qu'il s'agit d'une erreur pendant la première phase d'analyse. Lorsque les deux analyses mènent toujours à deux résultats différents, l'utilisation d'une nouvelle méthode d'analyse, en particulier l'analyse du code source, est indispensable. Similairement au cas de vérification du type d'un élément, l'analyse du code source consiste à analyser les instructions de définition de l'élément sur l'IHM ainsi que le code d'interrogation des tables de la base de données dédiées à cet élément. Plus spécifiquement, nous analysons les instructions (les requêtes), d'insertion, de mise à jour ou de sélection de l'attribut concerné. Ces instructions peuvent expliquer davantage certaines transformations, codées par le développeur, pendant l'interrogation de la base de données. Certains choix subjectifs peuvent être pris en compte pendant

l'identification de cet attribut. Le résultat de cette analyse est comparé avec les résultats précédant afin de déterminer la prise en compte de cet attribut. L'identification de cet attribut implique également l'identification de ses propriétés. La figure 47 illustre le processus de vérification d'un attribut d'un élément en adoptant trois méthodes d'analyse (IHM, base de données et code source).

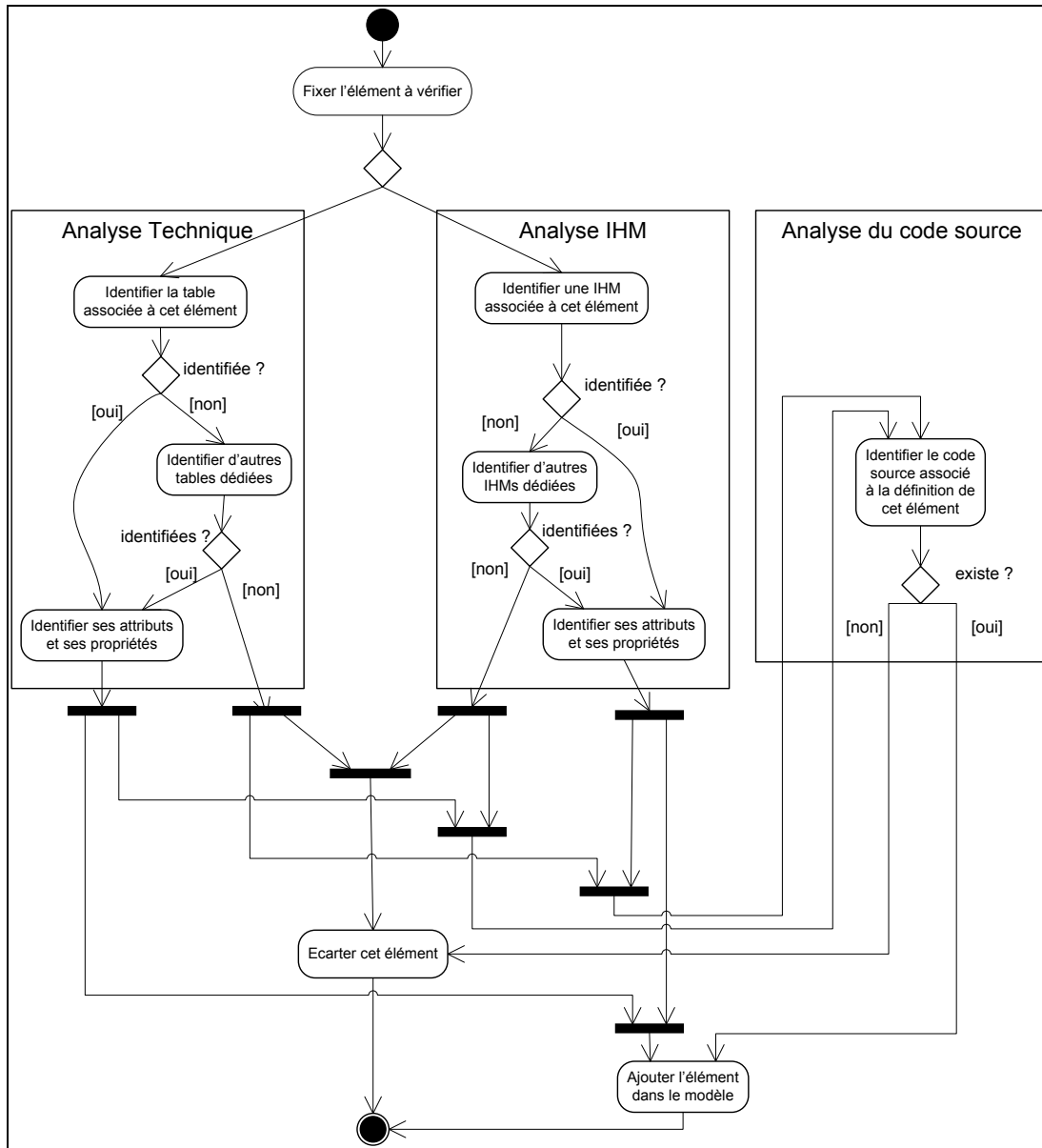


Figure 47 : Processus de vérification des attributs d'un élément

4.4.4.4. Vérification des multiplicités des relations

Les multiplicités entre deux éléments particuliers peuvent être différemment définies sur les deux modèles IHM et technique. Cette particularité nécessite la définition d'un processus pour leurs vérifications. Ce processus considère que les deux éléments sont déjà identifiés via les deux méthodes d'analyse centrées IHM et technique. Cependant, des nouvelles analyses (centrées IHM et technique), concernant l'association entre eux, doivent être rétablies. De point de vue IHM, le processus consiste à appliquer les étapes de la composition, définie au sein de l'analyse IHM, entre les deux éléments concernés. Les multiplicités doivent être identifiées en menant de nouvelles expérimentations par

l'interprétation des associations possibles entre plusieurs occurrences des éléments concernés. L'analyse de la base de données est restreinte à l'identification des tables dédiées. Ensuite, elle s'intéresse soit aux clés étrangères de ces tables soit aux tables définissant les relations entre les tables dédiées à ces deux éléments. Les multiplicités peuvent être ainsi déterminées par l'application des transformations du modèle logique vers le modèle conceptuel de données.

Les résultats des deux méthodes d'analyse sont ensuite comparés. Lorsque les deux méthodes mènent au même résultat, les multiplicités sont corrigées sur le modèle en question. Ensuite, elles sont représentées sur le métamodèle final. Lorsque les résultats sont différents, l'analyse du code source peut être utile. Nous analysons le code source d'interrogation des tables de ces éléments et des tables en relation (si elles existent) et également les instructions spécifiques à l'ajout de ces éléments et la définition des relations entre eux.

Les instructions du code source peuvent exprimer certains détails, codés implicitement par le développeur. Le résultat de cette analyse est généralement conforme à un des résultats précédents. La prise en compte des multiplicités finales peut être ainsi déterminée. La figure 48 illustre le processus de vérification des multiplicités des relations en adoptant les méthodes d'analyse IHM et technique (analyses de la base de données et du code source).

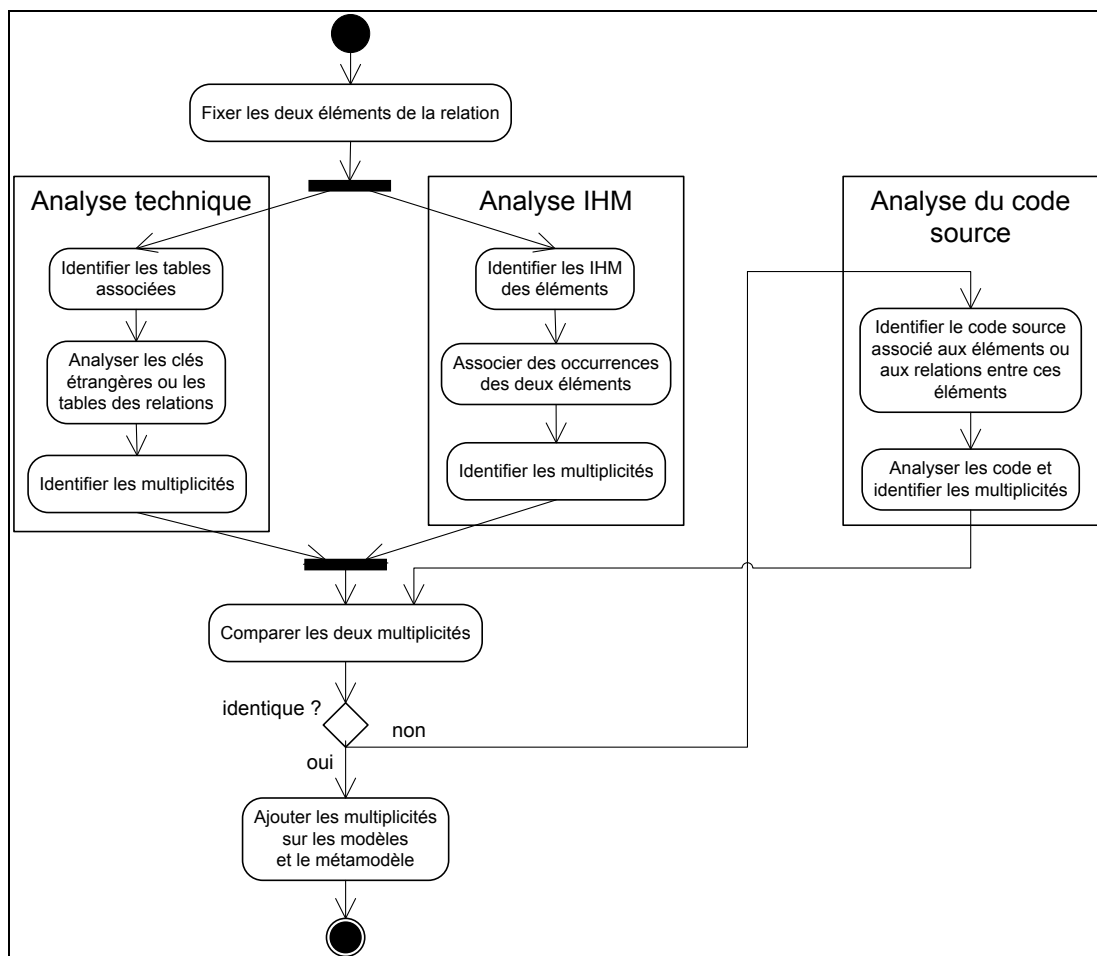


Figure 48 : Processus de vérification des multiplicités

4.5. Conclusion

Dans ce chapitre, nous avons spécifié la première partie de nos propositions qui porte sur le processus d'analyse, d'identification et de formalisation du langage de conception pédagogique d'une plateforme de formation.

L'objectif de ce processus est de formaliser ce langage à partir de deux points de vue différents et complémentaires : le point de vue IHM qui assure la partie du langage visible pour un utilisateur et le point de vue technique pour assurer la cohérence du langage avec son implémentation.

Les usagers de ce processus (analystes, ingénieurs de recherche ou pédagogiques, experts des plateformes, etc.) doivent avoir certaines compétences informatiques. La spécification du modèle technique (modèle conceptuel de données) n'est pas une tâche aisée. Elle nécessite (1) l'identification des tables relevant de la conception pédagogique d'une plateforme, (2) la génération du modèle logique de données et (3) la génération du modèle conceptuel de données par application des règles de retro-conception. Au cours de cette analyse, plusieurs difficultés techniques peuvent être rencontrées telles que l'identification des clés étrangères.

Dans le chapitre suivant, nous allons expérimenter ce processus sur deux plateformes de formation. Nous présentons l'application du processus sur un ensemble cohérent d'interfaces et de tables. Nous représenterons également certains cas particuliers rencontrés au sein de ces applications.

5

Application du processus sur les plateformes d'études

Sommaire

- 5.1. Introduction
- 5.2. Cas de la plateforme Moodle
- 5.3. Cas de la plateforme Ganesha
- 5.4. Conclusion

5.1. Introduction

Nous présentons dans ce chapitre l'application du processus d'identification et de formalisation du langage de conception pédagogique sur les deux plateformes Moodle et Ganesha, fréquemment utilisées au sein des établissements scolaires tels que l'université du Maine et l'IUT de Laval. Nous n'allons pas rendre compte avec exhaustivité de l'application complète du processus sur ces deux plateformes. Nous présentons une vue générique de ces applications, plus spécifiquement les étapes principales et les modèles résultants des différentes analyses. Nous proposons dans les annexes A et B un ensemble cohérent d'exemples pertinents dans le but d'illustrer toutes les spécificités et les particularités de ces applications ainsi que les solutions possibles proposées. L'objectif de ce chapitre est de justifier et d'argumenter certains choix dans notre proposition et de mettre en évidence certains cas particuliers, par le moyen des illustrations concrètes du processus et des modèles.

5.2. Cas de la plateforme Moodle

5.2.1. Introduction

La plateforme Moodle propose, à travers un nombre important d'interfaces, de nombreuses activités et types de ressources pour la spécification de situations d'apprentissage. L'application du processus d'identification et d'explicitation du métier de conception pédagogique sur cette plateforme a nécessité l'analyse complète de tous ses écrans et de toutes les tables de sa base de données. En fin de chaque analyse, nous présentons les modèles produits, et particulièrement le modèle du langage de conception pédagogique de cette plateforme.

5.2.2. Application de l'analyse centrée IHM sur Moodle

Nous rappelons que l'analyse centrée IHM consiste à identifier le langage de conception pédagogique du point de vue IHM : le langage visible à l'utilisateur. Elle est composée de trois analyses (IHM-macro, fonctionnelle et IHM-micro) et d'une étape de composition pour la formalisation du modèle centrée IHM.

5.2.2.1. Application de l'analyse IHM-macro sur Moodle

L'application de l'analyse IHM-macro sur Moodle consiste à identifier les interfaces liées à la conception des cours. Nous avons appliqué les deux stratégies d'analyse : (1) l'analyse d'une banque de cours disposée sur la plateforme puis (2) la spécification des nouveaux cours en termes de contenu et de paramètres. Tout d'abord, nous avons consulté le contenu des cours sur Moodle. Chaque élément du cours amène à une nouvelle interface, soit pour sa présentation ou pour sa modification. Nous avons identifié plusieurs IHM liées aux activités et aux ressources pédagogiques (leçon, atelier, forum, page, url, etc.). Puis, nous nous sommes intéressés à l'identification des éléments dédiés (comme le cas des discussions, sous-éléments du forum). Ces IHMs sont représentées sur le modèle IHM-macro par leurs concepts principaux qui sont identifiés grâce à l'analyse des titres et des chemins de navigation. Ensuite, nous nous intéressons à l'application de la deuxième stratégie qui peut être appliquée en amont ou en aval de la première stratégie. Nous avons ainsi spécifié le contenu d'un nouveau cours parmi les éléments proposés par la plateforme (des activités, des ressources, etc.). Lorsqu'un élément est choisi,

nous sommes redirigés vers l'interface de paramétrage. Leur concept principal est ainsi déterminé avant d'être vérifié sur le modèle IHM et si besoin ajouté. Nous avons remarqué que les IHMs de paramétrage représentent la description des éléments en termes d'attributs et de propriétés. Le processus propose leur analyse fine au sein de l'analyse IHM-micro. Ce choix répond à notre premier objectif qui est d'encadrer, dans un premier temps, le champ d'analyse en identifiant les interfaces relevant de la conception pédagogique et dans un deuxième temps s'intéresser davantage aux spécificités des éléments de conception pédagogique. Les concepts principaux identifiés sont représentés sur le modèle IHM-macro et ensuite associés en se basant sur les relations entre les interfaces permettant. La figure 49 illustre le modèle IHM-macro de la plateforme Moodle.

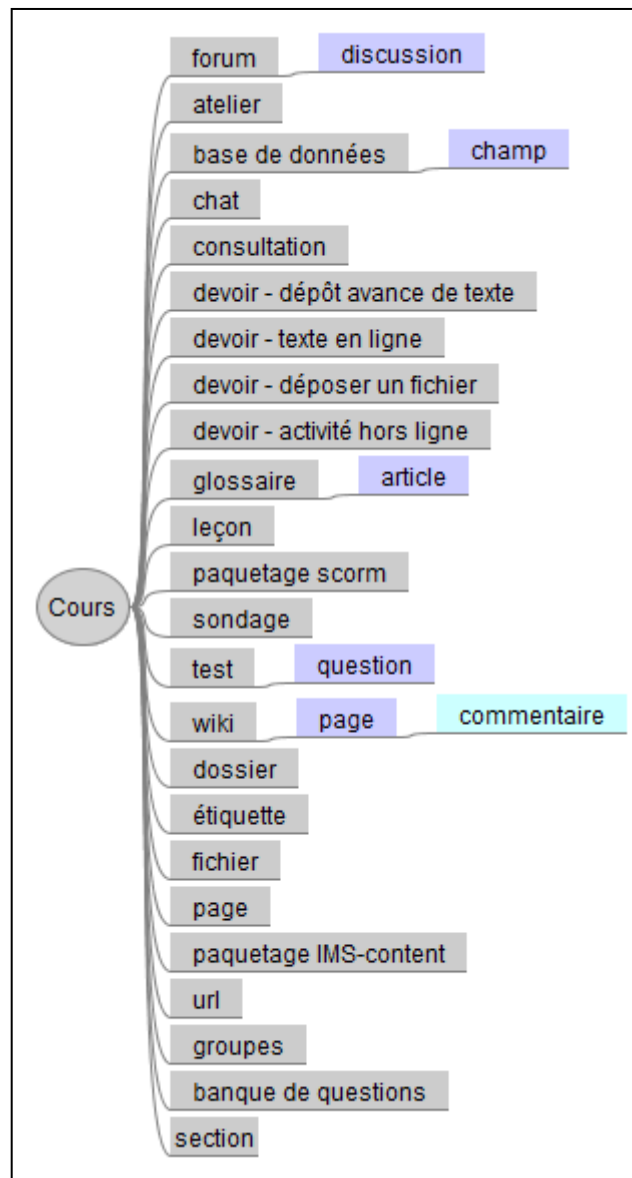


Figure 49 : Modèle IHM-macro de Moodle

5.2.2.2. Application de l'analyse fonctionnelle

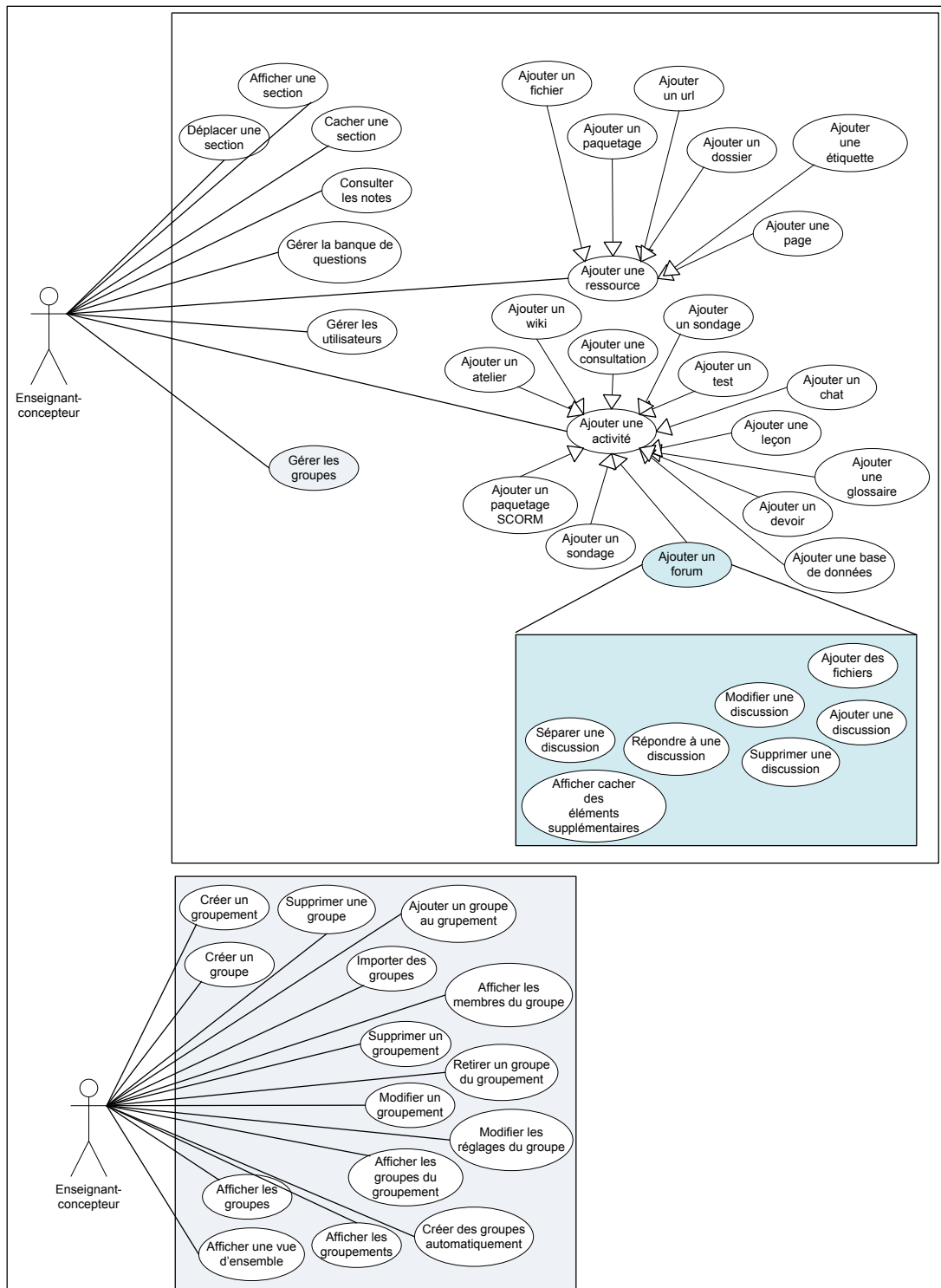


Figure 50 : Extrait du modèle fonctionnel final

L'analyse fonctionnelle consiste à identifier les fonctionnalités relevant de la conception pédagogique. L'application concrète de cette analyse sur les interfaces les plus significatives et les plus pertinentes pour la démonstration est détaillée au sein de l'annexe A. Au cours de cette analyse, nous avons tout d'abord décomposé chaque IHM en plusieurs zones. Ensuite, pour chaque zone, nous avons étudié les

composants de l'interface graphique (les widgets tels que les liens, les boutons, les icônes, etc.) pour identifier les fonctionnalités liées à la conception pédagogique (telles que '*ajouter une ressource*', '*ajouter une activité*', etc.) et pour écarter les fonctionnalités qui sont plus techniques que pédagogiques telles que les fonctions d'affichage. Les fonctionnalités des différentes zones sont représentées au sein du même modèle afin d'être associées au modèle principal. Cette analyse a permis la vérification de certaines hypothèses et déductions de l'analyse IHM-macro. Par exemple, les relations entre les IHMs du modèle IHM-macro sont interprétées par des relations entre les diagrammes du modèle fonctionnel (notion de fonctionnalités et de sous-fonctionnalités). La figure 50 illustre un extrait du modèle fonctionnel final. Nous avons choisi de séparer la représentation du modèle dédié à la gestion des groupes en lui attribuant la même couleur que la fonction '*gérer les groupes*', afin de ne pas complexifier la lecture du modèle global.

5.2.2.3. Application de l'analyse IHM-micro sur Moodle

L'application de l'analyse IHM-micro porte sur l'identification des caractéristiques des éléments de conception pédagogique. Nous avons remarqué que cette analyse est limitée aux interfaces de spécification ou de modification d'un élément/paramètre du cours. Elle aboutit à la spécification de plusieurs modèles IHM-micro dont chacun représente un élément. L'annexe A présente l'application de l'analyse IHM-micro sur un ensemble cohérent d'interfaces, en particulier les sections, les forums, les discussions et les groupes, illustrant les spécificités et les particularités de cette analyse.

Nous avons décomposé les IHMs concernées en plusieurs zones, souvent en reprenant la décomposition effectuée au sein de l'analyse fonctionnelle. Nous nous intéressons ensuite à l'analyse de chaque zone. Nous avons remarqué que plusieurs zones sont communes entre les IHMs de Moodle mais elles ne décrivent aucun élément de conception pédagogique. Elles sont ainsi analysées une seule fois. Souvent, l'analyse IHM-micro est consacrée à la zone présentée par un formulaire décrivant les attributs spécifiques d'un élément particulier.

Nous avons écarté plusieurs attributs que nous jugeons plus orientés techniques que pédagogiques. Par exemple, dans le cas du forum, nous avons écarté les attributs : '*la taille maximale de l'annexe*', '*le nombre maximale de l'annexe*', '*la durée de blocage*', '*le numéro d'identification*', etc. Nous avons pris en compte des attributs 'pédagogiques' suivants : '*nom du forum*', '*type du forum*', '*introduction au forum*', '*mode d'abonnement*', '*suivi des messages lus dans ce forum*', '*groupement*' et '*visible*'. Pour chaque attribut, nous avons analysé trois caractéristiques : attribut facultatif ou requis par l'interface, le type de l'attribut, les champs de valeur et la valeur par défaut. Ensuite, nous avons analysé les autres interfaces dédiées à cet élément que nous détaillons dans l'annexe A. La figure 51 représente le modèle IHM-micro dédié au forum.

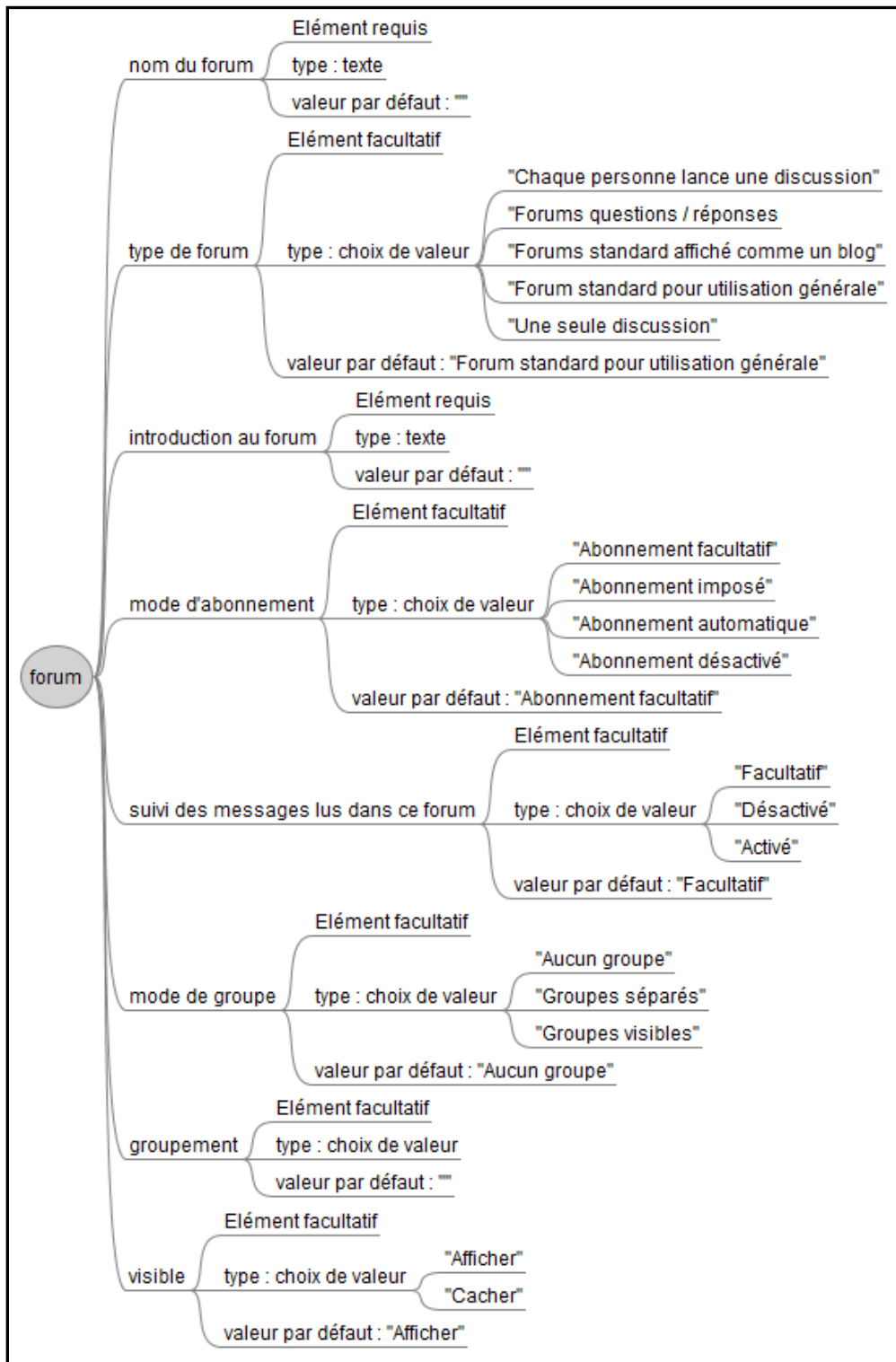


Figure 51 : Modèle IHM-micro d'un forum

5.2.2.4. Application de la composition

Une fois les modèles IHM-micro définis, la composition consiste alors à les associer pour formaliser le modèle IHM final du langage de conception pédagogique. Nous présentons dans l'annexe A un ensemble cohérent d'exemples concrets de la composition exposant ses spécificités ainsi que ses particularités. Les relations entre les modèles IHM-micro sont identifiées sur la base des modèles IHM-macro

et fonctionnel. En prenant le modèle IHM-micro d'un élément particulier, nous consultons le modèle IHM-macro afin de déterminer les éléments qui lui sont associés. Ensuite, en fixant un deuxième élément, nous consultons le modèle fonctionnel et nous vérifions s'il existe des fonctionnalités intermédiaires associées à d'autres éléments. Lorsqu'aucune fonctionnalité n'existe, nous spécifions la relation entre les modèles IHM-micro de ces deux éléments. Dans le cas contraire, le modèle IHM-micro du premier élément est associé au modèle IHM-micro de l'élément intermédiaire qui est également associé au modèle du deuxième élément. Ensuite, nous nous intéressons à la détermination des multiplicités des associations. Nous avons mené des expérimentations sur Moodle consistant à vérifier la possibilité d'associer plusieurs instances du premier élément de l'association à une instance du deuxième élément et inversement. Par exemple, pour déterminer la multiplicité entre les forums et les discussions, nous avons pris un forum existant, nous avons constaté que ce forum possède plusieurs discussions. Ainsi, la multiplicité est d'ordre (1,*). Ensuite, pour une discussion particulière, nous avons constaté qu'elle appartient à un seul forum. Ainsi, la deuxième multiplicité est d'ordre (1,1).

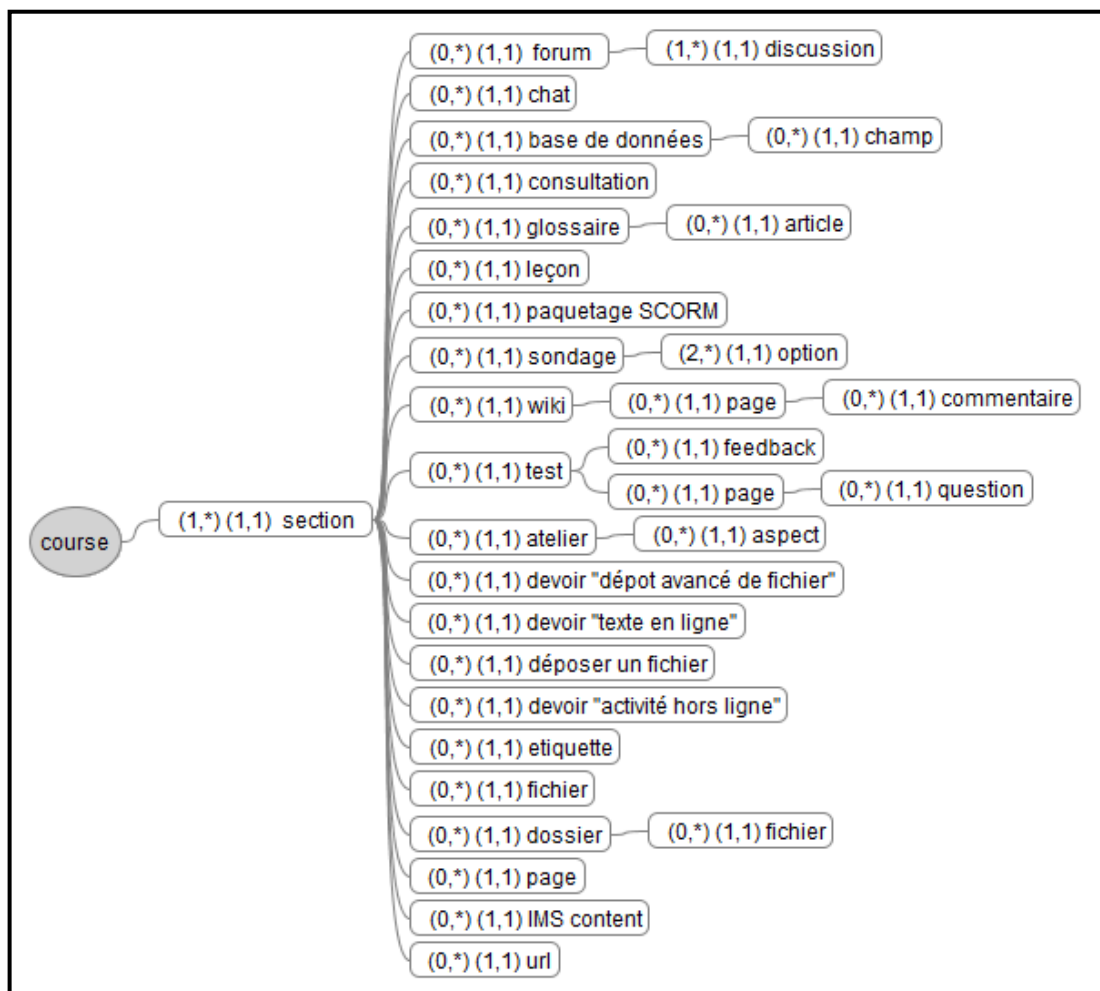


Figure 52 : Vue simplifiée du modèle IHM

Le modèle IHM est complètement défini lorsque toutes les associations, ainsi que les multiplicités, entre les modèles IHM-micro sont définies. La figure 52 illustre une vue simplifiée de ce modèle dont chaque élément/nœud désigne le modèle IHM-micro équivalent. Nous avons utilisé un outil gratuit (*FreeMind*) pour spécifier les modèles en format mindmap. Cependant, cet outil ne permet pas de re-

présenter les multiplicités. Pour cela, nous avons ainsi choisi d'annoter les nœuds du mindmap avec des multiplicités.

5.2.3. Application de l'analyse technique sur Moodle

L'analyse IHM a permis l'identification du langage de conception pédagogique de Moodle d'un point de vue centré IHM. La deuxième partie d'application du processus concerne l'analyse technique de la plateforme Moodle, plus spécifiquement l'analyse de sa base de données. Nous nous intéressons à l'identification de ce langage mais d'un point de vue technique permettant de valider la persistance des données spécifiques à ce langage. Dans l'annexe A, nous présentons un ensemble cohérent d'exemples pertinents dans le but d'illustrer toutes les spécificités et les particularités de cette analyse.

Cette analyse consiste à spécifier le modèle conceptuel de données restreint aux tables/colonnes en relation avec la conception pédagogique. Dans un premier temps, nous avons parcouru toutes les tables de la base de données. L'analyse sémantique des titres (des tables et des champs) a permis (1) d'identifier un premier dénombrement des tables liées à la conception pédagogique d'un cours en termes d'activités et de ressources pouvant être intégrées et (2) d'écarter toutes les tables qui portent sur des spécifications techniques ou qui ne possèdent pas d'enregistrements telles que les tables de gestion des utilisateurs, de traces des apprenants, etc. Une première hypothèse de l'analyse technique est d'encadrer les tables concernées. Cependant, la liste de ces tables n'est pas toujours définitive. De nouvelles tables peuvent être identifiées via des dépendances entre les tables de la base de données (par exemple via les clés étrangères, etc.) bien que d'autres peuvent être écartées. Par exemple, certaines tables identifiées par l'analyse sémantique de leur titre sont écartées parce qu'elles interviennent pour des finalités différentes à la conception pédagogique d'un cours telles que les tables *forum_read* et *forum_track_prefs*, pour le suivi des traces des apprenants au sein d'un forum. Nous avons analysé les champs des tables concernées dans le but d'identifier des attributs 'pédagogiques' et d'écarter les attributs 'techniques'.

Dans un deuxième temps, nous avons spécifié le modèle logique de données. Nous avons exploité les clés étrangères pour définir les relations entre ces tables. La spécification de ce modèle n'était pas aisée à cause des clés étrangères non précisément définies. Des exemples de difficultés sont présentés au sein de l'annexe A. Nous avons aussi constaté que plusieurs tables (telles que *groupings_groups*, *wiki_subwikis*, *glossary_entries_categories*, etc.), composées d'un identifiant et des clés étrangères, servent uniquement à définir des relations entre certaines tables. Ces tables non liées à la conception pédagogique seront automatiquement écartées pendant la génération du modèle conceptuel de données réduit de la base de Moodle (le modèle technique) par l'application des règles de rétro-conception. Cela justifie notre choix du format du modèle technique afin de cacher les détails et les contraintes techniques. Enfin, nous avons appliqué les règles de rétro-conception pour générer le modèle conceptuel de données réduit. Les clés étrangères ont permis la spécification des multiplicités requises. La figure 53 illustre un extrait du modèle technique final.

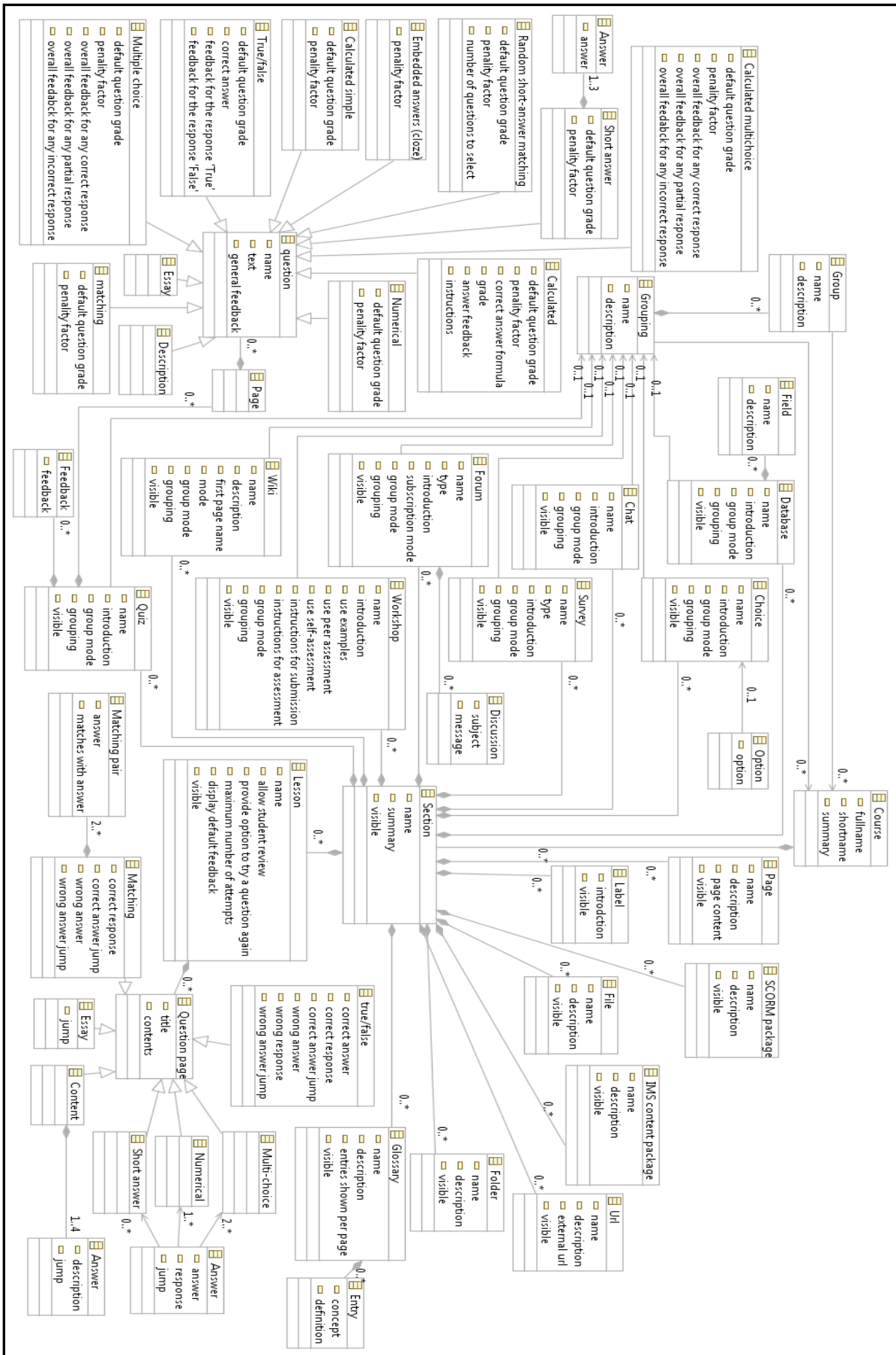


Figure 54 : Métamodèle de la plateforme Moodle réduit à la conception pédagogique

L'application de la confrontation pour tous les éléments pédagogiques de Moodle (activités, ressources, groupes, sections, etc.) est très similaire. Une présentation d'un ensemble cohérent d'exemples illustrant la confrontation de certains éléments ainsi que leurs attributs et leurs propriétés est située au sein de l'annexe A. Une fois que tous les éléments sont vérifiés et présentés, nous obtenons le métamodèle de conception pédagogique de Moodle, illustré par la figure 54.

5.3. Cas de la plateforme Ganesha

5.3.1. Introduction

La deuxième application du processus porte sur l'identification du langage de conception pédagogique de la plateforme Ganesha. Nous avons choisi cette seconde plateforme parce qu'elle dispose d'un paradigme de conception de cours différent de celui de Moodle : en termes d'activités et de ressources pédagogiques, de démarche de spécification du contenu du cours (appelé module), de profils d'utilisateurs, etc. Nous nous sommes intéressés particulièrement au profil 'intégrateur' ; le profil correspondant à un enseignant-concepteur. Dans cette section, nous présentons une vue générique de cette application. Par contre, un ensemble cohérent d'exemples illustrant les cas génériques et particuliers de cette application est donné dans l'annexe B.

5.3.2. Application de l'analyse centrée IHM sur Ganesha

Cette section est consacrée à l'application des différentes parties de l'analyse centrée IHM sur la plateforme Ganesha dans le but de spécifier le modèle de conception pédagogique de cette plateforme du point de vue IHM.

5.3.2.1. Application de l'analyse IHM-macro sur Ganesha

Pour identifier les interfaces relevant de la conception pédagogique d'un cours sur Ganesha (appelé module), nous avons appliqué dans un premier temps la première stratégie qui consiste à analyser des interfaces d'un module (un cours sur Ganesha) existant. En choisissant ce module, nous sommes amenés à une nouvelle interface qui représente son contenu. Le module est composé d'un ensemble d'éléments appelés séquences. Chaque séquence amène à un écran pour la consultation ou la spécification du contenu en termes d'activités. La spécification des activités amène également à un nouvel écran. Plusieurs activités sont proposées par Ganesha dont chacune dispose de sa (son) propre interface/écran. Nous avons suivi cette démarche pour l'identification d'autres interfaces sur cette plateforme telles que les interfaces des questions, des séries (de question), etc.

Dans un deuxième temps, nous avons suivi la deuxième stratégie d'analyse IHM-macro par la spécification d'un nouveau module. Nous avons tout d'abord paramétré les attributs du module proposés par l'interface (le nom, les objectifs, etc.). Ensuite, nous avons spécifié, indépendamment des modules, les séquences pouvant leur être associées. Dans chaque séquence, nous avons défini les activités requises parmi celles fournies par la plateforme (telles que les activités titre, les activités AICC, les activités collaboratives, les quiz, etc.). Par exemple, les quiz requièrent la spécification de série des questions avant d'y être associées.

Dans les deux stratégies, nous représentons chaque IHM identifiée au sein du modèle IHM-macro par leur concept principal. Puis, nous spécifions les relations entre les concepts en se basant sur les relations entre les interfaces. La figure 55 illustre le modèle IHM-macro de Ganesha.

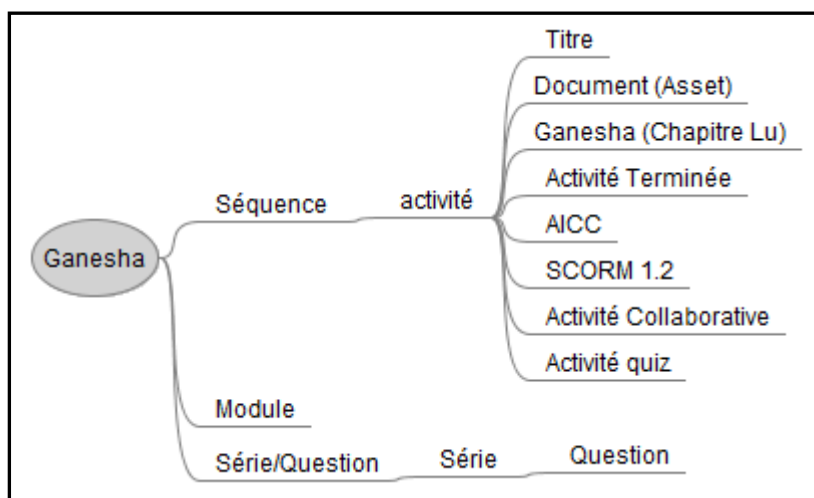


Figure 55 : Modèle IHM-macro de la plateforme Ganesha

Au cours de cette analyse, nous avons constaté que certaines parties (écrans) des interfaces ne sont pas liées à la conception pédagogique telles que les parties des statistiques, etc. Au cours de l'analyse IHM-macro, nous tenons en compte l'interface dans sa globalité. Ensuite, les analyses suivantes (fonctionnelle et IHM-micro) s'intéressent qu'aux zones relevant de la conception pédagogique.

5.3.2.2. Application de l'analyse fonctionnelle sur Ganesha

L'application de l'analyse fonctionnelle sur les IHMs identifiées consiste à identifier les fonctionnalités relevant de la conception pédagogique. Une description détaillée de cette analyse par un ensemble cohérent d'exemples est présentée dans l'annexe B.

En prenant une interface spécifique (celle des séquences par exemple (les figures B.5 et B.7 dans l'annexe B)), nous l'avons tout d'abord décomposé en trois zones. Ensuite, nous avons analysé les composants graphiques (widgets) de ces zones. Les widgets décrivant des fonctionnalités liées à la conception pédagogique sont pris en compte et les éléments techniques sont écartés.

Nous avons constaté que les interfaces de Ganesha disposent de zones communes qui sont analysées une seule fois (les zones 1 et 2). Egalement, plusieurs interfaces disposent des fonctionnalités pédagogiques liées à un élément particulier (le cas de séquences par exemple). Ces fonctionnalités sont représentées au sein du même modèle. La figure 56 représente un extrait du modèle fonctionnel final.

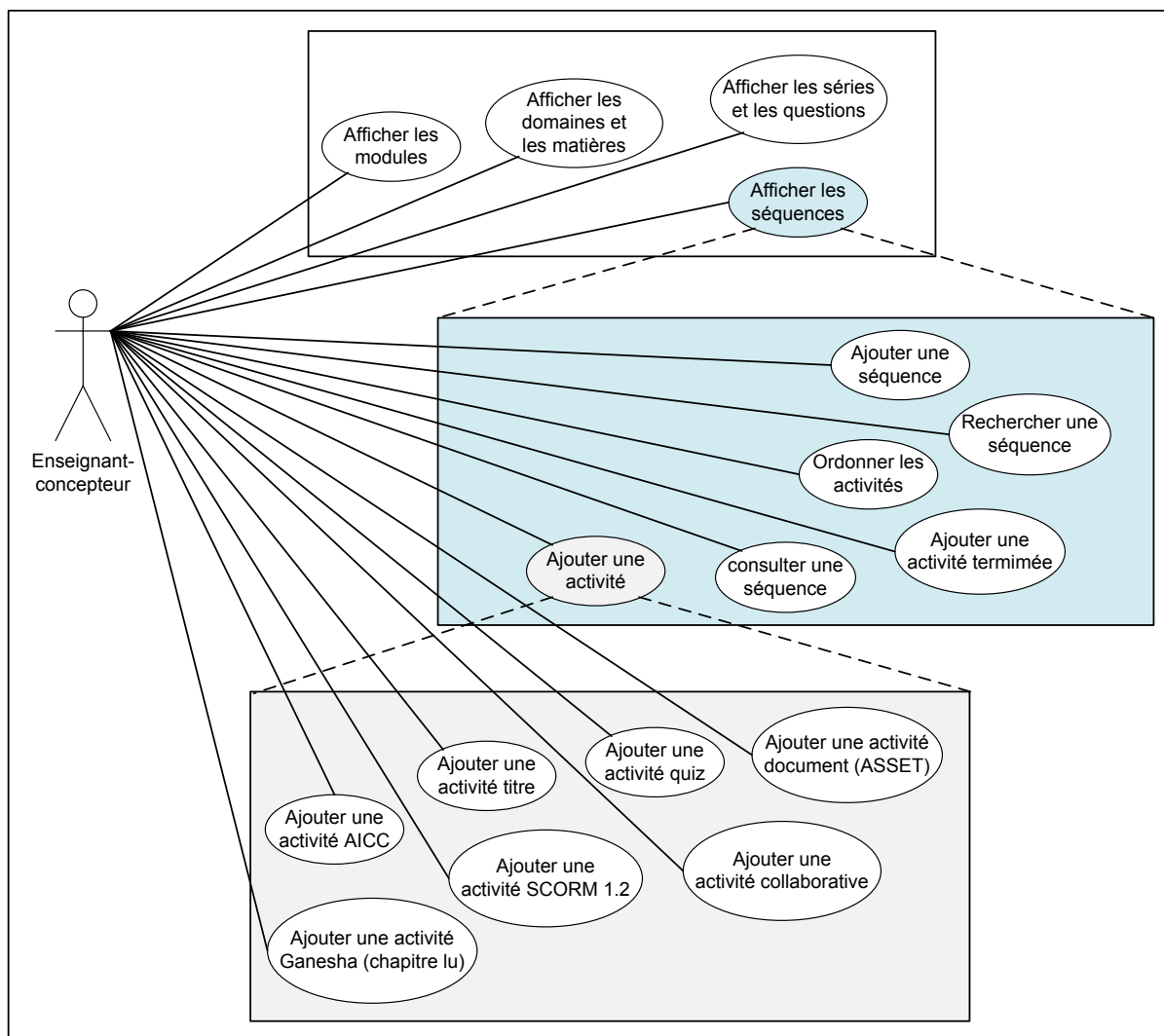


Figure 56 : Extrait du modèle fonctionnel de Ganesha

5.3.2.3. Application de l'analyse IHM-micro sur Ganesha

L'application de l'analyse IHM-micro sur la plateforme Ganesha consiste à identifier les propriétés, les caractéristiques et les attributs des éléments de conception pédagogique. Un ensemble cohérent d'exemples de cette application exposant ses généralités et ses particularités est présenté dans l'annexe B. Pour chaque élément de conception pédagogique (tels que les concepts du modèle IHM-macro), nous avons décomposé l'IHM associé en plusieurs zones. Souvent, nous avons repris la décomposition effectuée au sein de l'analyse fonctionnelle. Ensuite, nous avons analysé les zones proposant des formulaires de description des concepts pédagogiques. Nous avons identifié les attributs pédagogiques ainsi que leurs propriétés en termes d'élément requis ou facultatif, le type, les champs de valeur et la valeur par défaut. Nous les avons représentés au format mindmap. Toutefois, nous avons écarté plusieurs attributs que nous les avons jugés techniques. La figure 57 illustre un exemple du modèle IHM-micro dédié à l'élément quiz.

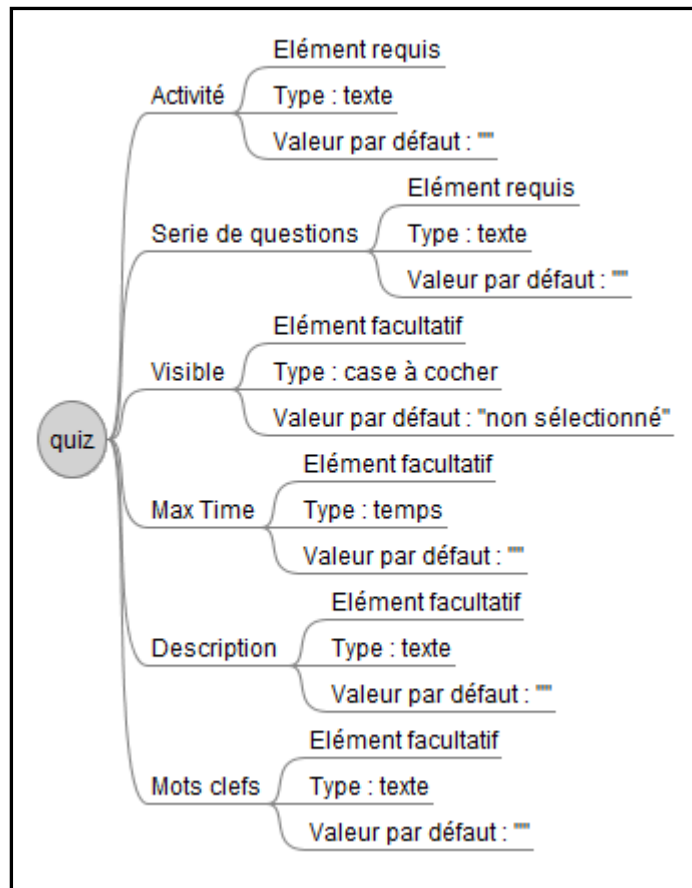


Figure 57 : Modèle IHM-micro d'un quiz

5.3.2.4. Composition et formalisation du modèle IHM de Ganesha

La composition consiste à associer les modèles IHM-micro, sur la base des relations au sein des modèles IHM-macro et fonctionnel, afin de formaliser le modèle IHM du langage de conception pédagogique de la plateforme Ganesha. L'annexe B expose d'un ensemble de cas génériques et aussi particuliers de la composition des modèles IHM-micro de Ganesha. En prenant un modèle IHM-micro spécifique (d'un premier concept), nous allons chercher les modèles qui peuvent lui être associés. Pour cela, nous consultons le modèle IHM-macro pour déterminer un concept associé au premier. Ensuite, nous vérifions sur le modèle fonctionnel, si les fonctionnalités de ces deux éléments sont associées ou bien il existe des fonctionnalités intermédiaires spécifiques à d'autres éléments. Sur la base de ces vérifications, les associations sont spécifiées. Enfin, nous nous intéressons à la définition des multiplicités. Pour cela, nous avons expérimenté la possibilité d'associer plusieurs instances du premier élément à une instance du deuxième élément et inversement.

Le modèle IHM est complètement défini lorsque toutes les associations et les multiplicités sont également définies. La figure 58 illustre une vue simplifiée de ce modèle dont chaque élément désigne le modèle IHM-micro équivalent. Cependant, nous avons annoté manuellement les nœuds du modèle avec des multiplicités parce que l'outil (*FreeMind*) ne propose pas d'annotations.

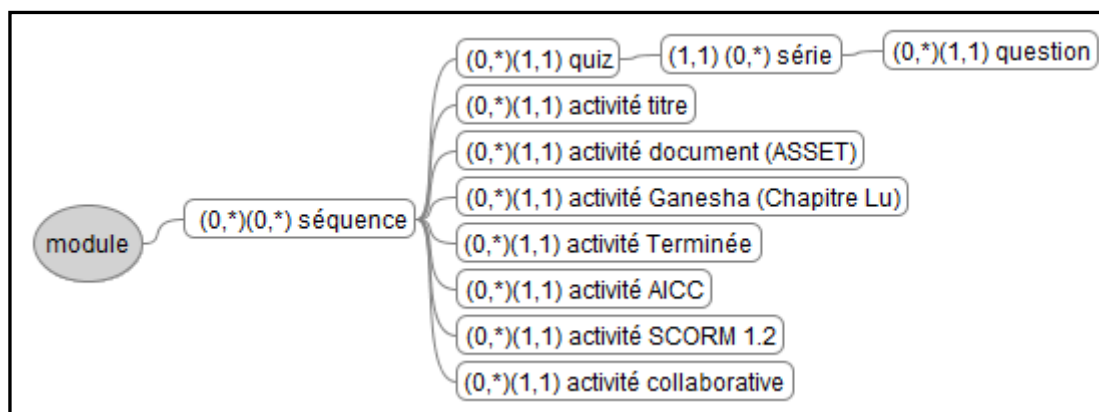


Figure 58 : Modèle IHM de Ganesha

Nous constatons ainsi que le modèle IHM est différent du modèle IHM-macro. Par exemple, il n'existe ni le modèle IHM-micro de l'élément activité ni l'association (séquence – activité). Les modèles IHM-micro des activités proposées par Ganesha (quiz, activité titre, etc.) sont directement associés au modèle IHM-micro de *séquence* contrairement au modèle IHM-macro où elles sont associées à l'élément générique '*activité*'.

5.3.3. Application de l'analyse technique sur Ganesha

L'application de l'analyse technique sur Ganesha consiste à identifier le langage de conception pédagogique de cette plateforme de point de vue technique. L'annexe B expose un exemple cohérent d'analyse technique en détaillant ses généralités et également ses particularités.

En premier lieu, nous avons tout d'abord parcouru toutes les tables de la base de données en analysant sémantiquement leurs titres. Nous avons écarté toutes les tables qui portent sur des spécifications techniques ou qui ne possèdent pas d'enregistrements. Ces tables sont créées pour des finalités différentes de la conception pédagogique (gestion des utilisateurs, traces des apprenants, etc.) telles que les tables '*stat_connexions*', '*prime_list_societe*', etc. Nous avons identifié certaines tables liées à la conception d'un élément pédagogique telles que '*list_sequences*', '*serie_question*', '*module*', etc.

En second lieu, nous avons analysé les enregistrements de ces tables qui représentent les détails des éléments spécifiés à partir de l'interface de la plateforme. Nous avons pris en compte ceux portant sur la définition des éléments ainsi que des clés étrangères référençant des tables spécifiques. Les éléments jugés plutôt techniques que pédagogiques (ont des valeurs par défaut ou sont des détails techniques à bas niveau) sont écartés. Nous avons représenté chaque table identifiée par un modèle décrivant les attributs retenus. Après avoir formalisé tous les modèles, nous avons généré le modèle logique de données. Les modèles sont reliés sur la base des clés étrangères. Ensuite, nous avons appliqué les règles de retro-conception pour spécifier le modèle conceptuel de données réduit, plus spécifiquement le modèle de conception pédagogique de Ganesha du point de vue technique. La figure suivante illustre ce modèle.

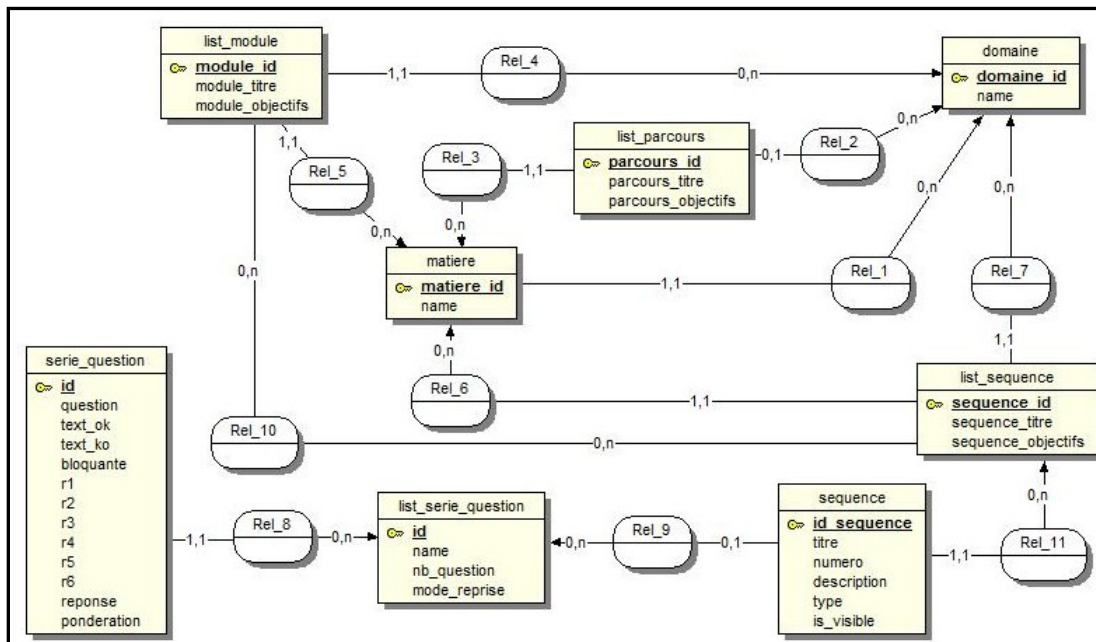


Figure 59 : Modèle conceptuel de donnée réduit de la plateforme Ganesha

5.3.4. Confrontation et formalisation

Nous avons appliqué le processus de confrontation et de formalisation du modèle final sur les deux modèles de conception pédagogique de la plateforme Ganesha, en particulier les modèles IHM et technique. Nous rappelons que l'annexe B expose un ensemble cohérent d'exemples de confrontation. Nous confrontons les éléments du modèle IHM avec ceux du modèle technique en termes d'existence et de définition. En choisissant un élément particulier du modèle IHM, plus spécifiquement son modèle IHM-micro, nous cherchons son équivalence sur le modèle technique. Souvent, les nominations des éléments sur les deux modèles sont similaires mais pas identiques. Pour cela, nous avons privilégié la nomination la plus significative et standard pour représenter le modèle final, souvent celle du modèle technique. Nous avons conservé le vocabulaire du modèle IHM pour la définition de la couche graphique des outils de conception. Nous avons appliqué les processus de vérification pour les éléments différemment définis sur les modèles. Puis, nous représentons les éléments confrontés et vérifiés par des méta-classes. Ensuite, nous nous intéressons à la vérification et à la représentation de leurs attributs par des méta-classes. Nous avons également mené des vérifications et des analyses spécifiques pour les éléments différemment définis. Enfin, nous nous intéressons à la définition des relations entre les méta-classes sur la base de relations dans les modèles IHM et technique. Lorsque toutes les méta-classes sont spécifiées, nous obtenons le modèle de conception pédagogique de la plateforme Ganesha. La figure 60 illustre ce modèle.

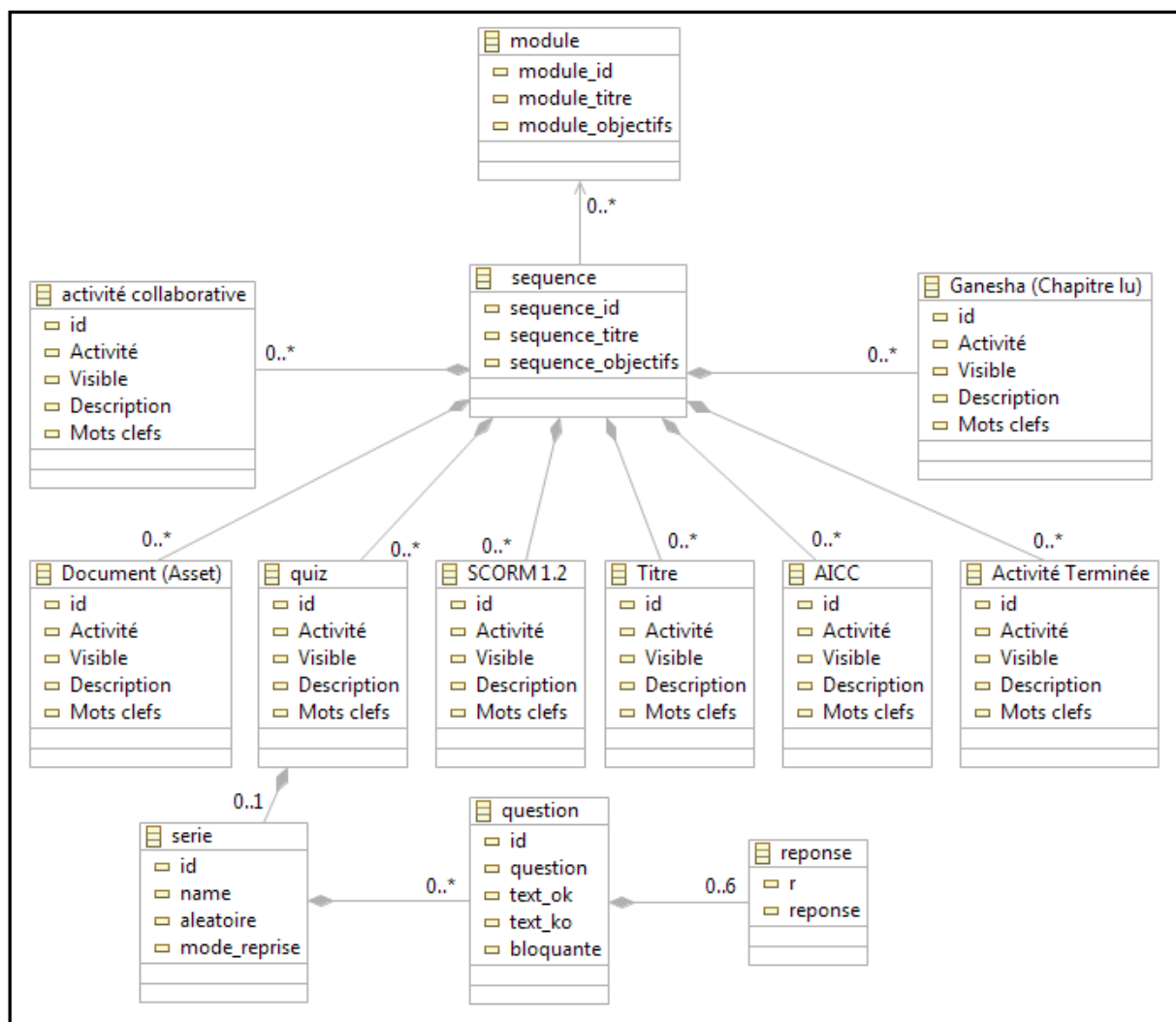


Figure 60 : Métamodèle de conception pédagogique de la plateforme Ganesha

5.4. Conclusion

Nous avons présenté dans ce chapitre une vue générale de l'application du processus d'identification et de formalisation du langage de conception pédagogique sur les plateformes Moodle et Ganesha. Toutefois, nous présentons dans les annexes A et B une description détaillée de ces applications.

L'expérimentation du processus a permis la mise à l'essai de nos propositions théoriques. Nous avons constaté que le processus définit les étapes primordiales d'identification de ce langage. Toutefois, il est flexible permettant la prise de décisions subjectives (par exemple, la nomination des éléments du modèle final). Le processus peut être appliqué pour plusieurs finalités telles que la comparaison des plateformes via la comparaison des métamodèles résultant. Ces métamodèles peuvent également servir pour d'autres champs d'application. Dans notre travail, ils servent comme base pour (1) le développement des outils de conception externe en suivant une approche de modélisation spécifique au domaine (*Domain Specific Modeling*) et (2) pour la spécification des modules de communication entre ces plateformes et les nouveaux outils de conception associés. Ces travaux de développement font l'objet du chapitre suivant.

6

Développement des outils et mise à l'essai

Sommaire

- 6.1. Introduction
- 6.2. Développement du service d'opérationnalisation
- 6.3. Développement des outils de conception
- 6.4. Mise à l'essai
- 6.5. Conclusion

Nous rappelons tout d'abord que l'objectif de notre travail est de simplifier l'opérationnalisation des scénarios pédagogiques sur les plateformes de formation. Pour cela, nous avons proposé d'identifier et de formaliser ce langage. Ensuite, nous l'utilisons comme base pour la spécification des éditeurs graphiques externes et pour le développement des modules d'opérationnalisation. Ces modules se chargent de l'import des scénarios spécifiés en dehors de plateformes, en particulier ceux spécifiés sur les éditeurs, et également de l'export des scénarios existants.

Dans ce chapitre, nous allons présenter les outils développés. La première partie de ce chapitre est spécifique au développement des modules d'opérationnalisation pour les plateformes de formation Moodle et Ganesha. Le développement des outils de conception fait l'objet de la deuxième partie de ce chapitre. Nous appliquons l'approche DSM outillée en utilisant deux frameworks particuliers du projet de modélisation d'Eclipse.

La troisième partie du chapitre concerne la mise à l'essai des outils développés. Nous allons décrire textuellement un scénario pédagogique. Nous allons utiliser les éditeurs pour le design de ce scénario. Ensuite, nous utiliserons les modules ajoutées aux plateformes pour son opérationnalisation. Enfin, nous montrons le résultat de cette opération en présentant les tâches de design à compléter sur les plateformes.

6.1. Développement du service d'opérationnalisation

6.1.1. Introduction

Après l'étude et l'analyse de certaines plateformes de formation à distance, nous avons constaté que plusieurs d'entre elles ne proposent pas des fonctionnalités d'import/export de situations d'apprentissages. Lorsqu'elles existent, le format d'échange est généralement complexe et composé de plusieurs informations réparties sur plusieurs fichiers tel que le cas du package utilisé par les fonctions *backup* et *restore* de la plateforme Moodle. Nous détaillerons ci-après ce package. Notre approche propose le développement des services d'opérationnalisation dédiés aux plateformes afin de les rendre capables d'importer et d'exporter des situations d'apprentissage (cours, scénarios pédagogiques, évaluations, etc.). Ces dernières doivent être conformes au langage de conception pédagogique des plateformes cibles. Dans cette section, nous présentons les services d'opérationnalisation développés pour les plateformes Moodle et Ganesha de façons différentes. Toutefois, nous avons pris en compte de préserver la sémantique des plateformes à l'opposé des travaux de [Caron 2007b] et [Abdallah 2009] qui proposent d'intégrer des nouveaux services web ou des modules de transformation.

6.1.2. Cas de la plateforme Moodle

6.1.2.1. Présentation de la fonctionnalité *backup/restore* du Moodle

La plateforme Moodle dispose de certaines fonctionnalités pour sauvegarder, restaurer et importer des cours. La fonctionnalité *backup* de Moodle (figure 61) permet aux enseignants-concepteurs de sauvegarder leurs cours dans un archive Zip spécifique (figure 62) qui contient plusieurs fichiers xml recueillant toutes les informations relatives au cours : pages web, forum et leurs messages, devoirs, etc. Ces fichiers représentent les données extraites de la base de données. Nous y retrouvons la description des éléments composant le cours. Nous y retrouvons également des détails techniques de bas niveau tels que les informations des blocs situés dans l'espace du cours, les informations dédiées aux éléments de définition d'un cours (*timecreated*, *legacyfiles*, *maxbytes*, *tags*, *requested*, etc.), les informations des utilisateurs et des rôles (*shortname*, *nameincourse*, *description*, *sortorder*, *archetype*, etc.)

La fonctionnalité *restore* de Moodle permet de restaurer un cours en cas de sinistre ou de le placer sur un autre site Moodle via son service de restauration. Plusieurs modes de restauration sont possibles tels que restaurer le cours actuel en supprimant le précédent, restaurer le cours actuel en y ajoutant des données, créer un nouveau cours, etc. Toutefois, les fonctionnalités *backup* et *restore* ne sont pas toujours accessibles aux enseignants-concepteurs ; leur apparition dans l'espace de cours dépend de la configuration de la plateforme.

La modification d'un concept ou d'un fichier enregistré dans ce backup n'est pas une tâche aisée. D'une part, elle nécessite la modification de tous les fichiers XML décrivant ce concept. D'autre part, la fonctionnalité *restore* requiert le même format du paquetage de sauvegarde. Il n'est pas facile de produire de tels paquetages par des outils externes. Toutes ces difficultés ont représenté les raisons majeures pour proposer un nouveau module, intégré à l'espace cours sur Moodle et proposant l'opérationnalisation et l'export des situations d'apprentissage dans un format simplifié.



Figure 61 : Les fonctionnalités de sauvegarde et de restauration de Moodle

activities			Folder
course			Folder
files			Folder
sections			Folder
completion.xml	79	63	XML Document
files.xml	1 280	430	XML Document
gradebook.xml	1 737	542	XML Document
groups.xml	86	64	XML Document
moodle_backup.log	0	0	Text Document
moodle_backup.xml	5 363	742	XML Document
outcomes.xml	83	65	XML Document
questions.xml	83	65	XML Document
roles.xml	693	289	XML Document
scales.xml	79	63	XML Document
users.xml	3 145	844	XML Document

Figure 62 : Exemple d'archive Zip de la sauvegarde d'un cours

6.1.2.2. Description du module d'opérationnalisation pour Moodle

Afin que des enseignants-concepteurs puissent importer et exporter des scénarios pédagogiques avec la plateforme Moodle, il est nécessaire de développer un module de communication et de l'intégrer dans l'espace de conception des cours de la plateforme [Abedmouleh et al. 2011b] [Abedmouleh et al. 2011c] [Abedmouleh 2010]. La figure 63 illustre ce module qui doit jouer le rôle de passerelle de communication entre les outils externes et la plateforme et garantir l'opérationnalisation des scénarios pédagogiques.

Le nouveau module est développé sous le format d'un *bloc*. Nous avons repris le langage de programmation utilisé pour le développement de la plateforme Moodle. Il s'agit du langage de programmation web le PHP. Nous avons également utilisé le langage XSLT (*eXtensible Stylesheet Language*

Transformations) pour la définition de certaines instructions du processus d'import et d'export. Nous présenterons ci-après les procédures développées et basées sur ce langage.

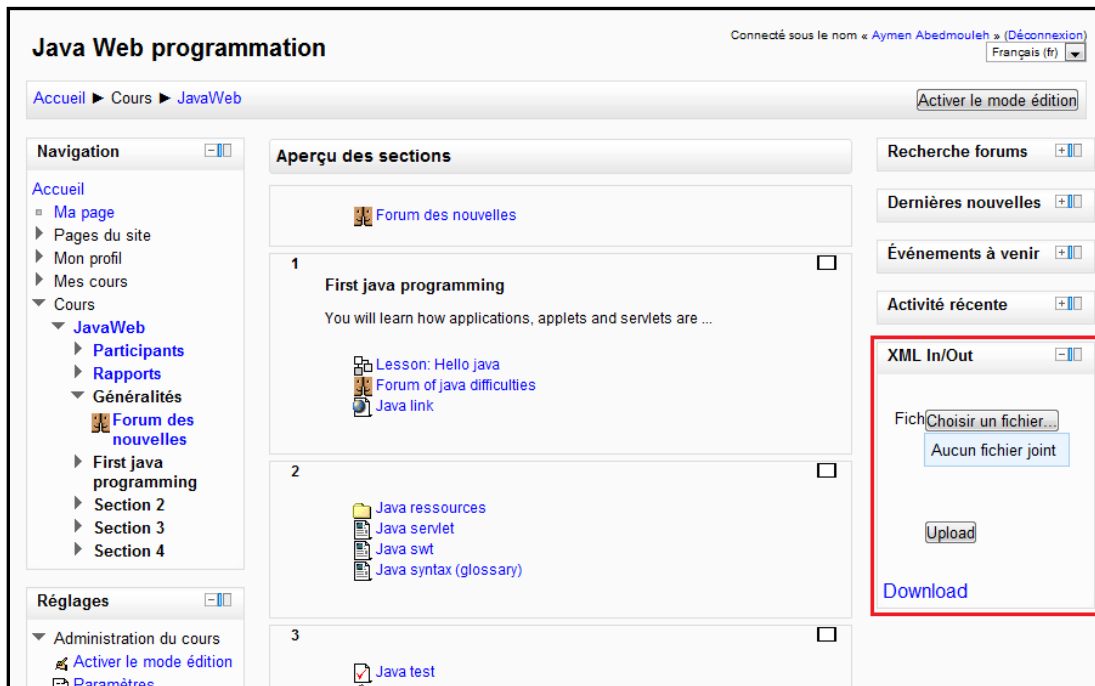


Figure 63 : Service d'opérationnalisation et d'export intégré à l'espace cours de Moodle

Pour définir le block d'import/export des cours dans Moodle, il est nécessaire de définir trois fichiers PHP principaux. D'autres fichiers peuvent être définis puis utilisés par les fichiers principaux. Tous les fichiers doivent être mis sous un répertoire bien spécifique *blocks/importandexportofcourse*. Les trois fichiers principaux sont les suivants :

block_importandexportofcourse.php : ce fichier présente la définition de classe pour le bloc. Il est utilisé pour gérer le bloc comme un plugin et pour le rendre à l'écran. Nous avons tout d'abord créé le fichier principal *importandexportofcourse.php*. Ensuite, nous avons défini les instructions nécessaires. Le développement impose plusieurs contraintes.

```
<?php
require_once('importandexportofcourse_up_form.php');

class block_importandexportofcourse extends block_base {

    function init() {
        $this->title = get_string('pluginname',
            'block_importandexportofcourse');
    }

    function get_content() {
        global $CFG;
        if ($this->content !== NULL) {
            return $this->content;
        }

        $this->content = new stdClass;
        $this->content->footer = '';
        $this->content->text = '<div class="xmlupform">';
```

Figure 64 : Extrait du fichier importandexportofcourse.php

Par exemple, la première ligne d'instruction, représentant la définition de la classe du bloc, doit être définie exactement de la manière indiquée sur la figure 64, à l'exception du nom du bloc. Chaque classe doit débiter par la méthode *init()* qui permet d'instancier les valeurs des variables de la classe. La figure 64 montre un extrait du fichier *block_importandexportofcourse.php* dont la fonction *init()* est consacrée à la définition du titre du bloc.

lang/en/block_importandexportofcourse.php : ce fichier est spécifique à la langue anglaise du *bloc*. Il est également possible de créer les *blocs* sous différentes langues. Ce fichier définit le nom du *bloc* qui sera affiché sur l'espace cours de la plateforme ainsi que tous les noms qui seront utilisés dans le bloc. La figure 65 illustre les instructions de ce fichier.

```
<?php
$string['upload'] = 'Upload';
$string['uploadfilename'] = 'File to upload :';
$string['download'] = 'Download';
$string['newxmlinoutblock'] = '(new XMLInOut block)';
$string['pluginname'] = 'XML In/Out';
?>
```

Figure 65 : Les instructions du fichier *lang/en/block_importandexportofcourse.php*

version.php : ce fichier est générique pour tous les blocs de la plateforme. Il possède une seule instruction consistant à définir la version du bloc développé. La figure 66 illustre un exemple de ce fichier.

```
<?php
    $plugin->version = 2011121010;
```

Figure 66 : Extrait du fichier *version.php*

Toutefois, la conception et le développement de ce module doit prendre en compte la contrainte liée à la modification et à l'adaptation des cours (existants sur la plateforme) à l'aide des outils externes. Ce choix apporte des éléments de réponses à une de nos problématiques : assurer une conception continue des situations pédagogiques. Ceci demande des utilisations successives de l'import/export. Après avoir exporté son cours de la plateforme, l'enseignant-concepteur peut effectuer des modifications nécessaires sur l'outil de conception avant de l'opérationnaliser à nouveau par le module développé. Lors du développement, nous avons choisi d'utiliser deux fonctionnalités existantes (*sauvegarde* et *restauration*) tout en ajoutant les instructions nécessaires pour leurs utilisations et en les adaptant à nos besoins.

Le module intègre alors deux processus : un processus d'export consistant à générer un document (XML) spécifiant toutes les informations liées à la conception pédagogique du cours et un processus d'import pour "opérationnaliser", dans l'espace cours, le scénario spécifié dans un fichier XML. Les fichiers des scénarios importés ou exportés ont l'obligation d'être conformes au schéma XML du formalisme métier de la plateforme Moodle.

Nous avons ainsi spécifié un schéma XML explicitant formellement ce langage. La figure 67 montre un extrait du schéma XML. Ce schéma est dérivé du métamodèle de langage conception pédagogique de Moodle (résultant de l'application de ce processus sur la plateforme Moodle (chapitre 4 et

5)). Il représente un format de communication avec des outils extérieurs à la plateforme. Il pourra aussi servir pour valider les documents XML qui seront importés/exportés. Ce schéma a été spécifié en utilisant le design pattern *russian dolls*⁸ parmi plusieurs patrons de conception de schéma XML tels que *Salami Slice Design*, *Venetian Blind Design*, *Garden of Eden Design*, etc. [Xfront 2012]. Aucun élément du document XML n'était partageables, il n'y a donc pas de nécessité d'avoir un mécanisme de réutilisation des éléments ; ce patron est très proche de la structuration du document et donc favorise sa compréhension.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  ...
  <xsd:element name="course">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="shortname" type="xsd:string"/>
        <xsd:element name="fullname" type="xsd:string"/>
        <xsd:element name="summary" type="xsd:string"/>
        <xsd:element name="sections">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="section" maxOccurs="unbounded">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="name" type="xsd:string"/>
                    <xsd:element name="summary" type="xsd:string"/>
                    <xsd:element name="activities">
                      <xsd:complexType>
                        <xsd:choice minOccurs="0" maxOccurs="unbounded">
                          <xsd:element name="resource">
                            ...

```

Figure 67 : Extrait du schéma XML équivalent au métamodèle du langage de conception pédagogique de Moodle

Afin d'assurer le bon fonctionnement de ces deux processus en termes de première création (premier import dans un cours vide) et d'adaptation (par ajouts et suppressions d'éléments), nous les avons testés dans un premier temps sur la base de deux concepts (forum et label). Puis, dans un deuxième temps, nous avons étendu le développement sur l'ensemble du métier explicité.

6.1.2.2.1. Description du processus d'export

Le premier processus (figure 68) se charge donc de la sauvegarde du cours et de sa formalisation conformément au métier que nous avons formalisé. Concrètement, il s'appuie sur la fonction de sauvegarde (*backup*) de Moodle pour générer le package du cours. Les données des fichiers du paquetage étant au format XML. Nous avons élaboré une transformation XSLT permettant de parcourir les différents fichiers XML décompressés du backup et de générer un fichier XML unique conformément au schéma XML (figure 67). La procédure XSLT est composée de deux parties. La première partie consiste à générer les règles pour la lecture des fichiers du backup. La deuxième partie consiste à générer les règles de la lecture du schéma XML. Pour chaque élément du schéma, une règle lui est associée.

⁸ Russian dolls est un patron de conception de schémas XML. Il représente le schéma qui reflète la structure des documents.

Le code spécifique à la procédure XSLT est reproduit dans l'annexe C. L'algorithme général de ce processus est le suivant :

- 1. Choisir le cours à exporter
- 2. Appeler la fonction backup de Moodle pour générer le paquetage du cours
- 3. Décompresser le paquetage du cours
- 4. Créer un fichier XML vide
- 5. Pour chaque élément du schéma XML de la plateforme,
 - 5.1. Lire les fichiers XML et identifier les informations dédiées à cet élément
 - 5.2. Ecrire ces informations au sein du fichier XML

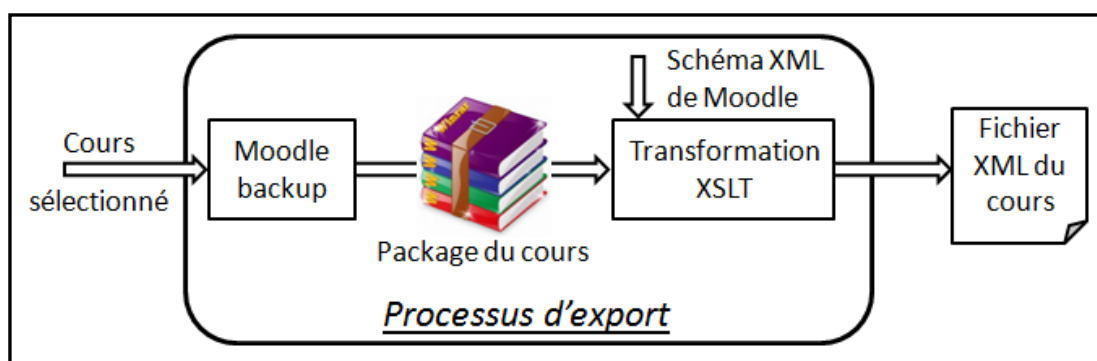


Figure 68 : Processus d'export de cours sur Moodle

Le fichier XML résultant du processus est directement manipulable par des outils de conception spécifiques. Ces outils font l'objet de la deuxième partie de ce chapitre. Ils sont développés sur la base du schéma XML/métamodèle du langage de conception pédagogique de la plateforme Moodle. Ils sont capables de charger les cours spécifiés dans un format XML conforme à ce schéma XML.

6.1.2.2.2. Description du processus d'import ou d'opérationnalisation

Le deuxième processus (figure 69) se charge de l'opérationnalisation (l'import) de situations pédagogiques spécifiées à l'externe des plateformes conformément au schéma XML. Nous avons choisi de nous appuyer sur la fonction *restore* existante dans Moodle pour concrètement opérationnaliser les situations pédagogiques. Pour cela, il faut réaliser deux opérations successives et complémentaires. La première consiste à compléter et transformer le scénario à importer dans le format requis par la fonctionnalité de *restore*. Ce format est un package compressé (format zip) dont la structure (figure 62) est conforme à celle du package de sauvegarde (nombreux dossiers et fichiers XML). La deuxième opération s'appuie sur la fonction *restore* pour établir l'opérationnalisation. Le processus est basé sur des transformations exprimées en langage XSLT. Ces règles de transformation modifient les aspects syntaxiques du scénario/cours à importer sans altérer leur sémantique.

A défaut de détailler techniquement ces règles, l'idée générale de l'algorithme de transformation que nous avons spécifié consiste à :

- 1/ réaliser le backup du cours concerné par l'import (via la fonction *restore*),
- 2/ compléter et modifier ce paquetage par les informations spécifiées dans le fichier XML du nouveau scénario (via la procédure XSLT développée),
- 3/ opérationnaliser le package modifié (via la fonction *restore*).

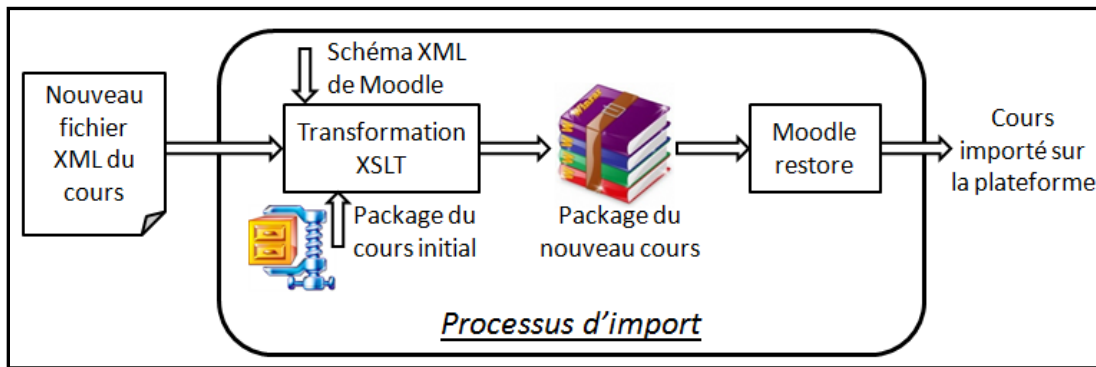


Figure 69 : Processus d'import de cours sur MOODLE

La procédure XSLT est basée sur le fichier XML du scénario à opérationnaliser ainsi que le schéma XML. Après avoir déterminé l'identifiant du cours (enregistré dans le fichier XML), la procédure réalise une sauvegarde du cours via la méthode *backup* de Moodle. Au sein de cette procédure sont définies les règles responsables de mettre à jour les données du packaging du cours selon les informations spécifiées dans le fichier XML. La procédure possède également des règles permettant la création des nouveaux fichiers dans le packaging lorsque des nouveaux éléments sont ajoutés dans le cours à opérationnaliser.

Le processus d'import prend en considération l'adaptation d'un cours existant et compatible avec un premier import. Concrètement, la transformation XSLT prend en compte quatre cas de figure :

- (1) la modification des informations d'un concept déjà existant dans le cours (ie. le concept apparaît dans le backup et dans le scénario à importer) ;
- (2) la création d'un élément de scénario non déjà présent dans le cours (ie. le concept n'apparaît pas dans le backup mais dans le scénario à importer) : ce point a nécessité de choisir des valeurs par défaut pour les identifiants des éléments ;
- (3) la suppression d'un élément du cours (ie. le concept apparaît dans le backup mais pas dans le scénario à importer) ;
- (4) l'omission d'un élément (ie. le concept apparaît dans le backup mais pas dans le scénario à importer mais l'outil externe ne gère pas cet élément métier) : ce point très complexe a nécessité l'introduction de l'information du « périmètre métier » abordé par l'outil externe à l'origine de la création du scénario. Nous avons pour cela choisi d'imposer aux outils externes de fournir le schéma XML formalisant le sous-ensemble du langage métier qu'ils manipulent, ce schéma ayant pour contrainte d'être un sous-ensemble de celui de la plateforme.

6.1.2.2.3. Expérimentation et évolution du service d'opérationnalisation

Nous avons expérimenté et testé le module d'opérationnalisation sur de nombreux cas afin de vérifier le bon fonctionnement des quatre possibilités de manipulations sur les scénarios/cours pour Moodle.

A titre d'illustration, voici quelques exemples. Dans le cas où l'enseignant a modifié une activité particulière du scénario telle que « leçon », le processus copie les parties XML du scénario initial en mettant à jour les données de cette activité. Par contre, si cette activité est supprimée, le processus ne copie pas les données de cette activité dans le nouveau package à opérationnaliser. Lorsqu'une activité est nou-

vement ajoutée, le programme est capable de générer les parties XML de cette activité et d'initialiser les données associées.

Pendant le développement du module d'opérationnalisation, nous avons rencontré plusieurs difficultés. A titre d'exemple, il était indispensable de distinguer les éléments nouvellement ajoutés aux cours de ceux qui existent déjà. Une des solutions que nous avons adoptée est d'initialiser l'identifiant des nouveaux éléments à zéro. Aussi, nous avons proposé une solution technique afin de palier au problème d'utilisation d'outils externes qui ne seraient capables de traiter que partiellement le langage métier que nous proposons pour la plateforme.

Les processus d'import et d'export ont l'avantage de palier au manque de données exclues pendant l'export initial du scénario vers les outils de conception externes puisque ces informations sont ajoutées au scénario avant sa restauration. Ainsi, le scénario à opérationnaliser est toujours complété en incluant tous les détails définis par défaut ou ajustés par le concepteur directement sur la plateforme.

6.1.3. Cas de la plateforme Ganesha

6.1.3.1. Description du module d'opérationnalisation

Nous avons choisi de mener une deuxième expérimentation sur la plateforme de formation Ganesha. Contrairement à la plateforme Moodle, Ganesha ne dispose pas de fonctionnalités de sauvegarde et de restauration de cours. Nous avons développé entièrement le service d'opérationnalisation (figure 70), intégré au sein de l'espace de cours, afin de la rendre capable de communiquer avec les outils de conception externes spécifiques. L'architecture de Ganesha est modulaire. L'espace administrateur de Ganesha propose l'ajout des extensions à cette plateforme via la configuration de certains paramètres (tels que l'adresse du serveur mail SMTP, le numéro du port SMTP, authentification, identifiant, mot de passe, etc.). Nous avons constaté que ces extensions sont difficiles à définir et ne conviennent pas à notre objectif pour la spécification d'un module d'import et d'export de cours.

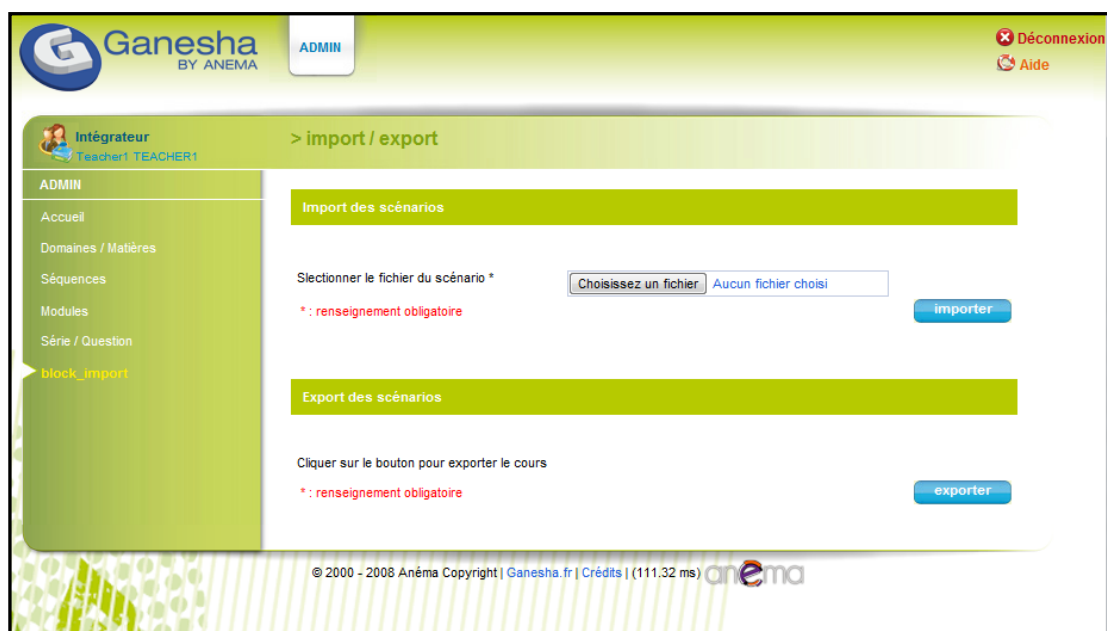


Figure 70 : Bloc import/export ajouté à Ganesha

Nous avons choisi d'intégrer ce module au sein de l'espace associé au profil d'un intégrateur parmi les profils proposés par la plateforme (admin, formateur, intégrateur, conseiller et gestionnaire). Ce profil est le plus proche du profil enseignant-concepteur de cours : l'utilisateur de nos propositions. Nous avons ainsi respecté deux contraintes principales concernant les droits associés à ce profil ainsi que la charte graphique de l'interface. Le langage de développement du module est le même langage de la plateforme Ganesha : le langage de programmation web PHP.

Le module d'opérationnalisation joue le rôle de passerelle de communication entre les outils externes et la plateforme. Il garantit l'opérationnalisation des informations des scénarios pédagogiques. Nous proposons un processus d'opérationnalisation des situations d'apprentissage (appelées module sur Ganesha). Nous proposons aussi un processus d'export afin de prendre en compte la possibilité de mener des modifications et des adaptations de ces situations à l'extérieur de la plateforme, en particulier avec les outils de conception développés.

Le module repose sur le schéma XML que nous avons spécifié en équivalence avec le métamodèle de langage de conception pédagogique de la plateforme Ganesha. La figure 71 représente un extrait de ce schéma XML. Le module intègre deux processus d'utilisation : un processus d'import pour opérationnaliser les scénarios spécifiés via des outils de conception externes et un processus d'exportation des cours (ou une partie) dans un format externe conforme au schéma.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
...
  <xsd:element name="Sequence" maxOccurs="unbounded">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Titre" type="xsd:string"/>
        <xsd:element name="Objectifs pedagogiques" type="xsd:string"/>
        <xsd:element name="id" type="xsd:nonNegativeInteger"/>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="SCORM 1.2" maxOccurs="unbounded">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="Activite" type="xsd:string"/>
                <xsd:element name="Description" type="xsd:string"/>
                <xsd:element name="Mots clefs" type="xsd:string"/>
                <xsd:element name="Visible" type="b01" default="1"/>
                <xsd:element name="id" type="xsd:nonNegativeInteger"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="Activite collaborative" maxOccurs="unbounded">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="Activite" type="xsd:string"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:choice>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
...

```

Figure 71 : Extrait du schéma XML de Ganesha

Contrairement au cas de Moodle, le processus de conception et d'opérationnalisation proposé pour Ganesha n'impose pas la contrainte de commencer par exporter le cours avant de spécifier ou modifier son contenu sur l'éditeur. Nous avons tenu en compte de cette contrainte dans Moodle afin de réutili-

ser nos outils pour la plateforme UMTICE⁹. Nous avons eu besoin de l'identifiant du cours requis par les fonctionnalités *backup* et *restore*.

6.1.3.2. Description du processus d'opérationnalisation (import)

Le processus d'opérationnalisation (import) consiste à implémenter les modèles de situations pédagogiques spécifiées en dehors des plateformes et conformes au schéma XML. La figure 72 illustre ce processus. L'idée générale de l'algorithme de ce processus consiste à parser le fichier XML du modèle et à insérer les données associées dans la base de données. Il ne requiert pas de transformations (XML) sur le modèle. Le processus prend en compte la contrainte d'adaptation d'un cours existant. Concrètement, le processus commence par parser le fichier XML du nouveau cours/module. Il se charge de la création des nouveaux concepts nouvellement ajoutés au cours (ceux qui apparaissent dans le scénario à importer mais pas dans le cours initial). Ce point a nécessité d'initialiser les identifiants des nouveaux éléments à zéro. Le processus se charge également de la mise à jour des informations des concepts existants (ie. le concept existe dans le cours mais il était modifié dans le nouveau cours). Nous avons également pris en compte le cas de la suppression d'un ou plusieurs élément(s) de cours (ie. le concept apparaît dans le cours initial mais pas dans le cours à importer). Le processus doit ainsi vérifier que certains éléments tels que les *séquences* n'appartiennent pas à un autre module avant leurs suppressions. Cette contrainte est due à la possibilité d'appartenance d'une séquence à plusieurs modules (prouvée par le métamodèle du langage de conception de Ganesha résultant de l'application du processus d'identification du langage de conception pédagogique sur Ganesha).

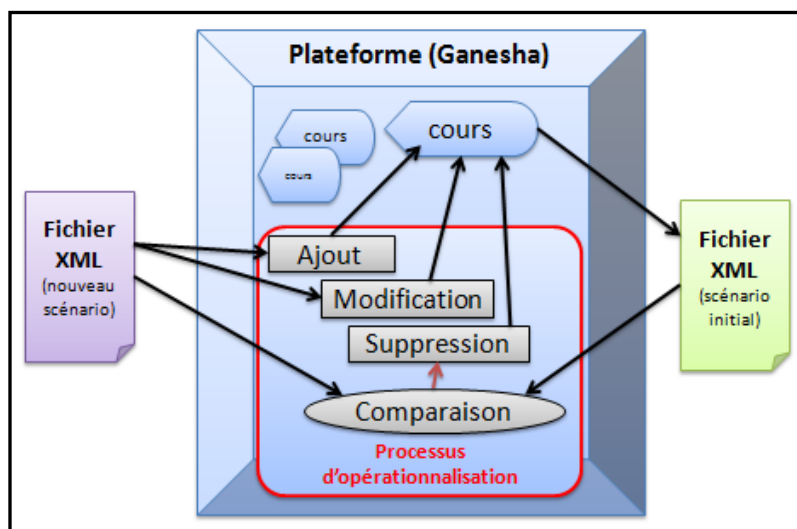


Figure 72 : Processus d'opérationnalisation développé pour Ganesha

6.1.3.3. Description du processus d'export

Les cours sur Ganesha sont désignés par le concept de module. Les modules sont associés à des matières qui sont également associées à des domaines. Le processus d'export (figure 73) permet de sauvegarder un module donné en format XML. Après avoir choisi un module, le processus se charge de générer le fichier XML correspondant. Dans le but de préserver la sémantique de la plateforme, nous avons tenu en compte la possibilité d'exporter les propriétés caractéristiques du domaine et de la ma-

⁹ Umtice est une plateforme de formation à l'origine de la plateforme Moodle. Elle est utilisée au sein de l'université du Maine

tière. Le processus interroge la base de données de Ganesha, il examine les tables qui embarquent le contenu du module. Les tables sont consultées en suivant la hiérarchie du schéma XML (par exemple la table des séquences est interrogée avant celle des activités). Ce processus aboutit à la génération d'une instance de document XML conforme au schéma XML de la plateforme et directement manipulable par des outils de conception spécifiques à Ganesha.

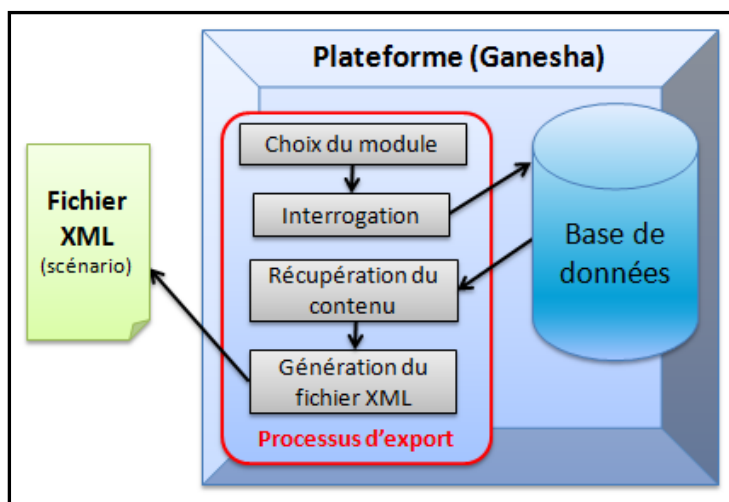


Figure 73 : Processus d'export développé pour Ganesha

6.1.4. Conclusion

Dans cette partie, nous avons exposé les propositions portant sur le développement de services d'opérationnalisation et d'export dédiés à des plateformes spécifiques. Ces processus prennent en charge la mise en place et l'export des scénarios sérialisés sous un format XML et conformes au schéma XML du langage de conception pédagogique de la plateforme cible. Ces scénarios peuvent être spécifiés et/ou modifiés via des outils de conception externes que nous présentons dans la section suivante.

6.2. Développement des outils de conception

Dans cette partie, nous allons exposer un cas de figure d'utilisation du langage de conception pédagogique d'une plateforme. Ce cas concerne le développement des prototypes/outils de conception pédagogique pour la spécification des situations d'apprentissages. Nous avons utilisé le *Eclipse Modeling Project* comme cadre pratique pour nos expérimentations en se focalisant sur l'approche DSM. Nous présentons ce projet, ses frameworks, ainsi que la façon dont nous les avons utilisés pour développer nos prototypes d'éditeurs graphiques spécifiques aux plateformes Moodle et Ganesha.

6.2.1. Présentation de l'outillage : Eclipse Modeling Project

L'*Eclipse Modeling Project* (EMP) est un projet spécifique dédié à la communauté Eclipse supportant l'évolution et la promotion des technologies de développement dirigées par les modèles. Ce projet propose un panel très large de frameworks pour faciliter le développement des outils/prototypes selon une approche dirigée par les modèles [EMP 2012]. Les deux frameworks principaux utilisés, dans le

cadre de notre travail pour le développement de outils de conception graphiques, sont les frameworks EMF (*Eclipse Modeling Framework*) et GMF (*Graphical Modeling Framework*). Ces deux frameworks facilitent le développement des prototypes (des éditeurs de modèles) grâce à la génération automatique du code source Java. Leur utilisation est basée principalement sur la définition d'un ensemble de modèles (modèle du domaine et les modèles de la couche graphique (*tooling*, *graph* et *mapping*)). Nous détaillerons la définition ainsi que l'utilisation de ces modèles dans la section 6.2.2.

6.2.1.1. Eclipse Modeling Framework

EMF est un framework dédié à Eclipse pour la modélisation et la génération automatique ou semi-automatique du code. Il permet de spécifier des métamodèles et de gérer des modèles conformes à ces métamodèles [Pelechano et al. 2006]. Les métamodèles peuvent être spécifiés en utilisant différentes syntaxes : Java annoté, schéma XML, diagramme de classe UML, etc. Une fois importés par EMF à partir de l'un de ces formats ou manuellement spécifiés sur le framework, les métamodèles sont transformés dans un format conforme au méta-métamodèle Ecore. Ce dernier, utilisé par EMF, est compatible avec le métamétamodèle MOF (*Meta Object Facility*) proposé par l'OMG (*Object Management Group*).

EMF intègre des éléments fondamentaux tels que EMF.Edit et EMF.Codegen :

- EMF.Edit comprend un ensemble de classes génériques et réutilisables pour la construction des éditeurs à partir des métamodèles EMF [Bézivin et al. 2004].
- EMF.Codegen fournit le service de génération de code pour construire un éditeur complet à partir d'un métamodèle EMF.

L'assistant graphique du framework EMF permet de spécifier les options de génération du code des éditeurs [Bézivin et al. 2004]. L'éditeur généré utilise les classes du framework EMF.Edit qui fournissent un accès aux données des modèles afin de les afficher dans des interfaces graphiques de base : par exemple des tableaux contenant l'ensemble des éléments d'un modèle ou des feuilles de propriétés adaptées à chaque élément d'un modèle. Le code généré par EMF n'est pas toujours suffisant. Pour cela, il est souvent nécessaire d'apporter des modifications sur ce code ou d'ajouter d'autres fonctionnalités [Kelly 2004].

Le framework EMF repose sur un générateur de code qui permet, à partir du modèle de domaine, de générer trois couches de l'outil d'édition. Ces couches sont :

- **la couche modèle** représente le modèle de domaine (le métamodèle) sous forme de classes Java ;
- **la couche du EMF.Edit** est l'ensemble de classes permettant la visualisation et l'édition des modèles ;
- **la couche éditeur** représente l'interface graphique de l'éditeur

La figure 74 illustre le processus de génération d'un outil de conception (éditeur) par le framework EMF. Elle représente également l'ensemble des relations qui existent entre les couches et les modèles.

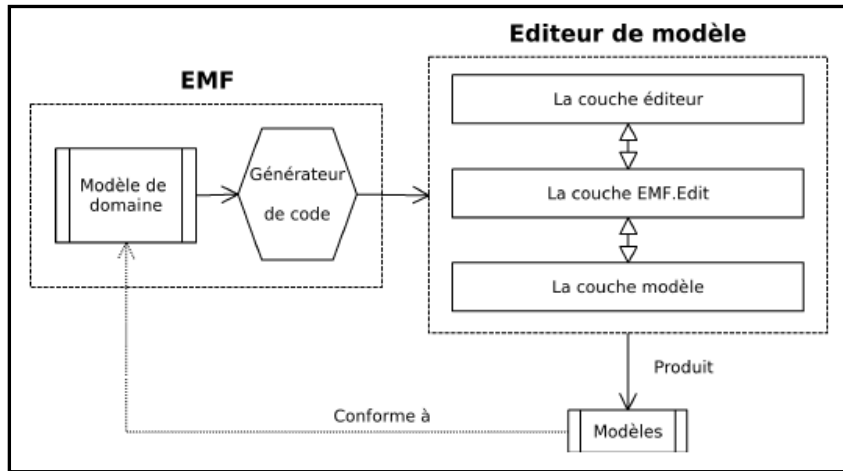


Figure 74 : Processus EMF [Zendagui 2010]

L'interface graphique de l'éditeur générée par EMF (figure 75) est très basique. Elle permet la création des éléments du modèle par l'intermédiaire d'une arborescence composée des éléments de base du modèle du domaine. Le framework GMF intervient pour surmonter la faiblesse de l'interface graphique générée par EMF en proposant des solutions pour générer des interfaces plus élaborées et surtout graphiques.

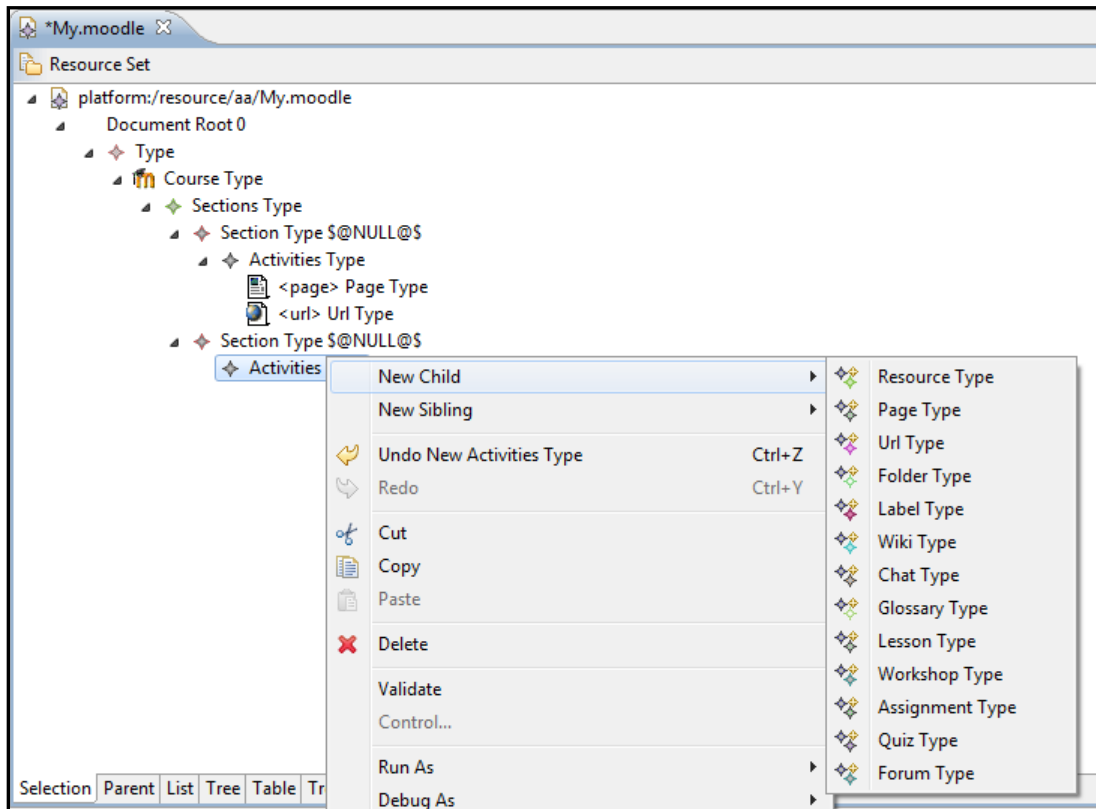


Figure 75 : Aperçu de l'éditeur généré par EMF

6.2.1.2. Graphical Modeling Framework

GMF est un framework spécifique pour le développement et l'exécution des éditeurs graphiques. Il se base sur les frameworks EMF et GEF (*Graphical Editing Framework*). En prenant le rôle de pont

entre ces deux frameworks, GMF fournit de nombreuses fonctionnalités permettant d'automatiser certaines tâches qui étaient réalisées à la main en utilisant directement les frameworks EMF et GEF. Les différents éléments créés au sein des frameworks EMF et GEF sont assemblés grâce à un modèle de mapping (*Mapping Model*).

GMF permet le développement d'éditeurs graphiques riches à partir d'un modèle de domaine formalisé avec EMF. Le développement d'outil DSM impose l'utilisation simultanée des frameworks EMF et GEF pour prendre en compte à la fois la modélisation des concepts d'un domaine et leur notation graphique. GEF fournit le support graphique requis pour la construction d'un éditeur de diagramme qui remplace la couche graphique générée par EMF [Kelly 2004].

Le framework GMF est composé de deux composants principaux : un ensemble d'outils et une infrastructure d'exécution. L'outillage est constitué des éditeurs permettant de créer/modifier des modèles décrivant la notation graphique, la sémantique et l'aspect outillage d'un éditeur graphique ainsi que d'un générateur de code des éditeurs graphiques. L'infrastructure d'exécution permet l'exécution des plugins générés, elle met à disposition des fonctionnalités permettant de produire des éditeurs graphiques universels et extensibles. Elle offre :

- un ensemble de composants réutilisables pour la construction des éditeurs graphiques, telles que l'impression, l'exportation d'image, les barres d'outils, etc. ;
- un modèle standardisé pour décrire les éléments du diagramme, séparant ainsi la sémantique (domaine) de la notation (graphique) des éléments ;
- une infrastructure de commande qui synchronise les commandes d'EMF et de GEF;
- une structure extensible qui rend les éditeurs graphiques générés ouverts et extensibles.

GMF propose la persistance des modèles créés via son éditeur dans un format XML synchronisé avec la vue graphique en cours d'édition. Ce mécanisme peut être configuré de manière à ce que le format XML ne soit pas du XMI mais du code XML. Cette caractéristique de GMF nous intéresse parce qu'elle permet la formalisation des futurs modèles produits en XML, conformément au schéma XML du modèle du domaine représentant le langage de conception pédagogique de la plateforme utilisée, et enfin supportable par les modules d'import/export.

GMF s'appuie sur des modèles principaux pour le développement des éditeurs graphiques. Ces modèles, basés sur le modèle de domaine, sont :

1. Le modèle de définition graphique (*Graphical Definition Model*), portant l'extension `.gmfgraph`. Il est généré à partir du modèle de domaine (`.ecore`) en utilisant l'assistant `GMFGraph`. Ce modèle embarque les informations de la représentation graphique (les figures, les nœuds, les compartiments, les liens, etc.). La couche graphique est composée de deux catégories d'éléments :

- les composants de base de l'interface graphique qui sont indépendants du modèle de domaine et communs pour toutes les interfaces générées par les frameworks EMF et GMF tels que les menus, la palette, les tableaux, etc.
- la notation graphique des éléments du modèle de domaine qui contient la définition de la représentation graphique des éléments du domaine. Elle dépend du modèle de domaine utilisé.

2. Le modèle de définition d'outillage (*Tooling Definition Model*), portant l'extension `.gmftool`. Il définit l'aspect outillage de l'éditeur graphique en utilisant l'assistant GMFTool. Il est utilisé pour la conception de la palette d'outils et d'autres outils périphériques comme les menus et les barres d'outils.

3. Le modèle de définition du Mapping (*Mapping Definition Model*), portant l'extension `.gmfmap`. Il définit en utilisant l'assistant GMFMap les liaisons entre le modèle du domaine, le modèle de la définition graphique et le modèle de la définition d'outillage. Il permet de lier un concept du domaine (défini dans le modèle de domaine) avec sa représentation graphique (définie dans le modèle de définition graphique) et les outils qui le gèrent (définis dans le modèle de définition de l'outillage).

4. Le modèle de génération du code de l'éditeur, portant l'extension `.gmfgen`. Il est généré à partir du modèle de définition de mapping. Il permet de paramétrer les détails d'implémentation qui seront utilisés lors de la génération du code final de l'éditeur.

La figure 76 illustre le processus GMF et les étapes nécessaires pour la génération des éditeurs graphiques. Elle représente également les modèles et les relations qui existent entre eux. Les différents modèles sont généralement incomplets et nécessitent plusieurs ajouts, modifications et paramétrages complétés à la main. Dans les sections suivantes, nous détaillons notre utilisation de ces frameworks et la définition des modèles pour le développement des outils de conception. Nous présentons également les différentes manipulations et modifications exercées sur ces modèles.

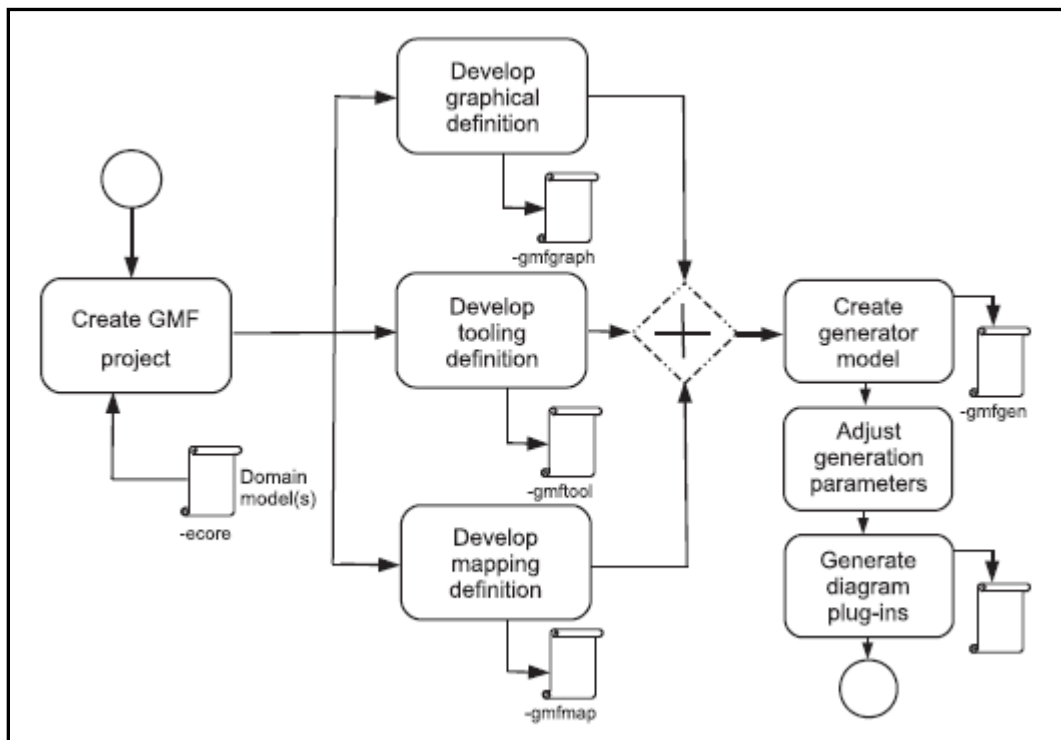


Figure 76 : Aperçu du processus de GMF [Laforcade 2010]

6.2.2. Mooditor : éditeur graphique pour Moodle

6.2.2.1. Introduction

Mooditor est un outil de conception de scénarios pédagogiques destiné aux enseignants-concepteurs utilisant la plateforme de formation à distance Moodle. Il consiste à proposer une interface graphique pour la conception des scénarios pédagogiques en se basant uniquement sur le langage de conception pédagogique spécifique à cette plateforme. Lors du développement du Mooditor, nous avons adopté l'approche *Domain-Specific Modeling*. Nous avons utilisé les deux frameworks présentés précédemment (EMF et GMF) comme un cadre pratique technique pour l'application de cette approche.

6.2.2.2. Les exigences des praticiens

Dans une première expérimentation, nous avons choisi de mettre l'accent sur certains objectifs et besoins des praticiens permettant la spécification et le développement d'un VIDL et d'un éditeur afin de vérifier notre approche DSM et son outillage [Abedmouleh et al. 2012d] [Laforcade et Abedmouleh 2012]. Nous avons identifié les exigences suivantes :

- Dessiner / modéliser graphiquement les sections des cours en les arrangeant dans l'espace de dessin sans spécifier un ordre définitif (des sections).
- Permettre le dessin des relations (sous forme de flèches) entre les sections pour représenter leurs futurs ordres sur la plateforme.
- Offrir, dans la palette de l'outil, les activités et les ressources de base proposées par la plateforme.
- Permettre l'ajout des activités et des ressources au sein des sections et la définition de leurs utilisations sans avoir à spécifier toutes les données habituelles et requises pour chacune d'elles.

Concrètement, les praticiens visent des outils de création (*authoring-tool*) orientés diagrammes, spécifiques aux plateformes qu'ils utilisent et qui leur permettront de se concentrer sur la conception globale du cours.

6.2.2.3. Architecture de Mooditor

L'architecture de Mooditor est composée de trois composants principaux : le composant IHM, le composant de gestion de modèles et de métamodèles et le composant du module d'importation. La figure 77 illustre l'architecture de Mooditor, les différents composants et les différentes interactions entre eux. Pour certains éléments de cette architecture tels que le composant IHM et les interactions entre les composants, le code a été généré automatiquement par les frameworks EMF et GMF. Cependant, nous avons développé certaines parties afin de palier certains manques et pour répondre à certains besoins.

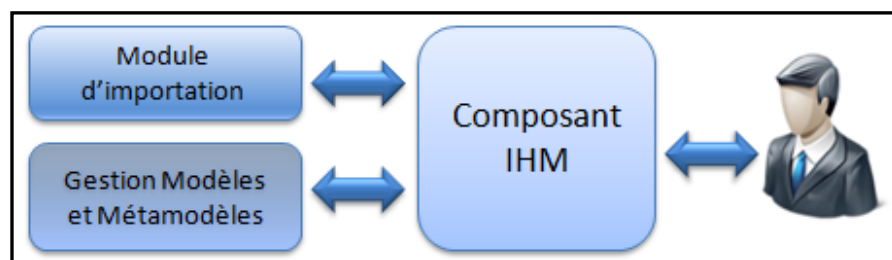


Figure 77 : Architecture simplifiée de Mooditor

6.2.2.5. Développement de la couche graphique de Mooditor

Cette partie est consacrée à la définition des éléments composant la couche graphique de l'éditeur. Elle nécessite la définition de plusieurs modèles dont certains sont dépendants du modèle du domaine (tels que le modèle d'outillage et de la définition graphique). La figure 79 illustre de façon simplifiée des modèles utilisés que nous détaillons dans les sections suivantes.

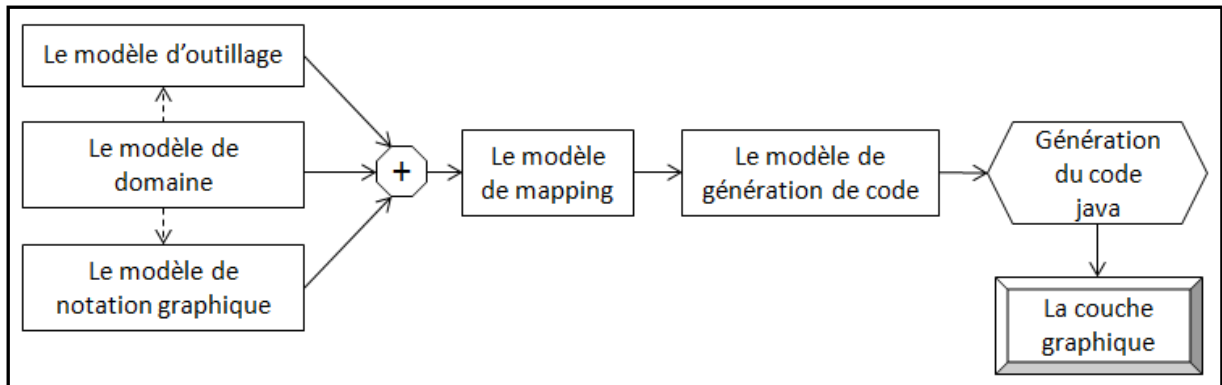


Figure 79 : les modèles de définition de la couche graphique

6.2.2.5.1. La définition de l'outillage : le modèle mooditor.gmftool

Le modèle d'outillage consistera à définir les éléments mis à disposition de l'enseignant-concepteur au sein de l'éditeur pour la spécification de son cours ou de son scénario pédagogique. Ces éléments sont définis au sein de la palette, des menus ou des barres d'outils de l'éditeur. L'éditeur que nous avons développé est caractérisé par une seule palette contenant tous les éléments de conception du cours spécifiés dans le modèle du domaine.

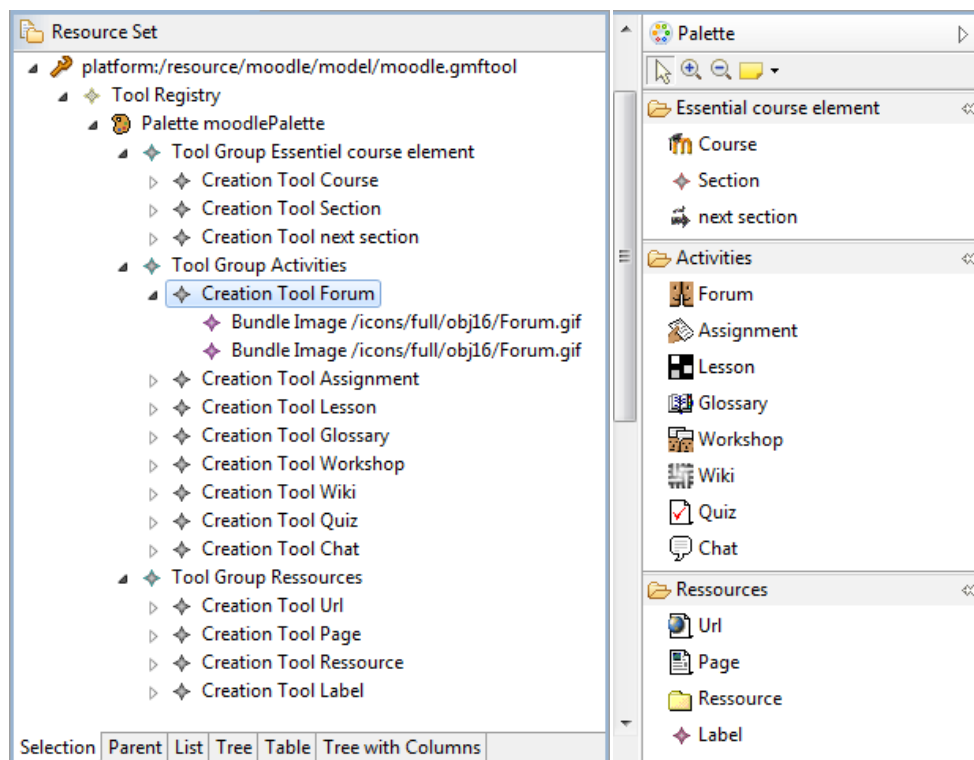


Figure 80 : Modèle de définition d'outillage et la palette de l'éditeur

Dans cette palette sont définis trois groupes d'outils. Le premier contient les outils de création des éléments de base (un cours, des sections, les groupes et le lien entre les sections). Les autres groupes contiennent les outils de création des concepts de cours en termes d'activités et de ressources définies dans le modèle du domaine. Afin de faciliter le design sur l'éditeur, nous avons associé une icône spécifique à chaque outil de création d'un élément du cours. Nous avons choisi de reprendre les mêmes icônes utilisées au sein de la plateforme Moodle. La figure 80 illustre le modèle de définition d'outillage (*mooditor.gmftool*) (la partie à gauche) et la palette graphique (la partie à droite).

6.2.2.5.2. La définition graphique : le modèle mooditor.gmfgraph

Le modèle de définition graphique (*mooditor.gmfgraph*) consiste à définir les éléments à utiliser au sein de l'espace de design de l'éditeur graphique. Quelques éléments de ce modèle, tels que les figures et les rectangles descriptifs, sont définis automatiquement par l'intermédiaire de l'assistant GMFGraph. Cependant, plusieurs améliorations doivent être apportées à la définition des éléments générés par l'assistant et les éléments manquants.

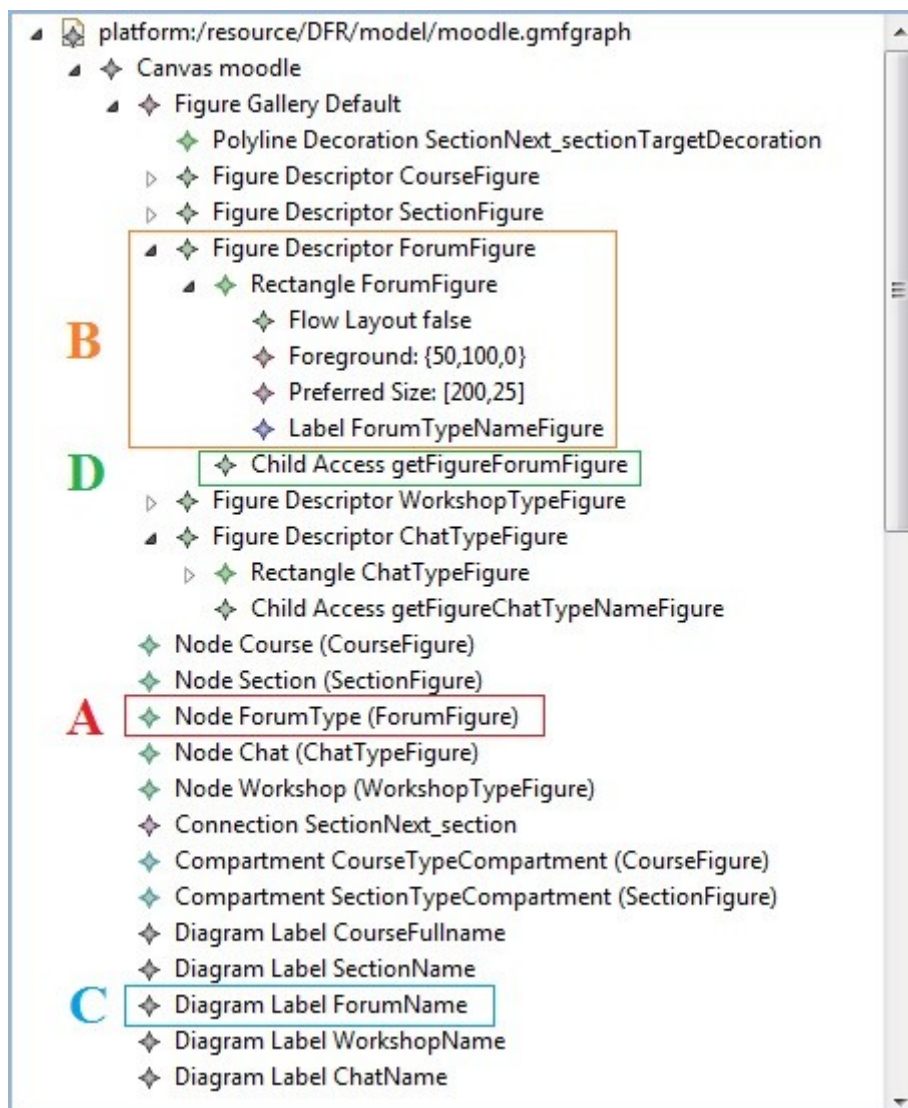


Figure 81 : Représentation graphique d'un forum

Le modèle *mooditor.gmfgraph* est conséquent. Dans ce document, nous ne pouvons pas présenter l'ensemble de la modélisation graphique que nous avons définie dans le modèle graphique. En revanche, nous illustrons en exemple la définition graphique du concept du forum. La figure 81 illustre un extrait des éléments du modèle *gmfgraph* associés au forum et que nous décomposons en quatre parties.

Chaque élément du modèle du domaine doit être défini par un nœud qui désigne la racine de la définition graphique. La partie entourée par le cadre (A) montre le nœud *forum*. Il servira pour le Mapping au sein du modèle *mooditor.gmfmap* (nous détaillerons cela dans la section suivante). Il est décrit par l'intermédiaire d'un élément de type *Figure Descriptor*.

Les éléments de type *Figure Descriptor* consistent à définir les détails de la représentation graphique. Chaque élément est composé de deux blocs. Par exemple sur la figure 81, le premier bloc entouré par le cadre (B) définit la structure de la représentation graphique du concept (*forum*). Elle est décrite par des éléments graphiques de base. A titre d'exemple, le forum est décrit par l'intermédiaire d'un rectangle et dispose d'un objet de type label. La couleur de fond et la taille du rectangle sont des exemples des propriétés du rectangle représentant le forum. Les labels (étiquettes) sont utilisés pour initialiser et afficher un texte et/ou une icône au sein du rectangle suite à l'ajout d'un élément (forum par exemple). Le contenu de ces objets peut être défini au moment de la définition du modèle graphique ou saisi par l'enseignant-concepteur au moment de l'édition du modèle (le cours). Le deuxième bloc entouré par le cadre (C) montre les labels utilisés pour la représentation graphique du concept forum.

La partie entourée par le cadre (D) représente la définition des méthodes qui permettent de fournir les valeurs des différents labels définis dans le premier bloc. Une méthode doit être définie pour chaque *label* ajouté dans le modèle. Ensuite, le code de ces méthodes est généré automatiquement par le framework GMF.

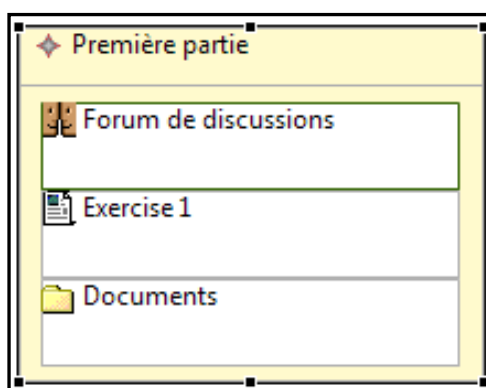


Figure 82 : représentation graphique du concept section

La figure 82 montre un exemple d'une section définie dans l'espace de définition graphique de l'éditeur. Cette section se présente comme un rectangle dont le fond et les bordures sont de couleur beige clair. La représentation graphique de la section est définie dans le modèle *mooditor.gmfgraph* en utilisant un élément graphique de type *Rectangle*. La section est munie d'un label représentant le nom de la section (*Première partie*). À l'intérieur de ce rectangle se trouvent les activités et les ressources

ajoutées par l'enseignant-concepteur en utilisant la palette de l'éditeur. Chaque élément tel que forum est muni d'un label pour spécifier son nom.

Afin de rendre les sections capables d'embarquer plusieurs éléments, nous avons associé à chaque section un élément appelé compartiment. Les compartiments nécessaires pour l'éditeur n'étaient pas générés par les assistants de GMF. Ils ont été définis manuellement dans la couche graphique de l'éditeur puis ils ont été associés aux éléments concernés.

6.2.2.5.3. Le mapping entre les modèles de base de GMF : le modèle mooditor.gmfmap

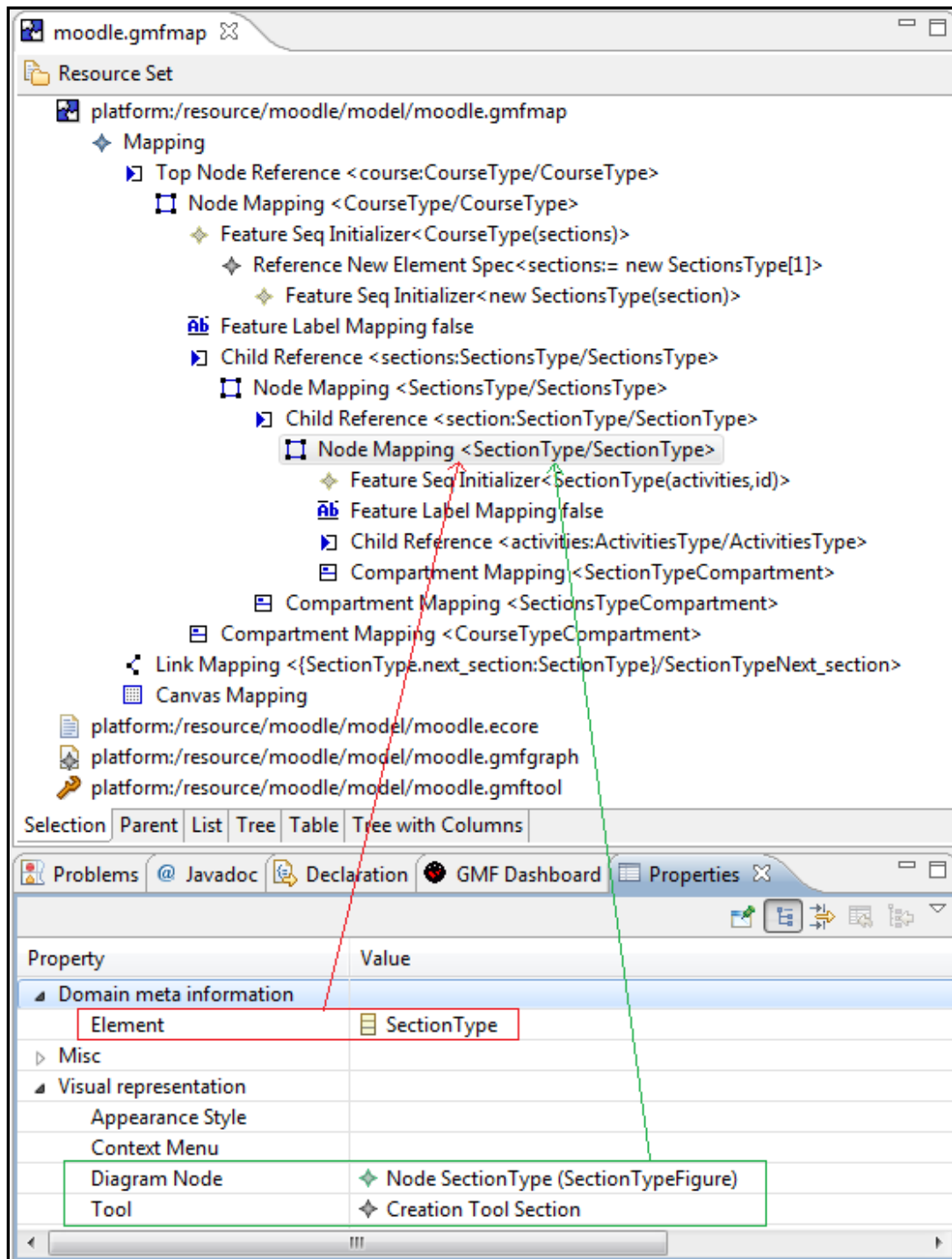


Figure 83 : Extrait du modèle mooditor.gmfmap

Le modèle *mooditor.gmfmap* consiste à définir les relations entre les modèles de domaine, de définition graphique et de définition de l'outillage. Il permet de relier un concept du modèle de domaine (défini dans le *.ecore*) avec leurs représentations graphiques (définies dans le *.gmfgraph*) et les outillages définis pour leurs créations (définis dans le *.gmftool*). L'assistant de GMF permet de générer une première version de ce modèle à partir des trois modèles précédents. Cependant, cette version n'est pas complète et nécessite l'ajout de plusieurs éléments manquants tels que les compartiments, les initialisations, le mapping aux labels, etc. La figure 83 illustre un extrait complet de ce modèle.

Nous avons spécifiquement présenté cet extrait du modèle parce qu'il montre deux types d'éléments importants. Le premier est *Feature Seq Initializer* qui consiste à initialiser certains éléments du modèle pendant leurs créations. Dans cet exemple, l'initialisation permet de créer une section dès la création d'un nouveau cours. Le deuxième élément est de type *Node Mapping*. Il permet de relier un élément du modèle de domaine avec sa représentation graphique et les outillages définis pour sa création. Dans cet exemple, l'élément du modèle de domaine *section* (cf. la partie de la figure 83 encadrée en rouge) qui est relié avec sa représentation graphique et l'outillage défini pour la création d'une nouvelle section (cf. la partie de la figure 83 encadrée en vert).

6.2.2.5.4. Génération du code Java de la couche graphique : le modèle *mooditor.gmfgen*

Après la définition des modèles requis par le processus GMF (modèle du domaine, modèle d'outillage, modèle de la définition graphique et le modèle de mapping), l'assistant de création de modèle de GMF permet de générer un nouveau modèle appelé le modèle générateur (*mooditor.gmfgen*). Nous avons apporté quelques modifications sur ce modèle en ajustant les paramètres responsables à la génération du contenu du modèle et de sa présentation graphique dans deux fichiers séparés. Le premier fichier représente la définition graphique du modèle spécifié. Il contient les propriétés graphiques du modèle, par exemple les coordonnées des différents objets. Le deuxième fichier représente les éléments du modèle de domaine, utilisés pour la spécification du cours. Il est conforme au modèle de domaine (correspondant au langage de conception pédagogique de la plateforme Moodle).

6.2.2.5.5. Mooditor : un plugin d'Eclipse

Après la définition de tous les modèles du processus GMF, il est possible de réaliser la génération automatique du code Java des éditeurs graphiques en tant qu'un plugin d'Eclipse ou une application *standalone*. La figure 84 représente une capture d'écran de l'interface de l'outil généré composé de trois parties principales :

- La partie entourée par le cadre (A) représente la palette de l'outil.
- La partie entourée par le cadre (B) représente l'espace de dessin et de conception. Dans cette partie, les modèles sont créés et édités sous forme graphique, par les enseignants-concepteurs.
- La partie (C) représente la grille de spécification des attributs et des propriétés des éléments du cours.

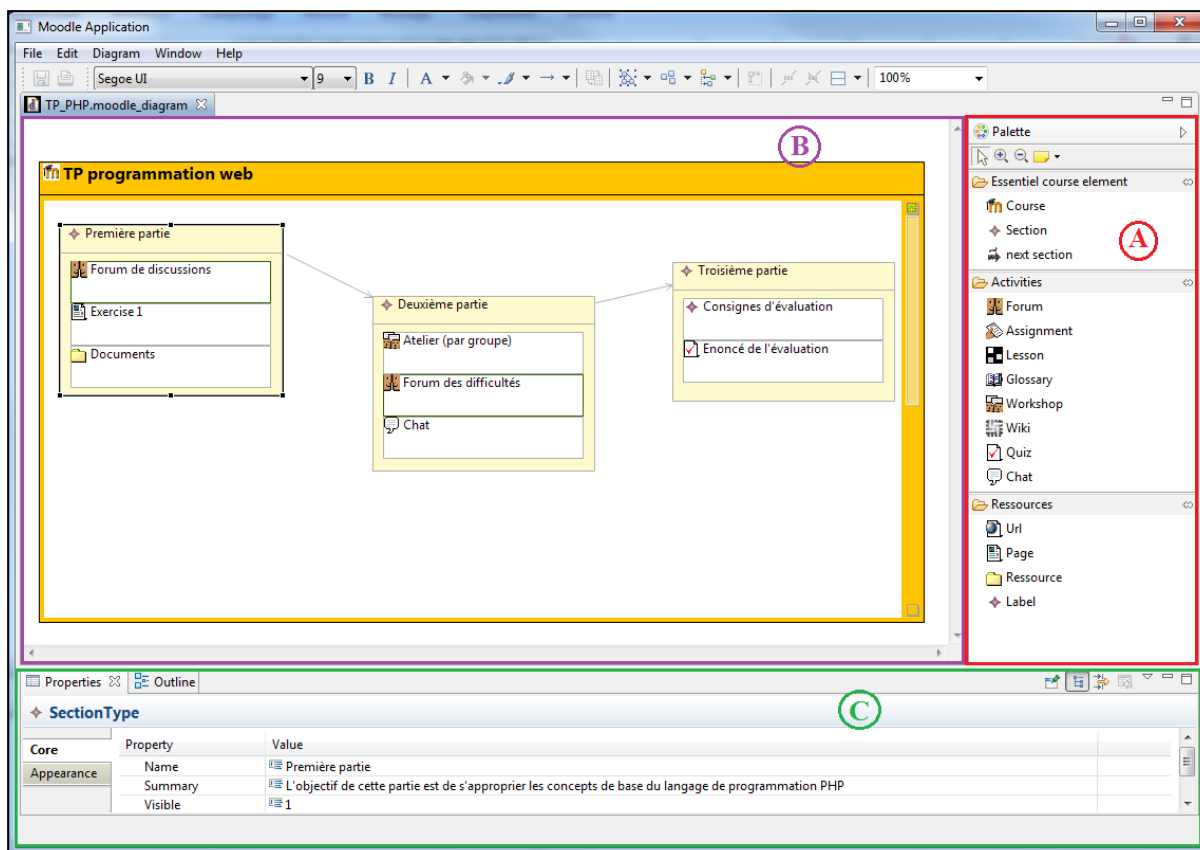


Figure 84 : Capture d'écran de Mooditor

L'éditeur final peut être utilisé (1) pour spécifier graphiquement des scénarios d'apprentissage en tant que modèles graphiques et (2) pour visualiser un scénario pédagogique à partir d'un autre outil, mais en satisfaisant la condition d'être conforme au schéma utilisé. Les modèles sont visualisés dans une vue orientée diagramme et sérialisés de façon synchrone en tant que fichier XML interprétable pas la machine conformément avec le schéma XML.

Cet éditeur graphique répond aux exigences des praticiens précédemment citées (section 6.2.2.2). Les enseignants-concepteurs peuvent compléter leur conception directement sur Moodle en spécifiant les détails de bas niveau qui ne sont pas traités par cet éditeur. Ils peuvent également exporter les cours de la plateforme pour les concevoir à nouveau sur l'éditeur sans perdre les détails de bas niveau (qui seront fusionnés pendant l'importation par le module spécifique de Moodle que nous avons élaboré).

6.2.2.6. Résolution de certaines issues

Après la génération automatique du code de l'éditeur via les frameworks EMF et GMF, en particulier la première version, nous avons constaté certains manques/défauts de conception nécessitant plusieurs améliorations. Afin de palier ces défauts, nous avons amené des modifications, des ajouts et des paramètres d'éléments dans les modèles de ces frameworks.

Concrètement, nous avons identifié certains défauts de design dont les trois principaux sont :

- (i) L'éditeur ne permet pas la définition des relations entre les sections de cours.

- (ii) Lors de l'importation d'un fichier XML d'un cours sur l'espace graphique de l'éditeur, spécifié via le même éditeur ou exporté via le module de communication ajouté à Moodle (cf. section 6.1.2.2), les relations entre les sections ne sont pas initialisées.
- (iii) Lors de l'enregistrement du cours spécifié via l'éditeur, les sections sont ordonnées par ordre de création et non par ordre pédagogique. En effet, l'enseignant a besoin parfois de spécifier les sections de son cours. Ensuite, il détermine leur ordre définitif par des flèches.

La logique de EMF consiste à cartographier (mapper) la notion d'ordre en accord avec la création des sections dans le compartiment d'un cours. Il s'agit concrètement d'un problème parce que les praticiens ne savent pas nécessairement à l'avance l'ordre concret des sections qu'ils conçoivent. Il est ainsi important de différencier l'ordre d'instanciation des sections de leur ordre pédagogique. Afin de remédier ce manque, nous avons apporté des modifications sur le modèle du domaine (le métamodèle de conception pédagogique de Moodle) et sur le code java généré par EMF [Abedmouleh et al. 2011c].

Dans un premier temps, les modifications ont consisté à :

- Ajouter une *EReference nextSection* dans la classe *section eClass* avec des bornes inférieure / supérieure défini à "0..1" (voir la relation *next_section* en rouge sur la figure 78).
- Définir l'attribut *transient* à vrai (*true*) dans le but d'informer le mécanisme de persistance EMF de ne pas traiter l'élément *nextSection*.
- Définir la notation correspondante (ligne avec une flèche) à l'élément *nextSection* dans le modèle graphique et la palette de l'éditeur.
- Associer les deux nouveaux éléments ensemble (la représentation graphique et l'élément de la palette associés à *nextSection*) en spécifiant le lien entre eux sur le modèle de *mapping*.
- Ajouter des contraintes OCL (*Object Constraint Language*) au modèle de Mapping, afin de ne pas autoriser (1) la relation réflexive à une section et (2) les cycles de sections.

Dans cette partie, nous n'avons pas modifié le code généré. Ce code, représentant le plugin de l'outil, peut être utilisable pour spécifier des modèles conformes au modèle de domaine.

Dans un deuxième temps et pour résoudre le problème de (ii) et (iii), nous avons apporté des modifications sur le code généré par EMF afin de redéfinir le comportement des méthodes de sauvegarde (*save*) et d'import (*load*). La méthode de sauvegarde se charge de l'enregistrement du modèle sous un format XML. Les instructions de code ajoutées permettent à cette fonction d'enregistrer les instances de sections ordonnées à partir de la section parent en tenant en compte les relations *nextSection* spécifiées. La figure 85 illustre un extrait du nouveau code associé à la fonction *save()*.

La méthode d'import *load()* doit définir l'élément *nextSection* sur l'espace graphique du cours en tenant en compte l'ordre des balises *section* parcourues à partir du fichier XML. Nous avons défini les instructions indispensables pour la spécification des relations entre les sections. La figure 86 représente la partie du code associée à la fonction *load()*.

```

1 public void save(Map<?, ?> options) throws IOException {
2     if (this.getContents() != null)
3         if (this.getContents().get(0) != null) {
4             MoodleType momo = (MoodleType) this.getContents().get(0);
5             SectionsType s = null;
6             if (momo.getCourse() != null)
7                 if ((s = momo.getCourse().getSections()) != null)
8                     s.reordonneSections2();
9         }
10    super.save(options);
11 }

1 /**
2  * @generated NOT
3  */
4 public void reordonneSections2() {
5     EList<SectionType> listeSection = getSection();
6
7     int ty = listeSection.size();
8     SectionType prems = null; //le premier qu'on va chercher
9
10    for (int i=0; i< ty ; i++){
11        prems = (SectionType)listeSection.get(i);
12        if ((prems.getPrevious_section()) == null){
13            //pas de precedent => c'est le premier (s'ils sont tous chainés)
14            while (prems.getNext_section() != null){
15                SectionType next = prems.getNext_section();
16                listeSection.move(listeSection.indexOf(prems), listeSection.indexOf(next));
17                prems = next;
18            }
19        }
20    }
21 }

```

Figure 85 : Extrait du nouveau code de la fonction save()

```

1 @Override
2 /**
3  * @generated NOT
4  */
5 public void load(Map<?, ?> options) throws IOException {
6     super.load(options);
7     if (this.getContents() != null)
8         if (this.getContents().get(0) != null) {
9             MoodleType momo = (MoodleType) this.getContents().get(0);
10            SectionsType s = null;
11            if (momo.getCourse() != null)
12                if ((s = momo.getCourse().getSections()) != null)
13                    s.createNextLinks();
14        }
15 }

1 /**
2  * @generated NOT
3  */
4 public void createNextLinks() {
5     if (this.getSection() == null)
6         return;
7     Iterator<SectionType> it = this.getSection().iterator();
8     if (!it.hasNext())
9         return;
10    int ty = getSection().size();
11    SectionType prec = null;
12    for (int i=0; i<ty; i++){
13        SectionType s = getSection().get(i);
14        if (prec != null && prec!=s)
15            prec.setNext_section(s);
16        prec = s;
17    }
18 }

```

Figure 86 : Extrait du nouveau code de la fonction load()

6.2.2.7. Conclusion

Cette section du chapitre a été consacrée à la représentation du premier outillage que nous avons développé afin d'instrumenter nos propositions théoriques. Le développement de l'éditeur pour la plateforme Moodle a représenté un nouveau terrain pour expérimenter l'approche DSM. Cet outil/prototype a représenté un cas de figure d'utilisation du métier d'une plateforme. Nous nous sommes basés sur le modèle de conception pédagogique de la plateforme Moodle, identifié grâce à l'application du processus que nous avons proposé. Ce métamodèle a servi comme base (le domaine métier) pour le développement de l'éditeur *Mooditor* : outil de conception des scénarios pédagogiques spécifiques à la plateforme Moodle. Ce métamodèle est utilisé par les frameworks EMF et GMF de l'*Eclipse Modeling Project* pour la génération du code Java de *Mooditor*. La première version de *Mooditor* a été générée automatiquement. Cependant, plusieurs modifications et ajouts ont été nécessaires afin de surmonter les défauts d'utilisation. Ces modifications ont concerné principalement le métamodèle de base, les modèles de la définition graphique et un nombre important d'instructions du code générés par les frameworks utilisés.

Nous avons également mené une deuxième expérimentation de l'approche DSM en développant un outil de conception de scénarios pédagogiques, similaire à celui de Moodle (section 6.2.2), mais spécifique pour la plateforme Ganesha. Nous n'allons pas montrer les étapes de développement parce qu'elles sont similaires à celle du développement de l'éditeur *Mooditor*. L'éditeur de Ganesha repose sur le langage de conception pédagogique dédié à cette plateforme, identifié via l'application du processus proposé (cf. section 5.3). Le développement de cet éditeur s'appuie également sur les frameworks EMF et GMF. Nous avons défini les modèles requis à la génération de cet éditeur (les modèles de domaine, d'outillage, de notation graphique, de mapping et de génération de code). Toutefois, plusieurs modifications sont apportées sur les modèles et le code généré automatiquement par les frameworks tels que la redéfinition des fonctions *save()* et *load()*. Dans de la section suivante, nous allons présenter une mise à l'essai de ces éditeurs.

6.3. Mise à l'essai

Dans cette section de chapitre, nous allons donner un exemple illustrant la mise en pratique de nos propositions sur les deux plateformes Moodle et Ganesha. Nous allons tout d'abord décrire textuellement un exemple de scénario pédagogique. Ensuite, nous allons utiliser les outils développés pour sa conception et son opérationnalisation. Nous détaillons les éléments utilisés pour la conception du scénario sur les éditeurs graphiques. Enfin, nous montrons l'opérationnalisation de chaque scénario sur la plateforme concernée.

6.3.1. Description du scénario pédagogique

Le scénario pédagogique que nous avons choisi pour illustrer cette mise à l'essai est une séance de travaux pratiques enseignée en présentiel dans le cadre de la formation DUT-INFO à l'IUT de Laval. Ce scénario s'adresse aux étudiants de la première année en promotion informatique. L'objectif du scénario est de proposer aux étudiants une formation de base sur le développement des applications web en php via la mise en pratique de certains exercices. L'apprenant est débutant en langage PHP, mais il a suivi des cours concernant les concepts de base de la programmation web en php. Dans une

séance de TP de quatre heures, les apprenants doivent écrire des scripts en php et commencer le développement des applications web simples. Le TP est composé de trois séquences principales :

- Séquence 1 : l'objectif de cette séquence est de s'approprier les concepts de base du langage de programmation php. Durant 45 minutes, les apprenants sont amenés à répondre individuellement à deux exercices. Le premier exercice consiste à récupérer des données envoyées via un formulaire web et les afficher au sein d'une nouvelle interface. Le deuxième exercice consiste à gérer les données d'une application web via l'utilisation d'une base de données. L'apprenant doit être capable de concevoir une base de données et d'interroger les données enregistrées. Les sujets des exercices sont mis à disposition des étudiants ainsi que certains documents de cours et de syntaxes de base de ce langage. Il est envisageable que les étudiants puissent s'entraider en communiquant entre eux ou avec leur tuteur.
- Séquence 2 : l'objectif de cette séquence est d'initier les étudiants à travailler en équipe sur un projet particulier. Durant 2h45min, les apprenants sont répartis en groupes de 4 personnes. Chaque groupe doit proposer une esquisse d'une application web pour la gestion des produits d'un centre commercial. Il doit définir une base de données pour l'enregistrement des produits en stock et les produits en promotion. Une partie de cette base est dédiée aux produits en rupture de stock pour aider à passer des commandes. L'application web est composée de plusieurs interfaces : d'authentification, de saisie de nouveaux produits, d'indication des produits en promotion, d'affichage de produits en rupture de stock, etc. Au cours de cette séquence, les groupes peuvent également échanger des idées et discuter certaines difficultés de développement rencontrées.
- Séquence 3 : la troisième partie consiste à évaluer individuellement les apprenants. Un test d'évaluation est prévu à la fin de la séance de cours pour une durée de 30 min. Avant de commencer l'évaluation, l'apprenant est invité à lire l'énoncé. Ensuite, il peut répondre aux questions dans l'ordre de son choix. Il est encouragé à les faire dans l'ordre préconisé par l'enseignant. Les discussions entre les apprenants ainsi qu'avec leur tuteur sont interdites.

6.3.2. Mise à l'essai : cas de la plateforme Moodle

La mise à l'essai des outils développés pour la plateforme Moodle consiste à suivre un processus bien spécifique. La figure 87 illustre ce processus qui doit être suivi par l'enseignant-concepteur afin d'effectuer la conception du scénario sur l'éditeur graphique et ensuite d'opérationnaliser le scénario résultat sur la plateforme Moodle.

A partir de l'espace du cours, l'enseignant-concepteur commence par exporter son cours via le nouveau module d'import/export (étape 1, figure 87). Il suffit qu'il clique sur le bouton *download* au sein du module ajouté. Les transformations des fichiers de sauvegarde générés par la fonction backup ainsi que la procédure développée s'exécute de façon implicite. Le cours est exporté sous forme d'un fichier XML contenant les informations conformes au schéma XML que nous avons spécifié. La figure 88 illustre un exemple d'un nouveau cours sur Moodle et le fichier XML associé et généré via le module de communication.

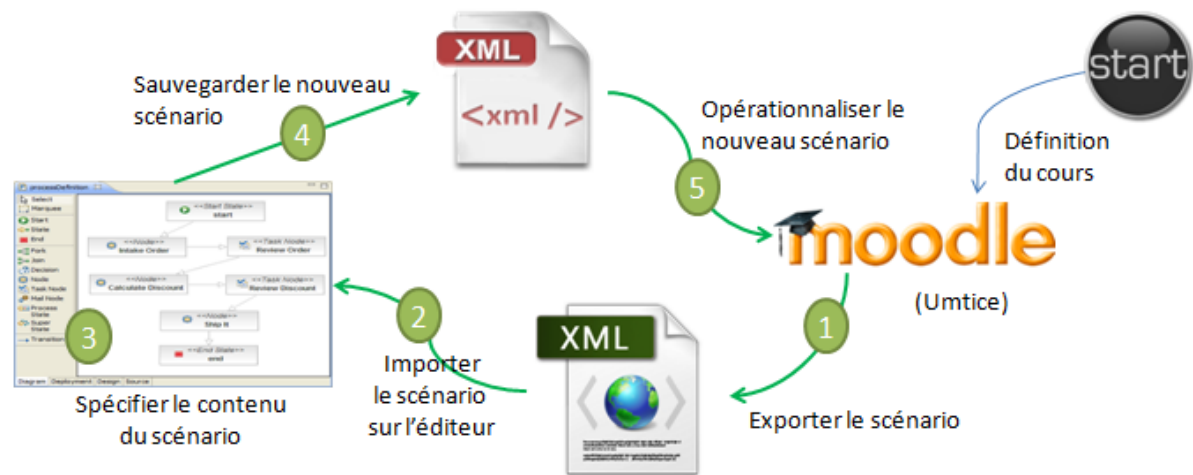


Figure 87 : Processus de conception et d'opérationnalisation d'un scénario sur Moodle

L'enseignant-concepteur se charge d'importer ce fichier XML sur l'éditeur (étape 2, figure 87) qui s'occupe de l'initialisation des éléments du cours (les concepts définis au sein du fichier XML). Lorsqu'il s'agit d'un nouveau cours, les sections sont initialement vides. Les relations entre elles sont définies en respectant leur ordre au sein du fichier XML. L'enseignant-concepteur spécifie ensuite le contenu du cours (étape 3, figure 87) en choisissant les activités, les services et les ressources adéquats pour chaque partie. Puis, il enregistre le cours soit au sein du même fichier soit au sein d'un nouveau fichier XML (étape 4, figure 87). Enfin, il utilise de nouveau le module développé pour importer le nouveau cours conçu sur la plateforme Moodle (étape 5, figure 87). Le processus se charge de créer implicitement le paquetage de cours requis par la fonction *restore* qui se charge de l'opérationnalisation.

```

<?xml version="1.0"?>
<moodle xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="moodle.xsd">
  <course id="16">
    <shortname>PW-PHP</shortname>
    <fullname>Programmation Web en PHP</fullname>
    <summary>Cours du php : lanagage de programmation</summary>
    <sections>
      <section id="1159">
        <name></name>
        <summary></summary>
        <activities>
          <forum id="401">
            <type>news</type>
            <name>News forum</name>
            <intro>General news and announcements</intro>
            <discussions/>
          </forum>
        </activities>
      </section>
      <section id="1160">
        <name></name>
        <summary></summary>
        <activities/>
      </section>
      <section id="1161">

```

Figure 88 : Exemple de cours initial sur Moodle et le fichier XML associé

6.3.2.1. Conception du scénario sur l'éditeur de Moodle

Dans cette section, nous allons décrire la conception du scénario, précédemment décrit, sur l'éditeur de Moodle. Il s'agit de la troisième étape du processus de conception et d'opérationnalisation (figure 87) du scénario. Le scénario dispose de trois séquences dont chacune correspond à une section sur Moodle.

Quant à la première séquence, nous allons décrire son objectif soit via la ressource étiquette (*label*), soit via la définition de l'attribut *introduction* de la section. Ensuite, nous allons utiliser, par exemple, l'élément *page* pour la description des sujets des exercices. Nous tenons en compte des ressources pédagogiques mises à disposition des apprenants tels que les documents de cours et de syntaxes de base de ce langage. Enfin, l'enseignant doit spécifier les outils de communication prévus pour cette section. Il peut utiliser les activités *forum* ou bien *chat*.

Quant à la deuxième séquence, l'enseignant peut spécifier son objectif avant de s'intéresser à son contenu. Comme pour la première séquence, il peut utiliser soit l'attribut *introduction* de la section, soit la ressource *étiquette (label)*. Ensuite, il crée les groupes d'étudiants. Au cours de cette étape, il définit les groupes sans attribuer leurs membres. Puis, il décrit le travail demandé, en particulier les détails de développement de l'application web, aux groupes en utilisant les activités *devoir* ou *atelier* ou les ressources *page* ou *fichier* ou *url*. Enfin, il définit les moyens de communication à mettre à disposition des groupes pour échanger des idées et discuter des difficultés de développement. Dans ce cas, il peut utiliser le *forum* ou le *wiki*.

Quant à la troisième séquence, l'enseignant doit spécifier un test d'évaluation. Il peut nommer la section "évaluation" et présenter certaines consignes telles que « lisez attentivement l'énoncé, il est conseillé de répondre aux questions dans l'ordre préconisé mais vous pouvez les faire dans l'ordre de votre choix, etc. ». Il choisit l'activité *test* et spécifie ses attributs.

Les deux figures 89 et 90 illustrent respectivement un aperçu du cours spécifié sur l'éditeur et un extrait du fichier XML associé.

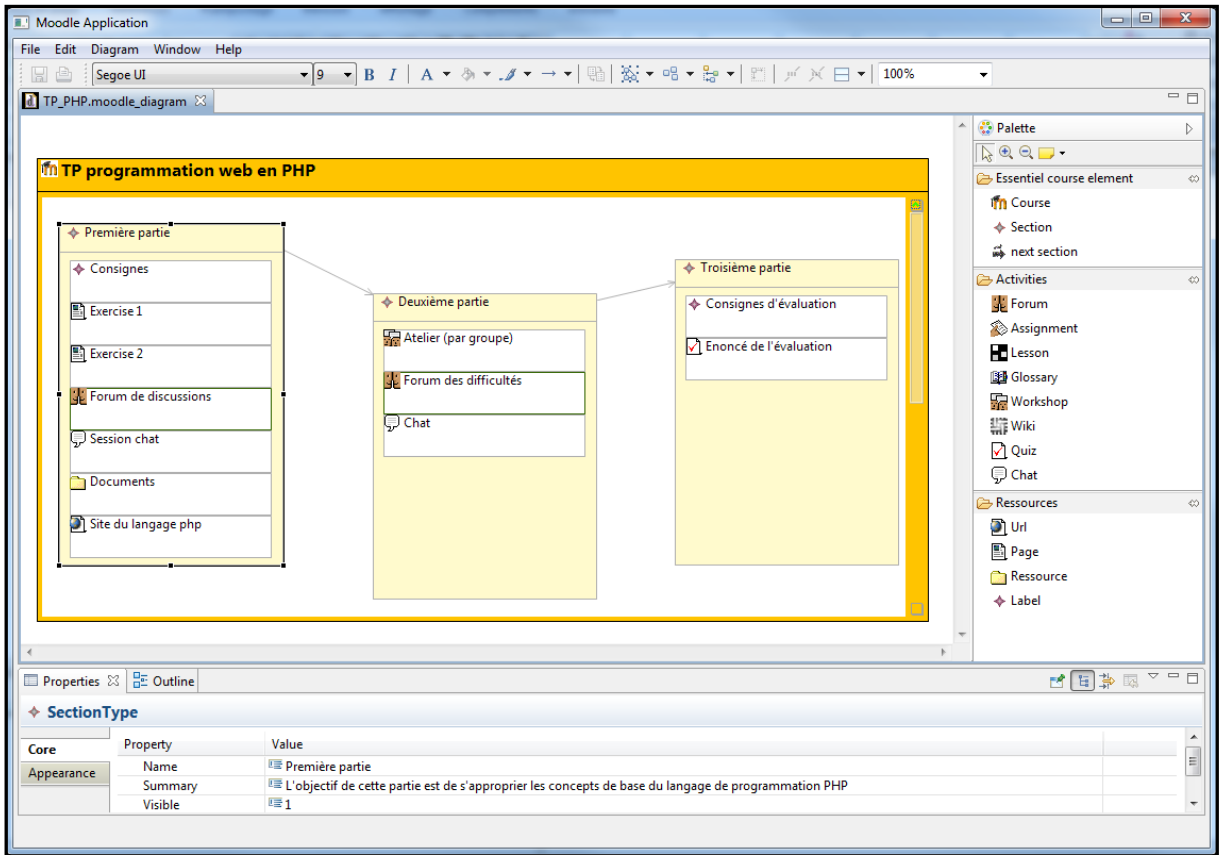


Figure 89 : Conception du scénario sur Mooditor


```

<?xml version="1.0" encoding="UTF-8"?>
<moodle xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="moodle.xsd">
  <course>
    <fullname>TP programmation web en PHP</fullname>
    <shortname>PW-PHP</shortname>
    <summary>TP d'initiation à la programmation web en PHP + évaluation</summary>
    <sections>
      <section>
        <name>Première partie</name>
        <summary>L'objectif de cette partie est de s'approprier
        les concepts de base du langage de programmation PHP</summary>
        <visible>1</visible>
        <activities>
          <label id="0">
            <name>Consignes</name>
            <introduction>Cette partie est composée de deux exercices à réaliser individuellement.
            Vous avez à disposition plusieurs documents (cours et autres)</introduction>
            <visible>1</visible>
          </label>
          <page id="0">
            <name>Exercice 1</name>
            <introduction>L'objectif est de manipuler les données envoyées via
            les formulaires</introduction>
            <content>Créer un script en PHP pour récupérer des données envoyées via un formulaire.
            Puis, ces données seront affichées au sein d'une nouvelle page.</content>
            <visible>1</visible>
          </page>
          <page id="0">
            <name>Exercice 2</name>
            <introduction>Gestion des données d'une base de données</introduction>
            <content>Créer un script php permettant d'insérer et d'afficher des
            données dans une base de données</content>
            <visible>1</visible>
          </page>
          <forum id="0">
            <name>Forum de discussions</name>
            <introduction>Si vous rencontrez des difficultés, vous pouvez les poster
            sur le forum pour avoir de l'aide</introduction>
            <type>Optional subscription</type>
            <groupmode>
            <visible>1</visible>
          </forum>
          <chat id="0">
            ...
          <ressource id="0">
            ...
          <url id="0">
            ...
          </activities>
        </section>
        <section id="0">
          <name>Deuxième partie</name>
          ...
          <workshop id="0">
            <name>Atelier (par groupe)</name>
            ...
          <chat id="0">
            <...
          <forum id="0">
            ...
          </section>
        <section id="0">
          <name>Troisième partie</name>
          ...
        </sections>
      </course>
    </moodle>

```

Figure 90 : Extrait du fichier XML du cours conçu

6.3.2.2. Opérationnalisation du scénario sur Moodle

Une fois la conception du scénario est terminée, il est enregistré dans un fichier externe en format XML. L'opérationnalisation du scénario est achevée par son import via le module de communication intégré au sein de l'espace de cours sur Moodle. La figure 91 illustre le résultat de l'import du cours.

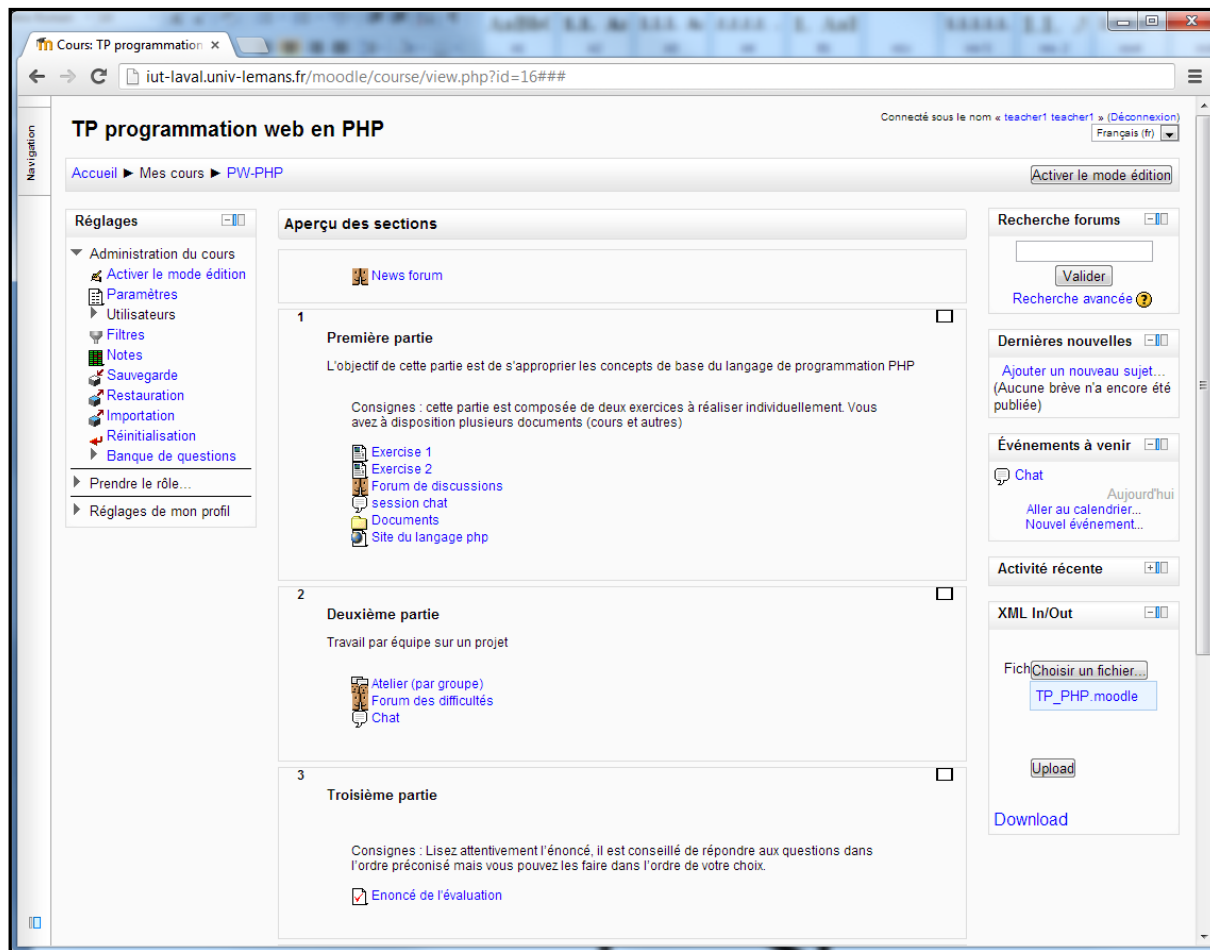


Figure 91 : Aperçu du scénario opérationnalisé sur Moodle

Bien que la conception initiale soit réalisée sur l'éditeur graphique, il est souvent nécessaire de continuer la spécification de certains paramètres/éléments directement sur l'interface de la plateforme. Par exemple, l'enseignant doit effectuer le dépôt concret des ressources et des documents. La gestion du temps n'est pas prise en compte au cours de la conception des éléments du scénario sur l'éditeur graphique. L'enseignant peut ainsi ajuster le temps associé aux activités concernées directement sur leurs interfaces.

6.3.3. Mise à l'essai : cas de la plateforme Ganesha

La deuxième mise à l'essai concerne l'expérimentation des outils que nous avons développés pour la conception et l'opérationnalisation du scénario exemple. Nous allons utiliser l'éditeur graphique développé pour concevoir ce scénario puis le module d'import/export développé pour l'opérationnaliser. Contrairement au cas de la plateforme Moodle, l'enseignant-concepteur peut utiliser directement l'éditeur graphique spécifique à la plateforme Ganesha pour concevoir des nouveaux scénarios péda-

gogiques (appelé aussi modules). Lorsque l'enseignant veut modifier un module déjà existant sur Ganesha, il doit ainsi l'exporter de la plateforme puis l'importer sur l'éditeur.

6.3.3.1. Conception du scénario sur l'éditeur de Ganesha

Cette section est consacrée à la description de la conception du scénario sur l'éditeur de Ganesha. L'enseignant concepteur commence par créer un nouvel espace de conception. Ensuite, il s'intéresse à la spécification du contenu de son scénario/module. Il doit ajouter un élément "module" et spécifier ses paramètres, en particulier le nom (requis) et l'objectif (facultatif). Comme il est textuellement décrit, le scénario est composé de trois séquences. Ainsi, l'enseignant peut ajouter trois séquences dans l'espace de design. Puis, il s'intéresse à la spécification de leur contenu et de leurs paramètres.

Quant à la première séquence, nous allons décrire son objectif par la spécification des attributs des séquences (nom et objectifs pédagogiques). Ensuite, nous allons utiliser les activités de type titre pour décrire les sujets des exercices. Nous spécifions les ressources pédagogiques mises à disposition des apprenants (le sujet des exercices, les documents de cours et de syntaxes de base de ce langage) en utilisant des activités de type document (ASSET).

Quant à la deuxième séquence, l'enseignant définit l'objectif de cette séquence via la spécification de leurs attributs (nom et objectifs pédagogiques). Ensuite, il doit décrire le travail demandé aux groupes d'étudiants en utilisant les activités dites collaboratives. Nous remarquons ici que le langage de conception de Ganesha, spécifique au profil intégrateur, ne propose pas de moyens de communication ni de la répartition des usagers du scénario en groupes. Ces éléments seront spécifiés directement sur la plateforme par un autre profil tels que celui de l'administrateur.

Quant à la troisième séquence, l'enseignant doit spécifier des évaluations aux apprenants. Tout d'abord, il peut définir les attributs de la séquence : le titre tel que "évaluation" et l'attribut objectifs pédagogiques tels que les consignes de l'évaluation (lire attentivement l'énoncé, répondre aux questions que vous connaissez, etc.). Ensuite, il choisit l'activité quiz pour spécifier le test. Il doit spécifier une série de questions avant de l'associer au quiz.

Les figures suivantes illustrent respectivement un aperçu du scénario spécifié sur l'éditeur ainsi qu'un extrait du fichier XML associé.

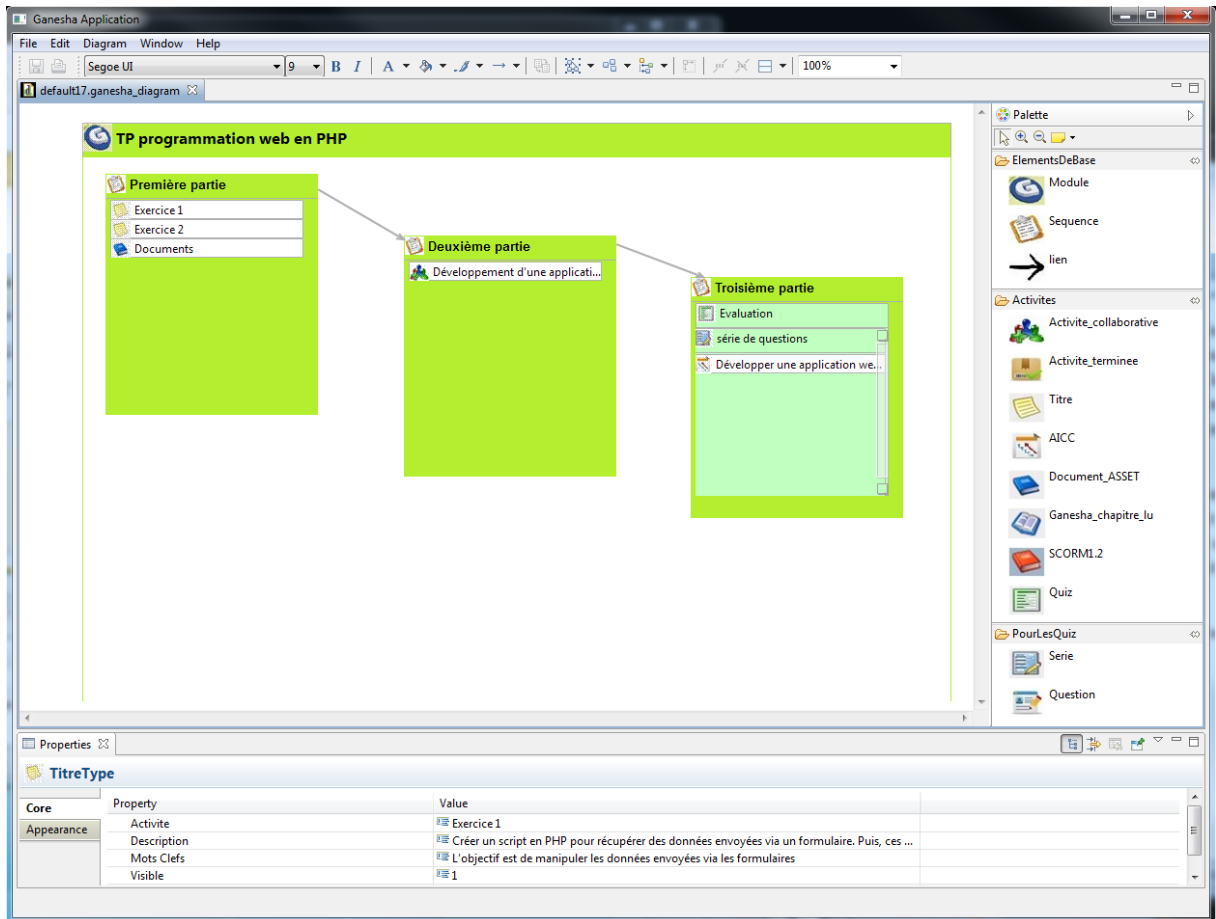


Figure 92 : Conception du scénario sur l'éditeur de Ganesha

```
<?xml version="1.0" encoding="UTF-8"?>
<ganesha>
  <Module>
    <id>0</id>
    <Titre>TP programmation web en PHP</Titre>
    <Sequence>
      <id>0</id>
      <TitreSequence>Première partie</TitreSequence>
      <ObjectifsPedagogiques>L'objectif de cette partie est de s'approprier
      les concepts de base du langage de programmation PHP</ObjectifsPedagogiques>
      <Titre>
        <id>0</id>
        <Activite>Exercice 1</Activite>
        <Description>Créer un script en PHP pour récupérer des données envoyées via un formulaire.
        Puis, ces données seront affichées au sein d'une nouvelle page.</Description>
        <MotsClefs>L'objectif est de manipuler les données envoyées via les formulaires</MotsClefs>
        <Visible>1</Visible>
      </Titre>
      <Titre>
        <id>0</id>
        <Activite>Exercice 2</Activite>
        <Description>Créer un script php permettant d'insérer et d'afficher des données dans une base de données</Description>
        <MotsClefs>Gestion des données d'une base de données</MotsClefs>
        <Visible>1</Visible>
      </Titre>
      <DocumentASSET>
        <Activite>Documents</Activite>
        <Description>Ci-joint des supports de cours et des tutoriaux du langage PHP</Description>
        <id>0</id>
        <Visible>1</Visible>
      </DocumentASSET>
    </Sequence>
    <Sequence>
      <id>0</id>
      <TitreSequence>Deuxième partie</TitreSequence>
```

Figure 93 : Extrait du fichier XML du scénario conçu

6.3.3.2. Opérationnalisation du scénario sur Ganesha

Après la conception du scénario sur l'éditeur, l'enseignant-concepteur se charge de l'opérationnalisation qui consiste à importer le fichier externe en format XML via le module de communication intégré au sein de l'espace de cours sur Ganesha. Toutefois, la conception peut se continuer sur les interfaces de la plateforme. Par exemple, la spécification des outils de communication (tels que forum ou chat) et des groupes est réalisée directement sur la plateforme et nécessite les droits différents à ceux d'un intégrateur. Les figures suivantes illustrent des extraits d'interfaces résultant de l'opérationnalisation du module sur la plateforme Ganesha.



Figure 94 : Ecran d'import du module sur Ganesha



Figure 95 : Ecran du module importé



Figure 96 : Ecran d'une séquence du module



Figure 97 : Ecran d'une activité (quiz) du module

6.4. Conclusion

Nous avons présenté dans ce chapitre l'ensemble de l'outillage que nous avons développé afin d'instrumenter nos propositions théoriques. Nous parlons ici des outils mis à disposition des enseignants-concepteurs utilisant les plateformes Moodle et Ganesha pour concevoir et opérationnaliser des scénarios pédagogiques. Ces outils se sont basés sur des modèles et des schémas très spécifiques correspondant aux métamodèles de conception pédagogique de ces plateformes, formalisés par l'intermédiaire du processus que nous avons proposé (cf. section 4 et 5).

Nous avons exploité ces métamodèles pour le développement des modules de communication. Ces modules intégrés au sein de l'espace cours de plateformes permettent les enseignants-concepteurs d'opérationnaliser et d'exporter des scénarios pédagogiques conformes au schéma XML équivalent aux métamodèles des plateformes utilisées.

Nous avons utilisé deux frameworks spécifiques (EMF et GMF) pour le développement des outils de conception graphiques (des éditeurs). Ces frameworks reposant sur plusieurs modèles ont facilité la génération automatique d'une partie importante du code de ces outils. Le modèle du domaine représente un des modèles primordiaux de ces frameworks. Il s'appuie sur le métamodèle de conception pédagogique de ces plateformes après avoir mener des modifications permettant de prendre en considération de certaines exigences de praticiens et de design. Nous avons également mené des modifications du code afin de satisfaire ces exigences et rendre l'outil de conception capable d'importer les modèles qui sont conformes au métamodèle de conception pédagogique.

Nous avons également présenté dans ce chapitre deux mises à l'essai des outils développés. Nous avons montré que les outils développés facilitent la conception pédagogique des scénarios pédagogiques et résolvent les difficultés d'opérationnalisation et des pertes d'information et de sémantique pendant cette étape. Toutefois, la conception peut être continue sur les interfaces des plateformes.

III

Conclusion générale

7

Conclusions et Perspectives

Sommaire

- 7.1. Bilan des travaux réalisés
- 7.2. Analyse critique des travaux réalisés
- 7.3. Perspectives
- 7.4. Conclusion

7.1. Bilan des travaux réalisés

Dans cette thèse, nous nous sommes intéressés à deux activités principales de la conception pédagogique. Il s'agit de la conception des scénarios pédagogiques et de leur opérationnalisation sur les plateformes de formation à distance. Le travail présenté dans ce mémoire est une contribution s'ajoutant aux différents travaux traitant ces deux activités. L'objectif de notre travail est de définir un cadre de réingénierie pour permettre à des enseignants-concepteurs et/ou praticiens de spécifier et d'opérationnaliser des situations d'apprentissage. Nous avons proposé une approche d'ingénierie et de réingénierie centrée plateformes ou systèmes LMSs. L'objectif de cette approche permet de faciliter et de garantir l'opérationnalisation des scénarios pédagogiques sans perte d'information ni de sémantiques sur les dispositifs d'apprentissage (plateformes et LMS).

Notre première proposition scientifique a consisté à définir une méthode spécifique pour identifier et formaliser le langage de conception pédagogique d'une plateforme. Ce langage englobe les activités, les ressources et les services mis à disposition des enseignants-concepteurs pour spécifier des situations d'apprentissage. Il englobe aussi d'autres éléments intervenant dans la conception de ces situations tels que les groupes d'apprenants, etc. Au cours de la définition de ce processus, nous avons pris en compte deux points de vue différents mais complémentaires. Le premier est centré sur le langage visible par l'utilisateur (enseignant-concepteur) que nous qualifions de langage centré IHM. Le deuxième est centré technique. Il appuie l'analyse sur des méthodes techno-centrées. Ce point est utile pour assurer que les futurs modèles (scénarios pédagogiques) conformes au langage formalisé seront complètement opérationnalisés. La dernière étape du processus (confrontation et formalisation du modèle final), propose un compromis entre les analyses IHM et technique. Elle permet la spécification du modèle final du langage de conception.

Après avoir identifié et formalisé ce langage, nous avons proposé des outils basés sur ce langage et facilitant l'opérationnalisation des scénarios pédagogiques spécifiés à l'extérieur de la plateforme. Pour cela, nous avons proposé d'exposer le langage métier de la plateforme sous un nouveau format spécifique d'import/export et d'étendre les plateformes par des nouveaux modules afin de les rendre capable d'importer/exporter des scénarios pédagogiques. D'un point de vue centré plateforme, l'opérationnalisation de ces scénarios est assurée par l'importation de leurs fichiers (XML). Ces scénarios doivent être formalisés conformément au schéma XML équivalent au métamodèle du langage de plateforme. Ces modules de communication ont permis d'assurer l'opérationnalisation sans perte de sémantique ni d'informations. Nous avons également pris en compte que ces modules permettent l'export des scénarios déjà existants sur la plateforme. Cela permet aux enseignants-concepteurs de mener des adaptations itératives de leurs scénarios ; ces derniers peuvent être mis à jour ou modifiés sur l'éditeur graphique et ensuite opérationnalisés de nouveau par le module d'import/export, sans perte des paramétrages/spécifications complémentaires réalisés sur la plateforme.

Nous avons également exploité ce langage de conception pédagogique de plateformes comme base (modèle métier ou du domaine) pour le développement de premiers outils de conception pédagogique extérieurs aux plateformes, en particulier des outils d'édition de scénarios. L'un des premiers objectifs de cette approche est d'offrir aux praticiens des VIDLs et des éditeurs externes dédiés basés sur le langage métier des LMS. L'objectif pour nos travaux consistait à expérimenter l'approche théorique

du DSM et les outils associés pour guider la spécification de ces VIDLs ainsi que le développement des éditeurs graphiques correspondants. Les scénarios spécifiés ou modifiés par ces outils sont totalement opérationnalisables parce qu'ils sont conformes au langage pédagogique rendu explicite de la plateforme. Par définition, il n'y a pas de pertes d'informations puisque aucune transformation de modèles n'est nécessaire. La sémantique des scénarios pédagogiques est préservée lors de l'opérationnalisation et également lors de l'exportation.

Nous avons expérimenté nos propositions théoriques sur deux environnements d'apprentissage. Il s'agit des deux plateformes de formation à distance Moodle et Ganesha, fréquemment utilisées au sein des établissements de formation. Nous avons ainsi développé pour chaque plateforme un module d'import/export et nous l'avons intégré au sein de l'espace de conception de cours. Nous avons également développé un éditeur graphique spécifique à chaque plateforme pour concevoir ou modifier des scénarios pédagogiques. Les outils développés ont été enfin expérimentés pour vérifier la faisabilité de notre approche.

7.2. Analyse critique des travaux réalisés

7.2.1. Apports de nos propositions

Par notre approche, nous voulions 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 :

- Le processus d'identification et de formalisation du langage de conception d'une plateforme que nous avons proposé permet de prendre en considération plusieurs points de vue complémentaires (IHM et technique). Ce processus peut être exploité à des fins de comparaison entre les plateformes sur la base de leur langage explicite.
- Contrairement à plusieurs éditeurs génériques, les éditeurs proposés permettent de travailler dans l'univers métier des dispositifs d'apprentissage, ce qui favorise leur appropriation et facilite aux enseignants la conception des scénarios pédagogiques. Ces éditeurs permettent de dépasser les contraintes et les détails techniques imposés par les interfaces de la plateforme. La conception est ainsi centrée sur les éléments que nous qualifions de pédagogiques.
- L'approche proposée permet aux enseignants de spécifier des scénarios pédagogiques avec un langage conforme à celui de la plateforme. De cette façon, les enseignants peuvent plus facilement surmonter les barrières conceptuelles et techniques posées par les interfaces des plateformes de formation. Ces interfaces sont basées sur des formulaires à longue liste d'attributs dont plusieurs entre eux sont des détails techniques à bas niveau. L'apport de cette approche est à deux niveaux :
 - certains paramètres (techniques) ont été écartés
 - avec le DSM, l'approche éditeurs/VIDL permet de jouer sur la représentation graphique pour réduire la surcharge cognitive de l'utilisateur lorsqu'il manipule les éléments.

- Les VIDLs que nous avons spécifiés et leurs éditeurs graphiques ont représenté de nouveaux terrains d'expérimentation de l'Ingénierie Dirigée par les Modèles, en particulier le DSM, dans le domaine de la conception pédagogique. Notre contribution est une application des méthodes et des techniques de l'IDM/DSM. Nous avons utilisé et expérimenté des frameworks spécifiques de DSM pour instrumenter nos propositions.
- Contrairement aux approches traditionnelles de la littérature impliquées dans notre problématique (telles que [Caron 2007], [Abdallah 2009], [de-Moura 2007], etc.) et portant sur des solutions techniques d'opérationnalisation partielle des scénarios pédagogiques, notre approche assure l'opérationnalisation de toutes les informations/données de ces scénarios sans perte sémantique.
- Les scénarios spécifiés par le moyen des éditeurs graphiques sont spécifiés dans un format externe (XML) ce qui ouvre le champ pour leur opérationnalisation sur d'autres plateformes après avoir effectué les transformations nécessaires.
- Le métamodèle de conception pédagogique, issu de l'application du processus sur une plateforme de formation spécifique, peut être utilisé dans le cadre d'autres travaux de recherche. Par exemple, [Laabidi 2013] a utilisé le métamodèle de Moodle pour intégrer des modèles d'accessibilité au sein de cette plateforme. Dans une approche IDM, le métamodèle de la plateforme désigne le modèle PDM (*Platform Description Model*) pour permettre la transformation du modèle PIM à celui du PSM.

7.2.2. Limites de notre travail

Le langage de conception pédagogique identifié dépend fortement de la vision de l'enseignant-concepteur et de l'analyste. Certains éléments peuvent être considérés techniques pour un enseignant et pédagogiques pour d'autres. Pour cela, nous recommandons que l'application du processus d'identification de ce langage soit réalisée en collaboration entre ces deux acteurs. Plusieurs paramètres et éléments peuvent être identifiés ou bien écartés. L'analyse IHM requiert une bonne interprétation des composants de l'interface graphique. Certains cas particuliers de cette analyse nécessitent la prise de décisions subjectives.

L'analyse technique s'avère un peu complexe. Plusieurs champs de la base de données sont représentés par des codes ou des libellés spécifiques (difficilement interprétables). Pour cela, il est souvent nécessaire d'analyser le code source associé à ces données/champs. Dans nos propositions, nous avons choisi de représenter le modèle technique par le format du modèle conceptuel de données (de la méthode Merise). Ce format permet la prise en compte de plusieurs détails non supportés par d'autres modèles (tels que le modèle logique de données) et de voiler les détails techniques liés à la base de données. Cependant, ce modèle risque d'être complexe à générer, en particulier pour les plateformes de taille importante. La confrontation entre le modèle IHM et le modèle technique peut être aussi complexe parce que certains attributs et/ou éléments peuvent être représentés sur les deux modèles de façon différente en termes de nomination, de définition, etc. Les éléments du modèle technique dépendent de l'architecture de la base de données tandis que ceux du modèle IHM dépendent de la structuration des informations sur les IHMs. L'élaboration du métamodèle peut nécessiter une description plus détaillée des possibilités d'expressivité en termes de type de relations entre les classes.

Notre approche est centrée plateforme. Le langage de spécification des scénarios pédagogiques est limité à celui proposé par la plateforme. Cela peut limiter l'expressivité pédagogique de ces scénarios parce que le modèle métier de l'enseignant ne fait pas partie du périmètre de notre travail. Nous avons spécifié des VIDLs centrés LMSs illustrant l'exploitation du métier pédagogique identifié. Leur avantage majeur est d'écarter les détails techniques non impliqués dans la conception pédagogique. Cependant, nous nous sommes limités à des premiers VIDLs afin d'étudier à un premier niveau, nécessaire, s'il était possible d'exploiter le métamodèle de la plateforme obtenu. Nous n'avons pas apporté de réelle valeur ajoutée à la conception intra-plateforme. Les interfaces graphiques des éditeurs proposent une notation graphique similaire à celle visible sur la plateforme mais en excluant les détails non pédagogiques. Notre objectif est de vérifier la faisabilité d'exploiter le métier de conception pédagogique identifié. Nous démontrons dans la section suivante qu'une de nos futures perspectives est de proposer des VIDLs plus adaptés aux praticiens, tant par la syntaxe concrète graphique mais également sémantique.

L'application de l'approche en générale sur les deux plateformes d'essai Moodle et Ganesha nous a permis de l'expérimenter dans un contexte réel, qui reste toutefois réduit. La mise à l'essai du processus d'identification et formalisation du langage de conception pédagogique sur ces plateformes doit encore être testée/expérimentée par des analystes et/ou des experts de plateformes.

La mise à l'essai des outils développés (les modules de communication et les éditeurs graphiques) a été également expérimentée à une échelle réduite. Nous avons utilisé les éditeurs graphiques pour concevoir des scénarios pédagogiques exemples, initialement sous un format textuel. L'objectif est de tester ces outils et de vérifier leur valeur ajoutée de point de vue conception pédagogique.

En ce qui concerne les modules d'import/export, ils dépendent de certains points tels que la version des plateformes, leur métier de conception pédagogique, la définition du schéma XML de ce métier, etc. Pour prendre en considération l'évolution du métier de la plateforme, ces modules devront être mis à jour par l'ajout ou la modification des règles d'opérationnalisation des nouveaux concepts. Dans le cas de la plateforme Moodle, le module développé dépend des fonctionnalités existantes *backup* et *restore*. L'évolution de ces fonctionnalités et du paquetage de sauvegarde exige de nouvelles modifications pour maintenir le fonctionnement du module d'import/export.

Toutefois, nous n'avons pas pu expérimenter les outils développés avec des enseignants-concepteurs ou des praticiens ni de valider nos propositions scientifiques à une échelle plus large et diversifiée. En effet, nous avons proposé par ce travail une approche nouvelle, mais il faut qu'elle soit utilisée pour valider son utilité et sa valeur ajoutée. Nous avons démontré la faisabilité de la mise 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.

7.3. Perspectives

7.3.1. Exploitation du métier identifié pour des VIDLs centrés LMSs et praticiens et pour l'interopérabilité entre les plateformes

Dans un cadre théorique et pratique du DSM, nous avons proposé une approche centrée plateformes. Toutefois, des expérimentations supplémentaires de nos propositions sont nécessaires. Notre proposition de conception doit être plus adaptée à une conception pédagogique spécifique qui tient compte des besoins et des pratiques d'enseignants-concepteurs plus complexes et plus abstraits (par rapport à certaines approches pédagogiques ou des théories éducatives). Il s'agit de dépasser les limites inhérentes à l'expressivité pédagogique concrète à la plateforme pour proposer des mécanismes de scénarisation plus proches de l'expressivité des pratiques de conception existantes ou souhaitées [Loiseau et Laforcade 2013]. Les scénarios doivent être traduits et situés dans un niveau d'abstraction compatible avec le langage de conception de la plateforme. Il est ainsi important d'identifier une relation forte entre l'expressivité pédagogique (les pratiques et les besoins de conception des enseignants) et la maîtrise de l'opérationnalisation (le paradigme et langage de conception pédagogique des plateformes utilisées).

Dans cette perspective, le projet GraphiT [Graphit 2013], débuté en février 2012 et mis en place par notre équipe Ingénierie des EIAH du LIUM, a pour but d'étudier cette question de recherche. Le projet se situe dans un cadre méthodologique mixte basé sur l'ingénierie des besoins des enseignants-utilisateurs de plateformes et sur les patrons de conception et de langages. Ce cadre est exploité pour la capture et l'explicitation des pratiques de conception et du métier de conception des plateformes ainsi que son exploitation pour la spécification de langages/éditeurs de scénarisation. La figure 98 illustre les contributions visées par le projet Graphit. Le cadre rouge (figure 98) illustre le périmètre des travaux de cette thèse qui seront prolongés à d'autres plateformes.

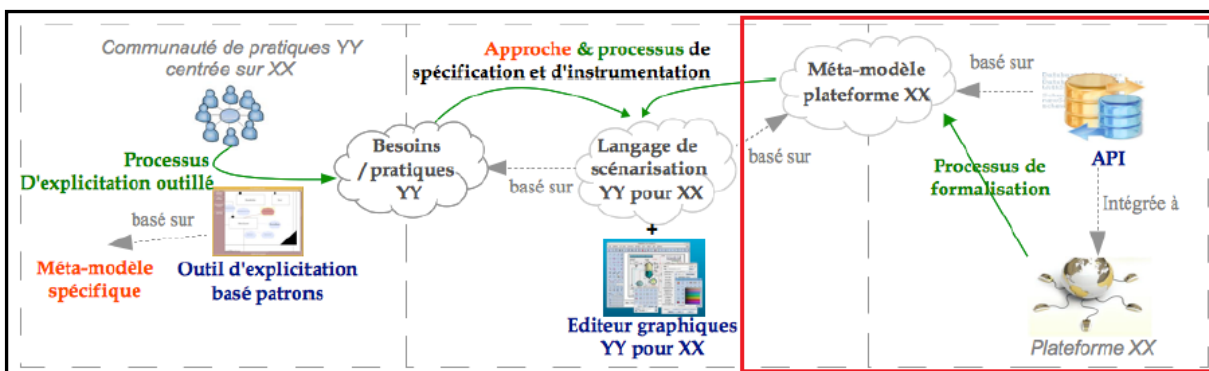


Figure 98 : Contributions visées par le projet GraphiT (outils en bleu, processus/méthodes en vert, modèles/techniques) [Loiseau et Laforcade 2013]

Dans le même contexte, les solutions actuelles de spécification des VIDLs centrés LMSs et orientés praticiens doivent être expérimentées afin de réduire l'écart de sémantique entre les langages de conception pédagogique et les plateformes de formation. L'utilisabilité et l'ergonomie des outils développés doivent être expérimentées avec les praticiens. Les VIDLs devront se focaliser sur des pratiques de conception spécifiques et sur un éventuel binding des scénarios produits. Les scénarios doivent être

sérialisés dans des formats propriétaires destinés à une interprétation machine pour une opérationnalisation sur des éventuels dispositifs.

Dans le cadre du projet Graphit, les langages de scénarisation graphiques visés se situent dans un niveau intermédiaire entre les VIDL à niveau spécification (conception encadrée par les pratiques de conception des enseignants) et ceux à niveau implémentation (conception limitée à ce que la plateforme permet). Il s'agit ainsi d'étudier jusqu'à quel niveau d'expressivité visuelle et sémantique proche des pratiques de conception des enseignants il est possible de s'élever sur la base d'un formalisme et d'une sémantique spécifiques à une plateforme.

Afin de spécifier des métamodèles / VIDLs à différents niveaux progressifs d'abstraction du métier de la plateforme, nous prévoyons d'étudier les théories et les techniques de l'Ingénierie Dirigée par les Modèles. En effet, les métamodèles de tissage et de composition semblent utiles pour la spécification des liens de sémantique entre les métamodèles qui seront ensuite utilisés pour la génération de règles déclaratives pour la transformation de modèles. Ce thème de recherche se situe à l'intersection de l'IDM et du domaine de conception pédagogique et n'a jamais été traité à ce jour. Les verrous suivants devront également être abordés afin d'atteindre nos objectifs :

- les limites d'expressivité des métamodèles (pour formaliser les syntaxes abstraites de VIDLs) et des notations visuelles (pour les syntaxes concrètes de VIDLs) spécifiées au-dessus de l'expressivité de la sémantique interne aux LMSs.
- les limites de la traduction automatique de modèles d'un VIDL à un autre sans perte d'informations ni de sémantique.
- les limites des possibilités de relations et d'expressivité entre deux VIDLs.
- le coût d'investissement et l'effort de programmation requis lorsque l'approche outillée DSM est suivie.
- l'utilisabilité, l'ergonomie (IHM) et d'autres aspects de validation des propositions par la communauté cible.

Dans la même perspective de travail, nous pensons qu'il est important d'étudier la possibilité d'assurer l'interopérabilité entre des plateformes de formation à distance pour lesquelles nos propositions auraient être réalisées (métamodèle de la plateforme et module d'import/export). Cette perspective requiert la définition des règles de transformation entre les métamodèles du langage des plateformes concernées afin de permettre l'opérationnalisation des modèles ou scénarios pédagogiques sur ces dispositifs. L'interopérabilité sera centrée sur l'opérationnalisation des scénarios transformés par le biais des modules d'import/export ajoutés aux plateformes. Toutefois, cette perspective nécessite l'étude des pertes pouvant être engendrées par ces opérations. Les pertes risquent d'être majeures lorsque les deux plateformes disposent de langage de conception pédagogique non similaire. La figure 99 illustre le principe d'interopérabilité entre deux plateformes exemples (Moodle et Ganesha) par la transformation entre les scénarios pédagogiques.

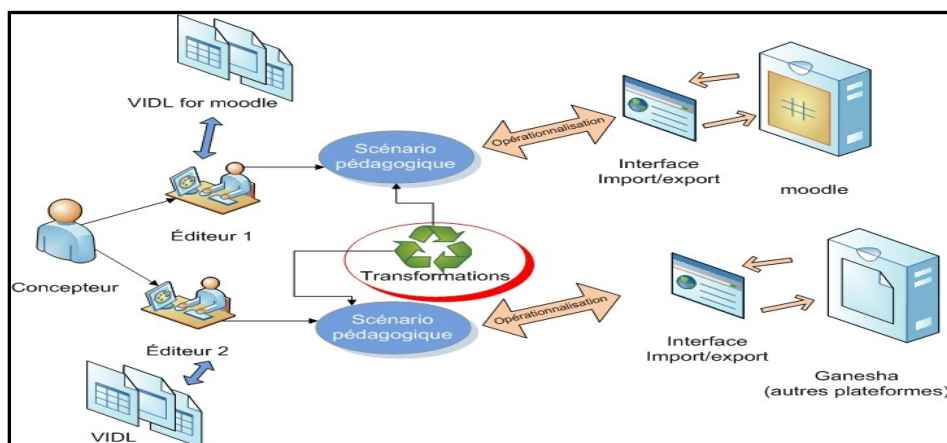


Figure 99 : Interopérabilité entre les plateformes par transformation entre les modèles

7.3.2. Extension du métier identifié par le processus

Ayant étudié la faisabilité d'opérationnaliser des scénarios pédagogiques spécifiés en dehors des plateformes, il est souvent important de revoir les détails fins qui sont déjà écartés (qui peuvent être considérés pédagogiques dans certains contextes) pendant l'identification du métier de conception pédagogique ainsi que d'autres aspects tels que le dépôt concret des documents et des ressources pédagogiques.

Nous avons déjà mentionné au sein du quatrième chapitre, plus spécifiquement dans la présentation du processus d'identification et de formalisation du langage conception pédagogique d'une plateforme, que le périmètre de notre travail ne prend pas en compte les traces d'exécution des scénarios pédagogiques. Cette hypothèse est due à notre intérêt principal portant sur la facilitation de l'opérationnalisation des scénarios produits à l'extérieur des plateformes. Cependant, les traces peuvent être utiles pour extraire des informations permettant d'améliorer les situations d'apprentissage spécifiées et exécutées. Dans ses travaux au LIUM, [Zendagui 2010] s'est intéressé à l'explicitation des besoins d'observation sur la base des scénarios. Les observations sont effectuées dans le double but de vérifier la cohérence de l'utilisation effective avec celle qui était prévue et de pouvoir détecter des utilisations émergentes afin d'améliorer la qualité des scénarios pédagogiques initiaux (conception continue).

Il nous semble ainsi intéressant d'étendre le processus pour capturer d'autres métiers de la plateforme liés à la conception pédagogique en général, et non spécifiquement à la scénarisation, comme par exemple le métier de l'observation (ce que la plateforme est capable de tracer, les moyens d'observation au niveau des traces, etc.). L'observation peut aider les enseignants-concepteurs à comprendre les écarts entre la conception qu'ils ont définie de la situation d'apprentissage et son déroulement réel et à apporter les améliorations nécessaires. Les informations liées à l'observation ainsi que les traces peuvent ainsi être identifiées et formalisées lorsqu'elles sont proposées par la plateforme. Cela ouvre le champ à la conception de l'observation en relation avec la conception pédagogique et selon une approche techno-centrée. Sur la base du nouveau métier identifié et formalisé, il serait alors possible de spécifier et de développer des langages/outils dédiés.

Dans cette même perspective, il nous semble aussi intéressant d'étudier la faisabilité d'intégrer au sein du processus d'autres aspects pour tenir compte d'autres langages utilisés pour la manipulation des traces et d'observations tels que le langage UTL (*Usage Tracking Language*). Ce langage permet de définir des indicateurs sous une forme proche des patrons de conception. Son intégration au sein du processus peut présenter une piste de recherche pour identifier les éléments pertinents spécifiques à la conception pédagogique ainsi que les éléments qui représentent des difficultés pour les usagers des situations d'apprentissage, en particulier les apprenants.

Table des figures

Figure 1 : Les principales activités de la conception pédagogique.....	27
Figure 2 : Processus itératif de conception d'un scénario pédagogique [Vantroys 2003].....	29
Figure 3 : Cycle de vie simplifié d'un scénario pédagogique	30
Figure 4 : Vue simplifiée du modèle EML [Pernin 2003].....	32
Figure 5 : Classification des exemples de VIDLs [Laforcade 2010].....	33
Figure 6 : Cycle de transformations appliqué en EIAH [Nodenot et al. 2005]	36
Figure 7 : Implémentation d'un logiciel à partir des termes de son domaine [Kelly et Tolvanen 2008].....	37
Figure 8 : Modèle conceptuel du langage IMS-LD [Ferraris et Lejeune 2007]	40
Figure 9 : Métamodèle simplifié de LDL [Ferraris et Lejeune 2007]	42
Figure 10 : Processus de conception pédagogique de CADMOS [Boloudakis et al. 2012].....	43
Figure 11 : Exemple de modèle conceptuel de CADMOS [Boloudakis et al. 2012]	43
Figure 12 : Exemple de modèle de flux de CADMOS [Katsamani et Retalis 2011]	44
Figure 13 : Ecran de l'environnement auteur LAMS [Dalziel 2010].....	45
Figure 14 : Paquetage du métamodèle CPM et extraits du profil CPM [Laforcade 2004].....	47
Figure 15 : Editeur de scénarios à base de patrons pédagogiques [De Moura Filho 2007].....	48
Figure 16 : Approche de conception dans le projet Bricoles [Caron 2007a].....	49
Figure 17 : Opérationnalisation manuelle versus automatisée des scénarios pédagogiques	54
Figure 18 : Processus de conception SCENARI [Bachimont et al. 2002].....	55
Figure 19 : Processus de conception, opérationnalisation et exécution d'un scénario LDL [Martel et Vignollet 2007].....	57
Figure 20 : Implémentation des unités d'apprentissage dans <e-LD> [Ortiz et al. 2007]	58
Figure 21 : Vue générale de l'approche proposée.....	68
Figure 22 : Les VIDLs selon le niveau d'abstraction.....	70
Figure 23 : Cycle de conception itérative	71
Figure 24 : La méthodologie de recherche adoptée.....	72
Figure 25 : Processus global d'analyse du langage de conception pédagogique d'une plateforme	78
Figure 26 : Analyse centrée IHM	80
Figure 27 : Diagramme d'activité de l'analyse IHM-macro.....	83
Figure 28 : Interface d'ajout d'un forum sur Moodle	84
Figure 29 : Extrait du modèle IHM-macro de Moodle.....	85
Figure 30 : Extrait d'interface de cours sur Moodle	86
Figure 31 : Diagramme d'activité de l'analyse fonctionnelle.....	87
Figure 32 : Vue générale du modèle fonctionnel.....	88
Figure 33 : Interface du cours sur Moodle	89
Figure 34 : Extrait du modèle fonctionnel de l'interface de cours sur Moodle	90
Figure 35 : Diagramme d'activité de l'analyse IHM-micro	91
Figure 36 : Interface d'ajout d'un forum dans Moodle décomposée en trois zones.....	92
Figure 37 : Interface d'ajout d'un forum	93
Figure 38 : Widgets graphiques dans une interface.....	94
Figure 39 : Extrait du modèle IHM-micro d'un forum de Moodle	95
Figure 40 : Diagramme d'activité de la composition.....	96
Figure 41 : Diagramme d'activité de l'identification des multiplicités	98
Figure 42 : Représentation des multiplicités sur le modèle IHM	99
Figure 43 : Diagramme d'activité de l'analyse de la base de données	100

Figure 44 : Diagramme d'activité de la confrontation et formalisation.....	107
Figure 45 : Processus de vérification de l'existence d'un élément	109
Figure 46 : Processus de vérification d'un type d'un élément	110
Figure 47 : Processus de vérification des attributs d'un élément.....	112
Figure 48 : Processus de vérification des multiplicités	113
Figure 49 : Modèle IHM-macro de Moodle.....	117
Figure 50 : Extrait du modèle fonctionnel final	118
Figure 51 : Modèle IHM-micro d'un forum.....	120
Figure 52 : Vue simplifiée du modèle IHM	121
Figure 53 : Extrait du modèle conceptuel de donnée réduit de la plateforme Moodle.....	123
Figure 54 : Métamodèle de la plateforme Moodle réduit à la conception pédagogique	124
Figure 55 : Modèle IHM-macro de la plateforme Ganesha	126
Figure 56 : Extrait du modèle fonctionnel de Ganesha.....	127
Figure 57 : Modèle IHM-micro d'un quiz	128
Figure 58 : Modèle IHM de Ganesha.....	129
Figure 59 : Modèle conceptuel de donnée réduit de la plateforme Ganesha.....	130
Figure 60 : Métamodèle de conception pédagogique de la plateforme Ganesha	131
Figure 61 : Les fonctionnalités de sauvegarde et de restauration de Moodle.....	135
Figure 62 : Exemple d'archive Zip de la sauvegarde d'un cours	135
Figure 63 : Service d'opérationnalisation et d'export intégré à l'espace cours de Moodle.....	136
Figure 64 : Extrait du fichier importandexportofcourse.php.....	136
Figure 65 : Les instructions du fichier lang/en/block_importandexportofcourse.php.....	137
Figure 66 : Extrait du fichier version.php	137
Figure 67 : Extrait du schéma XML équivalent au métamodèle du langage de conception pédagogique de Moodle.....	138
Figure 68 : Processus d'export de cours sur Moodle	139
Figure 69 : Processus d'import de cours sur MOODLE	140
Figure 70 : Bloc import/export ajouté à Ganesha.....	141
Figure 71 : Extrait du schéma XML de Ganesha	142
Figure 72 : Processus d'opérationnalisation développé pour Ganesha.....	143
Figure 73 : Processus d'export développé pour Ganesha	144
Figure 74 : Processus EMF [Zendagui 2010].....	146
Figure 75 : Aperçu de l'éditeur généré par EMF.....	146
Figure 76 : Aperçu du processus de GMF [Laforcade 2010].....	148
Figure 77 : Architecture simplifiée de Mooditor.....	149
Figure 78 : Extrait du modèle de domaine final	150
Figure 79 : les modèles de définition de la couche graphique	151
Figure 80 : Modèle de définition d'outillage et la palette de l'éditeur.....	151
Figure 81 : Représentation graphique d'un forum.....	152
Figure 82 : représentation graphique du concept section	153
Figure 83 : Extrait du modèle mooditor.gmfmap.....	154
Figure 84 : Capture d'écran de Mooditor	156
Figure 85 : Extrait du nouveau code de la fonction save().....	158
Figure 86 : Extrait du nouveau code de la fonction load().....	158
Figure 87 : Processus de conception et d'opérationnalisation d'un scénario sur Moodle.....	161
Figure 88 : Exemple de cours initial sur Moodle et le fichier XML associé.....	161
Figure 89 : Conception du scénario sur Mooditor.....	163
Figure 90 : Extrait du fichier XML du cours conçu	164

Figure 91 : Aperçu du scénario opérationnalisé sur Moodle	165
Figure 92 : Conception du scénario sur l'éditeur de Ganesha	167
Figure 93 : Extrait du fichier XML du scénario conçu.....	167
Figure 94 : Ecran d'import du module sur Ganesha	168
Figure 95 : Ecran du module importé.....	168
Figure 96 : Ecran d'une séquence du module	169
Figure 97 : Ecran d'une activité (quiz) du module	169
Figure 98 : Contributions visées par le projet GraphiT (outils en bleu, processus/méthodes en vert, modèles/techniques) [Loiseau et Laforcade 2013]	178
Figure 99 : Interopérabilité entre les plateformes par transformation entre les modèles.....	180

Références bibliographiques

[Abd-Ali et El-Guemhioui 2005]

J. Abd-Ali et K. El Guemhioui (2005). Horizontal transformation of PSMs. In: proceedings of the First European conference on Model Driven Architecture: foundations and Applications, (ECMDA-FA'05), Nuremberg, Germany, 7-10 novembre 2005.

[Abdallah 2009]

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

[Abedmouleh 2010]

Abedmouleh A. (2010). Approche Domain Specific Modeling pour l'opérationnalisation de scénarios pédagogiques sur des plateformes de formation à distance. In: Rencontres jeunes chercheurs en EIAH, RJC 2010, Lyon, France, 6-7 mai 2010, p.141-142.

[Abedmouleh 2012]

Abedmouleh A. (2012). Formalisation du langage de conception pédagogique implicite d'un LMS : motivations et processus. In : Quatrièmes Rencontres Jeunes Chercheurs en EIAH, Amiens, France, 23-24 mai 2012, p. 3-9.

[Abedmouleh et al. 2012a]

Abedmouleh A., Laforcade P., Oubahssi L. et Choquet C. (2012). Identification of LMSs Instructional Languages: an Analysis Process. In: IEEE International Conference on Advanced Learning Technologies, Rome, Italie, 4-6 juillet 2012, p. 367-368.

[Abedmouleh et al. 2012b]

Abedmouleh A., Oubahssi L., Laforcade P. et Choquet C. (2012). Expressing the implicit instructional design language embedded in an LMS: motivations and process. In: Computers and Advanced Technology in Education, Naples, Italie, 25-27 juin 2012, p. 774-064.

[Abedmouleh et al. 2012c]

Abedmouleh A., Oubahssi L., Laforcade P. et Choquet C. (2012). An analysis process for identifying and formalizing LMS instructional language. In: International Conference on Software and Data Technologies, Rome, Italie, 24-27 juillet 2012, p. 218-223.

[Abedmouleh et al. 2012d]

Abedmouleh A., Laforcade P. et Oubahssi L.. (2012). Specification of visual instructional design languages dedicated to Learning Management Systems. In: International Conference on Software and Data Technologies, Rome, Italie, 24-27 juillet 2012, p. 199-204.

[Abedmouleh et al. 2011a]

Abedmouleh A., Laforcade P., Oubahssi L. et Choquet C. (2011). A Model Driven and external approach for learning design upon Learning Management System. In: proceedings of the 3rd International Conference on Computer Supported Education (CSEDU'2011), vol.1, isbn : 978-989-8425-49-2, Noordwijkerhout, Netherlands, 6-8 mai 2011, p.398-401.

[Abedmouleh et al. 2011b]

Abedmouleh A., Oubahssi L., Laforcade P. et Choquet C. (2011). Operationalization of Learning Scenarios on Open and Distance Learning platforms: the case of the Moodle Platform. In: the 11th IEEE International Conference on Advanced Learning Technologies (ICALT), 2011, isbn : 978-1-61284-209-7 , Athens, USA, 6-8 juillet 2011, p.396-398.

[Abedmouleh et al. 2011c]

Abedmouleh A., Laforcade P., Oubahssi L. et Choquet C. (2012). Operationalization of learning scenarios on existent learning management systems: the Moodle case-study. In: International Conference on Software and Data Technologies In Proceedings of the 6th International Conference on Software and Data Technologies, isbn : 978-989-8425-77-5, Seville, Espagne, 18-21 juillet 2011, p.143-148.

[Bachimont et al. 2002]

Bachimont B., Cailleau I., Crozat S., Majada M. et Spinelli S. (2002). Le procédé SCENARI : Une chaîne éditoriale pour la production de supports numériques de formation. In: Actes du colloque, Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie (TICE'2002), Lyon, France, 13-15 novembre, p. 183-192.

[Berger et Kam 1996]

Berger C. et Kam R. (1996). Definitions of instructional design. Page web consultée le 01 février 2013, <http://www.umich.edu/~ed626/define.html>.

[Bézivin 2001]

Bézivin J. (2001). From object composition to model transformation with the MDA. In: the 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS39), USA, Santa Barbara, 29 juillet - 3 août 2001, p. 350-354.

[Bezivin 2004]

Bezivin J. (2004). Sur les principes de base de l'ingénierie des modèles. In: L'OBJET: Software, Databases, Networks. vol.10, Octobre 2004, p. 145-156.

[Bézivin et al. 2004]

Bézivin J., Blay-Fornarino M., Bouzeghoub M., Estublier J. et Favre J.M. (2004). Rapport de synthèse de l'AS CNRS sur le MDA (Ingénierie Dirigée par les Modèles), novembre 2004.

[Bézivin et Blanc 2002]

Bezivin J. et Blanc X. (2002). MDA : Vers un important changement de paradigme en genie logiciel. Développeur référence, juillet 2002, en ligne : <http://www.devreference.net>. <http://pagesperso.lina.univ-nantes.fr/~vailly-a/Enseignement/Documents/MDA.Partie1.JBXB.Last.prn.pdf>.

[Boehm 2007]

Boehm B.W. (2007). Software design and structuring. In: Software engineering, lifetime contributions to software development, management and research, edition 9 Addison-Wesley, 13 mars 2010, 792 pages, ISBN : 978-0137035151.

[Boloudakis et al. 2012]

Boloudakis M., Katsamani M., Retalis S. et Georgiakakis P. (2012). CADMOS: A learning design tool for Moodle courses. In: First Moodle research conference, Heraklion, Crete-Greece, 14-15 septembre 2012, p. 25-32.

[Botturi et al. 2006]

Botturi L., Derntl M., Boot, E. et Figl K. (2006). A classification framework for educational modeling languages in instructional design. In: the 6th IEE International Conference on Advanced Learning Technologies (ICALT-06), Kerkrade, The Netherlands, 5-7 juillet 2006, p.1216-1220.

[Botturi et Stubbs 2007]

Botturi L. et Stubbs S. (2007). Handbook of Visual Languages for Instructional Design: Theories and Practices. Information Science Reference, 2007.

[Blay-Fornarino et Franchi 2005]

Blay-Fornarino M. et Franchi P. (2005). Espace Technologique Langages et IDM. In: Synthèse de l'AS CNRS sur l'Ingénierie Dirigée par les Modèles, décembre 2005.

[Boulet et al. 2003]

Boulet P., Dekeyser J.L., Dumoulin C. et Marquet P. (2003). Mda for SoC embedded systems design, intensive signal processing experiment. In: SIVOES-MDA workshop at UML 2003, San Francisco, United States, 20-24 october 2003, p. 166-174.

[Burgos et al. 2005]

Burgos D., Arnaud M., Neuhauser M. et Koper R. (2005). IMS learning design : la flexibilité pédagogique au service des besoins de l'e-formation. <http://www.epi.asso.fr/revue/articles/a0512c.htm>, 2005. (Consulté en janvier 2013).

[Brassard et Daele 2003]

Brassard C. et Daele A. (2003). Un outil réflexif pour concevoir un scénario pédagogique intégrant les TIC. In: Environnements Informatiques pour l'Apprentissage Humain (EIAH'2003), Strasbourg, France, 15-17 avril 2003, p. 437-444.

[Caron 2007a]

Caron P.A. (2007). Bricoles : une approche dispositive des applications web 2.0 utilisables pour enseigner. In: Actes de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH 2007), Lausanne, Suisse, 27-29 Juin 2007, p. 137-142.

[Caron 2007b]

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 technologies de Lille.

[Caron 2007c]

Caron P.A. (2007). Contextualisation de dispositifs pédagogiques sur des applications web 2.0. In: Actes du colloque AREF : Actualité de la Recherche en Education et en Formation, Strasbourg, France, 28-31 août 2007, p. 1-11.

[Caron et al. 2005]

Caron P.A., Derycke A. et Le Pallec X. (2005). Bricolage and Model Driven Approach to design distant course. In: E learn 2005, world conference on E-learning in corporate Government, Healthcare & higher education, Association for the Advancement of Computing in Education (AACE), Vancouver, Canada, 24 octobre 2005, p. 2856-2864.

[Caron et al. 2007]

Caron P.A., Hoogstoel F., Le Pallec X. et Warin B. (2007). Construire des dispositifs sur la plateforme moodle - application de l'ingénierie bricolée. In: MoodleMoot 2007, Castres, France, 14 - 15 Juin 2007.

[Caron et Varga 2009]

Caron P.A. et Varga R. (2009). Artefacts malléables et perméables (AMP) pour mener des activités pédagogiques: L'utilisateur acteur et créateur de son dispositif numérique. Articles dans des revues avec comité de lecture listées par l'AERES (ACL). In : Distances et Savoirs, vol.7, n°2, p. 155-177.

[Charlier et Daele 2002]

Charlier B. et Daele A. (2002). Recre@sup: Réseau des centres de ressources pour l'enseignement supérieur. Rapport final. Projet de recherche : SCORATES-MINERVA 87788-CP-1-2000-1-BE-MINERVA-ODL.

[Chikofsky et Cross 1990]

Chikofsky J.E. et Cross J.H. (1990). Reverse engineering and design recovery: A taxonomy. In: Journal IEEE Software, vol.7, issue 1, 1990, p. 13-17.

[Choi et al. 2005]

Choi Y., Kim k. et Kim C. (2005). A design chain collaboration framework using reference models. In: the International Journal of Advanced Manufacturing Technology, Vol. 26, issue 1, 2005, p. 183-190.

[Choquet 2007]

Choquet C. (2007). Ingénierie et réingénierie des EIAH : l'approche REDiM. Habilitation à diriger des recherches en informatique, Université du Maine 2007.

[Clave 2004]

Clave A. (2004). D'UML à MDA en passant par les méta modèles. Rapport technique, La Lettre d'ADELI n°56, 2004.

[Coates et al. 2005]

Coates H., James R. et Baldwin G. (2005). A critical examination of the effects of learning management systems on university teaching and learning. In: Tertiary Education and Management, vol. 11, Issue 1, 2005, p. 19-36.

[Colace et al. 2003]

Colace F., De Santo M. et Vento M. (2003). Evaluating On-line Learning Platforms: a Case Study. In Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03). Big Island, Hawaii, 6-9 janvier 2003.

[Coppercore 2012]

Coppercore (2012). Coppercore, The IMS Learning Design Engine. Accessible à : <http://coppercore.sourceforge.net>, (consulté en decembre 2012).

[Corbière et Choquet 2005]

Corbière A. et Choquet C. (2005). Odp-rm : Un cadre de réingénierie des systèmes de formation. In: Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF 2005), vol. 12, 2005.

[Cottier et al. 2008]

Cottier P., Choquet C. et Tchounikine P. (2008). Repenser l'ingénierie des EIAH par des enseignants concepteurs. In: Usages, usagers et compétences informationnelles au XXIème siècle, Paris, France, 2008, p. 159-193.

[Crozat et Gebers 2009]

Crozat S. et Gebers E. (2009). Chaînes éditoriales Scenari et unité ICS : solutions pour la production documentaire numérique. In: Distances et savoirs, vol.7(3), juillet-septembre 2009, p. 421-442.

[Czarnecki et Helsen 2003]

Czarnecki K. et Helsen S. (2003). Classification of model transformation approaches. In: Workshop on the Generative Techniques in the Context Of Model-Driven Architecture (OOPSLA'03), Anaheim, California, USA, 2003.

[Dalziel 2003]

Dalziel J. R. (2003). Implementing Learning Design: The Learning Activity Management System (LAMS). In: Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education, Adelaide, Australia, 2003.

[Dalziel 2010]

Dalziel J. (2010). Visualising learning design in LAMS: a historical view. Proceedings of 2010 European LAMS & Learning Design Conference : sharing great ideas, vol.2, Sydney Australia, 15-16 juillet 2010, p. 21-35.

[David et al. 2008]

David J. P., George S., Godinet H. et Leclercq E.V. (2008). Scénariser une situation d'apprentissage collective instrumentée : réalités, méthodes et modèles, quelques pistes. In : International Journal of Technologies in Higher Education 4(2), 9 décembre 2008, p. 72-84.

[De-Moura Filho 2007]

De Moura Filho C. (2007). MDEduc : conceiving and implementing a language-oriented approach for the design of automated learning scenarios. Doctorat, Université des Sciences et Technologies de Lille.

[De-Moura Filho et Derycke 2007]

De Moura Filho C. et Derycke A. (2007). Concevoir des scénarios pédagogiques exécutables avec des patrons de conception pédagogiques. In : Actes de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH 2007), Lausanne, Suisse, p. 119-130.

[De-Vries et al. 2006]

De-Vries F., Tattersall C. et Koper R. (2006). Future developments of IMS Learning Design tooling. In: Educational Technology & Society, vol.9 (1), janvier 2006, p. 9-12.

[Dodero et Diez 2006]

Dodero J.M. et Díez D. (2006). Model-Driven Instructional Engineering to Generate Adaptable Learning Materials. In: the 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006), Kerkrade, The Netherlands, 5-7 juillet 2006, p.1188-1189.

[Dodero et al. 2007a]

Dodero J-M., Tattersall C., Burgos C. et Koper R. (2007). Non-representational authoring of learning designs : from idioms to model-driven development. [non publié].

[Dodero et al. 2007b]

Dodero J.M., Tattersall C., Burgos D. et Koper R. (2008). Transformational Techniques for Model-Driven Authoring of Learning Designs. In: proceedings of the 6th International Conference on advances in Web based Learning, Edinburgh (ICWL 2007), UK, 15-17 août 2007, p. 230-241.

[Durand 2006]

Durand G. (2006). La scénarisation de l'évaluation des activités pédagogiques utilisant les Environnements Informatiques d'Apprentissage Humain. Doctorat de l'Université de Savoie.

[Ecoutin 2001]

Ecoutin E. (2001). Fiche pratique no. 1 : les utilisations d'une plate-forme. Technical report, mars 2001. Accessible à : <http://capfoad.chez.com/plates%20formes/algptfor/utilisat.pdf>

[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.

[El-Kechaï et Choquet 2005]

El-Kechaï H. et Choquet C. (2005). Approche pragmatique de conception d'un EIAH : Réingénierie pédagogique dirigée par les modèles. In: actes de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH'2005), Montpellier, France, 25-27 mai 2005, p. 189-200.

[Ertoran 2006]

Ertoran P. (2006). Les Learning Content Management Systems (LCMS). Accessible à : <http://www.centre-inffo.fr/Les-Learning-Content-Management.html>, juillet 2006.

[Ferraris et al. 2005]

Ferraris C., Lejeune A., Vignollet L. et David J.P. (2005). Modélisation de scénarios pédagogiques collaboratifs. In: actes de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH 2005), Montpellier, France, 25-27 mai 2005, p. 285-296.

[Ferraris et al. 2007]

Ferraris C., Martel C. et Vignollet L. (2007). LDL for Collaborative Activities. Handbook of Visual Languages in Instructional Design: Theories and Practices. In: Instructional Design: Concepts, Methodologies, Tools and Applications, vol. 1, 2008.

[Ferraris et al. 2009]

Ferraris C., Ouari S., Vignollet L. et Martel, C. (2009). L'IDM pour la construction d'un environnement intégré support à la scénarisation pédagogique. In : Actes des 5èmes journées sur l'Ingénierie Dirigée par les Modèles (IDM 2009), Nancy, France, 25-26 mars 2009, p. 147-162.

[Ferraris et Lejeune 2007]

Ferraris C. et Lejeune A. (2007). Le couplage "scénario - traces". Exemple du langage LDL. In : Cinquième école thématique du CNRS sur les EIAH "Personnalisation des EIAH".

[George et Derycke 2005]

George S. et Derycke A. (2005). Conceptions et usages des plates-formes de formation. In: Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF 2005), vol. 12, 2005, p. 51-64.

[Gonzalez-Barbonea et Anido-Rifonb 2010]

Gonzalez-Barbonea, V. et Anido-Rifonb, L. (2010). From SCORM to Common Cartridge: A step forward. In: International Journal Computers & Education, vol. 54, issue 1, janvier 2010, p. 88-102.

[Gounon et al. 2005]

Gounon P., Leroux P. et Dubourg X. (2005). Describing learner support : An adaptation of IMS-LD Educational Modelling Language. Amsterdam (Pays-Bas). In: the 12th International Conference on Artificial Intelligence in Education (AIED'05), Supporting Learning through Intelligent and Socially Informed Technology, Amsterdam, The Netherlands, 18-22 juillet 2005, p. 807-809.

[Gounon et Leroux 2006]

Gounon P. et Leroux P. (2006). EdiScénE, un editeur de scénarios d'apprentissage et d'encadrement. In: actes de colloque Technologies de l'Information et de la Communication dans l'Enseignement Supérieur et l'Entreprise (TICE'2006), Toulouse, France, 25-27 octobre 2006, p. 369-376.

[Greenberg 2002]

Greenberg L. (2002). LMS and LCMS: What's the Difference? Accessible à : <http://www.learningcircuits.org/NR/exeres/72E3F68C-4047-4379-8454-2B88C9D38FC5.htm>

[Griffiths et al. 2005]

Griffiths D., Blat J., Garcia R., Vogten H. et Kwong KL. (2005). Learning Design Tools. In: Learning Design, A Handbook on Modelling and Delivering Networked Education and Training. ISBN: 978-3-540-22814-1, Springer-Verlag Berlin Heidelberg, p. 109-135.

[Gruau 2006]

Gruau C. (2006). Conception d'une base de données. Document technique. Accessible à : <ftp://ftp-developpez.com/cyril-gruau/ConceptionBD.pdf> (consulté en janvier 2013).

[Gruber et al. 2010]

Gruber M., Glahn C., Specht M. et Koper R. (2010). Orchestrating learning using adaptive educational designs in IMS learning design. In: 5th European Conference on Technology Enhanced Learning (EC-TEL 2010), Barcelona, Spain, 28 septembre – 1 octobre 2010, p. 123-138.

[Gutiérrez-Santos et al. 2008]

Gutiérrez-Santos S., Pardo A. et Delgado-Kloos C. (2008). Authoring courses with rich adaptive sequencing for IMS Learning Design. In: *Journal of Universal Computer Science*, vol. 14, no. 17 (2008), p. 2819-2839.

[Hause et al. 2005]

Hause M., Thom F. et Moore A. (2005). Inside SysML. In: *Computing & Control Engineering Journal*, vol. 16, issue 4, août-septembre 2005, p. 10-15.

[Henrard et al. 1996]

Henrard J., Hick J.M., Roland D., Englebert V. et Hainaut J.L. (1996). Techniques d'analyse de programmes pour la rétro-ingénierie de bases de données. In: *INFORSID'96*, Bordeaux, France, 4-7 juin 1996, p. 215-232.

[Hernandez-Leo et al. 2004]

Hernandez Leo D., Asensio Pérez J.I. et Dimitriadis. Y.A. (2004). IMS learning design support for the formalization of collaborative learning patterns. In: *proceedings IEEE International Conference on Advanced Learning Technologies (ICALT 2004)*, Joensuu, Finlande, 30 août – 1 septembre 2004, p. 350 - 354.

[Hernández-Leo et al. 2006]

Hernández-Leo D., Villasclaras-Fernández E.D., Asensio-Pérez J.I., Dimitriadis Y., Jorin-Abellán I.M., Ruiz-Requies I. et Rubia-Avi. B. (2006). COLLAGE: a collaborative learning design editor based on patterns. In: *Educational Technology & Society*, 2006, vol. 9(1), p. 58–71.

[Hlaoui et al. 2012]

Hlaoui Y.B., Ayed L.J.B., Fraj I.B. et Benyounes A. (2012). A BPEL4WS Extension to Develop and Execute Grid Service Workflow Application. In: *IEEE 36th Annual Computer Software and Applications Conference Workshops (COMPSACW)*, Izmir, Turkey, 16-20 juillet 2012, p. 224-229.

[Iksal et al. 2010]

Iksal S., Choquet C. et Phâm Thi Ngoc D. (2010). A generic modeling of indicator with utl - The Collaborative Action Function Example. In: *proceedings of 2nd International Conference on Computer Supported Education*, edited by INSTICC, isbn : 978-989-674-024-5, Valencia, Spain, 7-10 avril 2010, p.114-119.

[IMS-LD 2012]

IMS-LD (2012). IMS learning design information model, ims. Technical report, IMS Global Learning Consortium. Accessible à : <http://www.imsglobal.org/learningdesign/index.html>, consulté en Mars 2012.

[Jézéquel et al. 2012]

Jézéquel J.M., Combemale B. et Vojtisek D. (2012). *Ingénierie dirigée par les modèles : des concepts à la pratique*. ISBN : 9782729871963.

[John et Muthig 2002]

John I. et Muthig D. (2002). Tailoring Use Cases for Product Line Modeling. In : International Workshop on Requirements Engineering for Product Lines (REPL 2002), Essen, Germany. ISBN: 0-9724277-0-8.

[Jouault et al. 2006]

Jouault F., Bézivin J., Consel C., Kurtev I. et 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, Nantes, France, 3 juillet 2006.

[Katsamani et Retalis 2011]

Katsamani M. et Retalis S. (2011). Making Learning Designs in Layers: The CADMOS Approach. In: proceedings of the IADIS Multi Conference on Computer Science and Information Systems 2011, Rome, Italy, 20-26 juillet 2011, p. 305-312.

[Kelly 2004]

Kelly S. (2004). Comparison of Eclipse EMF/GEF and MetaEdit+ for DSM. In : 19th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, Workshop on Best Practices for Model Driven Software Development, Vancouver, Canada, 24-28 octobre 2004,.

[Kelly et al. 1996]

Kelly S., Lyytinen K. et Rossi M. (1996). MetaEdit+: A Fully Configurable Multi-User and Multi-Tool CASE Environment. In: Proceedings of 8th International Conference, CAiSE'96, Heraklion, Crete, Greece, 20–24 mai 1996, p. 1-21.

[Kelly et Tolvanen 2008]

Kelly S. et Tolvanen J.P. (2008). Domain-Specific Modeling. ISBN: 978-0-470-03666-2, mars 2008. Wiley-IEEE Computer Society Press.

[Koper et Tattersall 2005]

Koper R. et Tattersall C. (2005). Preface to Learning Design: A Handbook on Modelling and Delivering Networked Education and Training. In: Journal of Interactive Media in Education 2005(18), 24 août 2005.

[Kulaka et al. 2010]

Kulaka O., Cebib S. et Kahraman C. (2010). Applications of axiomatic design principles: A literature review. In: Expert Systems with Applications, vol. 37, issue 9, september 2010, p. 6705–6717.

[Laabidi 2013]

Laabidi M. (2013). Une approche dirigée par les modèles pour l'intégration de l'accessibilité dans les systèmes e-learning. Doctorat de l'université de Sfax

[Laforcade 2004]

Laforcade P. (2004). Méta-modélisation UML pour la conception et la mise en oeuvre de situations-problèmes coopératives. Doctorat de l'Université de pau.

[Laforcade 2007]

Laforcade P. (2007). Graphical representation of abstract learning scenarios: the UML4LD experimentation. In: Proceedings of the Seventh International Conference on Advanced Learning Technologies, Kerkrade, The Netherlands, 18-20 juillet 2007, p. 477–479.

[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. In: Journal of Visual Languages and Computing, vol.21 issue 6, décembre 2010, p.347-358.

[Laforcade et Abedmouleh 2012]

Laforcade P. et Abedmouleh A. (2012). Improving the design of courses thanks to graphical and external dedicated languages: a Moodle experimentation. In: Moodle Research Conference 2012, Heraklion, Greece, 14-15 septembre 2012, p. 94-101.

[Laforcade et al. 2003]

Laforcade P., Barbier F., Nodenot T. et Sallaberry C. (2003). Profiling cooperative problem-based learning situations. In: proceedings of the 2nd IEEE International Conference on Cognitive Informatics (ICCI'2003), London, UK, 18-20 août 2003, p. 32-38.

[Laforcade et al. 2005]

Laforcade P., Nodenot T. et Sallaberry C. (2005). Un langage de modélisation pédagogique basé sur UML. In: Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF 2005), vol.12, 2005, p.89-115.

[Laforcade et al. 2007]

Laforcade P., Barré V. et 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 de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH'07), Lausanne, Suisse, 27-29 juin 2007, p. 257-268.

[Laforcade et al. 2008]

Laforcade P., Zendagui B. et Barré V. (2008). A Domain-Specific-Modeling Approach to Support Scenarios-Based Instructional Design. In: proceedings of the 3rd European Conference on Technology Enhanced Learning (ECTEL'08), Maastricht, The Netherlands, 16-19 septembre 2008, p. 185-196.

[Laforcade et al. 2011]

Laforcade P., Oubahssi L. et Piau-Toffolon C. (2011). Quel accompagnement des équipes enseignantes : Moyens, usages, reconnaissance des compétences. In: Séminaire Vivaldi – Pays de la Loire. Le Mans, 5-6 avril 2011.

[Ledeczi et al. 2001]

Ledeczi A., Bakay A., Maroti M., Volgyes, P., Nordstrom G., Sprinkle J. et Karsai G. (2001). Composing Domain-Specific Design Environments. In: Computer journal, novembre 2001, vol. 34, issue 1, p. 44-51.

[Lengyel et Herdon 2009]

Lengyel P. et Herdon M. (2009). Implementing learning design by LAMS to improve teaching and learning. In: 4th Aspects and Visions of Applied Economics and Informatics, Debrecen, Hungary, 26-27 mars 2009.

[Le Pallec et al. 2005]

Le Pallec X., Renaux E. et Moura. C.O. (2005). Modx - a graphical tool for mof metamodels. In: Model Driven Architecture - Foundations and Applications, First European Conference (ECMDA-FA 2005), Nuremberg, Germany, 7-10 novembre 2005, ISBN 3-540-30026-0.

[Le pallec 2012]

Le Pallec X. (2012). Intégration de métriques de qualité des diagrammes et des langages dans l'outil ModX. In : Conférence en Ingénierie du Logiciel (CIEL 2012), Rennes, France, 2012, p. 1-6.

[Loiseau et Laforcade 2013]

Loiseau E. et Laforcade P. (2013). Spécification de langages de scénarisation graphiques centrés sur les plateformes de formation à distance Etude et expérimentation d'approches DSM pour Moodle. In : Environnements Informatiques pour l'Apprentissage Humain (EIAH'2013), 29-31 mai 2013, Toulouse, France.

[Marca et McGowan 1987]

Marca D.A., et McGowan C.L. (1987). SADT: structured analysis and design technique, New York, USA, ISBN:0-07-040235-3.

[Marchand et Depover 2002]

Marchand L. et Depover C. (2002). E-learning et formation des adultes en contexte professionnel, Bruxelles, édition De Boeck Université, 2002, 260 pages, ISBN : 9782804141066.

[Martel et Vignollet 2007]

Martel C. et Vignollet L. (2007). Learning Design Language to model the activity supported by services. Service Oriented Approaches and Lifelong Competence Development Infrastructures. In: proceedings of the 2nd TENCompetence Open Workshop. Manchester, UK, 11-12 janvier 2007.

[Mekpiroona et al. 2008]

Mekpiroona O., Tammarattananonta P., Buasrounga N., Apitiwongmanita N., Pravalpruka B. et Supnithia T. (2008). SCORM in Open Source LMS : A case study of LEARNSQUARE. In: 16th international conference on computer in Education (ICCE2008), Taipei, Taiwan, 2008.

[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. 2003]

Nodenot T., Laforcade P., Marquesuzaà C. et Sallaberry C. (2003). Knowledge modelling of cooperative learning situations: Towards a UML profile. In: proceedings of the 11th International Conference on Artificial Intelligence in Education (AIED'2003), Sydney, Australia, 20-24 juillet 2003.

[Nodenot et al. 2005]

Nodenot T., LePallec X. et Marino O. (2005). La manipulation des modèles : modélisation, méta-modélisation, transformation. In: Troisième école thématique du CNRS sur les Environnements informatiques pour l'apprentissage humain (EIAH) – Modèles, architectures logicielles et normes pour le développement et l'intégration des EIAH", organisée par le CNRS - Département STIC - RTP39, Au-trans, France, juillet 2005.

[Nodenot et al. 2008]

Nodenot T., Caron P.A., Le Pallec X. et Laforcade P. (2008). Applying Model Driven Engineering techniques and tools to the planets game learning scenario. In: Journal of Interactive Media in Educa-tion, 2008(23).

[OMG 2003]

Object Management Group (2003). Technical guide to model driven architecture : The mda guide v1.0.1. Technical report, juin 2003. Accessible à : <http://www.omg.org/docs/omg/03-06-01.pdf>.

[Ortiz et al. 2007]

Ortiz I. M., Ger P.M., Rodríguez, J.L.S. et Manjón B.F. (2007). Supporting the authoring and opera-tionalization of educational modelling languages. In: Journal of Universal Computer Science, vol. 13, no. 7 (2007), p. 938-947.

[Ortiz et al. 2008]

Ortiz I.M., Sierra, J.L. et Fernandez-Manjon B. (2008). A flow-oriented visual language for learning designs. In: proceedings of the Seventh International Conference on Web-based Learning. Jinhua, China, 20-22 août 2008, p. 486-496.

[Ortiz et al. 2009]

Ortiz I.M., Sierra J.L. et Fernández-Manjón B. (2009). Enhancing IMS LD Units of Learning Com-prehension. In: the 4th International Conference on Internet and Web Applications and Services. Ven-ice, Italy, 24-28 mai 2009, p. 561-566.

[O'Rourke 2012]

O'Rourke M. (2012). A handbook for the development of lessons, case studies, scenarios and simula-tions in the Moodle Virtual Learning Environment for self directed learning.

[Oubahssi 2005]

Oubahssi L. (2005). Conception de plates-formes logicielles pour la formation à distance, présentant des propriétés d'adaptabilité à différentes catégories d'usagers et d'interopérabilité avec d'autres envi-ronnements logiciels. Doctorat de l'Université René Descartes – Paris V.

[Oubahssi et al. 2009]

Oubahssi L., Laforcade P. et Cottier P. (2009). Ré-ingénierie d'un EIAH existant : le cas du Livret Électronique d'Apprentissage. In: Actes de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH 2009). Le Mans, France, 23-26 juin 2009, p. 311-319.

[Oubahssi et al. 2010]

Oubahssi L., Laforcade P. et Cottier P. (2010). Re-engineering of the Apprenticeship Electronic Booklet : Adaptation to new users requirements. In: IEEE International Conference on Advanced Learning Technologies (ICALT), Sousse, Tunisia, 5-7 juillet 2010, p. 511-515.

[Ouraiba 2012]

Ouraiba E.A. (2012). Scénarisation pédagogique pour des EIAH ouverts - Une approche dirigée par les modèles et spécifique au domaine métier. Doctorat de l'université du Maine.

[Panet et al. 1994]

Panet G., Letouche R. et Tardieu H. (1994). Merise 2 : Modèles et techniques Merise avancés. 360 pages. ISBN : 2708116533. Edité par Organisation Eds D'.

[Paquette 2002]

Paquette G. (2002). Modélisation des connaissances et des compétences, un langage graphique pour concevoir et apprendre. Presses de l'Université du Québec, 2002, 357 pages, ISBN : 2-7605-1163-4.

[Paquette 2005]

Paquette G. (2005). Apprentissage sur l'Internet : des plateformes aux portails à base d'objets de connaissance. In: S. Pierre (dir.), Innovations et tendances en technologies de formation et d'apprentissage. Presses de l'école polytechnique de Montréal, p. 1-30.

[Paquette et al. 1997]

Paquette G., Crevier F. et Aubin C. (1997). Méthode d'ingénierie d'un système d'apprentissage. Revue Informations in Cognito, vol.8, mars 1997.

[Paquette et al. 2008]

Paquette G., Léonard M. et Lundgren-Cayrol K. (2008). The MOT+ Visual Language for Knowledge-Based Instructional Design. In: Botturi et Stubs (dirs.), Handbook of Visual Languages for Instructional Design Languages – Theories and Practices, chapitre VIII. Hershey, PA : IGI Global.

[Parrend 2006]

Parrend P. (2006). Introduction à MDA : Principe. Accessible à : <http://pparrend.developpez.com/tutoriel/mda-intro/>.

[Pelechano et al. 2006]

Pelechano V., Albert M., Muñoz J. et Cetina C. (2006). Building tools for model driven development comparing microsoft DSL tools and eclipse modeling plug-ins. In : Proceedings of the 11th Conference on Software Engineering and Database (JISBD'06), Sitges, Spain, 3-6 octobre 2006.

[Pernin 2003]

Pernin J.P. (2003). Objets pédagogiques : unités d'apprentissage, activités ou ressources ?. In: Revue "Sciences et Techniques Educatives" Hors série 2003 : "Ressources numériques, XML et éducation", avril 2003, éditions Hermès, p. 179-210.

[Pernin 2004]

Pernin J.P. (2004). A propos d'objet d'apprentissage, d'activité, de ressource et de document. In : séminaire DOCSI, ENSSIB, Lyon, France, mars 2004.

[Pernin et Lejeune 2004]

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 Technologies de l'information et de la connaissance dans l'enseignement supérieur et l'industrie (TICE2004), Compiègne, France, p. 407-414.

[Randriamalaka 2008]

Randriamalaka N. (2008). Assistance à la modélisation et à l'analyse des traces d'usage dans un EIAH : une approche par la réingénierie. Doctorat de l'Université du Maine, Université du Maine.

[Rath et al. 2005]

Rath I., Schmidt A. et Vigo D. (2005). Automated model transformations in domain specific visual languages, Scientific Students' Associations Report, Budapest, Hongrie, octobre 2005.

[Rawlings et al. 2002]

Rawlings A., Rosmalen P.V., Koper R., Artacho M.R. et Lefrere P. (2002). Survey of Educational Modelling Languages (EMLs). In: Learning Technologies Workshop Version 1, CEN/ ISSS WS/LT, 19 septembre 2002.

[Reload 2006]

Reload (2006). Reusable eLearning Object Authoring Delivery. Accessible à : <http://www.reload.ac.uk/>, consulté en septembre 2012.

[Rodríguez Artacho et Verdejo Maíllo 2004]

Rodríguez Artacho M. et Verdejo Maíllo M. (2004). Modeling educational content: The cognitive approach of the PALO language. In: International Journal of Educational Technology Society, Vol. 7 (3), p. 124-137.

[Ryu et Yücesan 2007]

Ryu K. et Yücesan E. (2007). CPM: A collaborative process modeling for cooperative manufacturer. In: Journal Advanced Engineering Informatics, vol. 21, issue 2, avril 2007, p. 231-239.

[Schneider 2003]

Schneider D. (2003). Conception and implementation of rich pedagogical scenarios through collaborative portal sites. In: International Conference on Open and Online Learning (ICOOL 2003), Colloque de Guéret 2003, University of Mauritius, p. 243-268.

[Steel 2009]

Steel C. (2009). Reconciling university teacher beliefs to create learning designs for LMS environments. The University of Queensland. In: Australasian Journal of Educational Technology (AJET2009), vol. 25(3), p. 399-420.

[Taenzer 2006]

Taentzer G. (2006). Towards generating Domain-Specific Model editors with complex editing commands. In: Proc. International Workshop Eclipse Technology eXchange (eTX), Satellite Event of ECOOP, Portland, Oregon, 22-23 octobre 2006.

[Tardieu et al. 2000]

Tardieu H., Rochfeld A. et Coletti R. (2000). La méthode Merise. Principes et outils Edition 2000. 344 pages. ISBN : 2708124730 Edité par Organisation Eds D', juillet 2000.

[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. Rapport de l'Action Spécifique du département STIC du CNRS : Fondements théoriques et méthodologiques de la conception des EIAH. Accessible à : <http://membres-liglab.imag.fr/tchou-nikine/Articles/Platon-1.pdf>, (consulté en février 2013).

[Thi-Lan-anh et al. 2005]

Thi-Lan-anh D., Olivier G. et Houari S. (2005). Gestion de modèles : définitions, besoins et revue de littérature. In Premières Journées sur l'Ingénierie Dirigée par les Modèles, Paris, France, 30 Juin- 1 Juillet 2005, p. 1-15.

[Vantroys 2003]

Vantroys T. (2003). Du langage métier au Langage technique, une plateforme flexible d'exécution de scénarios pédagogiques. Doctorat de l'université Lille.

[Zendagui 2010]

Zendagui B. (2010). Modélisation de l'observation dans un contexte de réingénierie – Une approche dirigée par les modèles. Doctorat de l'université du Maine.

[Muehlen et Recker 2008]

Muehlen M.Z. et Recker J. (2008). How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In: the 20th international conference on Advanced Information Systems Engineering (CAISE 2008), Montpellier, France, June 16-20, 2008, p. 465-479.

Références sur le WEB

[Claroline 2012]

<http://www.claroline.net/>. Consulté le 13/01/2012

[DSM 2012]

DSM forum (2012). <http://www.dsmforum.org/>. Consulté le 15/09/2012

[EMP 2012]

Eclipse Modeling Project (2012). Accessible à : <http://www.eclipse.org/modeling/> (consulté en décembre 2012).

[Ganesha 2013]

<http://www.ganesha.fr/>

[GMF 2013]

<http://www.eclipse.org/modeling/gmp/>. Consulté le 25/01/2013.

[Graphit 2013]

Site du projet GraphiT. Accessible à : <http://www-lium.univ-lemans.fr/~laforcad/graphit/>, (consulté en mars 2013).

[MicrosoftDsl 2013]

<http://msdn.microsoft.com/vstudio/DSLTools/>. Consulté le 25/01/2013

[OLF 2013]

Office de la Langue Française (OLF) (2013). Accessible à : <http://granddictionnaire.com> (consulté en janvier 2013).

[ParisMoodle 2009]

http://mediabox.parisdescartes.fr/medias/NDE5NTg%3D/900313Presentation_pour_les_enseignants.pdf?id=8

[W3C 2000]

W3C. Simple object access protocol (soap). Technical report, mai 2000.

[Wikipedia 2012]

http://en.wikipedia.org/wiki/Function_model

[Xfront 2012]

<http://www.xfront.com/GlobalVersusLocal.html#FirstDesign>

Les annexes

Annexe A

Application du processus sur la plateforme Moodle

Sommaire

1. Introduction
2. Analyse centrée IHM
3. Analyse technique
4. Confrontation et formalisation du langage de conception
pédagogique
5. Conclusion

1. Introduction

Dans cette annexe, nous détaillons l'application du processus d'identification et de formalisation du langage de conception pédagogique de la plateforme open source Moodle (présenté brièvement dans le cinquième chapitre). Cette plateforme propose, à travers un nombre important d'interfaces, de nombreuses activités et types de ressources pour la spécification de situations d'apprentissage.

Cette application a nécessité l'analyse complète de tous les écrans et de toutes les tables de la base de données de cette plateforme. Cependant, nous n'allons pas rendre compte avec exhaustivité de cette application complète du processus. Nous allons seulement présenter et détailler un ensemble cohérent d'exemples pertinents dans le but d'illustrer toutes les spécificités et particularités du processus d'identification et d'explicitation du métier de conception pédagogique de Moodle. Les cas similaires, redondants, à ces exemples ne seront pas représentés dans ce chapitre. En fin de chaque analyse, nous présentons les modèles produits et illustrons ainsi les particularités des formalismes de représentation que nous proposons de suivre dans le processus. Avant de conclure cette annexe, nous présentons le modèle final représentant le langage de conception pédagogique de la plateforme Moodle.

Nous allons détailler chaque partie du processus par le biais d'une structure spécifique décrite par le contexte, l'objectif et l'application concrète du processus. Nous utilisons la légende suivante :



Ce pictogramme identifie des problèmes rencontrés.



Ce pictogramme identifie les solutions que nous proposons.



Ce pictogramme identifie des rappels ou des remarques ou des constats.

2. Analyse centrée IHM

Rappel : l'analyse centrée IHM consiste à identifier le langage de conception pédagogique de point de vue IHM : le langage visible à l'utilisateur. Comme décrite au sein du chapitre 4, cette analyse est composée de trois analyses (IHM-macro, fonctionnelle et IHM-micro) et d'une étape de composition pour la formalisation du modèle centrée IHM.

2.1. Analyse IHM-macro

Contexte : l'analyse IHM-macro est la première étape de l'analyse centrée IHM. Nous présentons dans cette partie l'application de cette analyse sur la plateforme Moodle.

Objectif : identifier les interfaces relevant de la conception pédagogique, plus précisément les interfaces de spécification des situations d'apprentissage.

Application de l'analyse IHM-macro : nous avons choisi d'illustrer cette application par un ensemble cohérent d'exemples génériques et particuliers.

A. Cas général de l'analyse IHM-macro

Nous avons tout d'abord commencé par nous approprier la plateforme Moodle de point de vue enseignant-concepteur. Ensuite, nous avons appliqué les deux stratégies d'analyse proposées.

Application de la première stratégie : cette stratégie de l'analyse IHM-macro est basée sur l'analyse d'une banque de cours disposée sur la plateforme. L'analyse a consisté à :

- prendre un cours particulier sur Moodle. La figure A.1 montre un exemple de cours sur Moodle, un cours de programmation web en java.

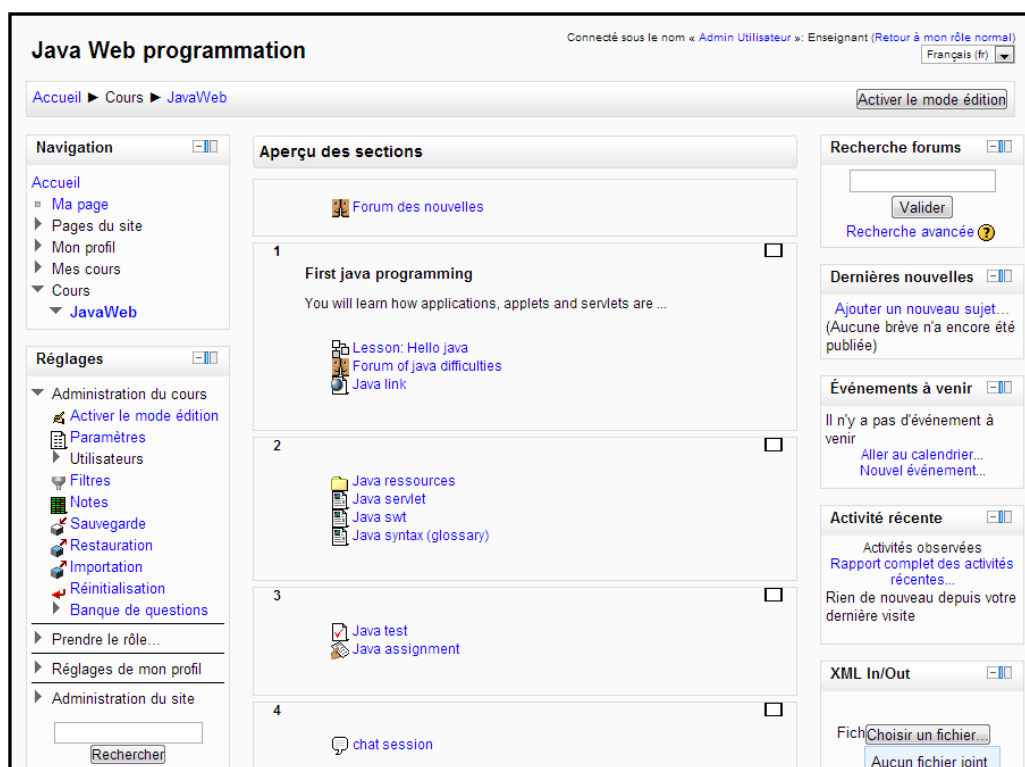


Figure A.1 : Exemple de cours sur Moodle

- étudier le contenu de ce cours.
- choisir un élément particulier de ce cours tel que le '*forum des difficultés du java*'. Nous sommes amenés à un nouvel écran représentant cet élément et proposant sa modification (figure A.18).



Les interfaces de spécification ou modification d'un élément de cours sont relevant de la conception pédagogique. Moodle dispose de plusieurs activités telles que leçon, atelier, wiki, devoir, etc. et de plusieurs ressources telles que fichier, url, page, etc. La figure A.2 montre un aperçu des interfaces des activités et des ressources dont elles doivent être identifiées et analysées.

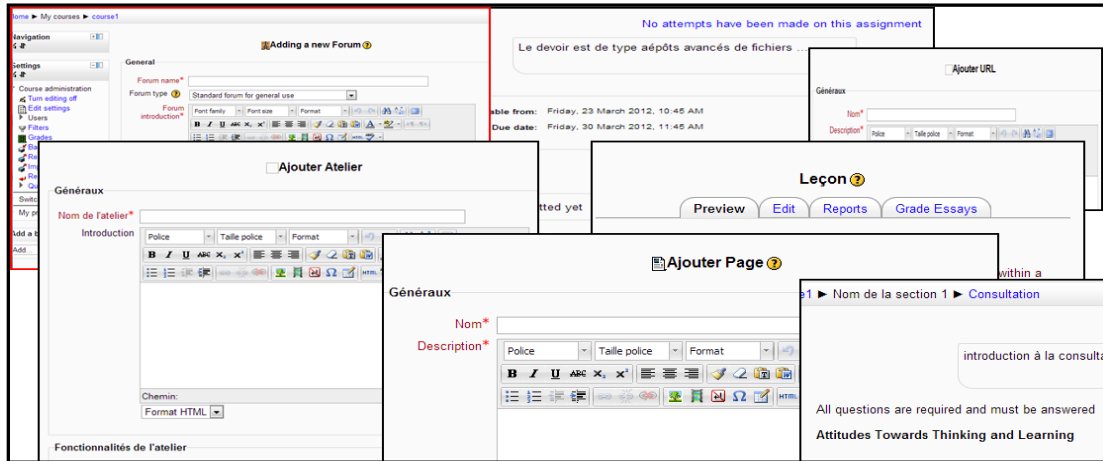


Figure A.2 : Aperçu des interfaces de Moodle

En prenant par exemple l'activité forum, nous avons identifié plusieurs interfaces dédiées à cette activité. La figure A.3 montre un extrait de ces interfaces.

- analyser sémantiquement et syntaxiquement le titre de l'interface consultée. Par exemple, l'analyse sémantique de l'interface du forum (encadrée en rouge et intitulée '*ajout forum*') affirme que cette interface est liée à la conception pédagogique du forum.



Ce constat sera confirmé ultérieurement au cours de l'analyse fonctionnelle en identifiant la fonctionnalité '*ajouter un forum*'.

- identifier de nouvelles interfaces proposant la spécification de nouveaux éléments dédiés à l'élément identifié. Par exemple, à partir des interfaces des forums, nous avons identifié des interfaces spécifiques à des discussions.



La consultation, la modification ou l'ajout d'une discussion particulière permet d'amener à des nouvelles interfaces.

- analyser les titres des interfaces identifiées. Par exemple, l'analyse des titres des interfaces de discussion a permis de déduire que 'discussion' est un élément de conception pédagogique, dédié au forum.
- essayer d'identifier des nouveaux éléments dédiés (par exemple aux discussions), s'ils existent.



Les interfaces des discussions ne proposent pas l'ajout de nouveaux éléments. Seulement, il est possible d'ajouter une nouvelle discussion, de répondre ou de supprimer une discussion existante. Cela nous a permis de déduire qu'il n'existe pas des éléments (des interfaces) dédiés aux discussions (ou à ses interfaces). Ce constat sera également approuvé pendant l'analyse fonctionnelle.

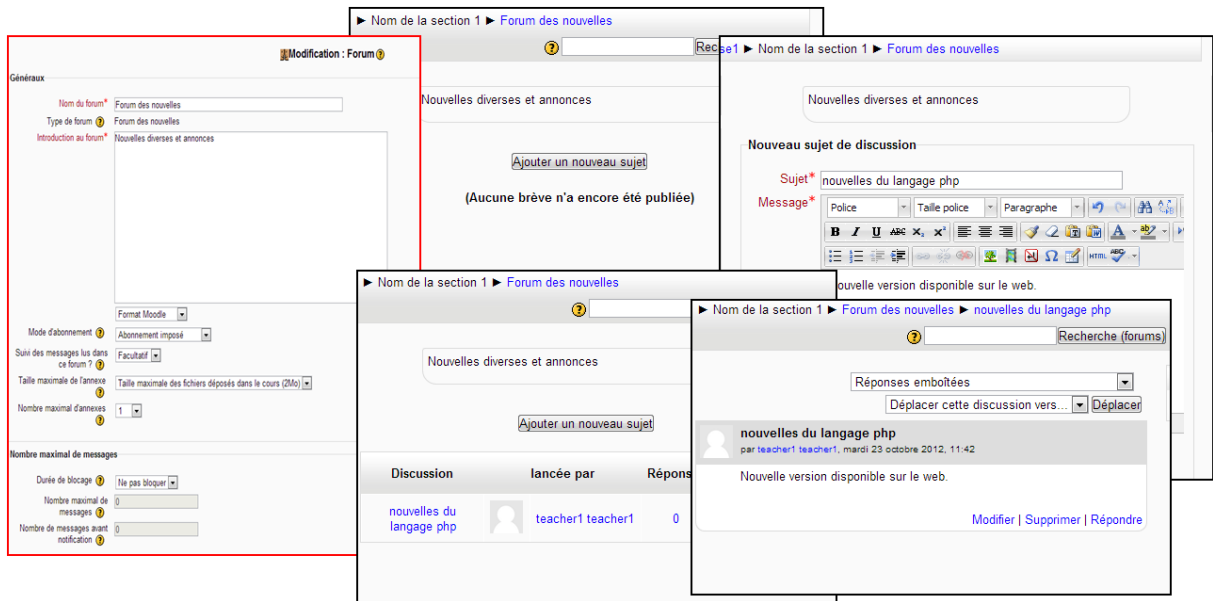


Figure A.3 : Extrait des interfaces dédiées au forum et ses éléments

Application de la deuxième stratégie : cette stratégie consiste à identifier les interfaces relevant de la conception pédagogique en spécifiant le contenu et les paramètres d'un nouveau cours.



La deuxième stratégie peut être appliquée en amont ou en aval de la première stratégie. Les deux stratégies sont complémentaires.

Cette application a consisté à :

- prendre un cours initialement vide.
- spécifier le contenu du cours parmi les éléments proposés par la plateforme Moodle.



Par exemple, Moodle propose de créer des activités pédagogiques telles que les forums, les leçons et également des ressources pédagogiques telles que les liens et les pages web.

- choisir un élément particulier sur l'espace de cours parmi les éléments proposés par Moodle en termes d'activités, de ressources, ou autres. Par exemple, en choisissant l'élément *forum* nous sommes redirigés vers une nouvelle interface intitulée '*ajouter forum*'.
- analyser sémantiquement le titre de l'interface de l'élément choisi pour déterminer si cette interface relève de la conception pédagogique sur Moodle. Par exemple, l'interface intitulée '*ajouter forum*' représente la description de l'élément forum en termes d'attributs et de spécifications. Elle est liée à la conception pédagogique.



En respectant la démarche du processus, cette interface sera ultérieurement analysée de façon plus fine, en particulier au cours de l'analyse IHM-micro. Ce choix répond à notre premier objectif : tout d'abord, encadrer le champ d'analyse en identifiant les interfaces relevant de la conception pédagogique et ensuite s'intéresser davantage aux éléments de conception pédagogique.



Nous détaillons un peu plus loin dans ce document, la représentation des interfaces identifiées (celles qui sont liées à la conception pédagogique) sur le modèle IHM-macro.

Remarques :



L'identification des interfaces dédiées au forum représente un exemple générique de l'application de l'analyse IHM-macro sur la plateforme Moodle¹⁰.



Le résultat de la première stratégie dépend directement du contenu de la banque de cours utilisée. Lorsque tous les éléments proposés par la plateforme sont utilisés dans la banque de cours, les deux stratégies mènent au même résultat. Cependant, lorsqu'un élément pédagogique proposé par la plateforme n'est pas utilisé au sein des cours, il est nécessaire de suivre la deuxième stratégie.



La première hypothèse de l'analyse IHM-macro consiste à parcourir tous les liens situés dans un espace de cours (nouveau ou existant) pour identifier les interfaces relevant de la conception pédagogique. Nous avons analysé les interfaces accessibles à partir de ces liens.

- Nous avons identifié des interfaces dédiées à des activités et à des ressources pédagogiques, à la gestion des utilisateurs, des rôles et des groupes, etc.
- Nous avons écarté les interfaces qui sont indépendantes de la conception de cours (par exemple celle de la sauvegarde ou de l'importation du cours, etc.).



Nous démontrons un peu plus loin que l'analyse fonctionnelle prouvera que les interfaces dédiées aux services de sauvegarde ou d'import n'interviennent pas dans la conception d'un cours.

B. Particularités de l'analyse IHM





Dans Moodle, les interfaces de gestion de paramètres du cours (groupes, etc.) sont accessibles depuis l'espace du cours, en particulier depuis le *bloc* réglages (figure A.1), et également depuis les interfaces du contenu de ce cours.



Ce cas particulier nous amène à prendre certaines décisions subjectives. Nous avons associé les interfaces de paramètres, en particulier les interfaces de spécification de groupes, au nœud *cours* parce que ces paramètres sont spécifiés au même niveau que le contenu du cours.

¹⁰ Moodle dispose des interfaces relevant de la conception pédagogique pour la spécification des activités (atelier, base de données, chat, consultation, devoirs, forum, glossaire, leçon, paquetage SCORM, sondage, test et wiki) et des ressources (dossier étiquette, fichier, page, paquetage IMS content et url).


	<p>Au sein du bloc ‘réglages’, Moodle dispose d’un onglet intitulé ‘<i>banque de questions</i>’. Ce dernier représente un lien vers les questions déjà spécifiées. Il propose aussi de créer de nouvelles questions en nous amenant sur les interfaces concernées. Cependant, ces interfaces sont également accessibles à partir des interfaces de l’activité <i>test</i>. Nous constatons ainsi que les interfaces de questions sont liées aux deux éléments : ‘<i>test</i>’ et ‘<i>banque de questions</i>’.</p>
	<p>Nous avons associé l’élément question au test. Nous justifions ce choix par l’utilisation des questions au sein des tests. Spécifier des questions sans les utiliser n’a pas d’intérêt pédagogique.</p>

C. Le modèle résultant

Le modèle IHM-macro, résultat de l’application de l’analyse IHM-macro sur Moodle, est construit au fur et à mesure de l’identification d’une interface relevant de la conception pédagogique. La figure A.5 illustre ce modèle. Sa spécification a consisté à :

- modéliser l’interface du concept principal du cours (figure A.1) par un nœud central intitulé *cours*.
- modéliser les interfaces identifiées, en particulier celles de spécification du contenu du cours (activités, ressources, paramètres, etc.) sur la plateforme Moodle, par des nœuds intitulés par le concept principal.

Par exemple, l’interface de la partie encadrée en rouge de la figure A.3 décrit le concept forum. Nous l’avons modélisé par un nœud, intitulé ‘forum’, sur le modèle IHM-macro.

	<p>Les interfaces dédiées à un élément spécifique (tel que forum) et ne proposant pas de nouveaux éléments de conception pédagogique sont représentées par un même nœud (forum).</p>
---	--

- définir les associations entre les nœuds en se basant sur les relations entre les interfaces.

Par exemple, nous avons représenté les interfaces accessibles à partir de l’espace du cours par des nœuds indiquant le concept principal et nous les avons associés au nœud cours tel le cas du forum. Ensuite, nous avons représenté les interfaces spécifiques aux éléments dédiés au forum sur le modèle IHM-macro tels que les interfaces de discussions (la partie droite de la figure A.3) que nous avons représentées par un nouveau nœud intitulé *discussion* et associé au nœud *forum*.

La figure A.4 montre un extrait du modèle IHM-macro représentant les interfaces des éléments *cours*, *forum* et *discussion*.

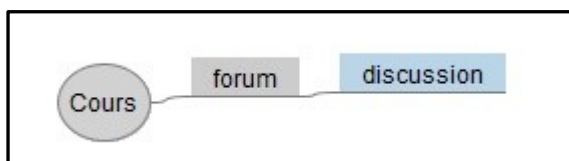


Figure A.4 : Extrait du modèle IHM macro représentant un forum et une discussion



Comme pour le cas du forum, les interfaces des activités et des ressources sont également représentées par des nœuds et associées au nœud du cours.

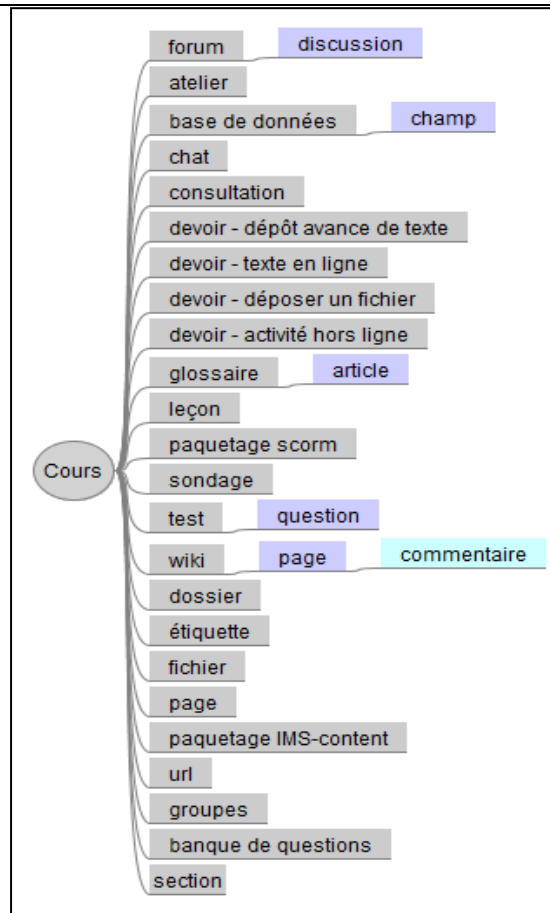


Figure A.5 : Modèle IHM-macro de Moodle

2.2. Analyse fonctionnelle

Contexte : nous présentons dans cette partie l'application de l'analyse fonctionnelle sur les interfaces les plus significatifs et les plus pertinents pour la démonstration

Objectif : identifier les fonctionnalités relevant de la conception pédagogique embarquées dans les différentes IHMs identifiées (celles du modèle IHM-macro).

Application de l'analyse fonctionnelle : nous présentons l'analyse fonctionnelle de certaines interfaces de la plateforme Moodle (dédiées au cours, forum, discussion). Ces analyses représentent un cas générique d'analyse des activités et des ressources (leçon, atelier, url, etc.) ainsi que les éléments dédiés. Les cas similaires et redondants ne seront pas présentés. Toutefois, nous représentons les cas particuliers de cette application.

A. Cas général : analyse fonctionnelle de l'interface du cours

Contexte : nous présentons dans cette partie l'application de l'analyse fonctionnelle sur l'interface du cours (figure A.6).

Objectif : identifier les fonctionnalités pédagogiques dédiées au cours.

Application de l'analyse fonctionnelle : l'analyse fonctionnelle de cette interface a consisté à :

- décomposer cette interface en quatre zones puis mener une analyse de chaque zone.
- analyser les widgets et les composants graphiques de chaque zone.
- identifier les fonctionnalités pédagogiques de chaque widget de chaque zone.
- représenter les fonctionnalités identifiées au sein du modèle fonctionnel.

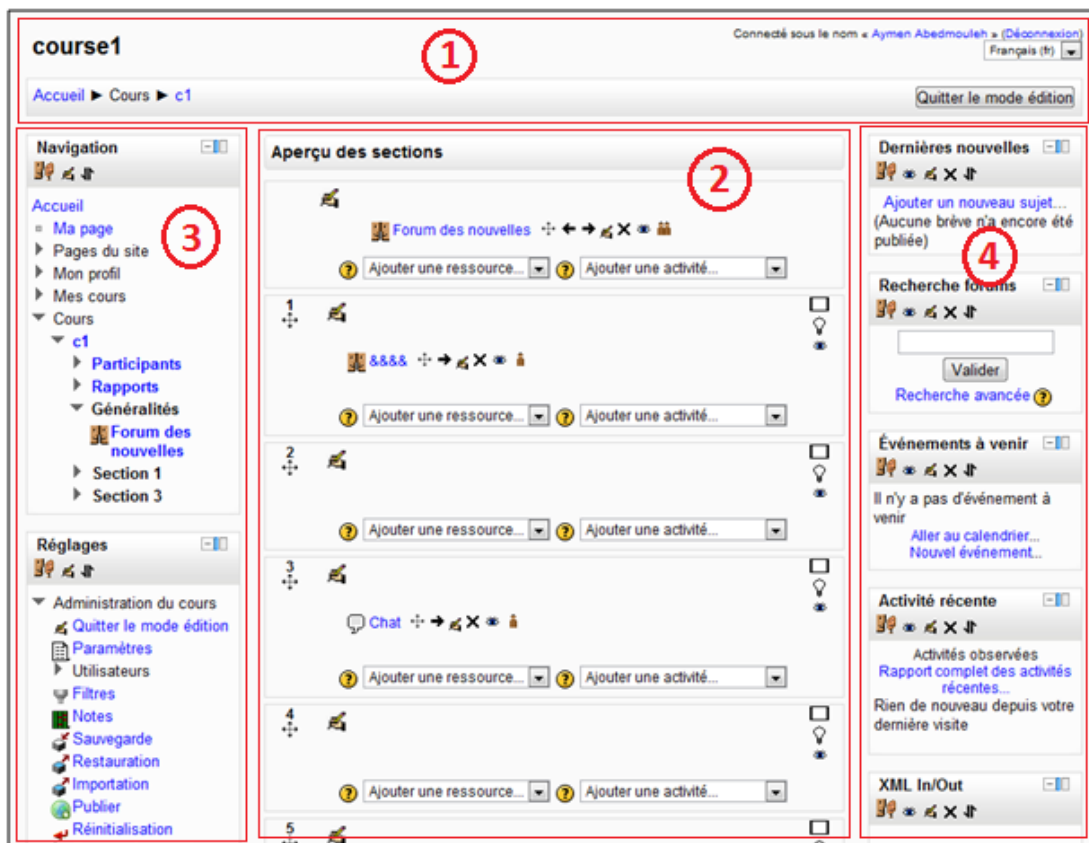


Figure A.6 : Répartition de l'interface en plusieurs zones

Analyse de la zone 1



La zone 1 est commune à toutes les interfaces de la plateforme Moodle. Elle présente certaines informations génériques. Nous y trouvons le nom du cours, l'enseignant connecté à ce cours, la langue choisie et le chemin de navigation (Accueil > Cours > c1).

Cette zone dispose d'un seul widget (un bouton) représentant la fonctionnalité 'Activer/quitter le mode édition'. Cette fonction a un impact sur le contenu de la zone 2 permettant leur édition. De point de

vue conception pédagogique, cette fonction n'influence pas le contenu du cours ou ses paramètres. Nous la considérons ainsi comme une fonction technique plus que pédagogique.

Analyse de la zone 3



La zone 3 est composée de deux principales parties : '*navigation*' et '*réglages*'. Nous avons choisi de conserver cette zone en une seule partie parce qu'elle est commune entre les interfaces du cours.

Nous avons analysé tous les liens, les widgets et les composants graphiques, mis à disposition de l'enseignant-concepteur, pour identifier les fonctionnalités 'pédagogiques' associées.

- Les widgets de la partie navigation indiquent soit les informations de l'enseignant-concepteur (le profile, les messages et les fichiers personnels, etc.) soit les liens pour consulter un cours particulier ou leur contenu (les messages et les discussions des forums associés, les rapports d'activités, etc.). Nous n'avons pas identifié des fonctionnalités pédagogiques dans cette partie parce que ces widgets (les éléments associés) n'interviennent pas dans la conception d'un cours.
- Les widgets de la partie réglages représentent des paramètres spécifiques à l'administration du cours.
 - L'onglet '*activer/quitte le mode édition*' permet d'activer ou désactiver l'édition du cours, en particulier la zone 2. Nous considérons cette fonction, déjà étudiée dans l'analyse de la zone 1, comme technique.
 - L'onglet '*paramètres*' représente un lien vers l'interface d'édition des paramètres génériques du cours tels que le nom du cours, résumé du cours, nombre de sections etc. Nous avons tenu compte de ce lien en lui attribuant la fonctionnalité '*spécifier les paramètres du cours*'.
 - L'onglet '*utilisateur*' décrit des éléments dédiés aux usagers du cours : '*utilisateurs inscrits*', '*groupe*', etc. Nous avons associé à ces deux éléments les fonctionnalités pédagogiques : '*gérer les utilisateurs*' et '*gérer les groupes*' permettant d'inscrire les apprenants au cours et de créer des groupes d'apprenants.
 - L'onglet '*notes*' permet à l'enseignant de consulter les notes des étudiants suite à des activités évaluées. Cette fonction peut être importante pour le suivi des apprenants mais nous ne prenons pas compte d'elle parce qu'elle n'influence pas le contenu du cours ou ses paramètres.
 - Les onglets '*sauvegarde*', '*restauration*', '*importation*' et '*réinitialisation*', permettent de sauvegarder, restaurer ou initialiser un cours. Ils n'interviennent pas à la conception pédagogique d'un cours.
 - L'onglet '*banque de questions*' permet d'accéder à l'espace des questions pour les consulter ou pour en définir des nouvelles. Nous avons lui associer la fonction '*gérer la banque de question*'.

Nous représentons les fonctionnalités pédagogiques identifiées au sein du modèle fonctionnel. La figure A.7 illustre un extrait de ce modèle.

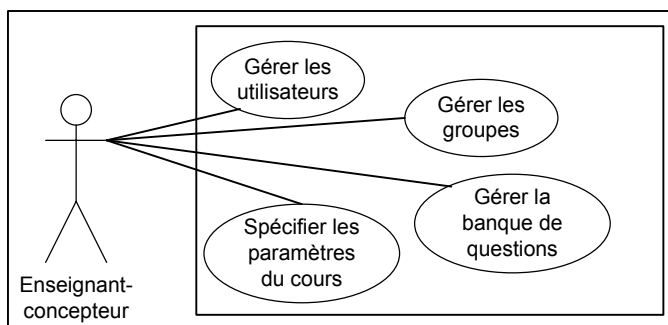


Figure A.7 : Extrait du modèle fonctionnel dédié à la zone 2

Analyse de la zone 4



La zone 4 (figure A.6) représente un ensemble de services mis à disposition des enseignants du cours par l'administration de la plateforme (le cas de la plateforme Umtice).

L'analyse fonctionnelle de cette zone a consisté à :

- analyser tous les widgets et les liens de cette zone. Certains d'eux sont liés à la conception pédagogique permettant de consulter le rapport des activités récentes, de chercher un forum, etc. Par exemple, le premier bloc (figure A.6) est dédié au forum défini par défaut sur Moodle : le forum des nouvelles.
- identifier des fonctionnalités pédagogiques telles que '*ajouter un nouveau sujet de discussion*', '*consulter les sujets de discussions de ce forum*' et '*consulter les discussions d'un sujet*'.
- représenter les fonctionnalités identifiées sur le modèle fonctionnel. La figure A.8 illustre un extrait du modèle fonctionnel dédié à cette zone.

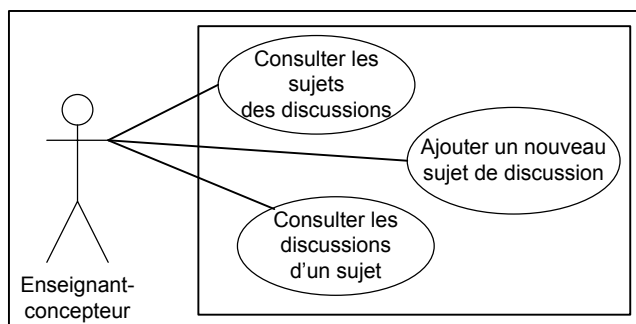


Figure A.8 : Extrait du modèle fonctionnel dédié à la zone 4



Nous montrons un peu plus loin que nous identifierons ces fonctionnalités pendant l'analyse fonctionnelle des interfaces dédiées au forum. Au sein du modèle fonctionnel complet, nous avons choisi que ces fonctionnalités seront dépendantes de la fonctionnalité globale '*ajouter un forum*' afin d'éliminer les représentations redondantes et de garantir une meilleure organi-

sation fonctionnelle.



Nous précisons que les zones 3 et 4 dépendent de l'instance de Moodle paramétrée par l'administration de la plateforme pour le profil d'enseignant-concepteur.

Analyse de la zone 2

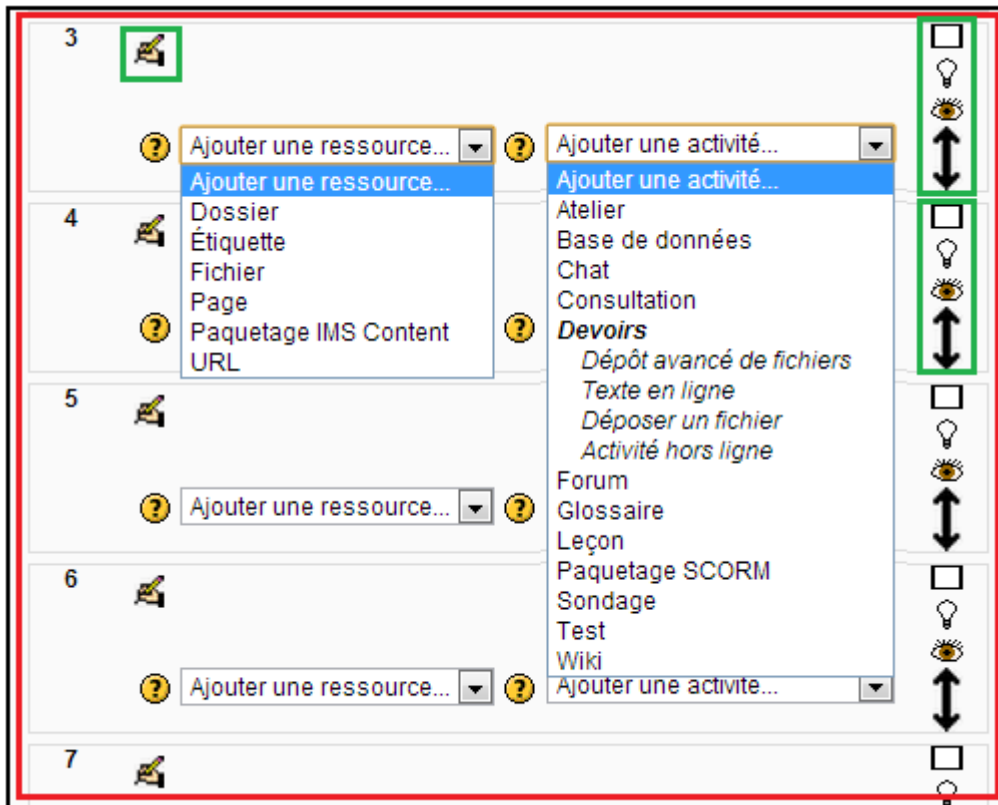


Figure A.9 : Extrait de la zone 2



La figure A.9 montre un extrait de la zone 2 de l'interface du cours sur Moodle. Cette zone dispose d'éléments répétitifs (les parties numérotées sur la figure A.9). L'analyse est ainsi centrée sur une seule partie.

L'analyse fonctionnelle de cette zone a consisté à :

- analyser les widgets d'une partie spécifique pour identifier les fonctionnalités pédagogiques associées. Nous avons identifié deux listes proposant d'ajouter une ressource ou d'ajouter une activité. Chaque élément de ces listes représente une spécification de ces deux fonctionnalités. Par exemple,
 - o L'élément forum désigne la fonctionnalité 'ajouter un forum' qui représente une spécification de la fonction 'ajouter une activité'.

B. Cas général : analyse fonctionnelle de l'interface du groupe/groupement

Contexte : les intervenants au cours représentent un élément de conception pédagogique. Nous présentons dans cette partie l'analyse fonctionnelle des interfaces de gestion des apprenants et des groupes d'apprenants. Nous nous intéressons aux zones liées à la conception de ces éléments telles que la zone 3 des interfaces des figures A.11 et A.12.

Objectif : identifier les fonctionnalités pédagogiques liées aux usagers du cours, en particulier les apprenants.

Application de l'analyse fonctionnelle :



L'interface (figure A.11) est accessible à partir de l'élément '*utilisateur inscrits*' de la zone 2. Cette zone permet d'afficher les utilisateurs déjà inscrits, les méthodes d'inscription et d'inscrire des utilisateurs (enseignants, apprenants, enseignants non éditeurs, etc.).

The screenshot shows the Moodle course interface for 'course1: 2 utilisateurs inscrits'. The top header (Zone 1) displays the course title and user information. The left navigation menu (Zone 2) includes options like 'Administration du cours', 'Paramètres', and 'Utilisateurs inscrits'. The main content area (Zone 3) features a table of registered users with columns for 'Prénom / Nom / Adresse de courriel', 'Dernier accès', 'Rôles', 'Groupes', and 'Méthodes d'inscription'. A table with 5 columns and 2 rows is visible:

Prénom / Nom / Adresse de courriel	Dernier accès	Rôles	Groupes	Méthodes d'inscription
studenta studenta studenta@univ-lemans.fr	204 jours 1 heure	Étudiant x	info1-ga x	Inscription manuelle dès le vendredi 23 mars 2012, 00:00
teacher1 teacher1 teacher1@gmail.com	3 secs	Enseignant		Inscription manuelle dès le mardi 16 octobre 2012, 00:00

Figure A.11 : Interface d'inscription des intervenants au cours

L'analyse fonctionnelle de l'interface de la figure A.11 a consisté à :

- décomposer cette interface en trois zones.
- analyser les widgets de la zone 3. Nous avons identifié un widget spécifique (un bouton) pour inscrire les intervenants du cours.
- représenter cette fonctionnalité pédagogique ('*inscrire les intervenants du cours*') dans le modèle fonctionnel.



L'interface (figure A.12) est accessible à partir de l'élément '*groupe*' de la zone 2 commune aux interfaces du cours. Elle est dédiée à la gestion des groupes intervenant dans un cours.

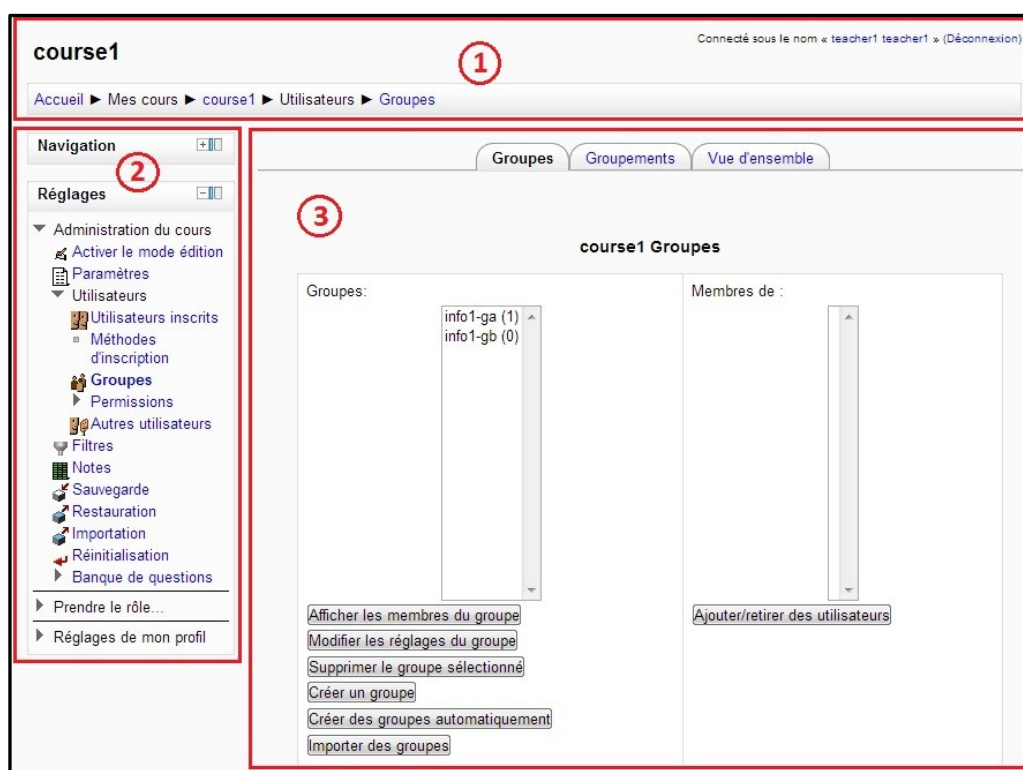




Figure A.12 : Interface de gestion des groupes

L'analyse fonctionnelle de l'interface de la figure A.12 a consisté à :

- décomposer l'interface en trois zones.
- analyser les widgets de la troisième zone (les zones 1 et 2 sont déjà analysées).

	En analysant la zone 3, nous avons constaté qu'elle est composée de trois parties : <i>groupes</i> , <i>groupements</i> et <i>vue d'ensemble</i> . Ce cas est particulier car chaque partie présente un écran de la zone 3. La figure A.12 représente la partie dédiée à l'élément <i>groupes</i> .
	L'analyse fonctionnelle de cette interface concerne ainsi ces trois parties. Nous analysons les widgets de ces parties.

- analyser la partie 'groupes' (zone 3, figure A.12):
 - o analyser les widgets de cette partie (les boutons).
 - o identifier les fonctionnalités pédagogiques associées à ces widgets telles que 'créer un groupement', 'afficher les groupes du groupement', etc.
 - o représenter ces fonctionnalités par le biais des cas d'utilisation au sein du modèle fonctionnel. La figure A.13 illustre ce modèle.

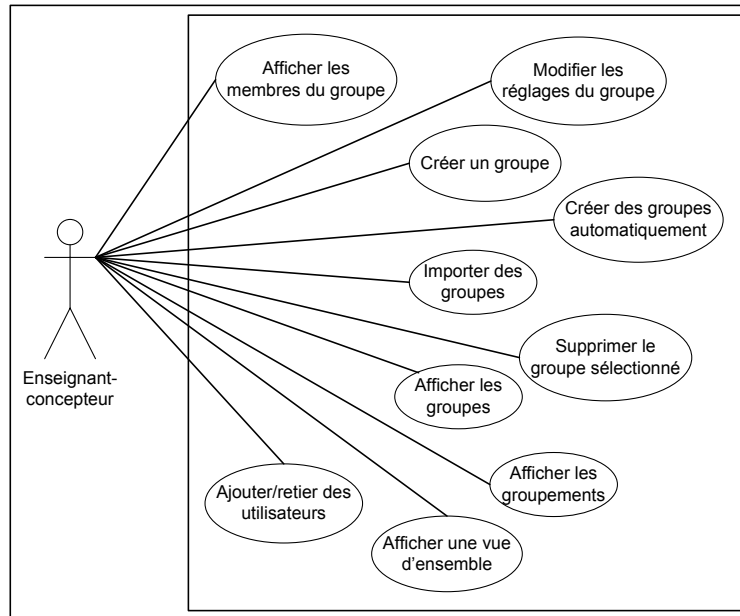


Figure A.13 : Modèle fonctionnel de gestion des groupes

course1 Connecté sous le nom « teacher1 teacher1 » (Déconnexion)

Accueil ► Mes cours ► course1 ► Utilisateurs ► Groupes ► Groupements

Navigation + | -

Réglages + | -

- Administration du cours
 - Activer le mode édition
 - Paramètres
 - Utilisateurs
 - Utilisateurs inscrits
 - Méthodes d'inscription
 - Groupes
 - Permissions
 - Autres utilisateurs
 - Filtres
 - Notes

Groupes Groupements Vue d'ensemble

Groupements

Groupement	Groupes	Activités	Modifier
groupement1	info1-ga	0	
groupement2	info1-gb	1	

Créer un groupement

Figure A.14 : Interface de gestion des groupements

- analyser la partie 'groupement' (zone 3 figure A.14):
 - o analyser les widgets et les composants de l'interface graphique de cette partie tels ceux encadrés en vert sur la figure.
 - o identifier les fonctionnalités pédagogiques associées à ces widgets telles que 'créer un groupement', 'afficher les groupes du groupement', etc.
 - o représenter ces fonctionnalités par le biais des cas d'utilisation au sein du modèle fonctionnel illustré par la figure A.15.
- analyser la partie 'vue d'ensemble' : nous avons analysé cette partie mais nous n'avons pas identifié de fonctionnalités pédagogiques.

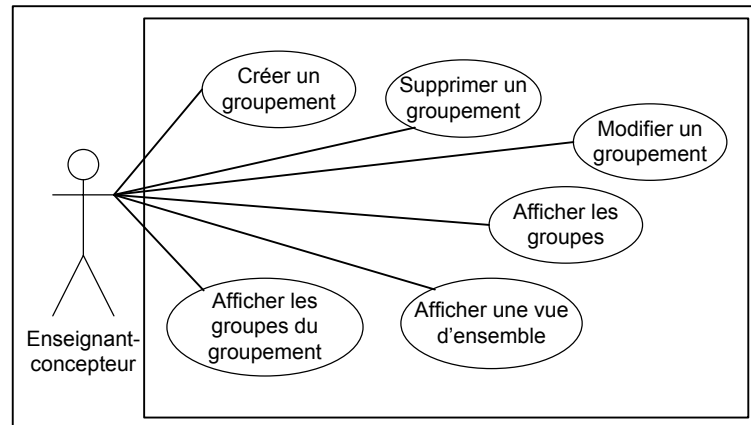


Figure A.15 : Modèle fonctionnel de gestion des groupements



Les deux modèles fonctionnels (figures A.13 et A.15) sont associés à une même interface. Ils représentent deux extraits du modèle fonctionnel final de cette interface. Cependant, ils disposent de certains cas d'utilisation partagés (dupliqués). Le modèle fonctionnel final représentera ces fonctionnalités sans redondance.



La fonctionnalité '*afficher les groupes du groupement*', identifiée pendant l'analyse de la partie '*groupement*', permet d'atteindre l'interface de la figure A.16. Nous avons appliqué l'analyse fonctionnelle sur cette interface.

- Nous avons identifié les fonctionnalités pédagogiques : '*ajouter des groupes au groupement*' et '*supprimer des groupes*'.
- Nous avons écarté la fonction '*retour aux groupements*' parce qu'elle est plutôt technique que pédagogique ; elle représente un lien entre deux écrans.
- Nous représentons ces fonctionnalités au sein d'un nouveau modèle fonctionnel (figure A.17) qui est ensuite associé au modèle fonctionnel du groupe.

Figure A.16 : Interface d'association entre les groupements et les groupes

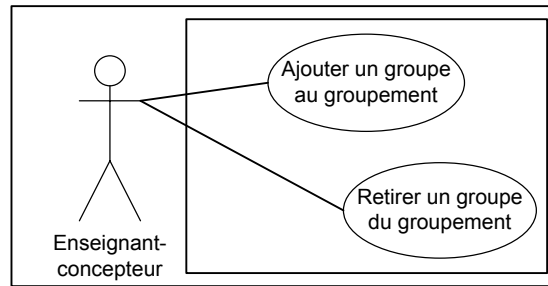


Figure A.17 : Modèle fonctionnel des associations entre les groupements et les groupes

C. Cas général : analyse fonctionnelle des interfaces du forum

Contexte : nous présentons dans cette partie l'application de l'analyse fonctionnelle sur les interfaces dédiées au forum. Cette application représente un exemple générique de l'analyse fonctionnelle des interfaces dédiées aux activités et aux ressources proposées par Moodle.

Objectif : identifier les fonctionnalités 'pédagogiques' dédiées aux forums.

Application de l'analyse fonctionnelle : comme pour l'analyse IHM-macro, nous avons suivi ici les deux stratégies d'analyse : l'analyse des interfaces de spécification d'un nouveau forum (figure A.18) et l'analyse des interfaces d'un forum existant (figure A.19).

C.1. Analyse des interfaces de spécification d'un nouveau forum

La capture d'écran montre l'interface 'Ajouter Forum' de Moodle. Elle est divisée en plusieurs sections :

- 1** : Le titre de la page 'course1' et le statut de connexion 'Connecté sous le nom « teacher1 teacher1 » (Déconnexion) sont situés en haut à droite.
- 2** : Le menu de navigation 'Administration du cours' est visible sur le côté gauche, incluant des options comme 'Paramètres', 'Utilisateurs', 'Notes', 'Sauvegarde', 'Restauration', 'Importation', 'Réinitialisation' et 'Banque de questions'.
- 3** : Le formulaire principal 'Ajouter Forum' est au centre, avec des champs pour 'Nom du forum*', 'Type de forum' (Forum standard pour utilisation générale), 'Introduction au forum*' (avec un éditeur de texte riche), 'Chemin', 'Mode d'abonnement' (Abonnement facultatif), 'Suivi des messages lus dans ce forum?' (Facultatif) et 'Taille maximale de l'annexe' (500Ko).

Figure A.18 : Extrait de l'interface d'ajout d'un forum

L'analyse fonctionnelle est appliquée sur l'interface de la figure A.18. Elle a consisté à :

- décomposer l'interface en trois zones.
- analyser les widgets de la zone 3 ; nous n'allons pas reprendre l'analyse des zones communes 1 et 2.
- identifier les fonctionnalités pédagogiques associées à ces widgets.



la zone 3 proposant un formulaire de spécification des attributs d'un forum ne dispose pas de fonctionnalités pédagogiques à l'exception de la fonctionnalité 'Afficher/cacher des éléments supplémentaires'. Nous avons considéré cette fonctionnalité comme pédagogique parce qu'elle permet de montrer de nouveaux attributs du forum.

- représenter les fonctionnalités identifiées au sein du modèle fonctionnel associé à forum.



Cette interface (figure A.18) sera analysée ultérieurement de façon plus fine, en particulier au sein de l'analyse IHM-micro. Elle décrit l'élément forum en termes d'attributs.

C.2. Analyse d'un forum existant

Nous avons appliqué l'analyse fonctionnelle sur l'interface de la figure A.19, en particulier la zone 3. Cette analyse a consisté à :

- analyser les widgets de la zone 3 ; cette zone dispose d'un seul widget (bouton).
- identifier si une fonctionnalité pédagogique est associée à ce widget ; nous avons identifié la fonctionnalité 'ajouter un nouveau sujet de discussion' que nous considérons comme pédagogique.
- représenter la fonctionnalité identifiée au sein du modèle fonctionnel associé à forum.



Figure A.19 : Interface d'un forum existant



La deuxième stratégie a permis l'identification de nouvelles fonctionnalités pédagogiques. Les deux stratégies sont complémentaires. Cela justifie notre proposition de suivre deux stratégies d'analyse d'une plateforme (c.f. section 4.1).



La fonctionnalité '*ajouter un nouveau sujet de discussion*' représente une sous-fonctionnalité d'*ajouter un forum*'. Elle permet d'atteindre l'interface de spécification d'une nouvelle discussion. Cela justifie la relation entre les interfaces forum et discussion, identifiée au sein de l'analyse IHM-macro.

D. Cas général : analyse fonctionnelle des interfaces de discussions

Contexte : nous présentons dans cette partie l'application de l'analyse fonctionnelle sur les interfaces dédiées aux discussions. Cette application représente un exemple générique de l'analyse fonctionnelle des interfaces des sous éléments des activités et des ressources pédagogiques sur Moodle.

Objectif : identifier les fonctionnalités 'pédagogiques' dédiées aux discussions.

Application de l'analyse fonctionnelle : nous avons suivi ici les deux stratégies d'analyse : l'analyse des interfaces de spécification d'une nouvelle discussion (figure A.20) et l'analyse des interfaces d'une discussion existante (figure A.21).

D.1. Analyse de l'interface de spécification d'un nouveau sujet de discussion

Nous avons appliqué l'analyse fonctionnelle sur l'interface de la figure A.20 qui a consisté à :

- décomposer cette interface en quatre zones (figure A.20).
- analyser les widgets des zones 2 et 3. Les zones 1 et 4 sont communes aux autres interfaces et ne disposent pas de fonctionnalités pédagogiques.



La zone 2 est commune aux interfaces de Moodle, mais spécifiquement pour cette interface, nous avons constaté qu'elle dispose de nouveaux éléments (des nouveaux liens situés dans la partie *réglages*)

- identifier les fonctionnalités pédagogiques associées aux widgets des zones 2 et 3.
 - o dans la zone 2, nous avons identifié les fonctionnalités : '*abonner ou désabonner l'enseignant au/du forum*' et '*afficher/modifier les abonnés à un forum*'.
 - o dans la zone 3, nous avons identifié la fonctionnalité : '*ajouter des fichiers*'.
- représenter les fonctionnalités identifiées au sein du modèle fonctionnel associé aux discussions.



la zone 3 propose un formulaire de spécification de nouvelles discussions. Elle décrit l'élément forum en termes d'attributs. Nous présentons un peu plus loin l'analyse IHM-micro des discussions qui sera centrée sur cette interface.

The screenshot shows the Moodle forum interface for 'course1'. The top navigation bar includes 'Accueil', 'Cours', 'course1', 'Section 1', and 'Forum'. The left sidebar contains 'Navigation' and 'Réglages' sections. The main content area is titled 'Nouveau sujet de discussion' and includes a text box for the forum's objective, a 'Sujet*' field, a 'Message*' field with a rich text editor, a 'Chemin:' field, an 'Abonnement' dropdown, an 'Annexe' section with an 'Ajouter...' button, and an 'Envoyer maintenant' checkbox. The right sidebar is titled 'Ajouter un bloc' and contains an 'Ajouter...' button. Red circles are placed over the course title, the navigation menu, the objective text box, and the 'Ajouter un bloc' sidebar.

Figure A.20 : Interface d'ajout d'un sujet de discussion



La figure A.20 représente l'interface de spécification de nouvelles discussions. Elle est accessible par la fonctionnalité 'ajouter un nouveau de sujet de discussion', identifiée dans l'analyse fonctionnelle spécifique aux forums. Cela confirme la relation entre les deux éléments *forum* et *discussion* sur le modèle IHM-macro.

D.2. Analyse des interfaces des discussions existantes

Nous avons également appliqué l'analyse fonctionnelle sur les interfaces des discussions existantes telles que celles des figures A.21, A.22 et A.23.

- La figure A.21 illustre l'interface de représentation des sujets de discussions postés sur un forum.
- La figure A.22 illustre les discussions réponses à un sujet de discussions particulier.
- La figure A.23 est à l'origine de la figure A.22 mais après avoir posté/spécifié de nouvelles discussions comme éléments de réponses aux discussions existantes.



Figure A.21 : Extrait de l'interface des nouveaux sujets de discussions

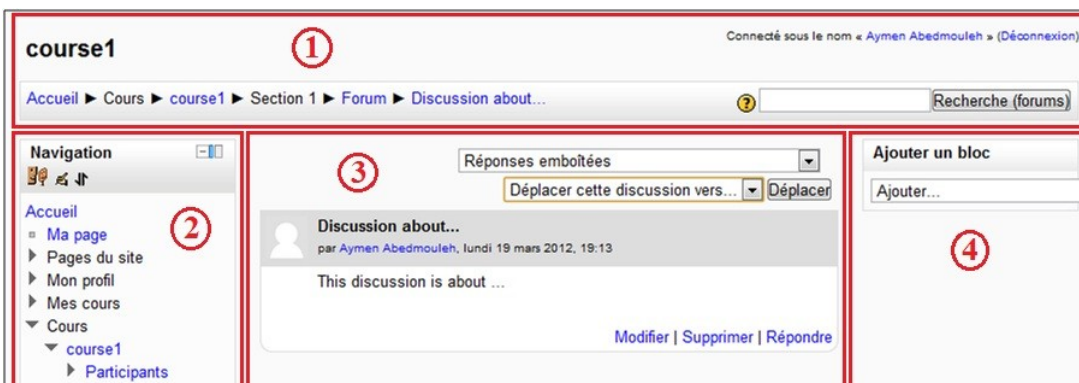


Figure A.22 : Extrait de l'interface d'un sujet de discussion

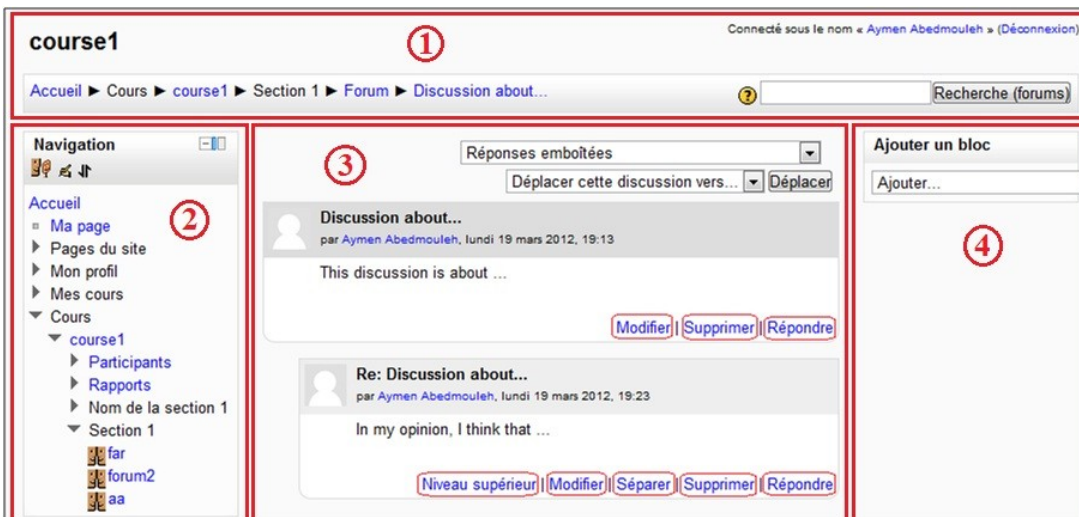


Figure A.23 : Extrait d'interface des discussions

L'analyse fonctionnelle a consisté à :

- décomposer les interfaces (des figures A.21, A.22 et A.23) en quatre zones.

- analyser les widgets (liens, boutons, icônes, etc.) embarqués dans ces interfaces, en particulier les zones numéro 3.
- identifier les fonctionnalités pédagogiques associées à ces widgets telles que ‘*ajouter une discussion*’, ‘*répondre à une discussion*’, ‘*séparer une discussion*’, etc.
- représenter les fonctionnalités identifiées au sein du modèle fonctionnel. La figure A.24 illustre ce modèle qui est ensuite associé au modèle fonctionnel final.

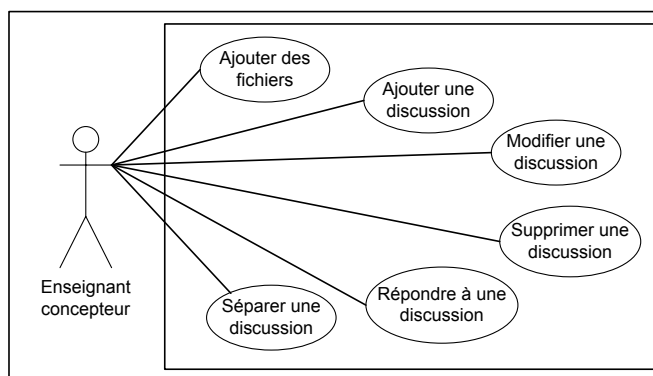


Figure A.24 : Modèle fonctionnel de la fonction ‘ajouter une discussion’



Les fonctionnalités dédiées à l’élément discussion représentent des sous-fonctionnalités de l’élément forum. Cela justifie la relation entre ces deux éléments.



Au cours de l’analyse IHM-macro, nous avons identifié des interfaces dédiées aux forums et aux discussions. Nous n’avons pas identifié des interfaces pour des sous-éléments des discussions.

Ensuite, l’application de l’analyse fonctionnelle sur ces interfaces a permis uniquement l’identification des fonctionnalités pédagogiques dédiées aux forums et discussions.

Cela justifie la non-existence ni des sous éléments des discussions ni des éléments intermédiaires entre forum et discussion.

E. Le modèle fonctionnel de Moodle

La figure A.25 illustre un extrait du modèle fonctionnel final. Ce modèle est le résultat de l’application de l’analyse fonctionnelle sur les interfaces de la plateforme Moodle, en particulier celles qui sont identifiées au sein de l’analyse IHM-macro.



Nous avons choisi de représenter séparément la partie modèle dédiée à la gestion des groupes afin d’éclaircir le modèle final. Toutefois, nous avons attribué la même couleur pour la fonction ‘*gérer les groupes*’ et le modèle associé.

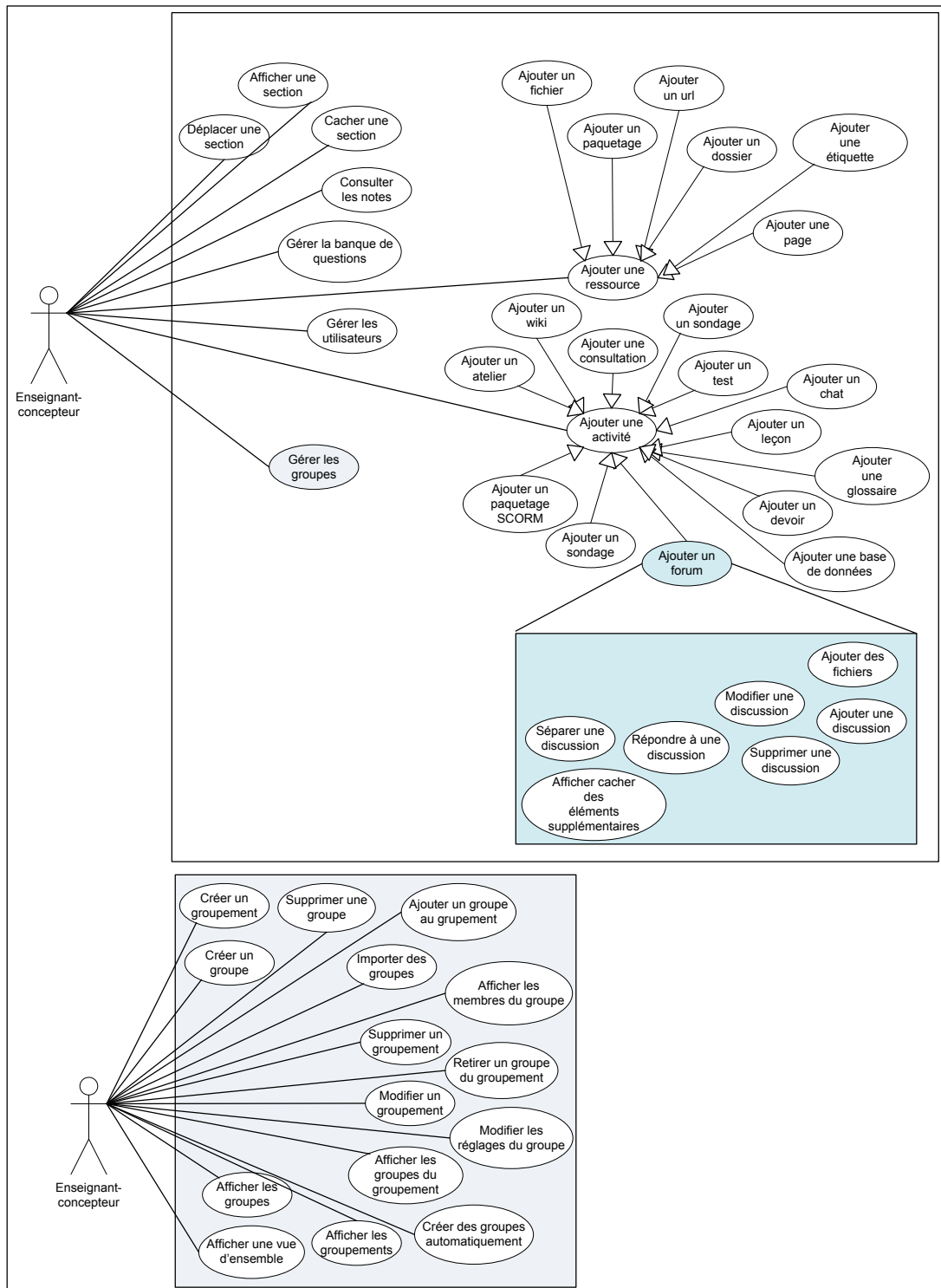


Figure A.25 : Extrait du modèle fonctionnel final

2.3. Analyse IHM-micro

Contexte : nous présentons dans cette partie l'application de l'analyse IHM-micro sur un ensemble cohérent d'exemples illustrant les spécificités et les particularités de cette analyse. Les cas similaires

ne seront pas représentés. Cette analyse concerne les interfaces de spécification ou de modification d'un élément ou d'un paramètre du cours.

Objectif : identifier les caractéristiques des éléments de conception pédagogique.

Application de l'analyse IHM-micro :

A. Cas général : Analyse IHM-micro de section

Contexte : les sections sont des éléments spécifiques pour la structuration des activités et des ressources dans un cours de Moodle.

Objectif : identifier les attributs 'pédagogiques' de l'élément section ainsi que leurs propriétés.

Application de l'analyse IHM-micro : cette application est centrée sur l'interface de la figure A.26.

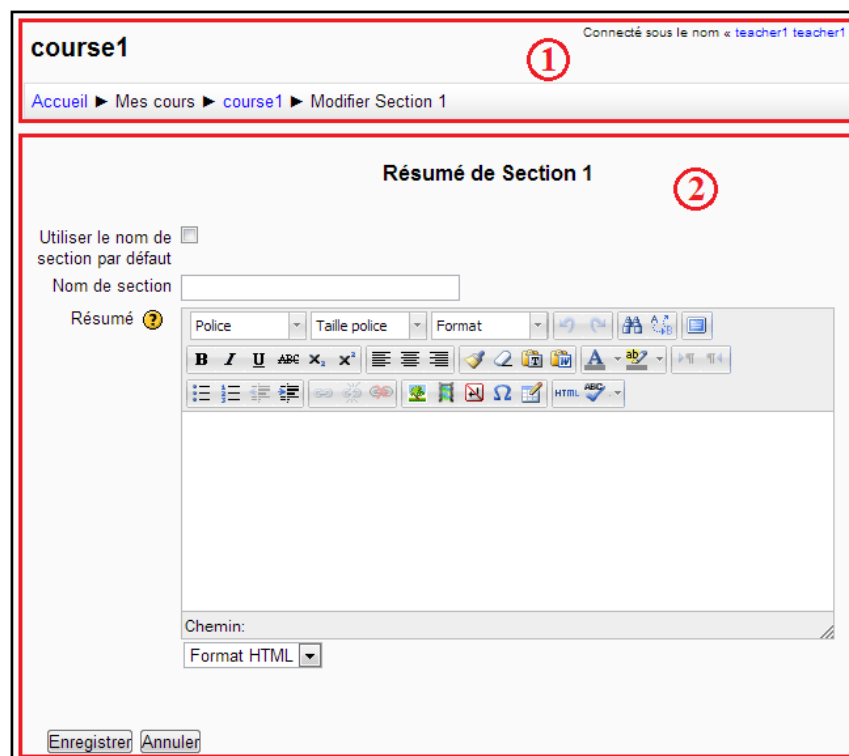


Figure A.26 : Interface d'une section

Cette analyse a consisté à :

- décomposer cette interface en deux zones.
- analyser les champs de description des sections dans chaque zone. La première zone ne dispose pas des éléments caractéristiques de sections. L'analyse est ainsi centrée sur la deuxième zone qui dispose de trois attributs.
- écarter les attributs que nous considérons comme technique plutôt que pédagogique tels que l'attribut '*utiliser le nom de section par défaut*' qui permet d'initialiser les attributs '*nom*' et '*résumé*' d'une section particulière.

- identifier les attributs pédagogiques (tels que ‘*nom de section*’ et ‘*résumé*’) ainsi que leur critère obligatoire ou facultatif, leur type et leur valeur par défaut.
- Représenter les attributs identifiés dans le modèles IHM-micro associé à l’élément *section*.

La figure A.27 illustre le modèle IHM-micro d’une section, composé des deux attributs : *Nom de section* et *résumé*.

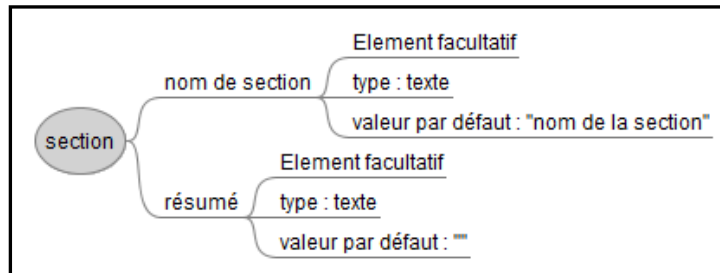


Figure A.27 : Modèle IHM-micro de section

B. Cas général : Analyse IHM-micro des forums

Contexte : les forums représentent une activité pédagogique sur Moodle. L’analyse IHM-micro du forum représente un cas général de l’analyse IHM-micro des activités et des ressources.

Objectif : identifier les attributs pédagogiques du forum ainsi que leurs types et leurs propriétés.

Application de l’analyse : malgré que plusieurs interfaces soient dédiées aux forums (telles que les figures A.18 et A.21), l’analyse IHM-micro est centrée seulement sur l’interface de la figure A.18 qui décrit le forum en termes d’attributs. Cette analyse a consisté à :

- décomposer l’interface en trois zones. Nous avons repris la décomposition en trois zones effectuée au sein de l’analyse fonctionnelle.
- analyser les zones de l’interface. Les zones 1 et 2, communes entre les interfaces de la plateforme, ne disposent pas des éléments caractéristiques d’un forum. L’analyse est ainsi centrée sur la troisième zone qui dispose de plusieurs attributs spécifiques au forum dont nous allons les analyser.
- écarter les attributs que nous jugeons plus orientés techniques que pédagogiques tels que ‘*la taille maximale de l’annexe*’, ‘*le nombre maximale de l’annexe*’, ‘*la durée de blocage*’, ‘*le numéro d’identification*’, etc.
- identifier les attributs ‘pédagogiques’ tels que : ‘*nom du forum*’, ‘*type du forum*’, ‘*introduction au forum*’, ‘*mode d’abonnement*’, ‘*suivi des messages lus dans ce forum*’, ‘*groupement*’ et ‘*visible*’.
- identifier pour chaque attribut les quatre propriétés suivantes : attribut facultatif ou requis par l’interface, le type de, les champs de valeur et la valeur par défaut.
- représenter les attributs du forum et leurs propriétés par un modèle IHM-micro. La figure A.28 illustre le modèle IHM-micro du forum.

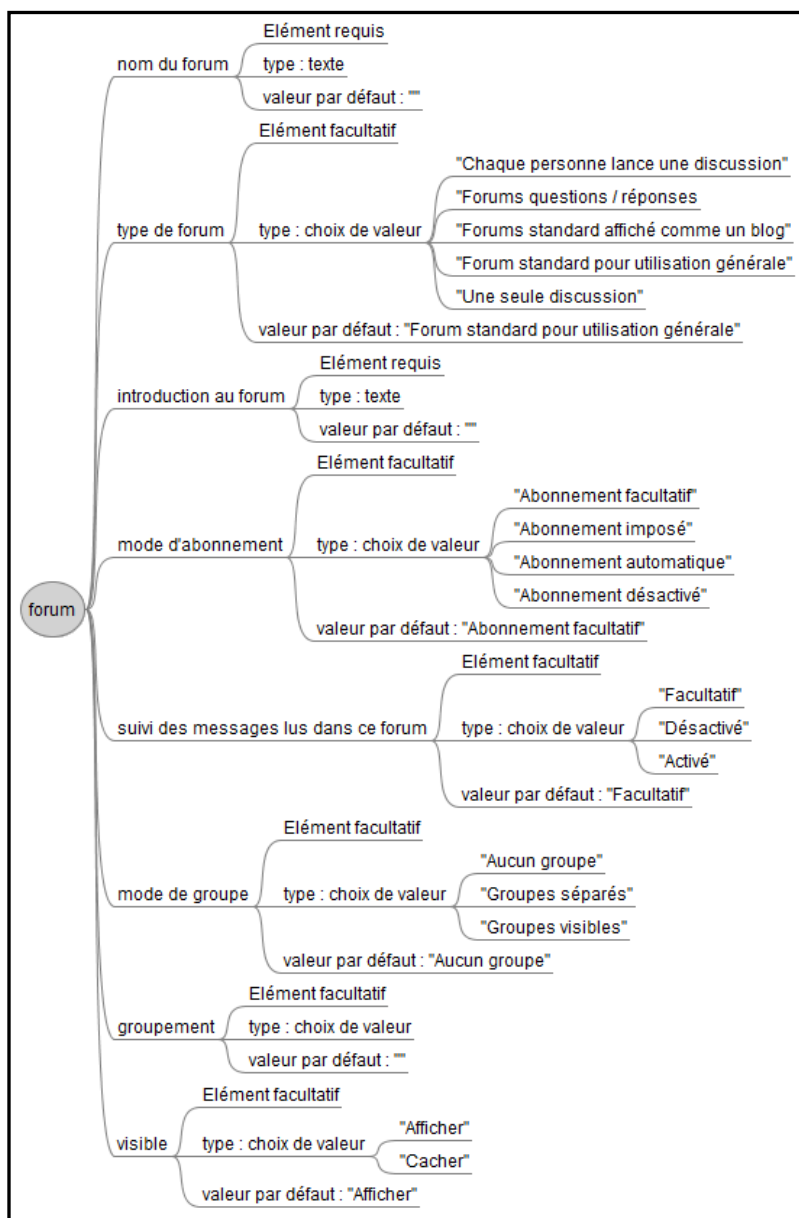


Figure A.28 : Modèle IHM-micro d'un forum

C. Cas général : analyse IHM-micro d'un élément dédié : le cas des discussions sous-éléments du forum

Contexte : pendant l'application de l'analyse IHM-macro, nous avons identifié des interfaces dédiées aux discussions, en relation avec les forums. Pendant l'application de l'analyse fonctionnelle sur ces interfaces, nous avons identifié des fonctionnalités spécifiques aux discussions qui représentent des sous-fonctionnalités au forum.

Objectif : identifier les attributs et les propriétés de l'élément *discussion*.

Application de l'analyse IHM-micro : cette analyse est similaire à l'analyse IHM-micro du forum. Elle a consisté à :

- décomposer l'interface de spécification des discussions en quatre zones (figure A.20). Nous avons repris la décomposition effectuée au sein de l'analyse fonctionnelle.
- analyser les zones de cette interface. L'analyse IHM-micro est centrée sur la zone 3 parce que les autres zones sont communes aux interfaces de la plateforme et elles ne disposent pas des éléments caractéristiques pour les discussions.
- écarter les attributs que nous considérons comme plus techniques que pédagogiques tels que l'attribut '*Envoyer maintenant*' qui permet d'envoyer une notification aux abonnés du forum.



Toutefois, l'analyste peut prendre en compte de cet attribut lorsqu'il est considéré plus pédagogique que technique par l'enseignant-concepteur collaborant avec lui.

- identifier les attributs pédagogiques et leurs propriétés. Deux attributs ont été identifiés : '*sujet*' et '*discussion*'.
- dresser le modèle IHM-micro d'une discussion. Dans ce modèle, nous retrouvons la description 'pédagogique' d'une discussion en termes d'attributs et de propriétés. La figure A.29 illustre ce modèle.

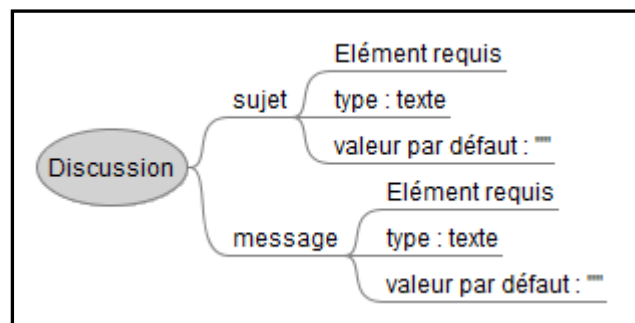


Figure A.29 : Modèle IHM-micro d'une discussion

D : Cas général : Analyse IHM-micro des groupes

Contexte : Les groupes d'apprenants représentent un élément de conception pédagogique représentant les apprenants d'un cours.

Objectif : identifier les attributs 'pédagogiques' de l'élément groupe ainsi que leurs propriétés.

Application de l'analyse IHM : l'analyse IHM-micro des groupes est aussi similaire à l'analyse du forum et de discussion. Elle a consisté à :

- décomposer l'interface de spécification d'un groupe en deux zones (figure A.30).
- analyser la deuxième zone qui décrit les éléments caractéristiques d'un groupe, en particulier la zone 2. La zone 1 ne dispose pas des éléments descriptifs des groupes.
- identifier les attributs 'pédagogiques' tels que '*Nom du groupe*' et '*Description du groupe*' dont nous avons identifié leur critère obligatoire ou facultatif, leur type et leur valeur par défaut.

- écarter les attributs plus orientés techniques que pédagogiques tels que ‘Clef d’inscription’.
- représenter le modèle IHM-micro des groupes (figure A.31).

Figure A.30 : Interface de spécification des groupes

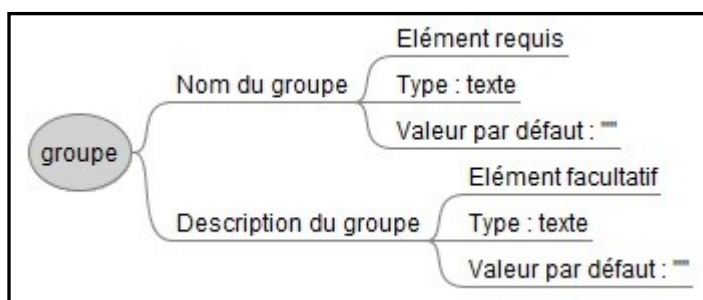




Figure A.31 : Modèle IHM-micro du groupe

E. Cas particulier 1 : Cas du test et feedback



Le test est un élément de conception pédagogique sur la plateforme Moodle. Au cours de l’analyse IHM-micro de son interface, nous avons identifié leurs attributs pédagogiques ainsi que leurs propriétés.

	<p>Nous avons identifié une particularité de cette interface, causée par la fonction ‘<i>ajouter 3 champs de feedback supplémentaires</i>’. Cette fonctionnalité permet d’ajouter plusieurs nouveaux attributs <i>feedback</i> au sein de l’interface du test. Ce cas est particulier parce que (1) feedback est un attribut de l’élément test, (2) le nombre de feedback est variable et (3) feedback possède également des attributs. La représentation du modèle IHM-micro, proposée par le processus que nous avons spécifié, ne permet pas de prendre en compte cette particularité.</p>
	<p>Nous avons ainsi proposé de représenter le feedback au sein d’un nouveau modèle IHM-micro séparément du test. Ensuite, l’étape de composition se charge de la définition des relations entre ces deux éléments/modèles. Les figures A.32 et A.33 illustrent les deux modèles IHM-micro associés au test et au feedback.</p>

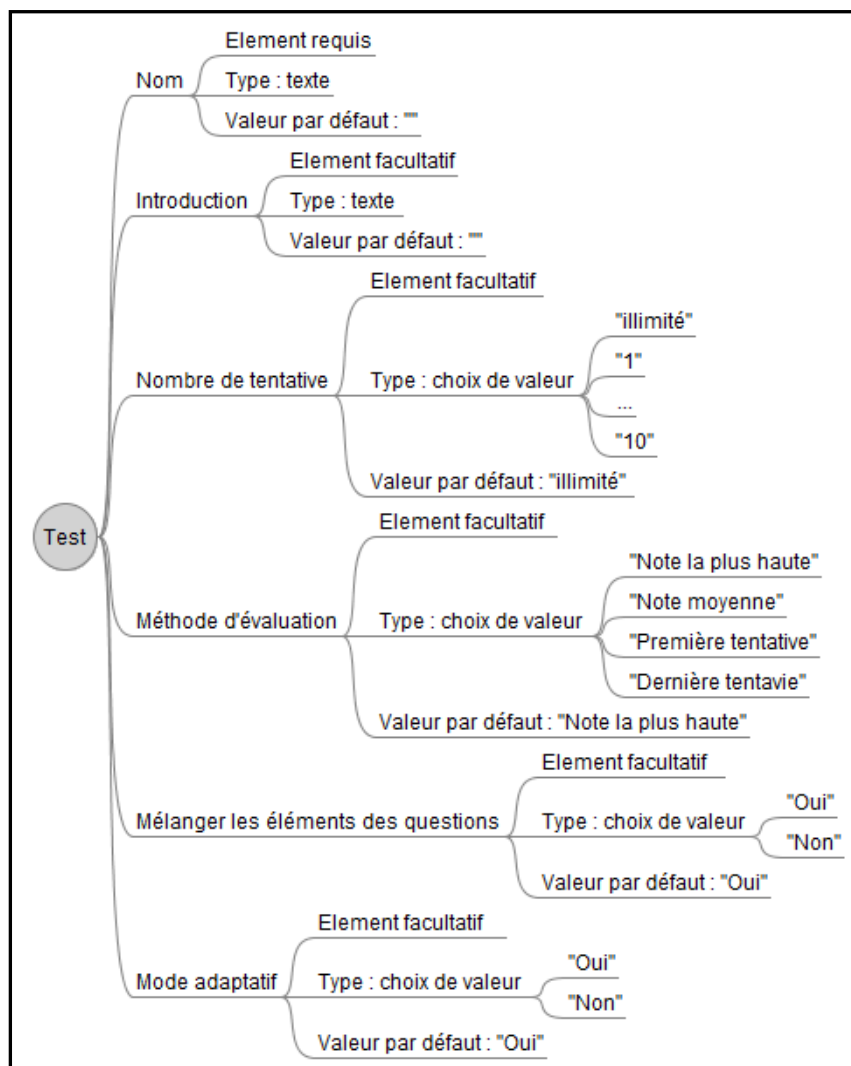


Figure A.32 : Modèle IHM-micro d'un test

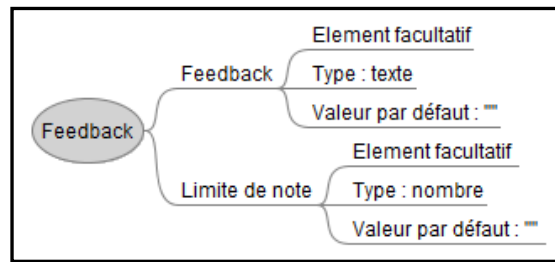


Figure A.33 : Modèle IHM-micro d'un feedback d'un test



G. Cas particulier 2 : cas de question et page

Figure A.34 : Interface de structuration de pages de question



Nous avons identifié un deuxième cas particulier pendant l'analyse IHM-micro des interfaces dédiées à l'élément test, en particulier celle de la figure A.34 qui présente un test composé de plusieurs pages de questions.

Une page regroupe un ensemble de questions. Elle représente un point intermédiaire entre l'élément 'test' et 'question'. Nous avons voulu prendre en compte de cette spécification en menant une analyse fine afin d'identifier les caractéristiques de cet élément (page).

	L'analyse IHM-micro a montré que 'page' ne possède pas d'attributs ou de propriétés.
	Son modèle IHM-micro sera représenté par un simple nœud intitulé 'page'. Il sera utilisé pendant la phase de composition présentée ultérieurement.

2.4. Composition du modèle IHM

Contexte : la composition est la dernière étape au sein de l'analyse centrée IHM.

Objectif : formaliser le modèle IHM final du langage de conception pédagogique de la plateforme Moodle en associant les modèles IHM-micro.

Application de la composition : nous présentons dans cette partie un ensemble cohérent d'exemples mettant en évidence les spécificités ainsi que les particularités de l'application de la composition.

A. Cas général 1 : composition des éléments cours et section

Définition de l'association (cours - section)

Rappel :

- Au sein de l'analyse IHM-macro, nous avons identifié des interfaces dédiées aux éléments cours et section ainsi qu'une relation entre elles.
- Au cours de l'analyse fonctionnelle, en particulier celle de l'interface principale du cours, nous avons identifié la fonctionnalité '*modifier le résumé*'. Cette fonctionnalité permet d'amener à l'interface spécifique aux sections. Nous n'avons pas identifié d'autres fonctionnalités intermédiaires et dédiées à d'autres éléments.
- Au cours de l'analyse IHM-micro, nous avons défini deux modèles IHM-micro dédiés aux cours et section.

Tous ces constats confirment l'association (cours-section). Leurs modèles IHM-micro doivent être associés.

Identification des multiplicités de l'association (cours - section)

- Le cours sur Moodle est composé de plusieurs sections. Le nombre de section est variable ; il est soit spécifié par l'enseignant pendant la création de l'espace cours, soit fixé par l'administration du cours (le cas de la plateforme *Umtice* au sein de l'université du Maine). Par exemple, la figure A.9 représente un cours composé de plusieurs sections. Cela permet de déduire que la branche de la multiplicité [cours– section] est d'ordre (1,*).

- En fixant une section particulière, cette dernière n'appartient qu'à un seul cours. Moodle ne propose aucun moyen ou outil pour associer une section à deux ou plusieurs cours. Ainsi, la branche de multiplicité [section – cours] est d'ordre (1,1).

La figure A.35 représente l'association entre les deux éléments cours et section ainsi que les multiplicités qui sont représentées en suivant la représentation du schéma entité-association de la méthode Merise.

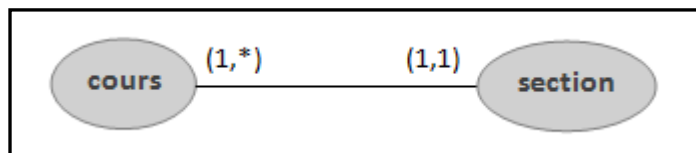


Figure A.35 : La composition : cours et section

B. Cas général 2 : composition des éléments section et forum

Définition de l'association (section - forum)

Rappel :

- Au cours de l'analyse IHM-macro, nous avons identifié des interfaces dédiées aux forums, sections et cours. Nous avons aussi défini une relation entre cours et section et aussi entre cours et forum. Les forums sont directement associés au cours.
- Au cours de l'analyse fonctionnelle, nous avons identifié des fonctionnalités pédagogiques dédiées à section et forum. Nous n'avons pas identifié des fonctionnalités intermédiaires dédiées à d'autres éléments.
- Au cours de l'analyse IHM-micro, nous avons identifié l'organisation du cours en sections. Le forum est un élément de conception pédagogique pouvant être défini au sein d'une section de cours. Nous avons également identifié deux modèles IHM-micro dédiés aux sections et aux forums.



Ainsi, le modèle IHM-micro du forum est associé au modèle d'une section et non au modèle du cours.



Dans Moodle, toutes les activités et les ressources des cours sont définies au sein des sections. Le cas général 2 (composition de section et forum) représente un exemple générique des compositions entre section et les activités et ressources.

Contrairement au modèle IHM-macro qui associe les sections, les activités et les ressources à l'élément cours, la composition prouvera que les activités et les ressources seront associées à *section* et non au *cours* parce qu'elles sont définies au sein des sections. Ainsi, leurs modèles IHM-micro seront associés au modèle de la *section* et non à celui du *cours*.

Identification des multiplicités de l'association (section - forum)

- En prenant un cours de Moodle, nous avons tout d'abord vérifié qu'il est possible d'associer plusieurs forums à une section. Cela nous a permis de déduire que la branche de la multiplicité [section – forum] est d'ordre (0,*).
- En fixant un forum particulier, nous avons déduit qu'il n'appartient qu'à une seule section. Moodle ne propose aucun moyen pour associer un forum à deux sections ou plusieurs. Ainsi, la branche de multiplicité [forum – section] est d'ordre (1,1).

La figure A.36 représente la composition entre les deux éléments *forum* et *section*.

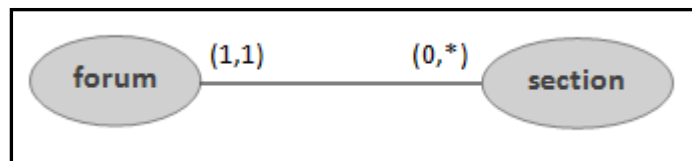


Figure A.36 : La composition : forum et section

C. Cas général 3 : composition des éléments forum et discussion**Définition de l'association (forum - discussion)**

Rappel :

- Au cours de l'analyse IHM-macro, nous avons identifié une relation entre les interfaces dédiées aux forums et aux discussions.
- Au sein de l'analyse IHM fonctionnelle, les fonctionnalités dédiées aux discussions ont représentées des sous-fonctionnalités de l'élément forum. Aucune fonctionnalité intermédiaire et dédiée à un autre élément n'était identifiée.

Cela confirme l'association entre forum et discussion. Leurs modèles IHM-micro, spécifiés au sein de l'analyse IHM-micro, doivent être associés.

Identification des multiplicités de l'association (forum - discussion)

- En choisissant un forum particulier, il est possible d'ajouter plusieurs discussions. Cela nous permet de déduire que la branche de la multiplicité [forum – discussion] est d'ordre (0,*).
- Dans la plateforme Moodle, une discussion, si elle existe, est associée à un unique forum. Cela nous permet de déduire que la branche de multiplicité [discussion – forum] est d'ordre (1,1).

La figure A.37 illustre la composition entre forum et discussion.

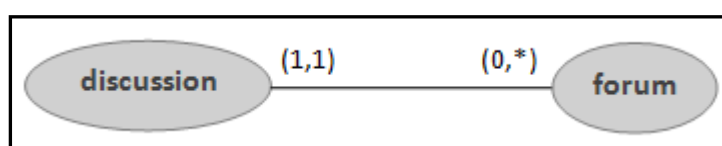





Figure A.37 : La composition : forum et discussion

D. Cas particulier 1 : composition du test – feedback

Définition de l'association (test - feedback)

	Ce cas particulier est identifié pendant l'analyse IHM-micro de l'interface du test. Il est dû à la fonctionnalité 'ajouter 3 champs de feedback' permettant d'ajouter plusieurs feedback au sein d'un test.
	Ce cas nous a amené à spécifier un modèle IHM-micro dédié au feedback séparément du modèle IHM-micro du test.
	Dans cette étape (la composition), les modèles IHM-micro du test et feedback doivent être associés.

Identification des multiplicités de l'association (test - feedback)

- Chaque test dispose de plusieurs feedbacks. Il en résulte que la branche de la multiplicité [test – feedback] est d'ordre (1,*).
- Par définition, chaque feedback est associé à un seul test. Moodle ne propose pas de moyens pour associer un feedback à deux ou plusieurs tests. Ainsi, la branche de multiplicité [feedback – test] est d'ordre (1,1).

La figure A.38 illustre la composition entre les éléments test et feedback.

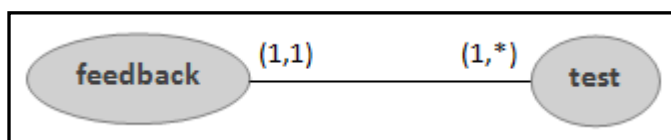





Figure A.38 : La composition : feedback et test

E. Cas particulier 2 : Les associations (test – page) et (page – question)

Définition des associations (test – page) et (page – question)

	Pendant l'analyse IHM-micro des interfaces du test, nous avons identifié un cas particulier qui concerne l'élément 'page' au sein du test. Les pages permettent de structurer les questions d'un test.
	Nous avons modélisé l'élément page par un modèle IHM-micro composé uniquement du nœud central page ; il ne possède pas d'attributs.
	Ce modèle doit être associé au modèle IHM-micro du test et également à celui d'une question.

Identification des multiplicités des associations (*test – page*) et (*page – question*)

- Par défaut, toutes les pages sont créées pour un test unique. La branche de multiplicité [page – test] est d'ordre (1,1).
- Au sein d'un test particulier, il est possible d'y ajouter plusieurs pages. Ainsi, la branche de la multiplicité [test - page] est d'ordre (0,*).
- Au sein d'une page, nous pouvons ajouter plusieurs questions. Ainsi, la branche de la multiplicité [page – question] est d'ordre (0,*).
- Quant à la multiplicité de la branche [question - page], Moodle permet d'ajouter aléatoirement une question déjà définie à une ou plusieurs page(s). Ainsi, la branche de multiplicité [question – page] est d'ordre (1,*).

La figure A.39 illustre une vue simplifiée de la composition des modèles de section, test, feedback, page et question.

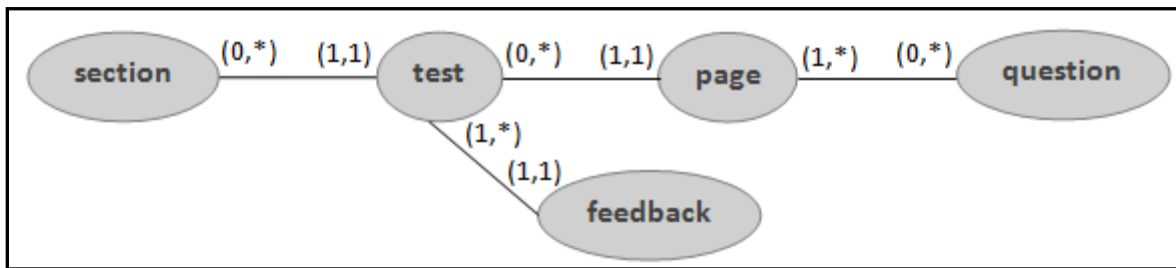


Figure A.39 : Vue générale des compositions entre section, test, feedback, page et question

F. Le modèle IHM de Moodle

La composition est la dernière étape de l'analyse centrée IHM permettant de formaliser le modèle IHM réduit de la conception pédagogique d'une plateforme. Ce modèle est complètement défini lorsque toutes les associations entre les modèles IHM-micro sont définies.

La figure A.40 illustre une vue simplifiée de ce modèle. Chaque élément/nœud de ce modèle désigne son modèle IHM-micro. Nous avons utilisé un outil gratuit pour faire les schémas de *mindmap*. Il s'agit de l'outil *FreeMind*. Cependant, cet outil ne permet pas de représenter les multiplicités. Nous avons ainsi choisi d'annoter les nœuds du mindmap avec des multiplicités.

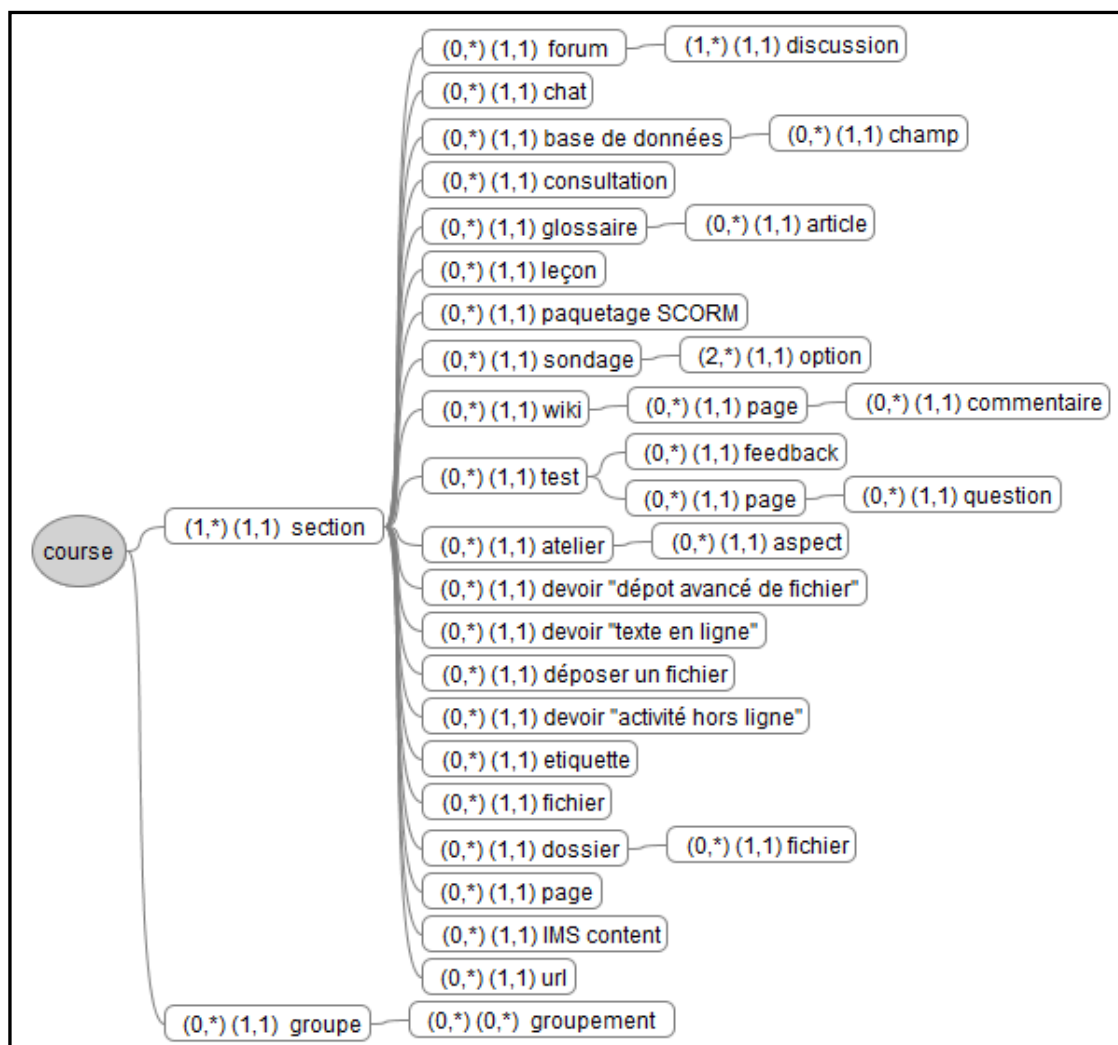


Figure A.40 : Vue globale du modèle IHM

3. Analyse technique : analyse de la base de données

Contexte : nous présentons dans cette partie l'application de l'analyse technique sur la plateforme Moodle, plus spécifiquement l'analyse de la base de données. Cette analyse est restreinte aux tables/colonnes en relation avec la conception pédagogique.

Objectif : spécifier le modèle conceptuel de données réduit à la conception pédagogique.

Application de l'analyse technique : nous présentons tout d'abord l'application générique de l'analyse technique sur Moodle. Ensuite, nous présentons un ensemble cohérent d'exemples pertinents dans le but d'illustrer toutes les spécificités et les particularités de cette analyse. Les cas similaires et redondants à ces exemples ne seront pas représentés. Cependant, nous représentons certaines difficultés rencontrées et certaines particularités de cette analyse.

L'analyse technique a consisté à :

- parcourir toutes les tables de la base de données de Moodle.

- analyser sémantiquement les titres des tables pour identifier les tables relevant de la conception pédagogique d'une situation d'apprentissage en termes d'activités et de ressources pouvant être intégrées.



Nous avons constaté que certaines tables renseignent sur des activités bien spécifiques telles que le cas des tables suivantes liées à l'activité forum : *forum*, *forum_discussions*, *forum_posts*, *forum_queue*, *forum_read*, *forum_subscriptions* et *forum_track_prefs*. Nous avons répertorié ces tables en plusieurs catégories (chacune est dédiée à un même concept) telles que la catégorie du forum.

- écarter les tables qui portent sur des spécifications techniques ou qui ne possèdent pas d'enregistrements. Ces tables sont créées pour des finalités différentes à la conception pédagogique (gestion des utilisateurs, les traces des apprenants, etc.) telles que les tables *course_completions*, *course_completion_aggr_methd*, *course_completion_criteria*, *course_display*, etc.



La base de données que nous avons analysée comporte les données d'une banque de cours. Elle dispose, à travers un nombre important de tables, de nombreuses activités et types de ressources dédiées à la spécification de situations d'apprentissage.

- écarter les tables non liées à la conception pédagogique parmi celles répertoriées au sein des catégories identifiées, telles que les tables de suivi des traces apprenants dans un forum (*forum_read* et *forum_track_prefs*, etc.).



Une première hypothèse de l'analyse technique est d'encadrer les tables concernées. La liste de ces tables n'est pas toujours définitive. D'autres tables en relation peuvent être identifiées via des dépendances entre les tables de la base de données (par exemple via les clés étrangères, etc.).

- identifier les attributs liés à la conception pédagogique embarqués dans les tables identifiées.
- spécifier le modèle logique de données dédié aux tables identifiées.
- générer le modèle conceptuel de données réduit de la base de données de Moodle par application des règles de retro-conception.

A. Cas général : Analyse de la table forum



Nous avons identifié des tables dédiées à l'élément 'forum' par l'analyse sémantique de leurs titres. Nous avons identifié une table principale intitulée *forum* et d'autres tables dont leurs titres sont composés de la concaténation du mot '*forum*' et une spécification de cet élément tels que *forum_discussions*, *forum_posts*, *forum_queue*, etc.

Nous avons écarté certaines tables non liées à la conception pédagogique d'un forum telles que *forum_queue*, *forum_read*, *forum_subscriptions*, *forum_track_prefs*, etc. Ces tables ne possèdent pas d'enregistrements malgré l'existence de plusieurs forums sur la plateforme. Elles sont spécifiques pour d'autres finalités telles que les traces des étudiants, etc.

Nous avons identifié les trois tables : *forum*, *forum_discussions* et *forum_posts*.

id	course	type	name	intro	introformat	assessed	assesstimestart	assesstimefinish	scale	maxbytes	maxattachments	forcesubscribe	trackingtype	rsstype
1	3	news	Forum des nouvelles	Nouvelles diverses et annonces		0	0	0	0	1	0	1	1	0
3	2	news	News forum	General news and announcements		0	0	0	0	0	0	1	1	0
4	2	general	Forum xyz	<p>forum xyz</p>		1	0	0	0	0	512000	9	0	1
5	2	general	forum 7	<p>Forum 7</p>		1	0	0	0	0	512000	9	0	1

Figure A.41 : Ecran de la table forum

Rappel : l'analyse de la table *forum* (figure A.41) consiste à analyser ses champs de point de vue conception pédagogique.

Objectif : identifier les champs les plus significatifs (les attributs pédagogiques) portant sur la définition des forums (ou des éléments caractéristiques d'un forum) et les clés étrangères des tables liées à la conception pédagogique.

Application de l'analyse : cette analyse a consisté à :

- écarter les champs (attributs) qui sont définis par défaut (valeur = 0 par exemple) ou des détails techniques à bas niveau tels que les champs :
 - de gestion d'affichage : *introformat*.
 - de gestion de temps : *assesstimestart* et *assesstimefinish*, *timemodified*, etc.
 - d'autres éléments : *blockperiod*, *completionreplies*, *studentlogs*, etc.
- identifier les champs principaux pour la spécification des forums tels que les champs : *type*, *name*, *intro*. Nous avons également identifié la clé étrangère *course* qui indique le cours associé au forum.
- représenter les champs identifiés dans le modèle de la table *forum*. La figure suivante illustre ce modèle.

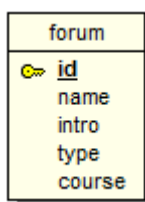


Figure A.42 : Modèle réduit de la table 'forum'



L'analyse technique des tables *forum* et *forum_discussions* (présentées ci-après) représente un exemple typique de l'analyse technique des tables dédiées aux activités et aux ressources (ainsi que les éléments) proposées par Moodle.

B. Cas général : Analyse de la table forum discussions

id	course	forum	name	firstpost	userid	groupid	assessed	timemodified	usermodified	timestart	timeend
1	2	4	discuss forum xyz	1	2	-1	1	1323952034	2	0	0
2	2	4	discuss 22	5	2	-1	1	1323948085	2	0	0
3	3	6	zzzzzzzz	10	2	-1	1	1324026634	2	0	0
4	3	6	eeeeeeeeee	11	2	-1	1	1324026701	2	0	0



Figure A.43 : Aperçu de la table forum_discussions

Contexte : la figure A.43 illustre la table *forum_discussions* visualisant un extrait des discussions postées sur un forum.

Objectif : identifier les champs les plus significatifs (les attributs pédagogiques) portant sur la définition des discussions et les clés étrangères des tables liées à la conception pédagogique.

Application de l'analyse : cette analyse a consisté à :

- analyser syntaxiquement et sémantiquement les champs de la table *forum_discussions*.
- identifier les attributs 'pédagogiques' tels que 'name' et des clés étrangères des tables liées à la conception pédagogique telles que les clés *course*, *forum* et *groupid* référençant les tables *course*, *forum* et *groupe*.
- écarter les champs techniques comme *assessed*, *timemodified*, *usermodified*, etc.
- représenter les champs identifiés dans le modèle de la table *forum_discussions*. La figure A.45 illustre ce modèle.

	Dans cette analyse, nous avons rencontré certaines difficultés concernant le champ <i>firstpost</i> . Les enregistrements correspondant à ce champ sont des valeurs numériques différentes.
	<p>Nous avons ainsi analysé le code source qui interroge la table <i>forum_discussions</i>, en particulier les instructions d'ajout d'une nouvelle discussion au sein d'un forum.</p> <p>La figure A.44 illustre un extrait de la fonction 'ajouter une discussion au sein d'un forum' spécifique à l'insertion de nouveaux enregistrements au sein de la table <i>forum_discussions</i>.</p> <p>En analysant ce code source, nous avons déduit que <i>firstpost</i> représente les sujets des discussions postés au sein d'un forum. Ainsi, nous avons tenu en compte de cet attribut. Nous réexpliquerons ci-après son utilité lorsque nous analysons la table <i>forum_posts</i>.</p>

```

function forum_add_discussion($discussion, $mform=null, &$message=null, $userid=null) {
    global $USER, $CFG, $DB;
    // The first post is stored as a real post, and linked
    // to from the discuss entry.
    $forum = $DB->get_record('forum', array('id'=>$discussion->forum));
    $cm     = get_coursemodule_from_instance('forum', $forum->id);
    $post = new stdClass();
    $post->discussion = 0;
    $post->parent     = 0;
    $post->subject    = $discussion->name;
    $post->message     = $discussion->message;
    $post->messageformat = $discussion->messageformat;
    $post->message trust = $discussion->message trust;
    $post->forum       = $forum->id; // speedup
    $post->course      = $forum->course; // speedup
    ...
    $post->id = $DB->insert_record("forum_posts", $post);
    // Now do the main entry for the discussion, linking to this first post
    $discussion->firstpost = $post->id;
    $discussion->timemodified = $timenow;
    ...
    $post->discussion = $DB->insert_record("forum_discussions", $discussion);
    // Finally, set the pointer on the post.
    $DB->set_field("forum_posts", "discussion", $post->discussion, array("id"=>$post->id));
    return $post->discussion;
}

```

Figure A.44 : Code source de la fonction ajout des discussions



Figure A.45 : Modèle réduit de la table 'forum_discussions'

C. Cas particulier : Analyse de la table forum_posts

id	discussion	parent	userid	created	modified	mailed	subject	message	messageformat	message trust	attachment	totalscore	mailnow
1	1	0	2	1323947819	1323947819	0	discuss forum xyz	<p>message discuss forum xyz</p>	1	0		0	0
2	1	1	2	1323947847	1323947847	0	Re: discuss forum xyz	<p>message Re: discuss forum xyz</p>	1	0		0	0
3	1	2	2	1323947866	1323947866	0	Re: discuss forum xyz	<p>message re re Re: discuss forum xyz</p>	1	0		0	0
4	1	1	2	1323947889	1323947889	0	Re: discuss forum xyz	<p>Message 2 Re: discuss forum xyz</p>	1	0		0	0

Figure A.46 : Aperçu de la table forum_posts

Contexte : la figure A.46 illustre la table *forum_posts* spécifique à la collecte des réponses et des messages des discussions.

Objectif : identifier les champs de définition des discussions et les clés étrangères des tables liées à la conception pédagogique.

Application de l'analyse : la table *forum_posts* est analysée de façon similaire aux tables précédentes (*forum* et *forum_discussions*). L'analyse a consisté à :

- identifier les éléments de spécification d'une discussion ou d'une réponse à une discussion. Nous avons identifié les champs : 'subject' et 'message'.
- identifier les clés étrangères spécifiques à des tables relevant de la conception pédagogique telle que la clé étrangère 'discussion'.
- écarter les champs techniques (d'affichage, utilisateurs, etc.) tels que 'created', 'modified', 'mailed', 'messagetrust', etc.
- représenter les champs identifiés dans le modèle de la table 'forum_posts'. La figure A.47 illustre ce modèle.



Nous allons présenter les particularités de l'analyse de cette table et nous allons expliquer davantage le champ 'firstpost' de la table 'forum_discussions'.

Difficulté :



La difficulté rencontrée concerne la prise en compte de l'élément *parent*. Cet élément, prenant des valeurs numériques, permet d'ordonner les discussions ou les messages réponses à une discussion particulière. Nous avons abouti à ce constat suite à deux vérifications.



Dans un premier temps, nous avons spécifié des nouvelles discussions sur un forum particulier. Ensuite, nous avons interprété les changements effectués sur le champ *parent*. Dans un deuxième temps, nous avons analysé le code source. Ce travail a consisté à localiser la partie du code concernée pour la table *forum_posts* et *forum_discussions*. Plusieurs fonctions ont été analysées et interprétées telles que la fonction d'ajout d'une discussion et la fonction de modification d'une discussion. Ces instructions se situent au sein du fichier "...moodle/mod/forum/lib.php".

La fonction *forum_add_discussion* (figure A.44) interroge les deux tables *forum_discussion* et *forum_posts*. Elle permet d'insérer les nouveaux sujets de discussion au sein de ces tables et également d'insérer les réponses aux discussions au sein de la table *forum_posts*.

L'élément *parent* est initialisé à zéro en insérant une nouvelle discussion. Ensuite, en ajoutant des réponses à une discussion particulière, ces dernières sont insérées dans la table *forum_posts* et l'élément *parent* prend la valeur de l'identifiant de la discussion et ainsi de suite. L'élément *parent* permet de structurer les discussions et leurs éléments de réponses. Nous avons ainsi tenu en compte de cet élément.

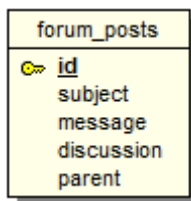


Figure A.47 : Modèle réduit de la table 'forum_posts'



En ce qui concerne l'élément *firstpost*, nous avons analysé la correspondance entre les deux tables. Nous avons rendu compte que *firstpost* est une clé étrangère qui renseigne sur des enregistrements au sein de la table *forum_posts*. Nous avons ainsi tenu en compte de ce champ.

D. Cas général : Analyse des tables dédiées aux groupes

Contexte : outre les tables dédiées aux activités et aux ressources proposées par Moodle, nous avons identifié des tables dédiées à la gestion des participants au cours, des groupes d'apprenants, etc. Nous présentons dans cette partie l'analyse des tables de gestion des groupes, en particulier les tables *groupings*, *groups*, *groupings_groups* et *groups_members*. La figure A.48 illustre un extrait de la table *groups* qui s'intéresse aux groupes d'apprenants.

id	courseid	name	description	descriptionformat	enrolmentkey	pict
1	1	groupe TD11	<p>Le groupe TD11 est ...</p>	1		
2	1	groupe TD12	<p>Le groupe TD12 est ...</p>	1		
3	1	groupe TD2	<p>Le groupe TD2 est ...</p>	1		

Figure A.48 : Extrait de la table groups

Objectif : identifier les champs de définition des groupes et les clés étrangères des tables liées à la conception pédagogique.

Application de l'analyse : l'analyse technique de ces tables a consisté à :

- identifier les éléments 'pédagogiques' pour la définition des groupes et les clés étrangères spécifiques à des tables liées à la conception pédagogique.
 - Au sein de la table 'groups', nous avons identifié les deux champs 'name' et 'description' et la clé étrangère 'courseid'.
 - Au sein de la table 'grouping', nous avons identifié les deux champs 'name' et 'description' et la clé étrangère 'course'.
 - Au sein de la table 'groupings_groups', nous avons identifié les deux clés étrangères 'groupingid' et 'groupid'. Cette table ne dispose pas d'éléments de conception pédagogique. Nous avons tenu compte de cette table parce qu'elle est fondamentale pour la définition des liens entre les deux tables 'groups' et 'groupings'.

- écarter les champs techniques tels que ‘*descriptionformat*’, ‘*enrolmentkey*’, ‘*picture*’, etc.
- représenter les champs identifiés dans les modèles associés. La figure A.49 illustre ces modèles.



Quant à la table *groups_members*, elle représente les membres d’un groupe. Nous avons analysé cette table mais nous sommes rendu compte qu’elle n’intervient pas à la conception d’un cours parce que nous considérons que les utilisateurs sont gérés par l’administration de la plateforme.

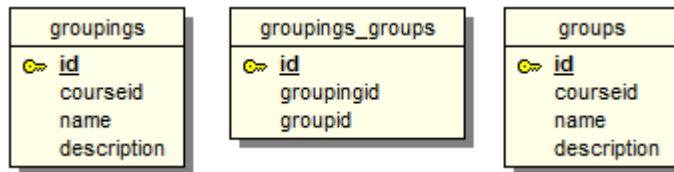


Figure A.49 : Modèles techniques des tables *groupings*, *groupings_groups* et *groups*

E. Cas particulier : spécificités des tables de Moodle

mdl23_folder
mdl23_forum
mdl23_forum_discussions
mdl23_forum_posts
mdl23_forum_queue
mdl23_forum_read
mdl23_forum_subscriptions
mdl23_forum_track_prefs
mdl23_glossary
mdl23_glossary_alias
mdl23_glossary_categories
mdl23_glossary_entries
mdl23_glossary_entries_categories
mdl23_glossary_formats
mdl23_grade_import_values
mdl23_grade_items
mdl23_grade_items_history
mdl23_grade_letters
mdl23_grade_outcomes
mdl23_grade_outcomes_courses
mdl23_grade_outcomes_history
mdl23_grade_settings
mdl23_groupings
mdl23_groupings_groups
mdl23_groups
mdl23_groups_members
mdl23_imsco
mdl23_label
mdl23_lesson
mdl23_lesson_answers
mdl23_lesson_attempts
mdl23_lesson_branch
mdl23_lesson_grades
mdl23_lesson_high_scores
mdl23_lesson_pages
mdl23_lesson_timer
mdl23_license
mdl23_log

Figure A.50 : Extrait des tables de la base de données de Moodle

La première particularité de l'application de l'analyse technique sur la plateforme Moodle concerne les titres des tables de la base de données. L'analyse sémantique de ces titres a permis l'identification de certaines tables dédiées à la conception générale du cours sur cette plateforme.




Nous avons remarqué un point commun entre ces titres. Pour chaque concept (forum par exemple) nous avons identifié une table principale dédiée et d'autres tables dont leurs titres sont formés par la concaténation du concept principal (*forum* par exemple) avec une spécification de la table comme '*forum_discussions*', '*forum_posts*', '*forum_queue*', '*forum_read*', '*forum_subscriptions*' et '*forum_track_prefs*'.

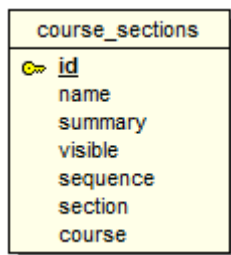
Cette particularité de la base de données de Moodle a facilité l'application de notre analyse technique parce que nous avons rapidement identifié les tables liées aux éléments de conception pédagogique. La figure A.50 montre un extrait des tables de la base de Moodle dont les parties encadrées sont les tables dédiées à un forum, à un glossaire et à une leçon.

E. Cas particulier : analyse de la table *course_sections*

id	course	section	name	summary	summaryformat	sequence	visible
1	2	0	NULL	NULL	1	10	1
2	3	0	Les lipides	Etudes des éléments de base	1	32,35	1
3	3	1	Interactions moléculaires		1		1
4	3	2	Les glucides	Les éléments de base et les macromolécules	1	36	1

Figure A.51 : Ecran de la table '*course_sections*'

	Nous avons appliqué l'analyse technique sur la table ' <i>course_sections</i> ' (figure A.51) afin d'identifier les champs les plus significatifs pour la définition des sections. Nous avons identifié les champs ' <i>name</i> ', ' <i>summary</i> ', ' <i>visible</i> ', ' <i>sequence</i> ', ' <i>section</i> ' et ' <i>course</i> '. La figure A.52 illustre le modèle retenu de la table ' <i>course_sections</i> '.
	Les deux difficultés principales de cette analyse concernent les deux champs ' <i>section</i> ' et ' <i>sequence</i> '. Nous sommes ainsi amenés à identifier leur usage.
	<p>Pour cela, nous avons analysé le code source, en particulier les instructions interrogeant la table '<i>course_sections</i>' et les requêtes d'insertion des nouvelles sections suite à l'ajout d'un nouveau cours. Nous avons déduit que :</p> <ul style="list-style-type: none"> - l'élément '<i>section</i>' ne représente pas une clé étrangère vers la table '<i>course_sections</i>'. Il représente le rang de la section au sein du cours d'identifiant '<i>course</i>'. - l'élément '<i>sequence</i>' référence des éléments de la table '<i>course_modules</i>' et il représente les modules spécifiés au sein d'une section. <p>Nous avons ainsi tenu en compte de ces éléments.</p>

Figure A.52 : Modèle réduit de la table *course_sections*

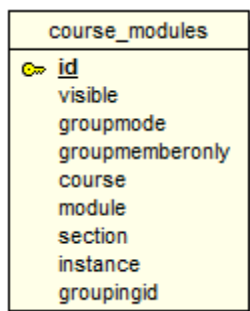
F. Cas particulier : analyse de la table *course_modules*

id	course	module	instance	section	idnumber	added	score	indent	visible	visibleold	groupmode	groupingid	groupmemberonly	completion
10	2	7	3	1	NULL	1323947763	0	0	1	1	0	0	0	0
11	2	7	4	14		1323947797	0	0	1	1	0	0	0	0
12	2	7	5	14		1323949934	0	0	0	1	0	0	0	0
32	3	7	6	2	NULL	1335279779	0	0	1	1	0	0	0	0

Figure A.53 : Ecran de la table '*course_modules*'

Application de l'analyse technique : l'analyse technique de la table '*course_modules*' (figure A.53) a consisté à :

- écarter les champs techniques tels qu'*idnumber*', '*added*', '*indent*', etc.
- identifier les champs 'pédagogiques' (*visible*', '*groupmode*', '*groupmemberonly*') et clés étrangères essentielles tels que '*course*', '*module*', '*groupingid*', etc.
- représenter les champs retenus dans le modèle de la table '*course_modules*'. La figure A.54 illustre ce modèle.

Figure A.54 : Modèle réduit de la table '*course_modules*'



Difficulté :



L'analyse sémantique des titres des tables de la base de données de Moodle a permis l'identification de la table '*course_modules*' (figure A.53). Le titre de cette table indique qu'elle représente les modules proposés par Moodle et/ou utilisés au sein d'un cours.

Cette table ne porte pas sur des éléments de conception pédagogique d'un cours sur Moodle. Elle ne permet que :

- de définir les liens entre certaines tables,
- d'associer les modules aux cours concernés,

	<ul style="list-style-type: none"> - de définir la visibilité des modules au sein d'un cours, - de définir les groupes associés aux modules, etc. <p>Les champs de cette table sont :</p> <ul style="list-style-type: none"> ▪ soient des clés étrangères telles que <i>'course'</i>, <i>'module'</i>, <i>'section'</i>, <i>'groupid'</i>, etc. ▪ soient des propriétés telles que <i>'visible'</i> et <i>'groupmode'</i>. ▪ soient des champs techniques tels que <i>'added'</i>, <i>'score'</i>, <i>'indent'</i>, <i>'completion'</i>, etc
	<p>Au cours de l'analyse de la table <i>course_modules</i>, nous avons rencontré des difficultés pour déterminer les usages des éléments <i>'module'</i> et <i>'instance'</i>. Plus spécifiquement, il s'agit de déterminer si (1) <i>module</i> représente une clé étrangère de la table <i>'modules'</i>, <i>'course_modules'</i> ou autres et (2) si <i>'instance'</i> représente une clé étrangère ou un attribut spécifique.</p>
	<p>Ces difficultés ont nécessité une analyse technique plus approfondie pour déterminer l'usage et la finalité de ces éléments. Nous avons ainsi analysé le code source qui interroge cette table.</p> <p>Nous avons déduit que <i>module</i> représente une clé étrangère vers la table <i>modules</i>. Cette table regroupe les activités et les ressources proposées par Moodle.</p> <p>L'identification du rôle de l'élément <i>'instance'</i> était plus difficile. Il représente une clé étrangère : il s'agit de l'identifiant de la table correspondante au module (table <i>modules</i>) d'identifiant <i>module</i>. Nous avons pris compte de cet élément. La figure A.55 illustre la relation entre les tables via les champs <i>module</i> et <i>instance</i>.</p>

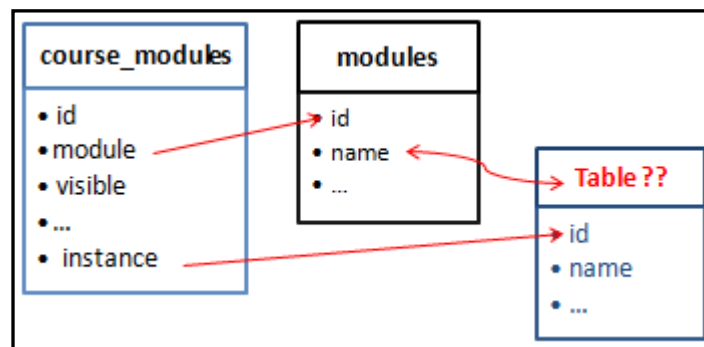


Figure A.55 : Schéma récapitulatif des clés étrangères


	<p>La figure A.56 illustre un exemple de relation entre les trois tables <i>'modules'</i>, <i>'course_modules'</i> et <i>'forum'</i>.) Le premier enregistrement de la table <i>'courses_modules'</i> possède les champs suivants : <i>'course'</i> égale à 2, <i>'module'</i> égale à 7 et <i>'instance'</i> qui vaut 3. Le module 7 représente le module forum défini au sein de la table <i>'modules'</i> par un identifiant égale à 7. Ainsi, l'instance 3 représente le <i>'forum'</i> d'identifiant égale à 3, défini au sein de la table <i>'forum'</i>, est qui est associé au cours d'identifiant 2.</p>
---	---

Table course_modules							
id	course	module	instance	section	idnumber	added	score
10	2	7	3	1	NULL	1323947763	0
11	2	7	4	14		1323947797	0
12	2	7	5	14		1323949934	0
32	3						0
33	5						0
34	4	3	2	news	News forum	General news and announcements	0
35	3						0
36	3	4	2	general	Forum xyz	<p>forum xyz</p>	0
37	3						0
38	5	5	2	general	forum 7	<p>Forum 7</p>	0
39	5	7	5	news	News	General news	0

Table modules						
id	name	version	cron	lastcron	search	visible
1	assignment	2010102600	60	1327663409		1
2	chat	2010080302	300	1327663409		1
3	choice	2010101300	0	0		1
4	data	2010100101	60	0		1
5	feedback	2010112302	0	0		0
6	folder	2010101400	0	0		1
7	forum	2010111500	60	1327663409		1
8	glossary	2010111501	0	0		1
9	imscp	2010101400	0	0		1
10	label	2010080300	0	0		1
11	lesson	2010122200	0	0		1

Figure A.56 : Exemples de relations entre les tables modules, course_modules et forum

G. Le modèle technique de Moodle

Le modèle technique est le résultat de l'application de l'analyse technique. Après avoir identifié les tables relevant de la conception pédagogique ainsi que leurs attributs, nous avons spécifié le modèle logique de données. Nous avons défini les relations entre les tables en tenant en compte des clés étrangères. Ensuite, nous avons appliqué les règles de retro-conception (*reverse engineering*) afin de générer le modèle conceptuel de données réduit. Ce modèle est spécifique à la conception pédagogique sur Moodle et représente le modèle technique. Les clés étrangères sont remplacées par des associations entre les concepts tout en spécifiant les multiplicités requis. La figure A.57 illustre un extrait du modèle technique.

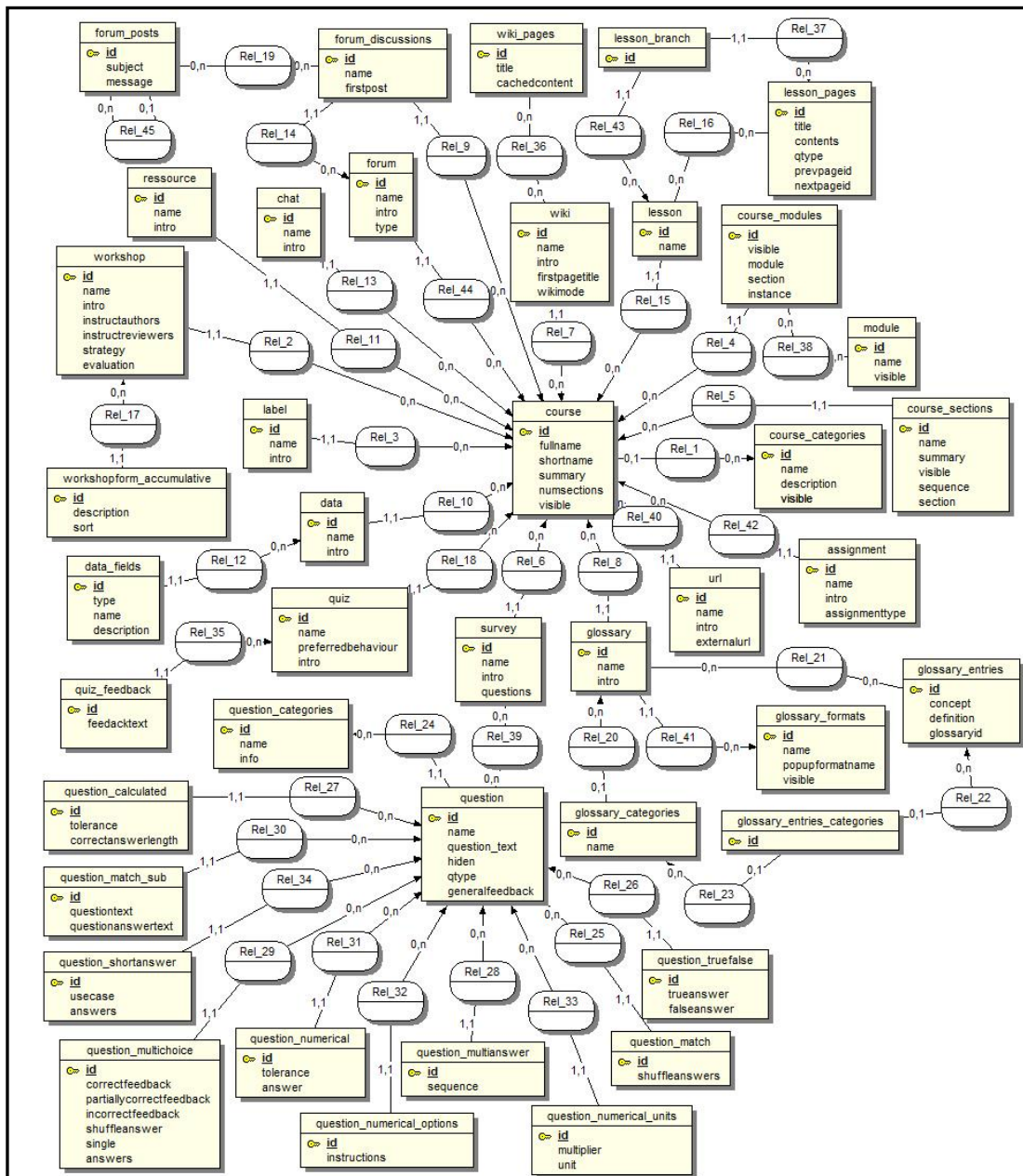


Figure A.57 : Modèle conceptuel de données réduit de la plateforme Moodle



Remarque 1 :

L'analyse technique de la plateforme Moodle était une tâche très fastidieuse. Nous avons surmonté plusieurs difficultés techniques pendant l'analyse des tables de la base de données. La spécification du modèle logique de données n'était pas aisée à cause des clés étrangères non précisément définies tels que le cas de la clé étrangère 'instance' de la table 'course_modules'.

Remarque 2 :

Le modèle logique de données représente à la fois les champs dédiés à la conception pédagogique d'un élément et les clés étrangères. Ainsi, il sera difficile de distinguer entre ces deux types de champs. Le format du modèle conceptuel de données (MCD) permet de cacher la mauvaise conception de la base de données ou les clés étrangères imprécisément définies.

	<p>Cela représente un des arguments du choix du format du modèle technique.</p> <p>Remarque 3 :</p> <p>Plusieurs tables de Moodle sont utilisées pour la définition des relations telles que les tables ‘<i>groupings_groups</i>’, ‘<i>wiki_subwikis</i>’, ‘<i>glossary_entries_categories</i>’, etc. Leurs champs sont composés de leur identifiant et des clés étrangères. Ces tables ne portent pas sur des éléments relevant de la conception pédagogique. Pour cela, elles doivent être écartées du modèle technique.</p> <p>La génération du modèle technique permet d’écarter ces tables. Cela justifie encore notre proposition de formaliser le modèle technique par le biais du modèle conceptuel de données afin de cacher les détails et les contraintes techniques.</p>
--	--

4. Confrontation et formalisation

Rappel : l’application des deux premières étapes du processus (les analyses centrées IHM et technique) ont permis la spécification de deux modèles de conception pédagogique de la plateforme Moodle selon deux point de vue différents : IHM et technique.


Objectif : confronter les modèles IHM et technique pour formaliser le modèle final du langage de conception pédagogique de la plateforme Moodle.

Application de la confrontation et la formalisation : nous présentons dans cette partie un ensemble cohérent d’exemples pour illustrer la confrontation entre les éléments des modèles par la vérification de l’identification et de la bonne définition des éléments relevant de la conception pédagogique. La confrontation concerne également les attributs et les propriétés de ces éléments. Nous appliquons le processus de vérification requis, parmi ceux définis au sein du chapitre 4, lorsqu’un cas de différence entre les deux modèles est identifié.


A. Cas général : confrontation de la représentation du forum

En ce qui concerne l’élément forum, la confrontation a consisté à :



- retrouver la représentation du modèle du forum sur le modèle IHM, plus spécifiquement son modèle IHM-micro.
- retrouver l’entité dédiée au ‘*forum*’ sur le modèle technique.



	<p>A ce stade, nous avons identifié le modèle IHM-micro du forum sur le modèle IHM et son entité sur le modèle technique. Cela confirme que forum, élément relevant de la conception pédagogique, est bien identifié.</p>
---	---

- représenter le forum par le biais d’une méta-classe, intitulée ‘*forum*’, au sein du métamodèle de conception pédagogique de Moodle.

	Ensuite, nous nous sommes focalisés à la vérification de ses attributs.
---	---

- retrouver les attributs du forum sur le modèle IHM qui sont : ‘*nom du forum*’, ‘*type de forum*’, ‘*introduction du forum*’, ‘*mode d’abonnement*’, ‘*suivi des messages lus dans ce forum*’, ‘*mode de groupe*’ et ‘*visible*’.
- retrouver les attributs du forum sur le modèle technique, en particulier dans l’entité forum. Les attributs sont : *name*, *intro* et *type*.

	Certains attributs (tels que ‘ <i>mode d’abonnement</i> ’, ‘ <i>suivi des messages lus dans ce forum</i> ’, ‘ <i>mode de groupe</i> ’ et ‘ <i>visible</i> ’) ne sont pas identifiés via l’analyse technique.
	Ce cas particulier nous amène à appliquer le processus de vérification des attributs. Nous allons ainsi : <ul style="list-style-type: none"> - reprendre l’analyse IHM-micro des interfaces dédiées aux forums qui a confirmé l’identification des attributs en question. - reprendre l’analyse technique dans le but d’identifier les attributs manquants. Nous avons ainsi identifié les attributs ‘<i>forcesubscribe</i>’, ‘<i>trackingtype</i>’, <i>rsstype</i> qui n’étaient pas identifié précédemment.



	Cependant, dans le modèle technique, les entités dédiées au forum ne disposent pas des attributs correspondants à ‘ <i>mode de groupe</i> ’ et à ‘ <i>visible</i> ’.
	Nous avons ainsi analysé les instructions spécifiques à l’enregistrement des nouveaux forums. La figure A.58 illustre un extrait des codes analysés. Nous avons déduit que ces attributs sont enregistrés au sein de l’entité ‘ <i>course_modules</i> ’ et ils sont désignés par les champs ‘ <i>groupmode</i> ’ et ‘ <i>visible</i> ’. Cette entité est associée à l’entité forum.

```
function forum_get_separate_modules($courseid) {

    global $CFG,$DB;
    $forummodule = $DB->get_record("modules", array("name" => "forum"));

    $sql = 'SELECT f.id, f.id FROM {forum} f, {course_modules} cm WHERE
    f.id = cm.instance AND cm.module =? AND cm.visible = 1 AND cm.course = ?
    AND cm.groupmode ='.SEPARATEGROUPS;
    return $DB->get_records_sql($sql, array($forummodule->id, $courseid));
}
```

Figure A.58 : Extrait du code associé aux attributs visible et groupmode

	<p>La question qui se pose : est ce que ces attributs doivent être représentés au sein de la méta-classe 'forum' ou dans une autre méta-classe (par exemple, une nouvelle méta-classe spécifique aux modules du cours) ? Au sein du modèle IHM, il n'existe pas d'élément spécifique aux modules.</p>
	<p>Nous avons ainsi choisi de représenter ces attributs au sein de la méta-classe du 'forum' ainsi que les attributs précédemment identifiés.</p>

- représenter les attributs de la méta-classe 'forum'. La figure A.59 illustre cette méta-classe.

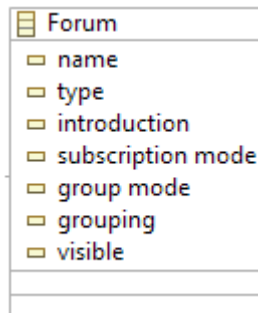





Figure A.59 : Méta-classe du forum


	<p>L'exemple de la confrontation spécifique à l'élément forum représente un cas générique de la confrontation des activités, des ressources et d'autres éléments pédagogiques (groupes, sections, etc.) proposés par Moodle.</p>
---	--


B. Cas particuliers de la confrontation

Cas particulier 1 :



	<p>En confrontant les deux modèles IHM et technique, nous avons remarqué tout d'abord que les éléments identifiés possèdent une nomination similaire mais pas identique.</p>
	<p>Le processus proposé n'impose pas une nomination spécifique. Pour cela, nous avons choisi la nomination la plus significative. Nous reprenons la nomination des éléments du modèle IHM pour la définition de la couche graphique des outils de conception.</p>

Cas particulier 2 :

	<p>Au sein du modèle IHM, les activités et les ressources sont associés à l'élément section. Par contre, sur le modèle technique, elles sont associées à l'élément section et également à l'élément cours.</p>
---	--

	<p>Ce cas particulier nous a amené à vérifier les relations entre les activités et les ressources d'une part et la section et le cours d'autre part. Nous adaptons la représentation IHM. Pour cela, nous avons associé les activités et les ressources à l'élément section seulement.</p>
---	--

Cas particulier 3 :

	<p>Certains attributs des éléments du modèle IHM ne sont pas identifiés dans les entités correspondantes du modèle technique (MCD) tels que le cas de l'attribut '<i>visible</i>'. Cet attribut est identifié dans tous les éléments, spécifiques aux activités et aux ressources, du modèle IHM par contre il n'a pas été identifié dans les entités correspondantes au niveau modèle technique.</p>
	<p>Pour cela, nous avons mené de nouvelles analyses IHM et technique avant d'adopter d'autres analyses, en particulier l'analyse du code. Nous avons constaté que '<i>visible</i>' est défini dans la table '<i>course_modules</i>' qui englobe tous les éléments des cours. Il indique si un élément du cours sera affiché ou caché au sein du cours. Cependant, nous avons choisi ainsi de préserver '<i>visible</i>' comme attribut de chaque élément.</p>

C. Métamodèle de conception pédagogique de Moodle

Le métamodèle de conception pédagogique de Moodle est le résultat de l'application du processus proposé sur cette plateforme. Ce métamodèle est formalisé au fur et à mesure de la confrontation et de la vérification des éléments de conception pédagogique. Ces éléments sont représentés via des méta-classes. Ensuite, les attributs ainsi que les relations entre les méta-classes sont spécifiées. La figure A.60 illustre le métamodèle de la plateforme Moodle.

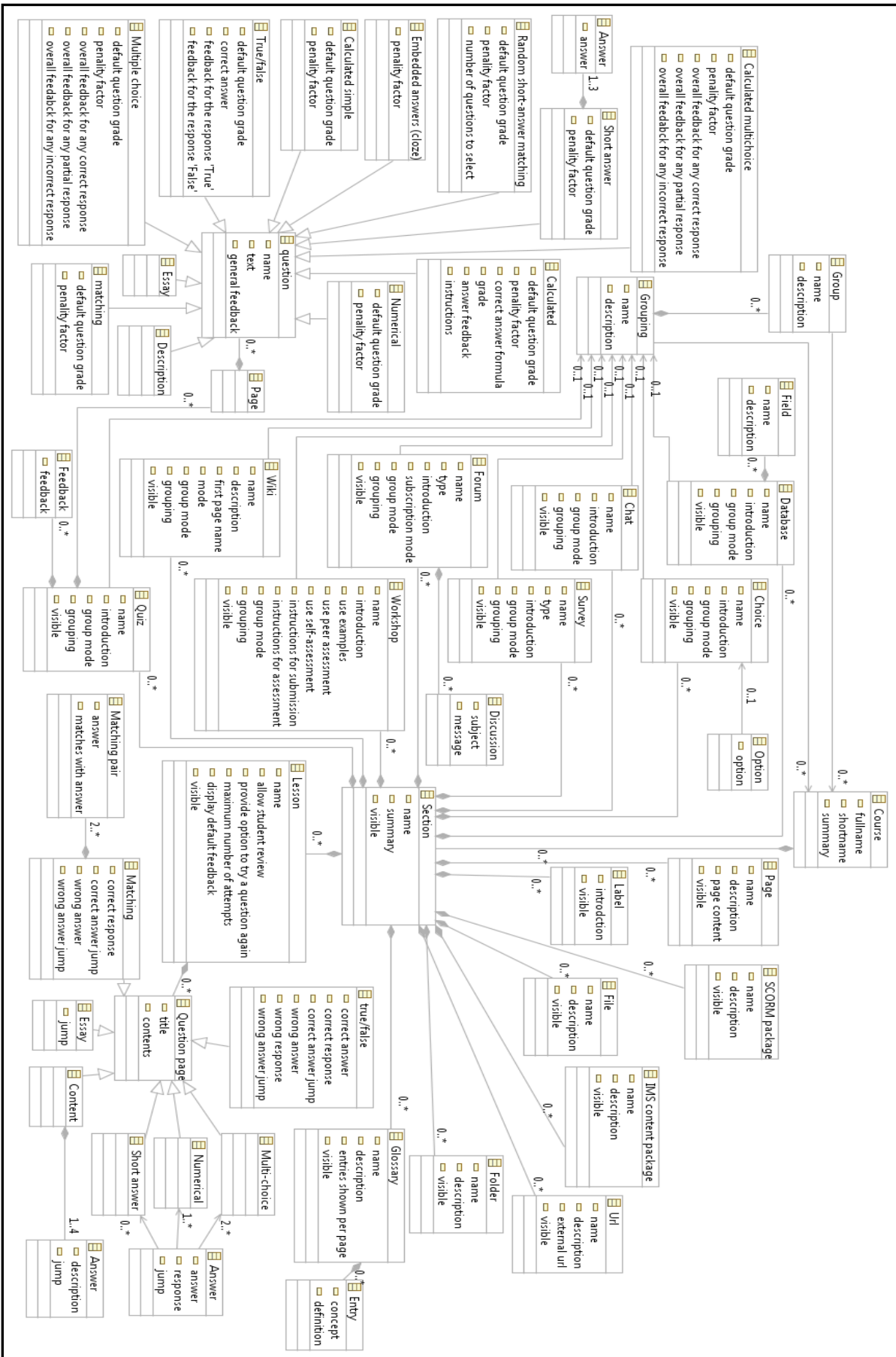


Figure A.60 : Métamodèle de la plateforme Moodle réduit à la conception pédagogique

5. Conclusion

La plateforme Moodle était un premier environnement pour expérimenter le processus proposé. Dans cette annexe, nous avons présenté l'application de ce processus sur les interfaces et les tables de la base de données, les plus significatives de cette plateforme. L'analyse centrée IHM a permis de spécifier le modèle de conception pédagogique de point de vue IHM. L'analyse technique a permis la spécification d'un deuxième modèle de conception pédagogique mais d'un point de vue différent : le point de vue technique. Ces deux analyses sont suivies par une étape de confrontation afin de formaliser un modèle compromis avec les deux autres modèles. Ce modèle représente le modèle final représentant le langage de conception pédagogique de la plateforme Moodle.

Nous avons remarqué que l'analyse technique a nécessité des compétences et des connaissances informatiques issues du domaine génie logiciel. L'analyse de la base de données, le *reverse engineering* du modèle de la base ainsi que la génération du modèle conceptuel de données requièrent des compétences informatiques spécifiques.

Table des figures

Figure A.1 : Exemple de cours sur Moodle.....	205
Figure A.2 : Aperçu des interfaces de Moodle.....	206
Figure A.3 : Extrait des interfaces dédiées au forum et ses éléments.....	207
Figure A.4 : Extrait du modèle IHM macro représentant un forum et une discussion.....	209
Figure A.5 : Modèle IHM-macro de Moodle.....	210
Figure A.6 : Répartition de l'interface en plusieurs zones.....	211
Figure A.7 : Extrait du modèle fonctionnel dédié à la zone 2.....	213
Figure A.8 : Extrait du modèle fonctionnel dédié à la zone 4.....	213
Figure A.9 : Extrait de la zone 2.....	214
Figure A.10 : Extrait du modèle fonctionnel de l'interface Moodle.....	215
Figure A.11 : Interface d'inscription des intervenants au cours.....	216
Figure A.12 : Interface de gestion des groupes.....	217
Figure A.13 : Modèle fonctionnel de gestion des groupes.....	218
Figure A.14 : Interface de gestion des groupements.....	218
Figure A.15 : Modèle fonctionnel de gestion des groupements.....	219
Figure A.16 : Interface d'association entre les groupements et les groupes.....	219
Figure A.17 : Modèle fonctionnel des associations entre les groupements et les groupes.....	220
Figure A.18 : Extrait de l'interface d'ajout d'un forum.....	220
Figure A.19 : Interface d'un forum existant.....	221
Figure A.20 : Interface d'ajout d'un sujet de discussion.....	223
Figure A.21 : Extrait de l'interface des nouveaux sujets de discussions.....	224
Figure A.22 : Extrait de l'interface d'un sujet de discussion.....	224
Figure A.23 : Extrait d'interface des discussions.....	224
Figure A.24 : Modèle fonctionnel de la fonction 'ajouter une discussion'.....	225
Figure A.25 : Extrait du modèle fonctionnel final.....	226
Figure A.26 : Interface d'une section.....	227
Figure A.27 : Modèle IHM-micro de section.....	228
Figure A.28 : Modèle IHM-micro d'un forum.....	229
Figure A.29 : Modèle IHM-micro d'une discussion.....	230
Figure A.30 : Interface de spécification des groupes.....	231
Figure A.31 : Modèle IHM-micro du groupe.....	231
Figure A.32 : Modèle IHM-micro d'un test.....	232
Figure A.33 : Modèle IHM-micro d'un feedback d'un test.....	233
Figure A.34 : Interface de structuration de pages de question.....	233
Figure A.35 : La composition : cours et section.....	235
Figure A.36 : La composition : forum et section.....	236
Figure A.37 : La composition : forum et discussion.....	236
Figure A.38 : La composition : feedback et test.....	237
Figure A.39 : Vue générale des compositions entre section, test, feedback, page et question.....	238
Figure A.40 : Vue globale du modèle IHM.....	239
Figure A.41 : Ecran de la table forum.....	241
Figure A.42 : Modèle réduit de la table 'forum'.....	241
Figure A.43 : Aperçu de la table forum_discussions.....	242
Figure A.44 : Code source de la fonction ajout des discussions.....	243

Figure A.45 : Modèle réduit de la table 'forum_discussions'	243
Figure A.46 : Aperçu de la table forum_posts	243
Figure A.47 : Modèle réduit de la table 'forum_posts'	245
Figure A.48 : Extrait de la table groups	245
Figure A.49 : Modèles techniques des tables groupings, groupings_groups et groups	246
Figure A.50 : Extrait des tables de la base de données de Moodle	246
Figure A.51 : Ecran de la table 'course_sections'	247
Figure A.52 : Modèle réduit de la table course_sections	248
Figure A.53 : Ecran de la table 'course_modules'	248
Figure A.54 : Modèle réduit de la table 'course_modules'	248
Figure A.55 : Schéma récapitulatif des clés étrangères.....	249
Figure A.56 : Exemples de relations entre les tables modules, course_modules et forum	250
Figure A.57 : Modèle conceptuel de donnée réduit de la plateforme Moodle	251
Figure A.58 : Extrait du code associé aux attributs visible et groupmode	253
Figure A.59 : Méta-classe du forum.....	254
Figure A.60 : Métamodèle de la plateforme Moodle réduit à la conception pédagogique	256

Annexe B

Application du processus sur la plateforme Ganesha

Sommaire

1. Introduction
2. Application de l'analyse IHM
3. Application de l'analyse technique
4. Confrontation et formalisation du langage de Conception
pédagogique
5. Conclusion

1. Introduction

Dans cette annexe, nous détaillons l'application du processus (analyse centrée IHM, analyse technique, et la confrontation et formalisation du langage de conception de Ganesha) sur la plateforme Ganesha, en particulier sur un ensemble cohérent d'exemples. Les cas similaires et redondants ne seront pas présentés. Nous présentons également les spécificités, les particularités et les modèles spécifiques à chaque partie du processus. Enfin, nous présentons le modèle de conception pédagogique de la plateforme Ganesha, résultat de l'application du processus.

Nous allons décrire chaque partie du processus par le biais d'une structure spécifique décrite par le contexte, l'objectif et l'application concrète du processus. Nous utilisons la légende suivante :



Ce pictogramme identifie des problèmes rencontrés.



Ce pictogramme identifie les solutions que nous proposons.



Ce pictogramme identifie des rappels ou des remarques ou des constats.

2. Analyse centrée IHM

Nous présentons dans cette section l'application des différentes parties de l'analyse centrée IHM appliquées sur la plateforme Ganesha. Pour cela, nous allons choisir le profil d'un itégrateur qui correspond à l'acteur de création de cours (modules). Cette analyse aboutit à la spécification du modèle IHM dédié à la conception pédagogique de cette plateforme.

2.1. Analyse IHM-macro

Contexte : nous présentons dans cette partie l'application de l'analyse IHM-macro sur la plateforme Ganesha qui représente la première étape de l'analyse centrée IHM.

Objectif : identifier les interfaces relevant de la conception pédagogique d'un cours/module ou d'une situation d'apprentissage.

Application de l'analyse IHM-macro : nous avons choisi d'illustrer cette application par un ensemble cohérent d'exemples génériques et particuliers.

A. Cas général

Nous avons tout d'abord commencé par nous approprier la plateforme Moodle de point de vue enseignant-concepteur. Ensuite, nous avons appliqué les deux stratégies d'analyse proposées.

Application de la première stratégie : cette stratégie de l'analyse IHM-macro est basée sur l'analyse d'une banque de cours disposée sur la plateforme. Nous avons ainsi choisi un module existant sur la plateforme Ganesha pour identifier les interfaces liées à la conception pédagogique.



Figure B.1 : Exemple de module sur Ganesha

Exemple : la figure B.1 montre un exemple de module (appelé 'les pattes') associé au domaine 'La gastronomie' et à la matière 'la cuisine italienne'. La partie encadrée en rouge de la Figure B.2 représente ce module en termes de contenu.

Pour le module choisi, nous avons testé tous les liens possibles qui peuvent amener à des interfaces de conception. Nous avons identifié plusieurs interfaces (telles que celles des modules, des séquences, des activités, etc.). Nous avons déterminé leur concept principal par l'analyse sémantique et syntaxique des titres des interfaces. Nous avons constaté que :

- le module est composé d'un ensemble de séquences.
- chaque séquence amène à un écran pour la consultation ou la spécification du contenu en termes d'activités.
- la spécification des activités amène également à un nouvel écran.
- plusieurs activités sont proposées par Ganesha (de type *titre*, *AICC*, etc.) dont chacune dispose de sa propre interface/écran. La figure B.2 illustre des extraits des interfaces identifiées.

Nous présentons chaque interface identifiée par son concept principal pour formaliser le modèle IHM-macro (nous détaillons cette partie un peu plus loin).

The screenshot displays three distinct administrative interfaces within the Ganesha platform:

- Module Management:** The top section shows the 'Gestion du module' interface. It features a sidebar with navigation options like 'Accueil', 'Domaines / Matières', 'Séquences', and 'Modules'. The main area lists three modules: '1 - Composition des pattes', '2 - Méthodes de préparation', and '3 - Présentation des pattes'. Each module row includes icons for adding, deleting, and searching, as well as a 'Gestion du module' button.
- Activity Management:** The middle section shows the 'Gestion des activités' interface. It lists five activities with columns for 'ACTIVITE', 'TYPE', and 'ACTIONS'. Activities include '1 - Inland Rules of the Road (HTML Format)', '2 - References and Lesson Objective', '3 - Steering & Sailing Rules', '4 - Conduct of Vessels in any Condition of Visibility', and '5 - Conduct of Vessels in Sight of One Another'. The 'TYPE' column shows 'Titre' or 'SCORM 1.2'.
- Question Management:** The bottom section shows the 'Gestions des questions' interface. It lists three series with columns for 'TITRE DE LA SERIE', 'Nombre de questions', 'Série Aleatoire', and 'Mode Reprise'. The series are 'Formation du cuisinier', 'Histoire américaine', and 'Histoire du 19ème siècle', all with 0 questions and 'NON' for the other two columns.

Figure B.2 : Extrait des interfaces relevant de la conception pédagogique

Application de la deuxième stratégie : nous allons identifier des interfaces relevant de la conception pédagogique par le biais de spécification d'un nouveau module. Pour cela, nous avons :

- paramétré les attributs proposés par l'interface de module tels que le nom, les objectifs, etc.
- spécifié indépendamment des modules, des nouvelles séquences pouvant lui être associées.
- défini pour chaque séquence ses activités. Ganesha dispose, à travers plusieurs interfaces, de plusieurs activités telles que les activités titre, les activités au format AICC, les activités collaboratives, les quiz, etc.



La spécification des quiz requiert qu'une série de questions soit déjà définie. Cette série peut être spécifiée sur des interfaces spécifiques.

La figure B.3 illustre des extraits d'interfaces identifiées. Ces interfaces, spécifiques à la création du contenu du cours, sont complémentaires aux interfaces identifiées par la première stratégie.

Figure B.3 : Extrait des interfaces de spécification d'un cours

B. Cas particulier



Au cours de cette analyse, nous avons constaté que certaines interfaces disposent de parties (ou des écrans) non relevant de la conception pédagogique telles que les parties des statistiques, etc. (voir figure B.2 par exemple).



Au cours de l'analyse IHM-macro, nous prenons en compte l'interface dans sa globalité. Ensuite, les analyses suivantes (fonctionnelle et IHM-micro) s'adressent uniquement aux zones relevant de la conception pédagogique.

C. Le Modèle IHM-macro

Le modèle IHM-macro résultant de l'analyse IHM-macro sur cette plateforme est construit au fur et à mesure de l'identification d'une interface liée à la conception pédagogique. Sa spécification a consisté à :

- identifier le concept principal qui décrit l'interface identifiée (tels que séquences, quiz, etc.).
- représenter chaque interface par un nœud qui désigne son concept principal.
- définir les relations entre les nœuds en se basant sur les relations entre les interfaces. Par exemple, les interfaces des activités sont accessibles à partir des séquences. Nous l'avons modélisé par le nœud activité puis nous l'avons associé au nœud séquence.



Contrairement à la plateforme Moodle, la conception pédagogique d'un module peut commencer par la spécification du contenu en termes de séquences d'activités ou d'une série de questions.



Pour, cela nous avons représenté les séquences et les séries de questions par deux nœuds situés au même niveau que les modules.

Nous avons reproduit la même démarche pour les interfaces liées à la conception pédagogique d'un module (activité, série, question, etc.). La figure B.4 illustre le modèle IHM-macro de Ganesha.

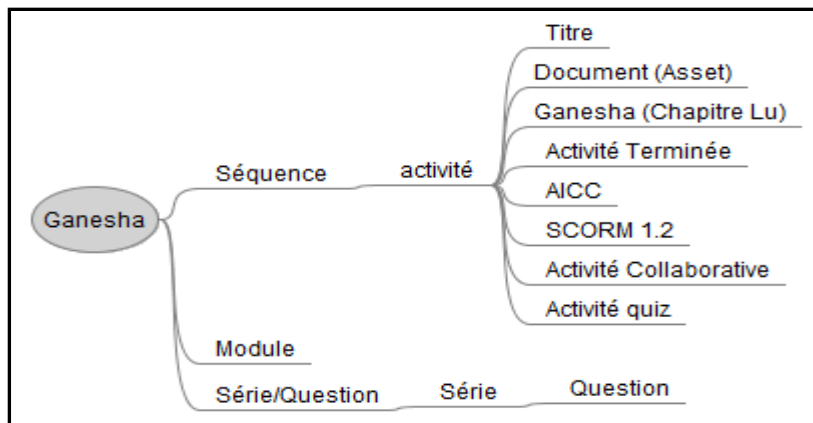


Figure B.4 : Modèle IHM-macro de la plateforme Ganesha

2.2. Analyse fonctionnelle

Contexte : nous illustrons l'application de l'analyse fonctionnelle sur Ganesha par le biais d'un cas générique (analyses des interfaces de séquences) et de quelques particularités identifiées. Nous ne présentons pas les cas similaires ou redondants.

Objectif : identifier les fonctionnalités relevant de la conception pédagogique embarquées dans les IHMs du modèle IHM-macro (identifiées pendant l'analyse IHM-macro).



Comme pour les interfaces, les fonctionnalités pédagogiques sont également moins nombreuses au sein de la plateforme Ganesha en comparaison avec la plateforme Moodle.

Application de l'analyse fonctionnelle :

A. Cas général : analyse des interfaces de séquences

L'analyse fonctionnelle de l'interface des séquences (figure B.5) a consisté à :

- décomposer l'intervalle en des zones principales. Exemple : la figure B.5 illustre les trois zones identifiées de cette interface et limitées par des cadres.
- identifier des fonctionnalités pédagogiques de chaque zone.
- Représenter les fonctionnalités pédagogiques identifiées au sein du modèle fonctionnel des séquences.
- associer le modèle fonctionnel complet dédié aux séquences au modèle fonctionnel principal.

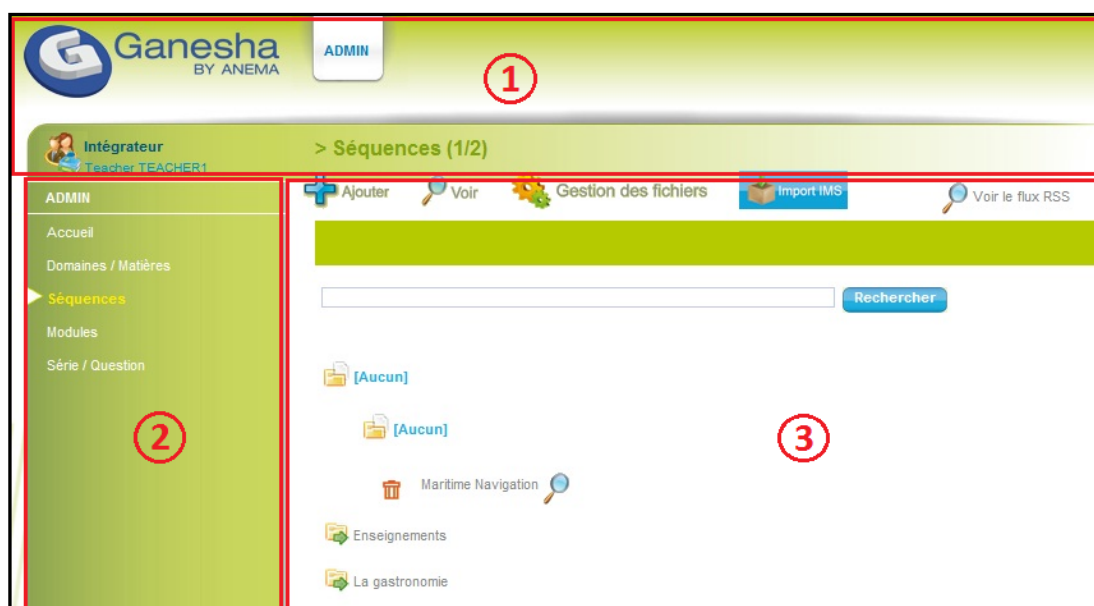


Figure B.5 : Interface de présentation des séquences sur Ganesha

Nous allons maintenant détailler davantage l'analyse fonctionnelle des zones de cette interface.

Analyse de la première zone :

Cette zone décrit le profil de l'enseignant concepteur et le nom de l'interface. Elle ne dispose pas de widgets représentant des fonctionnalités pédagogiques.

Analyse de la deuxième zone :

Cette zone définie sous la forme d'un menu représente des liens à des nouvelles interfaces. L'analyse a consisté à :

- analyser les widgets (liens hypertexte) permettant de montrer des écrans relevant de la conception pédagogique.
- identifier les fonctionnalités pédagogiques associées à ces widgets. Par exemple, nous avons identifié les fonctionnalités: '*consulter les séquences*', '*consulter les modules*' et '*consulter les séries / question*'.
- écarter les widgets désignant des fonctionnalités non liées à la conception pédagogique d'un module. Par exemple, nous n'avons pas pris en compte de '*Domaines / Matières*' parce qu'il n'intervient pas à la spécification du contenu d'un module. Il représente le cadre du module (cours).
- représenter les fonctionnalités identifiées sur le modèle fonctionnel.



Les zones 1 et 2 sont communes pour toutes les interfaces de la plateforme (dédiées au profil intégrateur). Elles sont analysées une seule fois. Dans la suite, l'analyse fonctionnelle sera centrée sur d'autres zones des interfaces relevant de la conception pédagogique telles que la zone 3 de la figure B.5.

Analyse de la troisième zone :

Cette analyse a consisté à :

- analyser les widgets de cette zone de point de vue fonctionnalités.
- tenir en compte uniquement les widgets indiquant des fonctionnalités pédagogiques telles que '*ajouter une séquence*', '*voir une séquence*', '*chercher une séquence*', etc.
- écarter les widgets décrivant des fonctionnalités techniques telles que '*voir le flux RSS*'.
- représenter les fonctionnalités identifiées sur un modèle fonctionnel. La figure B.6 représente le modèle fonctionnel partiel spécifique aux séquences.

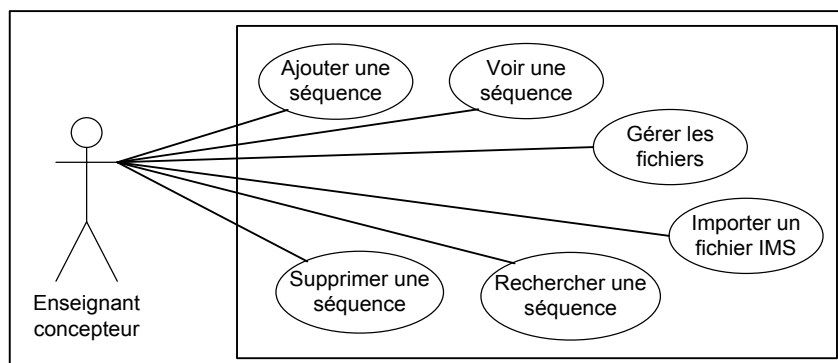


Figure B.6 : Modèle fonctionnel des séquences



L'analyse fonctionnelle concerne toutes les interfaces dédiées aux séquences. Outre l'interface de la figure B.5, l'interface de la figure B.7 doit être également analysée.

ACTIVITE	TYPE	ACTIONS
+ Ajouter une activité 🔍 Voir la séquence		
1 - Inland Rules of the Road (HTML Format)	Titre	
2 - References and Lesson Objective	SCORM 1.2	
3 - Steering & Sailing Rules	Titre	
4 - Conduct of Vessels in any Condition of Visibility	SCORM 1.2	
5 - Conduct of Vessels in Sight of One Another	SCORM 1.2	
6 - Conduct of Vessels in Restricted Visibility	SCORM 1.2	
7 - Lights & Shapes	SCORM 1.2	
8 - Sound & Light Signals	SCORM 1.2	
9 - Exam [75]	SCORM 1.2	

Figure B.7 : Interfaces de spécification du contenu des séquences

L'analyse fonctionnelle de cette interface est similaire à la précédente analyse. Les zones 1 et 2 de cette nouvelle interface ont été analysées. Ainsi, cette analyse est restreinte à l'analyse de la troisième zone. Pour cela, nous allons :

- analyser les widgets (les composants de l'interface graphique) pour identifier de nouvelles fonctionnalités liées à la conception pédagogique par rapport à celles identifiées lors de l'analyse de l'interface de la figure B.5. Nous avons identifié certaines fonctionnalités telles que 'ajouter une activité', 'supprimer une activité', 'déplacer une activité', etc.

- écartier les fonctionnalités techniques telles que : ‘passer en plein écran’, ‘passer tout en affichage normal’, ‘passer tout en affichage plein écran avec pop-up’, etc.
- représenter les fonctionnalités pédagogiques identifiées sans redondance telles que le cas de la fonctionnalité ‘voir une séquence’ sur le modèle fonctionnel des séquences. La figure B.8 représente le modèle fonctionnel complet, spécifique aux séquences

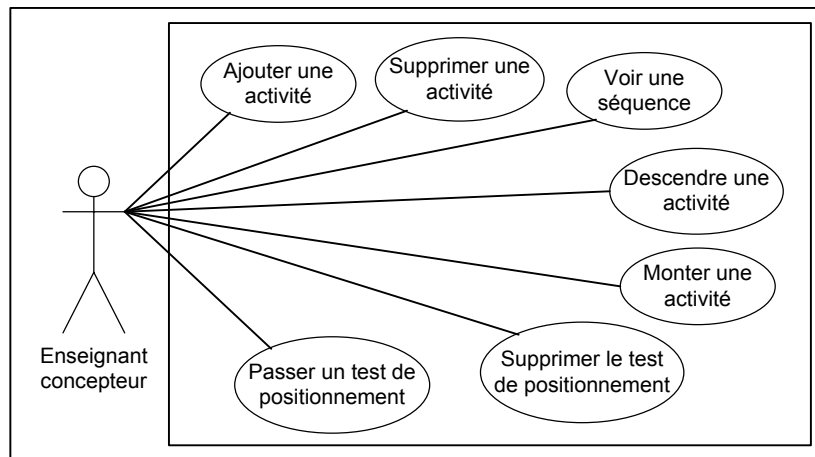


Figure B.8 : Modèle fonctionnel de l'interface 'Séquences'

B. Particularités de l'analyse fonctionnelle

Contexte : l'application de l'analyse fonctionnelle sur certaines interfaces ne requiert pas sa décomposition en plusieurs zones.

Exemple : l'interface de choix des activités (figure B.9). L'ajout d'une activité particulière sur Ganesha nécessite le passage par une interface intermédiaire consistant à choisir le type d'activité.

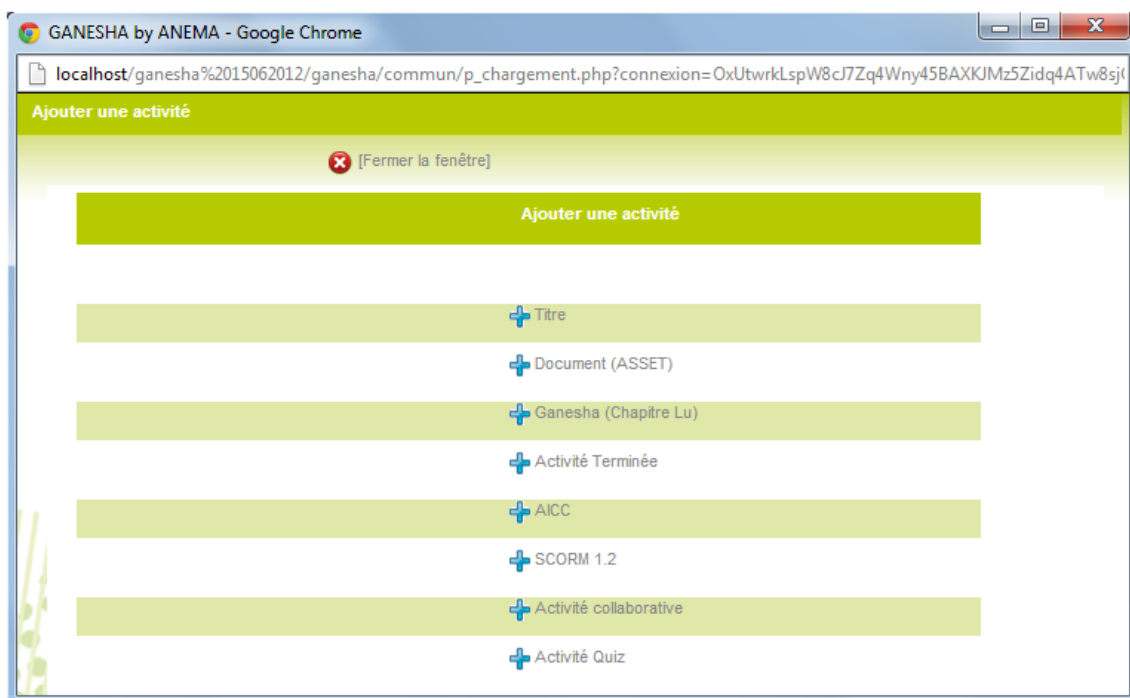



Figure B.9 : Interface d'ajout d'une activité

Analyse fonctionnelle : nous avons analysé les widgets de cette interface. Nous avons identifié huit fonctionnalités pédagogiques qui représentent des spécialisations de la fonction 'ajouter une activité'. Ensuite, nous allons représenter le modèle fonctionnel associé à l'ajout des activités. La figure B.10 illustre ce modèle.

 Cette interface ne dispose que de fonctionnalités liées à la conception pédagogique.

Le modèle fonctionnel ci-dessous représente les sous-fonctionnalités de la fonctionnalité 'ajouter une activité' que nous avons identifiée pendant l'analyse fonctionnelle de l'interface des séquences.

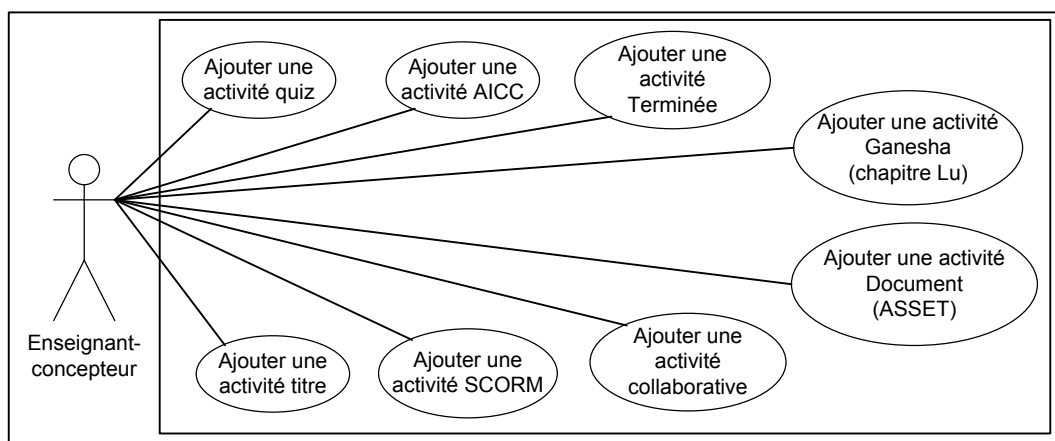


Figure B.10 : Modèle fonctionnel de l'ajout d'activité

C. Le modèle fonctionnel de Ganesha

La figure B.11 illustre un extrait du modèle fonctionnel final. Ce modèle est le résultat de l'application de l'analyse fonctionnelle sur les interfaces de la plateforme Ganesha, en particulier celles qui sont identifiées au sein de l'analyse IHM-macro.

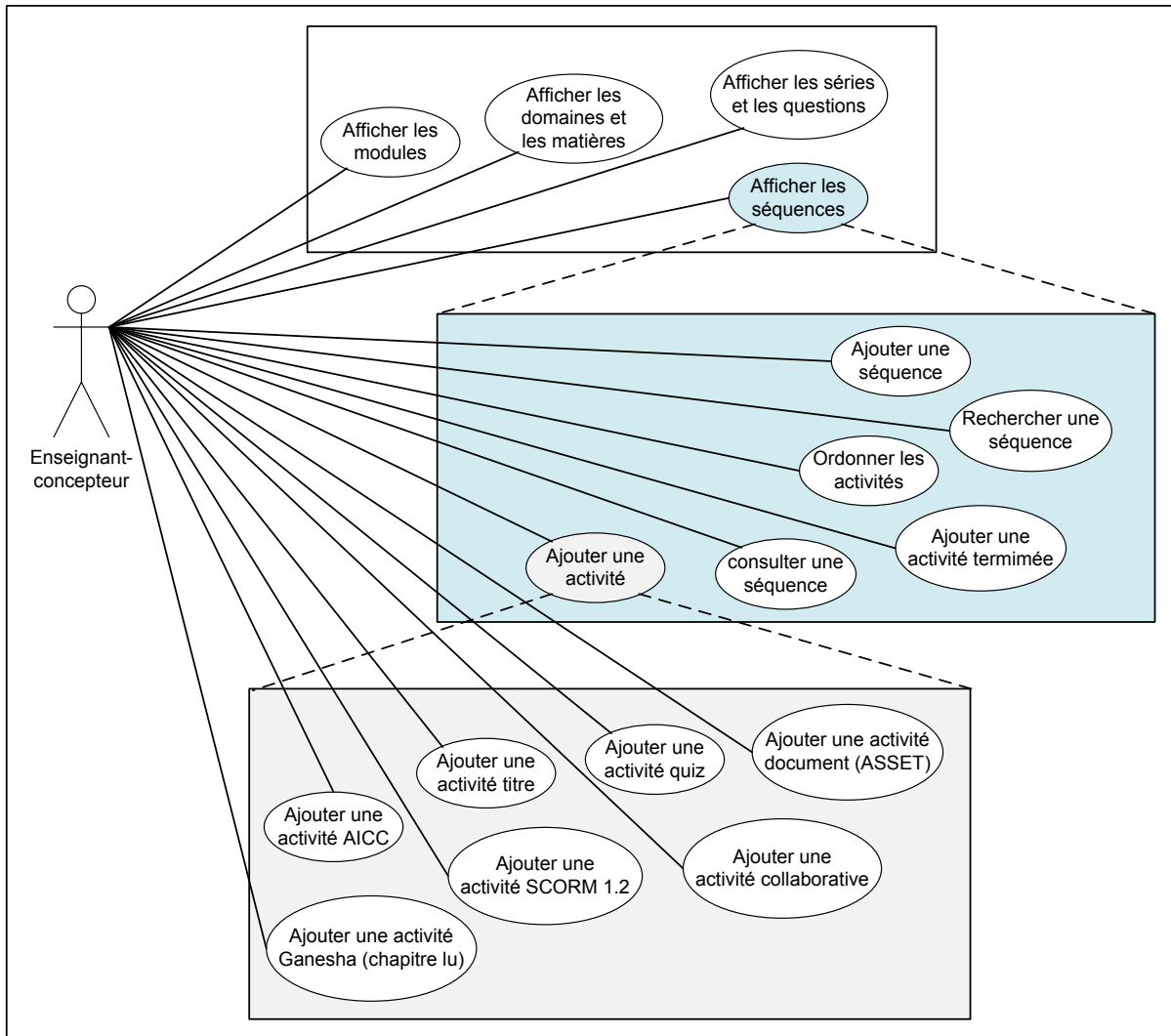


Figure B.11 : Extrait du modèle fonctionnel de Ganesha

2.3. Analyse IHM-micro

Contexte : nous présentons dans cette partie l'application de l'analyse IHM-micro sur un ensemble cohérent d'exemples génériques (l'activité quiz fréquemment utilisée au sein des cours de formation à distance) et particuliers (le cas de l'élément question). Les cas similaires ne sont pas présentés.

Objectif : identifier les attributs et les propriétés des éléments de conception pédagogique.

Application de l'analyse IHM-micro :

A. Cas général : analyse d'une activité (quiz)

Contexte : la spécification d'un élément pédagogique sur Ganesha (séquence, activité, série ou question) requiert le paramétrage du formulaire associé (figure B.12) et composé d'une longue liste d'attributs techniques et pédagogiques.

Figure B.12 : Interface d'ajout d'un Quiz

Application de l'analyse IHM-micro : cette application a consisté à :

- analyser les champs du formulaire qui représentent les attributs du quiz
- identifier les attributs pédagogiques du quiz. Les attributs pédagogiques que nous avons pris en compte sont : 'Activité', 'Serie de question', 'Visible', 'Max Time', 'Description' et 'Mots clefs'.
- identifier les propriétés de chaque attribut en terme d'élément requis ou facultatif, le type, les champs de valeur et la valeur par défaut.
- écarter les attributs que nous jugeons techniques tels que 'Activité de positionnement', 'Plein écran', 'Mastery', etc.
- représenter les attributs identifiés ainsi que leurs propriétés sur le modèle IHM-micro du quiz. La figure B.13 illustre ce modèle.

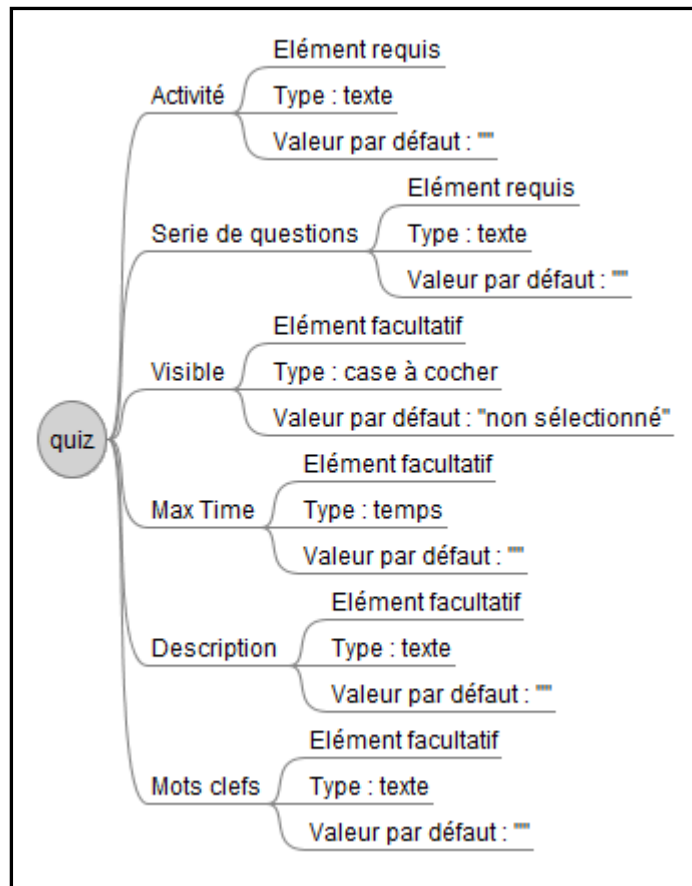


Figure B.13 : Modèle IHM-micro d'un quiz

B. Particularités de l'analyse IHM-micro

Nous présentons deux cas particuliers identifiés pendant l'analyse IHM-micro de l'interface dédiée à l'élément question (figure B.14).

Particularité 1 : l'interface des questions ne dispose pas d'attributs techniques. Nous allons ainsi :

- Prendre en compte tous ses attributs : *'Question'*, *'Texte correct'*, *'Texte incorrect'*, *'Bloquante'* et *'pondération'*.
- Identifier les propriétés de ces attributs en terme d'attributs requis ou facultatif, le type, la valeur par défaut et les champs de valeur.
- Représenter les attributs et les propriétés identifiés sur le modèle IHM-micro dédié à une question. La figure B.15 illustre ce modèle.

Particularité 2 : lors de la spécification d'une question, de nouveaux attributs apparaissent sur l'interface (la partie encadrée en rouge sur la figure B.14). Ces attributs sont spécifiques à un nouvel élément dédié : l'élément *réponse*.



Notre proposition du modèle IHM-micro ne permet pas la représentation de ce cas particulier parce que l'élément réponse dispose des attributs qui doivent être identifiés.



Nous avons analysé et représenté cet élément séparément des questions. La figure B.16 illustre le modèle IHM-micro de la réponse, composé de deux attributs : Réponse et Bonne réponse. Pendant l'étape de composition, nous définissons les associations entre les modèles de questions et réponses.

QUESTION

[Fermer la fenêtre]

MODIFIER UNE QUESTION

Question :
255 caractères max,
7 lignes max *

Texte correct :
235 caractères max,
4 lignes max *

Texte incorrect :
235 caractères max,
4 lignes max *

Bloquante

Pondération * 1

Enregistrer

REPONSE 1
60 caractères max,
2 lignes max

Bonne Reponse

REPONSE 2
60 caractères max,
2 lignes max

Bonne Reponse

REPONSE 3
60 caractères max,
2 lignes max

Bonne Reponse

REPONSE 4

Figure B.14 : Interface d'ajout d'une question

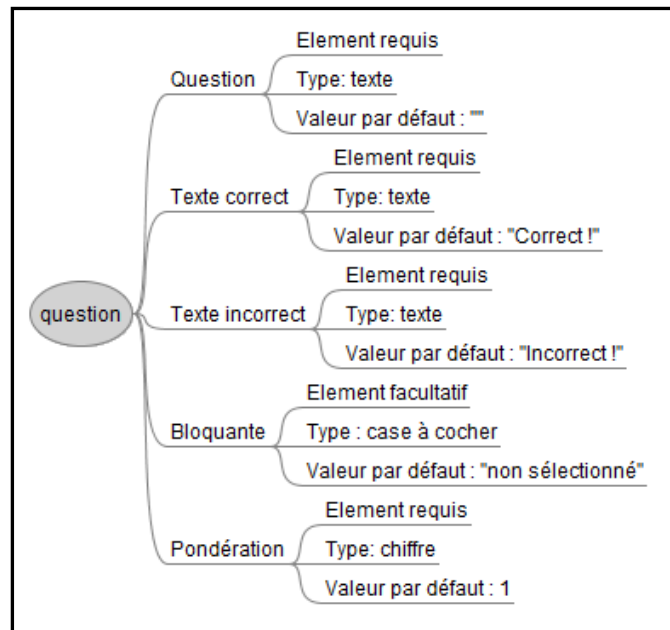


Figure B.15 : Modèle IHM-micro d'une question

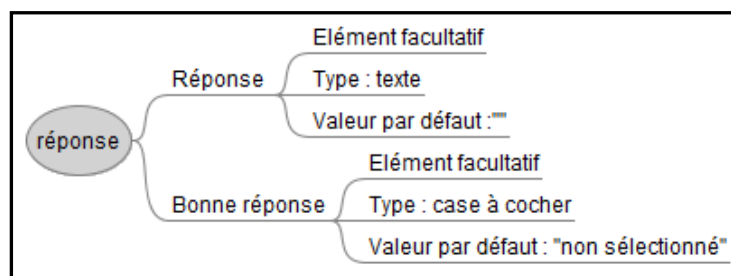


Figure B.16 : Modèle IHM-micro d'une réponse

2.4. Composition et formalisation du modèle IHM

Contexte : nous illustrons dans cette partie l'application de composition des modèles et la formalisation du modèle IHM du langage de conception pédagogique de la plateforme Ganesha par un ensemble de cas génériques et particuliers.

Objectif : définir les associations entre les modèles IHM-micro précédemment formalisés. Ces associations sont identifiées en se basant sur les modèles IHM-macro et fonctionnel.

Application de la composition :

A. Cas général : Composition des séquences et quiz

Définition de l'association (séquence – quiz)

Rappel :

- L'analyse IHM-micro a permis la formalisation de deux modèles IHM-micro dédiés aux séquences et aux quiz.

- Au cours de l'analyse IHM-macro, nous avons identifié des interfaces dédiées à ces éléments. La relation entre ces deux éléments est assurée par l'intermédiaire de l'élément 'activité'.
- Au sein de l'analyse IHM fonctionnelle, la fonctionnalité 'ajouter un quiz' est une spécification de la fonction '*ajouter une activité*' qui est une sous-fonctionnalité parmi les fonctionnalités de l'élément *séquence*.



L'élément activité en générale ne possède pas de modèle IHM-micro. Nous avons formalisé des modèles IHM-micro pour les activités spécifiques.



Les modèles IHM-micro des activités spécifiques tels celui du quiz, sont ainsi associés au modèle des séquences.

Définition des multiplicités de l'association (séquence – quiz)

- L'interface de spécification du contenu des séquences permettent d'ajouter plusieurs activités (telles que quiz) par le biais de la fonction '*ajouter une activité*'. Ainsi, la branche de multiplicité [séquence - quiz] est d'ordre (0,*).
- En prenant un quiz particulier, ce dernier ne peut être associé qu'à une seule séquence. Ainsi, la branche de multiplicité [quiz - séquence] est d'ordre (1,1).

La figure B.17 illustre une représentation simplifiée de l'association (séquence – quiz) dont chaque élément désigne son modèle IHM-micro.

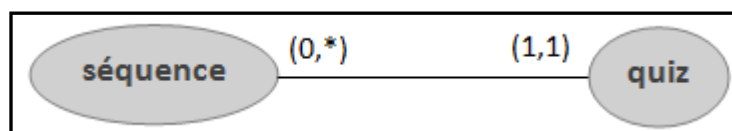


Figure B.17 : Composition entre séquence et quiz

B. Particularités de la composition

Au cours de l'application de la composition, nous avons identifié deux cas particuliers.

Cas n°1 : L'association (module – séquence)



Ce cas est équivalent à l'association (cours - section) sur Moodle mais il est différent en terme de multiplicité. Sur la plateforme Moodle, toutes les sections sont associées à un seul cours. Cependant, sur Ganesha, les séquences existantes peuvent être réutilisées au sein de plusieurs modules.



Au cours de la composition des modèles, nous avons défini l'association (module - séquence) en adoptant la même démarche suivie pour l'association (séquence - quiz).

Identification des multiplicités :

- Un module sur Ganesha est composé d'une ou plusieurs séquences (i.e. séquence est un sous-élément du module).
- Les séquences sont tout d'abord définies séparément des modules. Ensuite, elles sont associées aux modules concernés.

⇒ Ainsi les deux couples de multiplicité de cette association sont d'ordre (0,*).

La figure B.18 illustre une représentation simplifiée de l'association (séquence – module) dont chaque élément désigne son modèle IHM-micro.

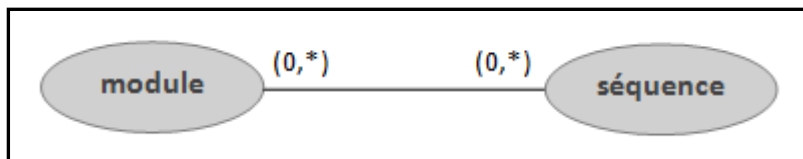


Figure B.18 : Composition entre module et séquence

Cas n°2 : L'association (question – réponse)**Définition de l'association :**

La spécification de nouvelles questions entraîne la spécification d'une ou plusieurs réponses à ces questions. Au sein de l'analyse IHM-micro, nous avons séparé la représentation des réponses de celle des questions. Les deux modèles des questions et des réponses doivent être associés ce que justifie l'association (question - réponse).

Identification des multiplicités :

- L'interface de spécification des questions permet de spécifier au maximum 6 réponses à une question. La branche de multiplicité [question – réponse] est d'ordre (0,6)
- La multiplicité [réponse – question] est d'ordre (1,1) parce que toutes les réponses sont associées par défaut à une seule question.

La figure B.19 illustre une représentation simplifiée de l'association (question – réponse).

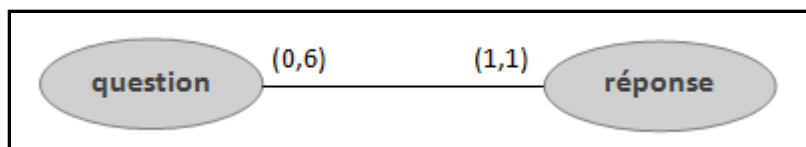


Figure B.19 : composition entre question et réponse

C. Le modèle IHM

Le modèle IHM du langage de conception pédagogique de la plateforme Ganesha est le résultat de l'étape de confrontation et de formalisation. Il est composé des modèles IHM-micro après avoir défini les associations nécessaires entre eux.

La figure B.20 illustre une vue simplifiée de ce modèle sous la forme d'une cartographie d'éléments de conception pédagogique de cette plateforme. Chaque élément du modèle désigne son modèle IHM-micro. Nous avons annoté les associations entre les éléments par les multiplicités identifiées.

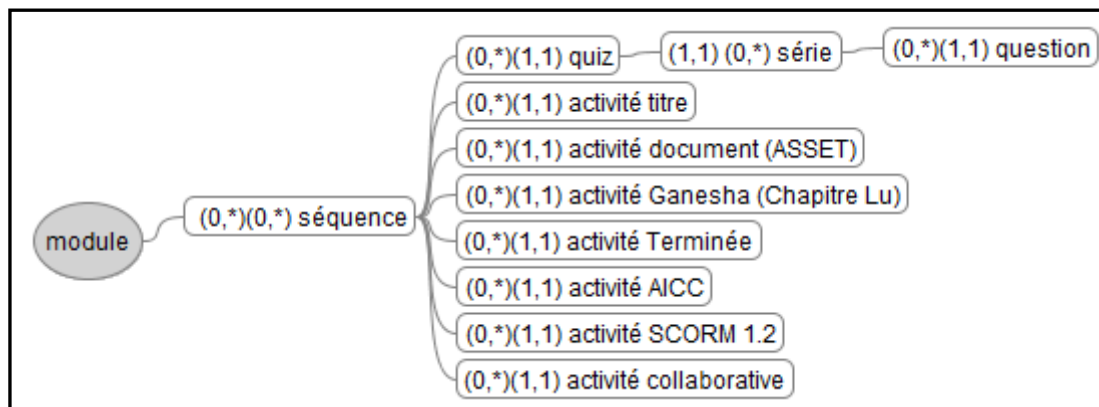


Figure B.20 : Modèle IHM de Ganesha



Nous avons constaté que le modèle IHM est différent du modèle IHM-macro. Nous retrouvons certains points communs entre les deux tels que l'association module – séquence. Par contre, les modèles IHM-micro des activités proposées par Ganesha (quiz, activité titre, etc.) sont directement associées au modèle IHM-micro de l'élément séquence contrairement au modèle IHM-macro où elles sont associées à l'élément *activité*.

3. Application de l'analyse technique : analyse de la base de données

Contexte : dans cette section, nous présentons un ensemble cohérent d'exemples d'analyse des tables liées à la conception pédagogique tout en explicitant les généralités et également les particularités de cette analyse technique.

Objectif : spécifier le modèle conceptuel de données réduit aux tables/colonnes en relation avec la conception pédagogique.

Application de l'analyse technique : cette application a consisté à :

- parcourir toutes les tables de la base de données de Ganesha.
- analyser sémantiquement les titres de tables.



La base de données que nous avons analysée comporte une banque de cours.

- écarter toutes les tables qui portent sur des spécifications techniques ou qui ne possèdent pas d'enregistrements. Ces tables sont créées pour des finalités différentes à la conception pédagogique (gestion des utilisateurs, les traces des apprenants, etc.) telles que les tables '*stat_connexions*', '*prime*', '*list_societe*', etc.

- identifier les tables liées à la conception pédagogique des situations d'apprentissage telles que les tables 'list_sequences', 'serie_question', 'module', etc.



Cependant, la liste des tables identifiées n'est pas définitive. De nouvelles tables peuvent être identifiées par le biais des clés étrangères. Des tables identifiées par l'analyse des titres seront écartées parce qu'elles ne sont pas en relation avec la conception de cours.

Dans la suite, l'analyse technique est consacrée aux tables que nous considérons comme relevant de la conception pédagogique pour formaliser le modèle technique.

A. Cas général : Analyse de la table list_sequence

sequence_id	sequence_titre	sequence_objectifs	sequence_duree	repertoire	auteur_id	code	domaine_id	matiere_id	sequence_acces	sequence_type
1	Maritime Navigation		0		-1	B0	0	0	0	0
2	Le vin français	La découverte du vin français à travers son histo...	0		-1	SEQ-AN1	3	5	0	0

Figure B.21 : Ecran de la table list_sequence

Rappel : la table *list_sequence* (figure B.21) considérée relevant de la conception pédagogique d'un cours (module) est identifiée par l'analyse sémantique de son titre.

Objectif : identifier les champs, non techniques, les plus significatifs pour la définition des séquences.

Application du processus d'analyse : l'application du processus d'analyse technique sur la table *list_sequence* a consisté à :

- analyser les enregistrements existants sur cette table. Ces enregistrements représentent les détails des séquences créées à partir de l'interface de la plateforme.
- écarter les champs qui sont soit définis par défaut (plusieurs enregistrements ont la même valeur (par exemple 'repertoire'), soient des détails techniques à bas niveau tels que 'sequence_access'.
- identifier les champs portant sur la définition des séquences ('sequence_titre' et 'sequence_objectifs').
- identifier les clés étrangères ('domaine_id' et 'matiere_id') qui référencent des tables liées à la conception pédagogique. Ces clés sont nécessaires pour définir les associations entre les tables concernées.
- représenter les éléments identifiés sur le modèle de la table 'list_sequence'. La figure B.22 illustre ce modèle.

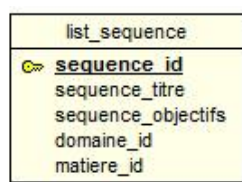




Figure B.22 : Modèle réduit de la table list_sequence




B. Particularités de l'analyse technique

Au cours de l'application de l'analyse technique sur la plateforme Ganesha, nous avons rencontrés plusieurs difficultés telles que :



Difficulté n°1 :

	Il était difficile d'identifier les tables associées par des clés étrangères. La table 'module', à titre d'exemple, contient la clé étrangère 'sequence_id'. Cependant, la base de données de Ganesha possède deux tables 'sequence' et 'list_sequence', dont chacune est représentée par la clé primaire 'sequence_id'. Ainsi, l'identification de la table référencée par cette clé étrangère n'était pas aisée.
	Pour cela, nous avons analysé le code source qui interroge la table module. Nous avons déduit que cette clé référence la table 'list_sequence'. Elle représente les séquences qui sont associées à un module particulier.

Difficulté n°2 :

	La table module consiste à définir les relations entre les modules et les séquences. Du point de vue génie logiciel, sa clé principale doit être composée de la concaténation des deux clés principales des tables correspondantes ('list_sequence' et 'list_module'). Cette règle n'était pas respectée.
	Au niveau modèle conceptuel de données, cette table sera remplacée par une association entre les entités 'list_module' et 'list_sequence'.
	Cela justifie notre choix du formalisme du modèle technique. Le modèle conceptuel de données est capable de cacher les détails techniques non liés à la conception pédagogique.

Difficulté n°3 :

	L'analyse syntaxique du titre de la table séquence indique qu'elle est susceptible de représenter les enregistrements des séquences.
	Nous avons analysé les champs de cette table ainsi que le code qui l'interroge. Nous avons déduit que cette table représente les activités définies au sein d'une séquence.

Difficulté n°4 :

id	serie_question_id	question	text_ok	text_ko	bloquante	r1	r2	r3	r4	r5	r6	reponse	ponderation	ordre
1	1	question 1	Correct ! ok	Incorrect ! ko	0	rep1	rep2					1	1	1
2	1	Quel est la différence entre Java et JavaScript ?	La différence entre les deux est ...	Pas de différence.	0								1	2
3	1	Qui a découvert l'amérique ?	Christophe Colombe	toutes autres réponses	0								1	3

Figure B.23 : Ecran de la table serie_question



En analysant la table `serie_question` (figure B.23), nous avons identifié certains attributs spécifiques intitulés `'r1'`, `'r2'`, `'r3'`, `'r4'`, `'r5'` et `'r6'`.



Nous sommes ainsi amenés à analyser le code source de l'interface de spécification de questions, en particulier les instructions qui interrogent la table `serie_question`. La figure B.24 illustre un extrait de code spécifique à ces attributs. Nous avons constaté que les `'r1'`, `'r2'`,... et `'r6'` représentent les éléments réponses à une question donnée.

```

$tab = array() ;
$tab['question'] = read_post_str('text1') ;
$tab['text_ok'] = read_post_str('text2') ;
$tab['text_ko'] = read_post_str('text3') ;
$tab['bloquante'] = read_post_chk('text4') ;
$tab['ponderation'] = read_post_int('text5') ;

// traitement des reponses...
$tab['reponse']='' ;
$tab['r1'] = read_post_str('text6') ;
$tab['reponse'] .= read_post_chk('text7')==1&&strlen($tab['r1'])>0?'1':'';
$tab['r2'] = read_post_str('text8') ;
$tab['reponse'] .= read_post_chk('text9')==1&&strlen($tab['r2'])>0?'2':'';
$tab['r3'] = read_post_str('text10') ;
$tab['reponse'] .= read_post_chk('text11')==1&&strlen($tab['r3'])>0?'3':'';
$tab['r4'] = read_post_str('text12') ;
$tab['reponse'] .= read_post_chk('text13')==1&&strlen($tab['r4'])>0?'4':'';
$tab['r5'] = read_post_str('text14') ;
$tab['reponse'] .= read_post_chk('text15')==1&&strlen($tab['r5'])>0?'5':'';
$tab['r6'] = read_post_str('text16') ;
$tab['reponse'] .= read_post_chk('text17')==1&&strlen($tab['r6'])>0?'6':'';

$clsAdmin->modifierUneQuestion(ID, IDQ, $tab);
location(PAGE.'&action=win_close&id='.$ID);
die();

```

Figure B.24 : Extrait du code dédié aux questions



D'autre part, plusieurs attributs ou clés étrangère ne sont pas conçus(es) précisément telles que le cas de la clé étrangère `'serie_question_id'` au sein de la table `'serie_question'`. Cette clé peut référencer soit la table `'serie_question'` soit la table `'list_serie_question'`.



Cela a nécessité une analyse du code qui interroge la table `'serie_question'` afin d'identifier la table référencée par cette clé. Nous avons déduit qu'elle référence la table `'list_serie_question'` et permet d'organiser les questions d'une série donnée.

A l'exception de l'élément ordre, les attributs de la table `'serie_question'` ne sont pas techniques. Nous avons ainsi pris en compte ces attributs pour la définition des questions (`'question'`, `'text_ok'`, `'text_ko'`, `'question'`, `'r1'`, `'r2'`, `'r3'`, `'r4'`, `'r5'`, `'r6'`, `'réponse'` et `'pondération'`) et de la clé étrangère (`'serie_question_id'`). La figure B.25 illustre le modèle retenu pour cette table.

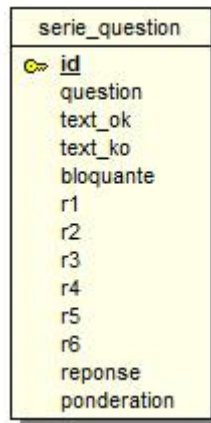


Figure B.25 : Modèle réduit de la table serie_question

C. Le modèle technique

Après avoir analysé et identifié les tables et les attributs liés à la conception pédagogique et surmonté toutes les difficultés rencontrées :

- nous avons spécifié le modèle logique de données. Nous nous sommes basées sur les clés étrangères pour spécifier les relations entre les tables.
- nous avons appliqué les règles de retro-conception dans le but de générer le modèle technique. Ce modèle représente le modèle conceptuel de données de la base de données de Ganesha réduit à la conception pédagogique de cette plateforme. La figure B.26 représente ce modèle technique.

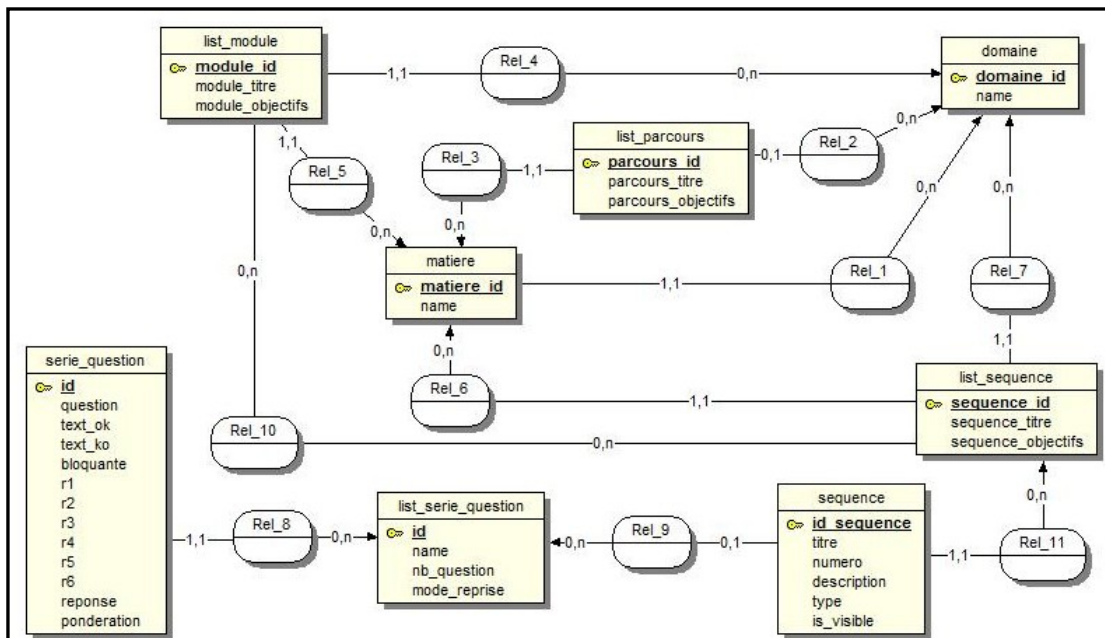


Figure B.26 : Modèle conceptuel de donnée réduit de la plateforme Ganesha

4. Confrontation et formalisation

Contexte : nous rappelons qu'à ce stade, nous avons identifié le langage de conception pédagogique de la plateforme Ganesha selon deux points de vue différents et complémentaires.





Objectif : formaliser le modèle final via la confrontation entre les deux modèles spécifiés (IHM et technique).

Application de la confrontation et formalisation : nous avons appliqué dans cette partie le processus de confrontation sur un ensemble cohérent d'exemples.

A. Cas général de la confrontation



L'élément '*module*' désigne l'élément cours sur cette plateforme. Dans ce cas générique, nous allons confronter sa représentation sur les deux modèles (IHM et technique).

1. Confrontation de la définition des éléments (cas du module)

	Sur le modèle IHM, nous retrouvons la représentation du module, plus spécifiquement son modèle IHM-micro intitulé ' <i>module</i> ' qui le décrit en termes d'attributs et de propriétés.
	Sur le modèle technique, nous retrouvons la table ' <i>list_module</i> ' décrivant les modules.
	L'élément module est bien défini sur les deux modèles. Cependant, sa nomination sur les deux modèles est similaire mais pas identique.
	Pour cela, nous avons choisi la nomination la plus significative (souvent celle du modèle technique). Toutefois, nous avons conservé la nomination des éléments du modèle IHM pour la définition de la couche graphique des outils de conception. Nous représentons l'élément module par une méta-classe intitulée ' <i>module</i> '.

2. Confrontation de la définition des attributs des éléments

Maintenant, nous nous intéressons à la vérification et à la représentation des attributs du module.

	Au sein des deux modèles (IHM et technique), nous retrouvons les deux attributs indiquant le titre et les objectifs du module.
	Nous reprenons la nomination des attributs du modèle technique, en particulier ' <i>module_titre</i> ' et ' <i>module_objectifs</i> '. La figure B.27 illustre la méta-classe de l'élément module.

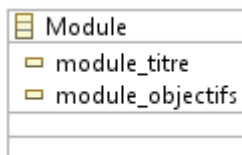






Figure B.27 : Méta-classe du module

La démarche appliquée à l'élément module est reproduite pour les autres éléments du modèle IHM (séquence, série, question et toutes les activités).

3. Définition des relations entre les méta-classes (cas du module et séquences)

Une fois toutes les méta-classes définies, nous nous intéressons à la spécification de leurs associations.



	Sur le modèle IHM, les modules sont associés aux séquences. Les multiplicités de cette association sont d'ordre (0,*) et (0,*).
	Sur le modèle technique, nous avons retrouvé l'association entre les deux tables list_module et list_sequence décrite par les multiplicités (0,n) et (0,n).
	Ainsi, nous définissons l'association entre les deux méta-classes 'list_module' et 'list_sequence'.
	Sur le métamodèle, nous avons représenté les relations définies au sein des deux modèles (IHM et technique) et dont leurs multiplicités sont : - (0,*) et (0,1) par des associations tel que le cas du module et séquence - (0,*) et (0,1) par des compositions tel que le cas de séquence et quiz.

Nous avons repris la même démarche pour la spécification des associations entre les méta-classes.

B. Particularités de la confrontation

Au sein de cette étape de confrontation, nous avons dû traiter plusieurs cas particuliers. Dans cette section, nous allons présenter quatre cas particuliers.

Difficulté n°1 : Cas des activités

	Au sein du modèle IHM, nous retrouvons toutes les activités proposées par la plateforme Ganesha (quiz, titre, activité collaborative, etc.). Cependant, lorsque nous vérifions ces activités sur le modèle technique, nous ne retrouvons pas d'entités correspondantes.
	Pour cela, nous sommes amenés à analyser le code source spécifique aux activités, en particulier les instructions de code qui interrogent la base de données suite à l'ajout de nouvelles activités. 1. Nous avons identifié ces instructions au sein des deux fichiers "\ganesha\class\sequence.class.php" et "\ganesha\admin\sequence.php". Les deux figures B.28 et B.29 illustrent des

extraits de code pour l'ajout de nouvelles activités.

2. Nous avons analysé ce code et nous avons constaté que toutes les activités de Ganesha sont enregistrées au sein de la table sequence.

3. Pour être plus représentatif, nous avons choisi de représenter ces activités par des méta-classes spécifiques au sein du métamodèle du langage de conception de Ganesha.

```
function add($titre,$code,$aut,$support,$objectifs,$duree,$sequence_acces=0,$sequence_type=0) {
    global $db, $clsMembre;

    $support = str_replace(array('../','/..','..\','\..\'),'', $support);
    if(GUSERTYPE != TYPE_ADMIN && MODE_ASP) {
        $aut = GUSERID;
    }

    // la sequence existe ?
    if ( defined('WS_GENDEP') && strlen(trim($code))>0 ) {
        $$SQL = "SELECT sequence_id FROM ".$this->table." WHERE code='".$code."'";
        $db->query($$SQL) ;
        if ( $db->next_record() ) {
            $ret = $db->f('sequence_id');
            $$SQL = "UPDATE ".$this->table." SET sequence_titre='".$titre."' ,auteur_id='".$aut." ";
            $$SQL = "WHERE sequence_id='".$ret;
            $db->query($$SQL) ;
            return $ret ;}
        }

        $$SQL = "INSERT INTO ".$this->table." (sequence_titre,sequence_objectifs,";
        $$SQL.= "sequence_duree,code,auteur_id,sequence_acces,sequence_type) ";
        $$SQL.= "VALUES ('".$titre."','".$objectifs."','".$duree."','".$code."','".$;
        $$SQL.= $aut."','".$sequence_acces."','".$sequence_type."'";
        $db->query($$SQL) ;

        $$SQL = "SELECT MAX(sequence_id) FROM ".$this->table." ";
        $db->query($$SQL) ;
        $db->next_record();
        ...
    }
```

Figure B.28 : Extrait des instructions d'ajout d'activités

```
function Sequence() {
    global $TBL ;
    parent::GBDD() ;
    $this->table = $TBL['list_sequence'];
    $this->table_seq = $TBL['sequence'];
}
```


Figure B.29 : Définition de la table séquence

Difficulté n°2 : Cas des réponses d'une question




Le deuxième cas particulier que nous avons rencontré concerne les réponses d'une question.






Au sein du modèle IHM, nous avons spécifié deux modèles IHM-micro spécifiques aux éléments question et réponse. Puis, nous avons défini l'association entre ces deux modèles ainsi que les multiplicités. Au niveau modèle technique, nous avons déjà représenté les réponses d'une question par les attributs r1, r2, r3, r4, r5 et r6 conformément à la table serie_question.

	En confrontant ces deux éléments, nous devons choisir leur représentation au niveau méta-modèle.
	Nous avons ainsi choisi de spécifier deux méta-classes, question et réponse. Ces deux méta-classes sont reliées par une association de composition, au sens méta-modélisation, dont la multiplicité maximale vaut 6.

Difficulté n°3 : Cas de l'association séquence et liste série question

	Au sein du modèle IHM, les séries de questions sont associées uniquement à l'activité quiz. En consultant le modèle technique, nous constatons l'existence d'une association entre les deux entités <code>sequence</code> et <code>list_serie_question</code> . Quant à l'entité séquence, elle regroupe tous les enregistrements des activités de Ganesha. Cette association n'est valable que lorsque la séquence est de type quiz.
	Nous avons choisi les différentes activités par des méta-classes. Ainsi, les méta-classes de quiz et série de questions doivent être associées. Cette association est représentée sur le métamodèle final.
	Cette contrainte justifie encore notre choix de distinguer les méta-classes associées aux différentes activités.

Difficulté n°4 : Cas des éléments non existants au sein du modèle IHM

	Pendant la confrontation entre les deux modèles IHM et techniques, nous avons identifié un cas particulier concernant la non existence de certains éléments sur le modèle IHM. Ce cas concerne par exemple les deux éléments matière et domaine qui sont présentés par deux entités (<code>domaine</code> et <code>matiere</code>) au sein du modèle technique mais ils ne sont pas représentés au sein modèle IHM.
	Ce cas nous a amené à appliquer le processus de vérification d'existence d'un élément (section 4.4.4.1). Nous avons ré-établi l'analyse IHM en ce qui concerne ces deux éléments. Nous avons identifié des IHM et des fonctionnalités spécifiques à ces éléments. Cependant, nous avons constaté que ces deux éléments n'interviennent pas à la conception d'un module (cours). Leur objectif est d'organiser les modules (les cours). Ainsi, nous ne prenons pas en compte ces deux éléments au sein du métamodèle du langage de conception d'un cours de Ganesha.
	Il est très probable que le modèle technique possède des éléments non relevant de la conception pédagogique. Cela justifie notre choix de mener deux analyses différentes (IHM et technique) et ensuite de confronter les deux modèles produits.

C. Le métamodèle de conception pédagogique de Ganesha

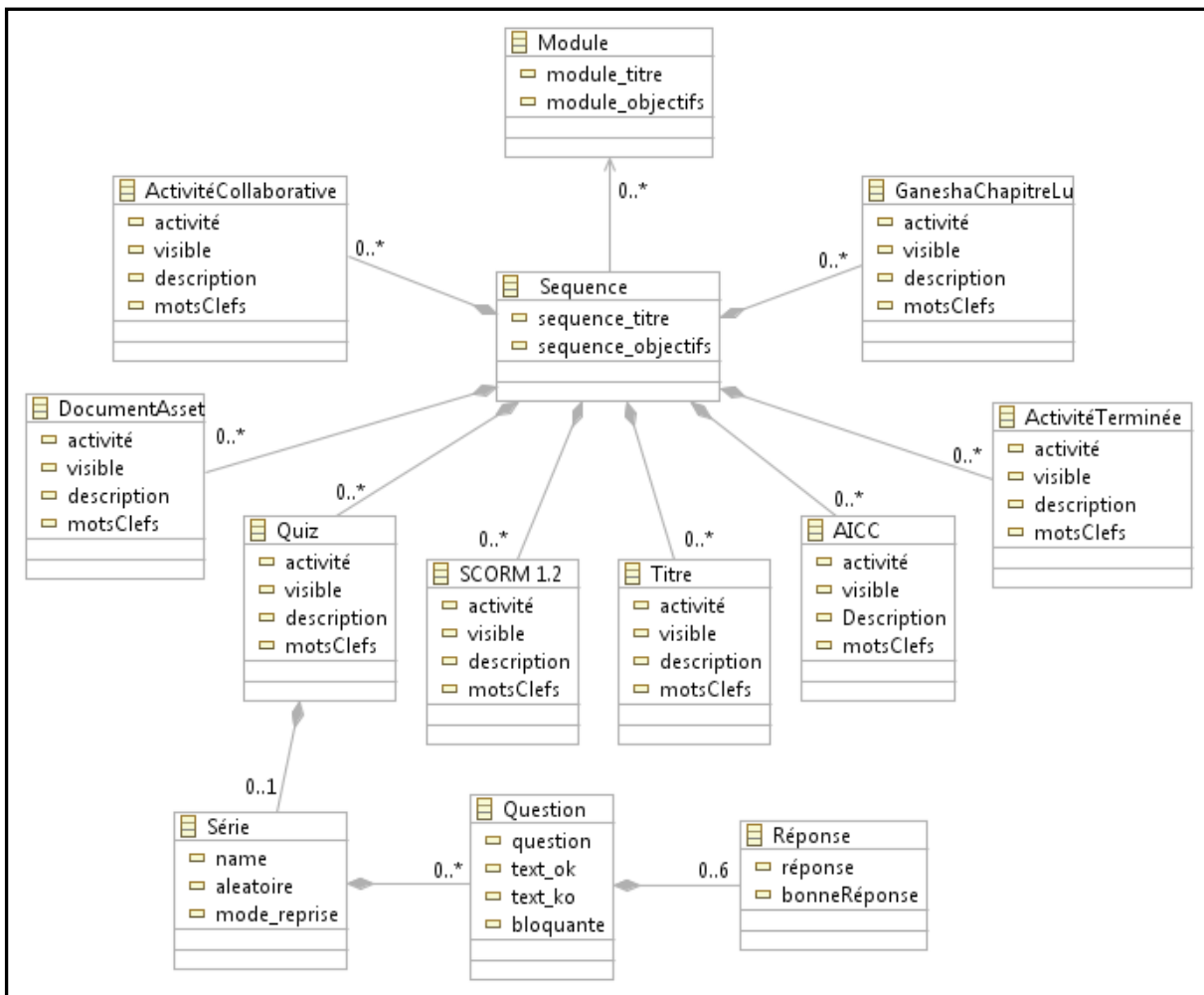


Figure B.30 : Métamodèle de conception pédagogique de la plateforme Ganesha

Le métamodèle du langage de conception pédagogique de Ganesha est le résultat de l'application du processus proposé. La figure B.30 illustre ce métamodèle.

Comme pour l'exemple générique de la confrontation, nous avons formalisé ce métamodèle au fur et à mesure de la vérification de l'existence d'un élément du modèle IHM avec son équivalent sur le modèle technique. Nous avons représenté chaque élément vérifié par une méta-classe. Ensuite, nous nous sommes intéressés à la vérification de leurs attributs.

Nous avons appliqué les processus de vérification pour les éléments et les attributs qui sont différemment représentés sur les modèles IHM et technique. Le paragraphe 4.2 illustre des exemples partiels que nous avons identifiés.

5. Conclusion

La plateforme Ganesha a représenté un deuxième environnement pour expérimenter le processus proposé. Cela a représenté un terrain de validation de nos propositions scientifiques et de nos choix. Nous

avons analysé la partie de conception pédagogique de cette plateforme de point de vue IHM et technique. Ensuite, la confrontation des deux modèles technique et IHM a servi pour la formalisation du métamodèle du langage de conception pédagogique.

Ce métamodèle peut servir pour la comparaison de son langage pédagogique avec ceux d'autres plateformes. Dans notre travail, nous l'avons utilisé comme base pour (1) le développement des outils de conception externe en suivant une approche de modélisation spécifique au domaine (*Domain Specific Modeling*) et (2) pour la spécification des modules de communication entre ces plateformes et les nouveaux outils de conception associés. Ces travaux de développement sont représentés au sein du sixième chapitre.

Au sein de l'application du processus sur la plateforme Ganesha, nous avons identifié et surmonté certaines issues et cas particuliers. Par exemple, l'analyse technique requiert des compétences informatiques en génie logiciel, en particulier le reverse engineering du modèle de la base, la génération du modèle conceptuel de données et l'analyse du code source de la plateforme.

Table des figures

Figure B.1 : Exemple de module sur Ganesha	263
Figure B.2 : Extrait des interfaces relevant de la conception pédagogique	264
Figure B.3 : Extrait des interfaces de spécification d'un cours	265
Figure B.4 : Modèle IHM-macro de la plateforme Ganesha	266
Figure B.5 : Interface de présentation des séquences sur Ganesha	267
Figure B.6 : Modèle fonctionnel des séquences	269
Figure B.7 : Interfaces de spécification du contenu des séquences	269
Figure B.8 : Modèle fonctionnel de l'interface 'Séquences'	270
Figure B.9 : Interface d'ajout d'une activité	270
Figure B.10 : Modèle fonctionnel de l'ajout d'activité	271
Figure B.11 : Extrait du modèle fonctionnel de Ganesha	272
Figure B.12 : Interface d'ajout d'un Quiz	273
Figure B.13 : Modèle IHM-micro d'un quiz	274
Figure B.14 : Interface d'ajout d'une question	275
Figure B.15 : Modèle IHM-micro d'une question	276
Figure B.16 : Modèle IHM-micro d'une réponse	276
Figure B.17 : Composition entre séquence et quiz	277
Figure B.18 : Composition entre module et séquence	278
Figure B.19 : composition entre question et réponse	278
Figure B.20 : Modèle IHM de Ganesha	279
Figure B.21 : Ecran de la table list_sequence	280
Figure B.22 : Modèle réduit de la table list_sequence	280
Figure B.23 : Ecran de la table serie_question	281
Figure B.24 : Extrait du code dédié aux questions	282
Figure B.25 : Modèle réduit de la table serie_question	283
Figure B.26 : Modèle conceptuel de donnée réduit de la plateforme Ganesha	283
Figure B.27 : Méta-classe du module	285
Figure B.28 : Extrait des instructions d'ajout d'activités	286
Figure B.29 : Définition de la table séquence	286
Figure B.30 : Métamodèle de conception pédagogique de la plateforme Ganesha	288

Annexe C Procédure d'import/export de Moodle et Ganesha

Annexe C : Procédure d'import/export de Moodle et Ganesha

1. Procédure d'import/export de Moodle

Dans la première partie de cette annexe, nous détaillons la procédure développée pour le processus d'export pour Moodle. La procédure se compose de deux parties :

- La première partie fixe, appelée *out_of_Moodle_fixed.xsl* est conçue pour exposer un modèle simple d'un cours Moodle. Elle consiste à transformer le paquetage de sauvegarde de Moodle dans un seul fichier XML conforme à schéma XML du langage de conception pédagogique de Moodle (*moodle.xsd*).
- La deuxième procédure, appelée *out_of_Moodle_generate.xsl*, consiste à générer la composante modulaire de la procédure *out_of_moodle.xsl*. Elle permet la lecture des informations à partir d'un schéma.

Nous allons présenter ci-dessous le code source de ces deux parties :

out of Moodle fixed.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Last tested version of Moodle : 2.0 RC1 (Build: 20101012) -->
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
xmlns:exsl="http://exslt.org/common" extension-element-prefixes="exsl"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!-- Parameter to know where to look for sub-files -->
  <xsl:param name="path" select=""/>
  <xsl:output method="xml" indent="yes"/>
  <xsl:strip-space elements="*/>

  <!-- Main rule -->
  <xsl:template match="/moodle_backup">
    <moodle>
      <!-- Add an attribute to specify which schema goes with the output document. -->
      <xsl:attribute name="xsi:noNamespaceSchemaLocation">
        <xsl:value-of select="'moodle.xsd'"/> <!-- Just to prevent spaces. -->
      </xsl:attribute>
      <xsl:apply-templates select="//course"/>
    </moodle>
  </xsl:template>

  <!-- Should happen only once. -->
  <xsl:template match="course">
    <course>
      <!-- Load the course sub-file -->
      <xsl:apply-templates mode="doc" select="document(concat($path, directory,
'/course.xml'))/course"/>
    </course>
  </xsl:template>
</xsl:transform>
```



```

<sections>
  <!-- Iterate on each section, ordering by the number hidden in each sub-file. -->
  <xsl:apply-templates select="../sections/section">
    <xsl:sort data-type="number" select="document(concat($path, directory,
'/section.xml'))/section/number"/>
  </xsl:apply-templates>
</sections>
</course>
</xsl:template>

<!-- Each section of the main file... -->
<xsl:template match="section">
  <section>
    <xsl:apply-templates mode="doc" select="document(concat($path, directory,
'/section.xml'))/section"/>
    <activities>
      <xsl:apply-templates select="../activities/activity[sectionid = current()/sectionid]"/>
    </activities>
  </section>
</xsl:template>

<!-- Here happens the logic to get activity contents from sub-files... Comment in case of not support-
ed module ! -->
<xsl:template match="activity">
  <xsl:variable name="content">
    <xsl:apply-templates mode="doc" select="document(concat($path, directory, '/', modulename,
'.xml'))/activity/*[name() = current()/modulename]"/>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test="exsl:node-set($content)/*[@id]">
      <xsl:copy-of select="$content"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:comment>
        <![CDATA[Module not handled : ]]>
        <xsl:copy-of select="modulename"/>
      </xsl:comment>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<!-- In case of unknown elements in sub-files, don't fail, just do nothing... -->
<!-- All others "doc" rules are generated w.r.t to the schema -->
<xsl:template mode="doc" match="*" />

<!-- Special rule to copy an id attribute -->
<xsl:template name="set-id">
  <xsl:attribute name="id">
    <xsl:choose>
      <xsl:when test="../@moduleid">
        <xsl:value-of select="../@moduleid"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="@id"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:attribute>

```

```

</xsl:choose>
</xsl:attribute>
</xsl:template>
<!-- default rule, do nothing -->
<xsl:template match="*" />
</xsl:transform>

```

out of Moodle generate.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:oxsl="http://not.important.at/all" xmlns:xsd="http://www.w3.org/2001/XMLSchema" exclude-
result-prefixes="xsd">

  <!-- Prepare the xsl output -->
  <!-- In the following, all oxsl: prefixed will be output, all xsl: will be processed -->
  <xsl:namespace-alias stylesheet-prefix="oxsl" result-prefix="xsl"/>
  <xsl:param name="path" select="."/ />
  <xsl:output method="xml" indent="yes" encoding="UTF8"/>

  <!-- On the root of the schema, just prepare the layout of the future transformation -->
  <xsl:template match="/">
    <oxsl:transform version="1.0">
      <!-- Th new transform will include the fixed part -->
      <oxsl:include href="{ $path }out_of_moodle_fixed.xml"/>
      <oxsl:output method="xml" indent="yes"/>

      <!-- And then we process every element declaration of the schema (no ordering) -->
      <xsl:apply-templates select="//xsd:element"/>
    </oxsl:transform>
  </xsl:template>

  <!-- For each element that has a name specified (but nothing else...) -->
  <xsl:template match="xsd:element[@name]">
    <!-- We generate a rule that will be used in sub-files -->
    <oxsl:template mode="doc" match="{ ../..../@name }/{ @name }">
      <!-- just copy itself -->
      <oxsl:copy>
        <!-- and recurse... -->
        <oxsl:apply-templates mode="doc"/>
      </oxsl:copy>
    </oxsl:template>
  </xsl:template>

  <!-- For each element that has a name specified and needs an id attribute -->
  <xsl:template match="xsd:element[@name][./xsd:complexType/xsd:attribute[@*='id']]">
    <!-- We generate a rule that will be used in sub-files -->
    <oxsl:template mode="doc" match="{ @name }[@id]">
      <!-- prepare the call to the relevant rule to get the id -->
      <oxsl:call-template name="set-id"/>
      <!-- and make recurse... -->
      <oxsl:apply-templates mode="doc"/>
    </oxsl:template>
  </xsl:template>

```

```

<!-- Same but only for stuff inside an activity ! -->
<!-- For each element that has a name specified and needs an id attribute -->
<xsl:template
match="xsd:element[@name][./xsd:complexType/xsd:attribute[@*='id']][ancestor::*[@name='activiti
es']]">
  <!-- We generate a rule that will be used in sub-files -->
  <oxsl:template mode="doc" match="{@name}{@id}">
    <!-- and copy and recurse... -->
    <oxsl:copy>
      <!-- prepare the call to the relevant rule to get it -->
      <oxsl:call-template name="set-id"/>
      <oxsl:apply-templates mode="doc"/>
    </oxsl:copy>
  </oxsl:template>
</xsl:template>

<!-- If the schema specify this element has a type, then prepare a rule to just copy it -->
<!-- all others will be discarded -->
<!-- Beware that any @type can be specified, but only those which can be represented as a string are
relevant. So don't define (and reference) complexType in the schema without modifying that... -->
<xsl:template match="xsd:element[@name][@type]xsd:element[xsd:simpleType]" priority="2">
  <oxsl:template mode="doc" match="{../..../@name}/{@name}">
    <oxsl:copy-of select="."/>
  </oxsl:template>
</xsl:template>
</xsl:transform>

```

2. Procédure d'import/export de Ganesha

Dans la deuxième partie de cette annexe (C), nous présentons un extrait du code source que nous avons spécifié pour le processus d'import/export dédié à la plateforme Ganesha. Les instructions sont codées en même langage de cette plateforme : le langage PHP.

La partie ajoutée dans le menu intégrateur au sein du fichier ganesha\inc\admin.inc.php est coloré en gris.

```

...
elseif (GUSERTYPE==TYPE_INTEGRATEUR) {
    if ( $ClsACL->autorise($PAGE_ADMIN[$i],ACL_LIST) ) {
        $ret['menu'][$index]['texte'] = getSousMenuTitre($i) ;
        $ret['menu'][$index]['link'] = $url;
        $ret['menu'][$index]['selected'] = $select;
        $index++;
        $ret['menu'][$index]['texte'] = "block_import";
        $ret['menu'][$index]['link'] = HTTP_ADMIN.'newimport.php'?connexion='.CNX.'&
action=liste';
        $ret['menu'][$index]['selected'] = 1;}}
...

```

Le code ci-dessous représente les fonctionnalités que nous avons développé pour opérationnaliser et exporter les scénarios pédagogiques dédiés à la plateforme Ganesha.

```

<?php
// les prérequis
require_once("../anema.inc.php");
require_once(RP_CLS.'site.class.php');
require_once(RP_INC.'bouton.inc.php') ;
require_once(RP_INC.'admin.inc.php');
require_once (RP_CLS.'admin.class.php');
require_once(RP_CLS.'module.class.php');
require_once(RP_CLS.'domaine.class.php');
require_once(RP_CLS.'matiere.class.php');
require_once(RP_CLS.'sequence.class.php');
require_once(RP_CLS.'membre.class.php');
require_once(RP_CLS.'serie_question.class.php');
require_once(RP_CLS.'chapitre.class.php');
require_once(RP_CLS.'scorm.class.php');

debugf(__FILE__);
define('SUBMENU', SMENU_MODULE);
define('ACTION', read_get_str('action',"liste"));
define('ID', read_get_int('id'));
define('MENU', MMENU_ADMIN);
define('SUBMENU', SMENU_ACCEUIL);
define('PAGE',HTTP_ADMIN."newimport.php?menu=".MENU."&submenu=".SUBMENU."
&connexion=".CNX);
define('PAGE2',HTTP_ADMIN."module.php?menu=".MENU."&submenu=".SUBMENU."& con-
nexion =".CNX);

```

```

function init() {
    global $ClsACL, $PAGE_ADMIN, $ClsDomaine, $ClsMatiere, $ClsMembre, $ClsSequence,
    $ClsModule, $ClsParcours, $ClsRSS, $ClsACL, $PAGE_ADMIN;
    if ( !$ClsACL->autorise($PAGE_ADMIN[SUBMENU],ACL_LIST)) {
        $page = $ClsACL->defaultPage() ;
        if ( $page == " ) {
            PageErreur('action=acl') ;}
        location($page."?connexion=".CNX);
        exit() ;}}

```

```

function doActionWin($action) {}

```

```

function ajouterUnDomaine($name,$code,$directory){
    global $ClsDomaine ;
    $ret=$ClsDomaine->add($name,$code,$directory);
    return $ret;}

```

```

function modifierUnDomaine($domaine_id,$name,$code,$directory){
    global $ClsDomaine ;
    $ClsDomaine->update($domaine_id,$name,$code,$directory);
    return(OK);}

```

```

function ajouterUneMatiere($matiere,$idd,$code){

```

```
global $ClzMatiere ;
$ret= $ClzMatiere->add($matiere,$idd,$code);
return $ret;}
```

```
function modifierUneMatiere($idm,$matiere,$code){
    global $ClzMatiere ;
    $ClzMatiere->update($idm,$matiere,$code);
    return(OK);}
```

```
function ajouterUneSequence($titre,$code,$aut,$support,$objectifs,$duree,$bar=0) {
    global $ClzSequence ;
    return($ClzSequence->add($titre,$code,$aut,$support,$objectifs,$duree,$bar));}
```

```
function updateNumeroChapitre($id_sequence, $numero, $nouveauNumero){
    global $db ;
    $sSQL = "UPDATE sequence SET numero='".$nouveauNumero.'" ";
    $sSQL.= "WHERE id_sequence='".$id_sequence.'" AND numero='".$numero.'" ";
    $db->query($sSQL);}
```

```
function updateChapitresSuivantes($id_sequence, $numero){
    global $db ;
    $tab=$this->chercherActivitesSuivantes($sequence_id, $numero);
    for ($i=0; $i<$tab['cnt'];$i++){
        $sui=0;
        $sui=$tab[$i]+1;
        $sSQL = "UPDATE sequence SET numero='".$sui.'" ";
        $sSQL.= "WHERE id_sequence='".$id_sequence.'" AND numero='".$tab[$i]."'";
        $db->query($sSQL);
        echo 'upChap'.$tab[$i].'  
';}}
```

```
function chercherActivitesSuivantes($sequence_id, $numero){
    global $db ;
    $tab = Array() ;
    $sSQL = "SELECT numero FROM sequence where numero>".$numero." order by numero DESC ";
    $this->_db->query($sSQL);
    $i=0;
    $tab['cnt']= $this->_db->num_rows();
    while( $this->_db->next_record() ) {
        $tab [$i] = $this->_db->f('numero');
        $i++;}
    return $tab;}
```

```
function doAction($action) {
    switch($action) {
        case 'liste': return($this->liste()); break;
        case 'modif': return($this->modif()); break;
        case 'exporter': return($this->exporter()); break;
        case 'afficheModuleParAuteur': return($this->afficheModuleParAuteur()); break;
```

```
default : die('Erreur action '.$action);}}
```

```
function chercherDomaineEtMatiereParModuleEtAuteur($module_id, $auteur_id) {
  $ret = Array() ;
  $where= "where auteur_id='".$auteur_id.'"and module_id='".$module_id.'" ";
  $$SQL = "SELECT domaine_id, matiere_id FROM list_module ";
  $$SQL.= $where;
  $this->_db->query($$SQL);
  $ret['cnt']= $this->_db->num_rows();
  if ( $this->_db->next_record() ) {
    $ret['domaine_id'] = $this->_db->f('domaine_id');
    $ret['matiere_id'] = $this->_db->f('matiere_id');
  }
  return ($ret);}
```

```
function chercherDomaineDuModule($domaine_id) {
  $$SQL = "SELECT name FROM domaine where domaine_id='".$domaine_id.'" ";
  $this->_db->query($$SQL);
  if ( $this->_db->next_record() ) return ( $this->_db->f('name') );
  else return "";}

```

```
function chercherMatiereDuModule($matiere_id) {
  $where= "where matiere_id='".$matiere_id.'" ";
  $$SQL = "SELECT name FROM matiere ";
  $$SQL.= $where;
  $this->_db->query($$SQL);
  if ( $this->_db->next_record() ) return ( $this->_db->f('name') );
  else return "";}

```

```
function chercherSequenceParDomaine($auteur_id, $domaine_id) {
  $tSequence = Array() ;
  $where="where auteur_id='".$auteur_id.'"and domaine_id='".$domaine_id.'" ";
  $$SQL = "SELECT sequence_id,sequence_titre,repertoire,auteur_id,code,";
  $$SQL.= "domaine_id,matiere_id,sequence_objectifs,sequence_duree " ;
  $$SQL.= "FROM list_sequence " ;
  $$SQL.= $where;
  $this->_db->query($$SQL);
  while( $this->_db->next_record() ) {
    $tSequence['id']          = $this->_db->f('sequence_id');
    $tSequence['titre']      = $this->_db->f('sequence_titre');
    $tSequence['repertoire'] = $this->_db->f('repertoire');
    $tSequence['code']       = $this->_db->f('code');
    $tSequence['domaine_id'] = $this->_db->f('domaine_id');
    $tSequence['matiere_id'] = $this->_db->f('matiere_id');
    $tSequence['objectifs']  = $this->_db->f('sequence_objectifs');
    $tSequence['duree']     = $this->_db->f('sequence_duree');
    $tSequence['auteur_id'] = $this->_db->f('auteur_id');
    $tSequence['loc']       = $this->loc;}
  return($tSequence);}
```

```
function chercherSequenceParDomaineOuMatiere($auteur_id, $domMat, $id) {
```

```

$Sequence = Array() ;
$where="where auteur_id='".$auteur_id.'" and $domMat = '".$id.'" ";
$$SQL = "SELECT sequence_id,sequence_titre,repertoire,auteur_id,code,";
$$SQL.= "domaine_id,matiere_id,sequence_objectifs,sequence_duree " ;
$$SQL.= "FROM list_sequence ";
$$SQL.= $where;
$this->_db->query($$SQL);
$Sequence['cnt'] = $this->_db->num_rows();
$i = 0 ;
while( $this->_db->next_record() ) {
    $Sequence[$i]['id']           = $this->_db->f('sequence_id');
    $Sequence[$i]['titre']       = $this->_db->f('sequence_titre');
    $Sequence[$i]['repertoire']  = $this->_db->f('repertoire');
    $Sequence[$i]['code']        = $this->_db->f('code');
    $Sequence[$i]['domaine_id']  = $this->_db->f('domaine_id');
    $Sequence[$i]['matiere_id']  = $this->_db->f('matiere_id');
    $Sequence[$i]['objectifs']   = $this->_db->f('sequence_objectifs');
    $Sequence[$i]['duree']       = $this->_db->f('sequence_duree');
    $Sequence[$i]['auteur_id']   = $this->_db->f('auteur_id');
    $Sequence[$i]['loc']         = $this->loc;
    $i++ ;}
return($Sequence);}

```

```

function chercherSequenceParModule($auteur_id, $module_id) {
    $NumSequence = Array();
    $sql1 = "SELECT sequence_id FROM module ";
    $where= "where module_id='".$module_id.'" ";
    $sql1.= $where;
    $this->_db->query($sql1);
    $NumSequence['cnt']=$this->_db->num_rows();
    $j=0;
    while( $this->_db->next_record() ) {
        $NumSequence[$j]['sequence_id']=$this->_db->f('sequence_id');
        $j++;}
    $Sequence = Array() ;
    $Sequence['cnt']=$NumSequence['cnt'];
    for ($i=0; $i<$Sequence['cnt'];$i++){
        $$SQL = "SELECT sequence_id,sequence_titre,repertoire,auteur_id,code,";
        $$SQL.= "domaine_id,matiere_id,sequence_objectifs,sequence_duree " ;
        $$SQL.= "FROM list_sequence ";
        $where="where sequence_id='".$NumSequence[$i]['sequence_id'].'" ";
        $$SQL.= $where;
        $this->_db->query($$SQL);
        while( $this->_db->next_record() ) {
            $Sequence[$i]['id']           = $this->_db->f('sequence_id');
            $Sequence[$i]['titre']       = $this->_db->f('sequence_titre');
            $Sequence[$i]['repertoire']  = $this->_db->f('repertoire');
            $Sequence[$i]['code']        = $this->_db->f('code');
            $Sequence[$i]['domaine_id']  = $this->_db->f('domaine_id');
            $Sequence[$i]['matiere_id']  = $this->_db->f('matiere_id');
            $Sequence[$i]['objectifs']   = $this->_db->f('sequence_objectifs');
            $Sequence[$i]['duree']       = $this->_db->f('sequence_duree');
            $Sequence[$i]['auteur_id']   = $this->_db->f('auteur_id');
            $Sequence[$i]['loc']         = $this->loc;}
    }
}

```

```

    }
    return($tSequence);}

```

```

function chercherChapitreParSequence($sequence_id){
    global $ClsChapitre;
    return $ClsChapitre->read($sequence_id,$idu=-1,$idg=-1);}

```

```

function chercherModuleParAuteur($auteur_id){
    $tListModuleAuteur = Array();
    $sql1 = "SELECT module_id, module_titre, module_objectifs, domaine_id, matiere_id FROM
    list_module ";
    $where= "where auteur_id='".$auteur_id.'" ";
    $sql1.=$where;
    $this->_db->query($sql1);
    $tListModuleAuteur['cnt']= $this->_db->num_rows();
    $j=0;
    while( $this->_db->next_record() ) {
        $tListModuleAuteur[$j]['module_id']=$this->_db->f('module_id');
        $tListModuleAuteur[$j]['module_titre']=$this->_db->f('module_titre');
        $tListModuleAuteur[$j]['module_objectifs']=$this->_db->f('module_objectifs');
        $tListModuleAuteur[$j]['domaine_id']=$this->_db->f('domaine_id');
        $tListModuleAuteur[$j]['matiere_id']=$this->_db->f('matiere_id');
        $j++;}
    return $tListModuleAuteur;}

```

```

function chercherModuleParID($module_id){
    $tListModuleId = Array();
    $sql1 = "SELECT module_id, module_titre, module_objectifs, domaine_id, matiere_id FROM
    list_module ";
    $where= "where module_id='".$module_id.'" ";
    $sql1.=$where;
    $this->_db->query($sql1);
    $tListModuleId['cnt']= $this->_db->num_rows();
    while( $this->_db->next_record() ) {
        $tListModuleId['module_id']=$this->_db->f('module_id');
        $tListModuleId['module_titre']=$this->_db->f('module_titre');
        $tListModuleId['module_objectifs']=$this->_db->f('module_objectifs');
        $tListModuleId['domaine_id']=$this->_db->f('domaine_id');
        $tListModuleId['matiere_id']=$this->_db->f('matiere_id');}
    return $tListModuleId; }

```

```

function afficheModuleParAuteur(){
    $tListModuleAuteur=$this->chercherModuleParAuteur(GUSERID);
    $codeHTML.= " ";
    $codeHTML.= SubTitleAdmin("Export de module");
    $form = new HTML_Form(PAGE."&action=exporter&id=".ID,__("export des scénarios"));
    $form->addTitreText(__(" "));
    $choix = array();
    for($i=0;$i<$tListModuleAuteur['cnt'];$i++) {
        $choix[] = array('id'=>$tListModuleAuteur[$i]['module_id'], 'name'=>__(". $tList
        ModuleAuteur[$i]['module_titre']. " ". $tListModuleAuteur[$i]['module_id']));}

```



```

$form->addTabChoice($choix,0,__("Selectionner les modules à exporter"));
$form->addFreeRow();
$form->addSubmit("",'exporter 2/2');
$codeHTML.= $form->toHTML();
return $codeHTML;}

```

```

function liste(){
    global $CIsMembre, $CIsACL, $PAGE_ADMIN, $CIsimport;
    $codeHTML= SubTitleAdmin("import / export");
    /* _____ FORMULAIRE IMPORTER _____ */
    $form = new HTML_Form(PAGE."&action=modif&id=".ID,__("Import des scénarios"));
    $form->addTitreText(__(" "));
    $form->addTitreText(__(" "));
    $form->addTextFile(__("Slectionner le fichier du scénario"));
    $form->addSubmit("", __("importer"));
    $codeHTML.= $form->toHTML();
    /* _____ FORMULAIRE EXPORTER _____ */
    $form1 = new HTML_Form(PAGE."&action=afficheModuleParAuteur&id=".ID,__("Export
    des scénarios"));
    $form1->addTitreText(__(" "));
    $form1->addTitreText(__(" "));
    $form1->addTextInfo("Cliquer sur le bouton pour exporter le cours", "");
    $form1->addSubmit("",'exporter');
    $codeHTML.= $form1->toHTML();
    return $codeHTML;}

```

```

function exporter(){
    global $CIsSerieQuestion;
    $tab=array();
    $tab[0][0]='Activite'; $tab[1][0]='titre';
    $tab[0][1]='Description' ;$tab[1][1]='description';
    $tab[0][2]='MotsClefs'; $tab[1][2]='keyword';
    $tab[0][3]='Visible'; $tab[1][3]='is_visible';
    $tab[0][4]='Titre'; $tab[1][4]='module_titre';
    $tab[0][5]='Objectifs'; $tab[1][5]='module_objectifs';
    $tab[0][6]='Titre'; $tab[1][6]='titre';
    $tab[0][7]='Objectifs'; $tab[1][7]='objectifs';
    $tab[0][8]='Titre'; $tab[1][8]='name';
    $tab[0][9]='SerieAleatoire'; $tab[1][9]='aleatoire';
    $tab[0][10]='ModeReprise'; $tab[1][10]='mode_reprise';
    $tab[0][11]='Question'; $tab[1][11]='question';
    $tab[0][12]='TextCorrect'; $tab[1][12]='text_ok';
    $tab[0][13]='TextIncorrect'; $tab[1][13]='text_ko';
    $tab[0][14]='Bloquante'; $tab[1][14]='bloquante';
    $tab[0][15]='Ponderation'; $tab[1][15]='ponderation';
    $tab[0][16]='module'; $tab[1][17]='id';
    $tab[0][18]=""; $tab[1][18]="";
    $tab[0][19]=""; $tab[1][19]="";

    $document = new DomDocument('1.0', 'UTF-8');
    $document->formatOutput = TRUE;
    // récupérer l'ID du module à exporter
    $ganesha=$document->createElement('ganesha');

```

```

$domaine=$document->createElement('domaine');
$document->appendChild($ganesha);
$ganesha->appendChild($domaine);

$module_id=read_post_int('text1');
$module=$this->chercherModuleParID($module_id);
$moduleMatiere=$this->chercherDomaineEtMatiereParModuleEtAuteur($module_id, GUSERID);
$domaine->setAttribute("id", $moduleMatiere['domaine_id']);

$domaine_titre=$document->createElement('Titre');
$domaine->appendChild($domaine_titre);
$txt=$document->createTextNode($this->chercherDomaineDuModule($module['domaine_id']));
$domaine_titre->appendChild($txt);
// matiere
$matiere=$document->createElement('matiere');
$domaine->appendChild($matiere);
$matiere->setAttribute("id", $moduleMatiere['matiere_id']);
$name=$document->createElement('name');
$matiere->appendChild($name);
$txt=$document->createTextNode(utf8_encode($this->chercherMatiereDuModule($module['matiere_id'])););
$name->appendChild($txt);
// module
$module=$document->createElement('module');
$matiere->appendChild($module);
$module->setAttribute("id", $module_id);
$this->createElm($document, $module, $module, $stab[0][4], $stab[1][4]);
$this->createElm($document, $module, $module, $stab[0][5], $stab[1][5]);

// sequence
// chercher les séquences du module
$sequence=$this->chercherSequenceParModule(GUSERID, $module_id);
for ($i=0; $i<$sequence['cnt'];$i++)
{
    $sequence=$document->createElement('sequence');
    $module->appendChild($sequence);
    $sequence->setAttribute("id", $sequence[$i]['id']);
    $this->createElm($document, $sequence, $sequence[$i], $stab[0][6], $stab[1][6]);
    $this->createElm($document, $sequence, $sequence[$i], $stab[0][7], $stab[1][7]);

    //chercher les chapitre de la séquence
    $tchapitre=$this->chercherChapitreParSequence($sequence[$i]['id']);
    for($j=0;$j<$tchapitre['cnt'];$j++){ //
        if ($tchapitre[$j]['titre'] !='Support') {
            switch($tchapitre[$j]['type']) {
                case '0' : $activite=$document->createElement('titre');break;
                case '1' : $activite=$document->createElement('DocumentASSET');break;
                case '2' : $activite=$document->createElement('GaneshaChapitreLu');break;
                case '3' : $activite=$document->createElement('ActiviteTerminee');break;
                case '4' : $activite=$document->createElement('AICC');break;
                case '5' : $activite=$document->createElement('SCORM12');break;
                case '8' : $activite=$document->createElement('ActiviteCollaborative');break;
                case '9' : $activite=$document->createElement('Quiz');break;}

            $sequence->appendChild($activite);

```

```

$activite->setAttribute("numero", $tchapitre[$j]['numero']);
for($i=0;$i<4;$i++)
  $this->createElm($document, $activite, $tchapitre[$j], $stab[0][$i], $stab[1][$i]);
  if($tchapitre[$j]['type'] ==9){
    $serie=$document->createElement('serie');
    $activite->appendChild($serie);
    // il faut chercher la série concernée ; tjrs il ya une seule serie pour un quiz
    $serie_id= substr($tchapitre[$j]['launch'],3);
    $serie=$ClsSerieQuestion->read($serie_id);
    $serie->setAttribute("id", $serie_id);
    $this->createElm($document, $serie, $serie, $stab[0][8], $stab[1][8]);
    $this->createElm($document, $serie, $serie, $stab[0][9], $stab[1][9]);
    $this->createElm($document, $serie, $serie, $stab[0][10], $stab[1][10]);
    // les questions de la série
    $tquestion=$ClsSerieQuestion->lister($serie_id);

    for ($c=0;$c<$tquestion['cnt']; $c++){
      $question=$document->createElement('question');
      $serie->appendChild($question);
      $question->setAttribute("id", $tquestion[$c]['id']);
      $this->createElm($document, $question, $tquestion[$c], $stab[0][11],
$stab[1][11]);
      $this->createElm($document, $question, $tquestion[$c], $stab[0][12], $stab[1][12]);
      $this->createElm($document, $question, $tquestion[$c], $stab[0][13], $stab[1][13]);
      $this->createElm($document, $question, $tquestion[$c], $stab[0][14], $stab[1][14]);
      $this->createElm($document, $question, $tquestion[$c], $stab[0][15], $stab[1][15]);

      $rep=$tquestion[$c]['reponse'];
      $ri=explode(";", $rep);
      for ($k=1;$k<7;$k++){ //toujours 6 réponses au maximum
        $trouve=0;$r=0;
        if ($tquestion[$c]["r".$k]!=""){
          $reponse=$document->createElement('Reponse');
          while (($trouve==0)&& $r<count($ri)){
            if($ri[$r]==$k) $trouve=1;
            $r++;}
          $reponse->setAttribute("BonneReponse", $trouve);
          $question->appendChild($reponse);
          $txt=$document->createTextNode($tquestion[$c]["r".$k]);
          $reponse->appendChild($txt);}
        }
      }
    }
  }
}
$document->save('C:\Windows\Temp\ganesha.xml');//('C:\Users\MILES\Desktop\fichier9.xml');

// Constantes de configuration //
// Nom du fichier pour le système
define('CFG_SYSTEM_FILENAME', 'C:\Windows\Temp\ganesha.xml');
// Nom du fichier pour le navigateur
define('CFG_SEND_FILENAME', CFG_SYSTEM_FILENAME);

// Constantes à ne pas modifier //
define('CFG_FILESIZE', filesize(CFG_SYSTEM_FILENAME));
define('CFG_FILE_MD5', md5_file(CFG_SYSTEM_FILENAME));
define('CFG_DATE_FORMAT', 'D, d M Y H:i:s');

// Eléments nécessaires //

```

```

error_reporting(0);
ini_set('zlib.output_compression', 0);
/* Les en têtes nécessaires */
// Gestion du cache //
header('Pragma: public');
header('Last-Modified: ' . gmdate(CFG_DATE_FORMAT) . ' GMT');
header('Cache-Control: must-revalidate, pre-check=0, post-check=0, max-age=0');
// Informations sur le contenu à envoyer //
header('Content-Transfer-Encoding: none');
header('Content-Length: ' . CFG_FILESIZE);
header('Content-MD5: ' . base64_encode(CFG_FILE_MD5));
header('Content-Type: application/octetstream; name="' . CFG_SEND_FILENAME . '");
header('Content-Disposition: attachment; filename="' . CFG_SEND_FILENAME . '");
// Informations sur la réponse HTTP elle-même //
header('Date: ' . gmdate(CFG_DATE_FORMAT, time()) . ' GMT');
header('Expires: ' . gmdate(CFG_DATE_FORMAT, time()+1) . ' GMT');
header('Last-Modified: ' . gmdate(CFG_DATE_FORMAT, time()) . ' GMT');
/* Envoi du fichier */
readfile(CFG_SYSTEM_FILENAME);
} // fin exporter()

```

```

function modif(){
    $saut = GUSERID;
    global $ClsACL,$PAGE_ADMIN,$ClsImportSCN;
    // à revoir //if ( !$ClsACL->autorise($PAGE_ADMIN[SUBMENU],ACL_MOD)) { PageEr-
    reur('action=acl');}

    $filer= $_FILES['text1'];
    $filerName=$filer['name'];
    $filerTmpName=$filer['tmp_name'];
    $codeHTML = "";
    $codeHTML.= SubTitleAdmin("import / export");
    $form = new HTML_Form(PAGE2."&action=liste&id=".ID.__("Import / export des
    scénarios"));

    $form->addTitreText(__(" "));
    $form->addTitreText(__(" "));
    $form->addTextInfo(" Le cours est bien créé sur la plateforme ", "");
    $form->addSubmit("", __("voir le cours"));
    $dom = new DomDocument();
    $dom->load($filerTmpName);
    $ganesha= $dom->documentElement;
    $this->parcours2($dom, $ganesha, $saut);
    $codeHTML.= $form->toHTML();
    return $codeHTML;}

```

```

function parcours2($dom, $ganesha, $saut){

    global $domaine_id, $matiere_id, $ClsAdmin, $ClsModule;
    foreach ($ganesha->childNodes as $node){
        if($node->nodeType == XML_ELEMENT_NODE){

```

```

/* _____ DOMAINE _____ */
if ($node->tagName=='domaine') {
    $domaine_id=$node->getAttributeNode('domaine_id')->nodeValue;
    //$domaine_titre=$dom->getElementsByTagName('domaine_titre')->item(0)->nodeValue;
    foreach ($node->childNodes as $nodesDomaine){
        if ($nodesDomaine->tagName=='domaine_titre'){
            $domaine_titre=$nodesDomaine->firstChild->nodeValue;
            if ($domaine_id==0){
                /** Ajoute un nouveau domaine * @param string $domaine titre du domaine */
                $domaine_id = $this->ajouterUnDomaine($domaine_titre,"");}
            else{
                /** maj d'un domaine * @param integer $idd id du domaine , @param string
$domaine titre du domaine */
                $this->modifierUnDomaine($domaine_id,$domaine_titre,"");}
        }
    }
/* _____ MATIERE _____ */
elseif ($nodesDomaine->tagName=='matiere') {
    $matiere_id=$nodesDomaine->getAttributeNode('matiere_id')->nodeValue;
    $matiere_name=$nodesDomaine->firstChild->nodeValue;
    foreach ($nodesDomaine->childNodes as $nodesMatiere){
        if ($nodesMatiere->tagName=='name'){
            if ($matiere_id==0){
                /** Ajoute unz nouvelle matiere**/
                $matiere_id=$this->ajouterUneMatiere($nodesMatiere-
>nodeValue,$domaine_id,""); }
            else{
                $this->modifierUneMatiere($matiere_id,$nodesMatiere->nodeValue,"");}
        }
    }
/* si le module existe je propose de parcourir le module en entier en vérifiant l'existence de chaque
element dans lenouveau fichier. S'il n'existe pas, on supprime */

/* _____ MODULE _____ */
elseif ($nodesMatiere->tagName=='module'){
    $module_id=$nodesMatiere->getAttributeNode('module_id')->nodeValue;
    foreach ($nodesMatiere->childNodes as $nodesModule){
        if ($nodesModule->tagName=='module_titre'){
            $module_titre=$nodesModule->nodeValue; }
        elseif ($nodesModule->tagName=='module_objectifs'){
            $module_objectifs=$nodesModule->nodeValue;
            if ($module_id==0){ /*** Ajoute un nouveau module/
            $module_id=$ClsAdmin-
>ajouterUnModule($module_titre,"$aut,$module_objectifs);

            $ClsAdmin->modifierDomaineModule($module_id,$domaine_id);
            $ClsAdmin->modifierMatiereModule($module_id,$matiere_id);}
            else{ /*** mise à jour du module /
            $ClsAdmin-
>modifierUnModule($module_id,$module_titre,"$aut,$module_objectifs);
            $ClsAdmin->modifierDomaineModule($module_id,$domaine_id);
            $ClsAdmin->modifierMatiereModule($module_id,$matiere_id);} }

/* _____ SEQUENCE _____ */
elseif ($nodesModule->tagName=='sequence'){
    $position=0;$inserted=0;$modif=0;
    $sequence_id=$nodesModule->getAttributeNode('sequence_id')-

```

```

>nodeValue;
        foreach ($nodesModule->childNodes as $nodesSequences){
            if ($nodesSequences->tagName=='sequence_titre'){

                $sequence_titre=$nodesSequences->nodeValue;
            }
            elseif ($nodesSequences->tagName=='sequence_objectifs'){
                $sequence_objectifs=$nodesSequences->nodeValue;
                if ($sequence_id==0){ /** Ajoute un nouveau module /
                    $sequence_id=$ClsAdmin-
>ajouterUneSequence($sequence_titre,$code=-1,$aut,$support,$sequence_objectifs,$duree,0);

                    $ClsAdmin->modifierDomaineSequence($sequence_id, $do
maine_id);
                    $ClsAdmin->modifierMatiereSequence($sequence_id, $ma
tiere_id);
// pour relier les séquences à leurs modules
                    $ClsAdmin->ajouterUneSequenceModule($module_id,1,
                    $sequence_id,1,0) ;}
                else{ /** mise à jour du module /
                    $ClsAdmin->modifierUneSequence($sequence_id, $sequence
_titre,$code,$aut,$support,$sequence_objectifs,$duree,$bar=0);
                    $ClsAdmin->modifierDomaineSequence($sequence_id, $do
maine_id);
                    $ClsAdmin->modifierMatiereSequence($sequence_id, $ma
tiere_id);}}

                /* _____ACTIVITE_____ */
            }
            elseif (($nodesSequences->tagName=='activite')|| ($nodesSe
quences->tagName=='activitequiz')){
                $position++;
                $numero=$nodesSequences->getAttributeNode('numero')-
>nodeValue;
                $visible=$nodesSequences->getAttributeNode('visible')-
>nodeValue;

                foreach ($nodesSequences->childNodes as $nodesActivite){
                    if ($nodesActivite->tagName=='type'){
                        $type=$nodesActivite->nodeValue;
                        switch($type){
                            case 'Titre' : $ttype=0;break;
                            case 'Document(ASSET)' : $ttype=1;break;
                            case 'Ganesha(Chapitre lu)' : $ttype=2;break;
                            case 'Activite terminee' : $ttype=3;break;
                            case 'AICC' : $ttype=4;break;
                            case 'SCORM 1.2' : $ttype=5;break;
                            case 'Activite Collaborative' : $ttype=8;break;
                            case 'Activite Quiz' : $ttype=9;break; }
                    }
                    elseif ($nodesActivite->tagName=='activite_titre'){
                        $activite_titre=$nodesActivite->nodeValue;
                    }
                    elseif ($nodesActivite->tagName=='description'){
                        $description=$nodesActivite->nodeValue;
                    }
                    elseif ($nodesActivite->tagName=='motsClefs'){
                        $motsClefs=$nodesActivite->nodeValue;
                    }
                }
            }
        }
// après avoir acquérir tous les paramètres de l'activité on l'insère dans le dernier parametre récupéré

```

```

// a voir prerequis idscorm regroupe positionnement=0 ou 1

if ($numero==0){
    $inserted++;

    // il faut augmenter toutes les activités suivantes
    $this->updateChapitresSuivantes($sequence_id,
$position-1);

    $numero= $ClsAdmin->ajouterUnChapitre($sequence_id,$ttype,$activite_titre,$fic='aa',$window=2,$visible,1, $mastery=0,$maxtime=",$prerequis=",$idscore=",$description,$regroupe=",$motsClefs,$launch=",$positionnement=0, $tab_code_ifi=",$time_limit_action=",$fic_correct=",$correct_auto=0,$prerequis_mode=1);

    // il faut changer le numéro pour respecter l'ordre dans le fichier de l'éditeur
    $this->updateNumeroChapitre($sequence_id, $numero, $position); }

else{// mise à jour de l'activité
// faire la mise à jour que la premiere fois !!!!

    $ccc=$numero+$inserted;
    $ClsAdmin->modifierUnChapitre($sequence_id,$ccc,$ttype,$activite_titre,$fic=",$window=2,$visible,1, $mastery=0,$maxtime=",$prerequis=",$idscore=",$description=",$regroupe=",$motsClefs=",$launch,$positionnement=0,$tab_code_ifi=",$time_limit_action=",$fic_correct=",$correct_auto=0,$prerequis_mode=1);
    $modif++;}}

/* _____ SERIE _____ */

elseif ($nodesActivite->tagName=='serie'){
    $serie_id=$nodesActivite->getAttributeNode('serie_id')->nodeValue;
    foreach($nodesActivite->childNodes as $nodesSerie){
        if ($nodesSerie->tagName=='serie_titre'){
            $serie_titre=$nodesSerie->nodeValue;
            if ($serie_id==0){
                $serie_id=$ClsAdmin->ajouterUneSerie($serie_titre,$code=",$salea=0,$mode_reprise=0); }
            else{//mise à jour de la serie
                $ClsAdmin->modifierUneSerie($serie_id,$serie_titre,$code=",$salea=0,$nbq=0,$mode_reprise=0);}
                //mise à jour du chapitre pour lier la série au quiz
                $launch = 'id='.$serie_id;
                if ($modif==0) $cc=$numero;
                else $cc=$numero+$inserted;
                $ClsAdmin->modifierUnChapitre($sequence_id,$cc,$ttype,$activite_titre,$fic='aa', $window=2,$visible,1,$mastery=0,$maxtime=",$prerequis=",$idscore=",$description=",$regroupe=",$motsClefs,$launch,$positionnement

```

```

=0,$stab_code_ifi="$time_limit_action",$fic
_correct=", $correct_auto=0,$prerequis_mode
=1);}

/* _____ QUESTION _____ */

elseif ($nodesSerie->tagName=='question'){
  $question_id=$nodesSerie->getAttributeNode
  $ ('question_id')->nodeValue;
  $stab = array() ; $stab['question'] = "";
  $stab['bloquante'] = 0 ; $stab['text_ok'] = "";
  $stab['text_ko'] = ""; $stab['ponderation'] = 1 ;
  $stab['r1'] = " ; $stab['r2'] = " ; $stab['r3'] = " ;
  $stab['r4'] = " ; $stab['r5'] = " ; $stab['r6'] = " ;
  $stab['reponse'] = " ;

  foreach($nodesSerie->childNodes as $nodes
  Question){
    if ($nodesQuestion->tagName=='question
    _titre'){
      $question_titre=$nodesQuestion->node
      Value;
      $stab['question'] = $question_titre;}
    elseif($nodesQuestion->tagName=='text_
    ok'){
      $text_ok=$nodesQuestion->nodeValue;
      $stab['text_ok']=$text_ok;}
    elseif($nodesQuestion->tagName=='text_
    ko'){
      $text_ko=$nodesQuestion->nodeValue;
      $stab['text_ko']=$text_ko;}
    elseif($nodesQuestion->tagName==
    'ponderation'){
      $ponderation=$nodesQuestion->node
      Value;
      $stab['ponderation'] = $ponderation ;}
    elseif($nodesQuestion->tagName==
    'reponses'){
      $bonne_reponse=$nodesQuestion->getAt
      tributeNode('bonne_reponse')->node
      Value;
      $i=1;
      foreach($nodesQuestion->childNodes as
      $nodesReponses){
        if ($nodesReponses->tagName==
        'reponse'){
          $reponse=$nodesReponses->node
          Value;
          $stab['r'.$i] = $reponse ;
          if ($bonne_reponse==1) $stab
          ['reponse'].=$i." ";
          $i++;}}}}
    if ($question_id==0){
      $question_id=$ClsAdmin->ajouterUneQues
      tion($serie_id,$stab);}

```



```
else { // mise à jour de la question
    $ClsAdmin->modifierUneQuestion($serie_id,
    $question_id,$tab); } } } } } } } } } } } } } } } }
```

```
function doPost($action) {
    global $ClsACL, $PAGE_ADMIN;
    switch(ACTION) {
        case 'liste' : break;
        case 'modif' : if (!$ClsACL->autorise($PAGE_ADMIN[SUBMENU],ACL_MOD)) {
            PageErreur('action=acl'); }
        break;}}}
```

```
function doGet($action) {
    global $ClsACL, $PAGE_ADMIN;}
```

```
LoadPageAdmin(new PageImportScccc);
```