Analyses des apports du méta-standard, ODP-RM à la communauté EIAH, Instances sur un système de formation

Alain Corbière

Thèse de doctorat de l'Université du Maine

spécialité

Informatique

présentée par

Alain Corbière

pour l'obtention du titre de

Docteur de l'Université du Maine

Analyses des apports du méta-standard ODP-RM à la communauté EIAH

Instances sur un système de formation

Composition du jury :

Rapporteurs :		
	M. Alain Derycke	Professeur à l'Université de Lille I
	M. Mourad Chabane Oussalah	Professeur à l'Université de Nantes
Examinateur :		
	M. Thierry Nodenot	Maître de Conférences habilité à diriger des recherches à l'Université de Pau et des Pays de l'Adour
Co-encadrant :		
	M. Christophe Choquet	Maître de Conférences à l'Université du Maine
Directeur :		
	M. Pierre Tchounikine	Professeur à l'Université du Maine

Thèse préparée au sein du Laboratoire d'Informatique de l'Université du Maine (LIUM)

Remerciements

Tout d'abord, mes remerciements s'adressent à M. Alain Derycke, Professeur de l'Université de Lille I, pour m'avoir fait l'honneur de rapporter et de juger mon travail.

Je remercie M. Mourad Chabane Oussalah, Professeur de l'Université de Nantes, pour m'avoir également fait l'honneur de rapporter et de juger mon travail, ainsi que Mme Dalila Tamzalit, Maître de Conférences à l'Université de Nantes pour ses précieux conseils.

Je remercie également M. Thierry Nodenot, Maître de Conférences habilité à diriger des recherches à l'Université de Pau et des Pays de l'Adour, pour sa présence et pour avoir porté une grande attention à mon travail.

Je tiens à remercier Pierre Tchounikine, Professeur de l'Université du Maine, pour m'avoir accueilli dans le Laboratoire d'Informatique de l'Université du Maine (LIUM), pour m'avoir prodigué des conseils toujours très constructifs et très enrichissants et pour avoir accepté de présider le jury de ma soutenance de thèse.

Je tiens à remercier Christophe Choquet, Maître de Conférences à l'Université du Maine, pour m'avoir encadré et dirigé au cours de cette thèse, pour sa confiance et son soutien constant tout au long de ce travail.

Je tiens à remercier Maurice Henry, Président de l'Université du Maine, et Xavier Dubourg, Directeur de l'Institut Universitaire de Technologie (IUT) de Laval, pour m'avoir accordé leur confiance en allégeant ma charge d'enseignement et m'avoir permis ainsi d'effectuer ce travail dans les meilleures conditions.

J'adresse également mes remerciements à toute l'équipe de recherche du LIUM du site de l'IUT de Laval. Ces membres, Christophe Choquet, Vincent Barré, Philippe Cottier, Xavier Dubourg, Sebastien Iksal, Colin Schmidt, Pierre Laforcade et Claudine Piau-Toffolon, Maîtres de Conférences, Pariticia Gounon, Docteur et Hassina El-Kechaï et Noa Randriamalaka, étudiants en thèse, sont à l'origine des nombreuses initiatives créant la dynamique de recherche au sein du département Services et Réseaux de Communication (SRC) de l'IUT de Laval. Parmi ces initiatives, mon travail s'est plus particulièrement intégré dans le projet sur la Réingénierie des EIAH Dirigée par les Modèles (REDiM), dirigé par Christophe Choquet. Dans un tel contexte, j'ai pu participer aux nombreux échanges au sein de cette équipe et bénéficier ainsi d'un environnement favorable pour mener mes travaux scientifiques. Je les remercie pour leur écoute attentive et leur esprit d'entraide si vital dans la finalisation d'un travail de recherche scientifique.

Je tiens à remercier tout particulièrement Sarah Bertrand pour son expertise sur la rédaction en langue anglaise.

Je remercie l'équipe pédagogique du département SRC (Services et Réseaux de Communication) et les équipes administrative et technique de l'IUT de Laval, qui ont su créer les conditions optimales pour la réalisation d'un tel projet.

Enfin je remercie mes proches pour leur compréhension et leur soutien tout au long de ces années et tout particulièrement, ma compagne qui n'a jamais douté de l'aboutissement de mon projet et mon père pour la relecture de ce document.

Table des matières

Répertoire des figures

Répertoire des tableaux

INTRODUCTION

Le travail présenté dans ce document porte sur les problématiques d'ingénierie et de réingénierie dans les EIAH (Environnements Informatiques pour l'Apprentissage Humain). Notre contribution consiste à soumettre à la communauté EIAH un cadre unificateur, le cadre ODP-RM (Object Distributed Processsing - Reference Model). Ce cadre, proposé par l'ISO (International Organization for Standardization), se définit comme un méta-standard ayant pour objectif d'établir des normes qui permettent de tirer les avantages de la distribution de l'information dans un environnement possédant des unités hétérogènes de traitement de l'information et qui relèvent de domaines différents (ISO/IEC-10746-1 1998).

Dans ce document, nous utilisons le terme « système de formation » pour désigner un EIAH vu comme un ensemble structuré, organisé et composé de plusieurs éléments complexes.

Nous présentons trois constats qui constituent le point de départ de notre travail.

Tout d'abord, nous pouvons observer que les concepteurs d'une formation cherchent des modèles pour appréhender et communiquer ses pratiques pédagogiques. L'une des attentes des concepteurs d'un système de formation est de disposer d'un cadre unificateur définissant un ensemble de concepts pour spécifier les usages des technologies de l'information et de la communication. Selon [(Charlier, Daele et al. 2002), page 347], cette problématique est toujours d'actualité et au cœur de l'innovation pédagogique.

Le second constat se définit par l'arrivée de modèles de technologies éducatives proposés par les différents consortiums de normalisation. Il nous amène à nous interroger sur la capacité de ces modèles de spécification à supporter les différents échanges qui animent les communautés de concepteurs. Le concepteur, souhaitant se positionner au centre des innovations pédagogiques et s'adapter à de nouveaux outils, usages ou pratiques, s'interroge sur la pertinence de ces récentes propositions. Actuellement, les technologies éducatives normées apportent deux réponses : la première centrée sur la gestion des contenus et la seconde orientée sur le processus d'apprentissage. Dans le premier cas, ces technologies doivent intégrer les contraintes d'accessibilité, de modularité et d'interopérabilité (Friesen 2001). Dans le second cas, elles visent à optimiser les échanges sur les styles ou sur les approches d'apprentissage entre les concepteurs pédagogiques. Elles se situent au cœur des projets d'une démarche qualité dans un système de formation (Pawlowski 2002b).

Dès les premières propositions de technologies éducatives normées, de nombreuses ambiguïtés ont été identifiées. En particulier, les modèles de spécification produits sont présentés comme étant capables à la fois de définir

des modèles neutres d'un point de vue pédagogique et de créer des instances explicitant une sémantique pédagogique en formalisant les savoirs enseignés, les stratégies didactiques ou pédagogiques et le comportement de l'apprenant. De plus, les modèles sont considérés à la fois comme des modèles conceptuels permettant au concepteur d'exprimer ses intentions, et comme un cadre technique le contraignant pour expliciter le comportement des différentes composantes du système. Ces différents phénomènes amènent à poser des problèmes de cohérence et de considérer les nombreuses mises en pratique des technologies éducatives normées et plus particulièrement des communautés de logiciel libre produisant des EIAH donnant accès aux ressources de développement.

Ces trois constats nous amènent à chercher et à proposer un cadre unificateur définissant un ensemble de concepts utilisés dans le discours des concepteurs, à inscrire les propositions sur les technologies éducatives dans ce cadre et à expliciter les observations sur les pratiques des communautés de logiciel libre diffusant des EIAH.

Nous proposons alors un ensemble de termes définis par ce cadre aux concepteurs souhaitant collaborer sur les différents points de vue du comportement des différentes composantes d'un système de formation. Et nous vérifions la capacité d'un tel cadre à intégrer les modèles de spécifications issus des propositions sur les technologies éducatives normées, les structures logicielles des composantes des applications produites par les communautés du logiciel libre et à expliciter les mises à l'essai effectuées sur le site de l'IUT (Institut Universitaire de Technologie) de Laval (France). Ce travail nous permet de décrire dans la terminologie du cadre choisi des relations et des transformations formalisant les nouveaux éléments méthodologiques pour la conception d'un système de formation. En articulant ces différentes observations, le travail présenté dans ce mémoire montre les apports du méta-standard ODP-RM à la communauté EIAH.

Par conséquent, notre contribution vise à illustrer l'utilisation du cadre ODP-RM en présentant trois instances sur un système de formation. Sa présentation dans ce document se structure en cinq chapitres de la manière suivante : le premier présente le contexte général, la problématique de recherche et l'approche méthodologique choisie. Le deuxième analyse la synergie entre une communauté du logiciel libre et une approche dirigée par les modèles. Il propose un point de vue spécifique défini par le cadre sur un système de formation considéré. Le troisième étudie la synergie entre une communauté du logiciel libre et un processus de développement, son objectif est d'expliciter plus précisément le comportement des composants du modèle correspondant au point de vue considéré. La quatrième analyse la synergie entre une approche dirigée par les modèles et un processus de développement logiciel, elle nous amène à expliciter les échanges entre les composants du modèle en prenant en compte la sémantique du point de vue considéré. Le dernier chapitre du manuscrit conclut et présente les perspectives d'une telle contribution.

Le cadre ODP-RM est l'outil choisi pour guider les analyses que nous présentons dans ce document. Dans la littérature scientifique, il est présenté comme un outil d'analyse de systèmes complexes (Wegmann et Naumenko 2001).

Cela nous amène à considérer le domaine de la conception d'un système de formation que nous cherchons à formaliser (Durand 1979) comme un système organisé et constitué de composants caractérisés par leurs comportements et les échanges qui les mettent en relation.

Nous partons d'un travail d'observation de l'évolution de plusieurs outils qui correspond à une vision élargie du domaine de la conception d'un EIAH. Ce qui nous amène à analyser le travail effectué par les concepteurs appartenant aux communautés ayant produit ces outils et les liens identifiés dans leurs utilisations.

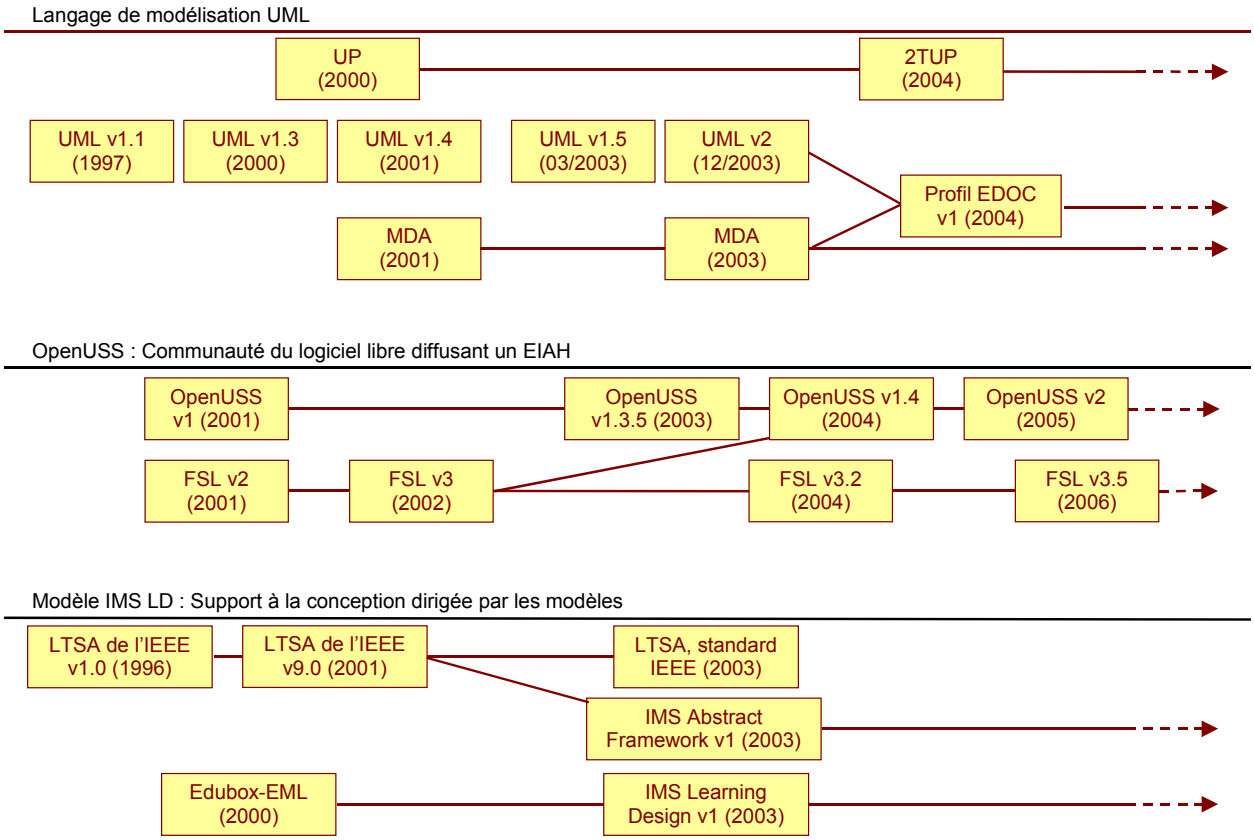


Figure 1 : Evolutions observées des outils tout au long de notre travail de thèse (pour le détail des abréviations, cf. page 163)

Le cadre se définit comme un méta-standard sur les systèmes distribués. Il est indépendant de tout type de domaine et supporte un processus de modélisation d'un domaine d'analyse en instanciant un ensemble de concepts pré-définis. Pour cela, ce cadre dispose de deux formes d'abstraction : la première porte sur une architecture à cinq points de vue mettant à disposition un langage pour spécifier un aspect particulier du système et la deuxième permet d'expliciter les cohérences entre ces différents points de vue.

L'ensemble des points de vue est la première forme d'abstraction qui permet de se concentrer sur des parties indépendantes de la spécification et sur les préoccupations particulières de chacun de ces aspects. La spécification d'un point de vue est obtenue en utilisant un ensemble spécifique de concepts d'architecture et de règles de structuration.

- le point de vue métier focalise sur les objectifs, le domaine d'application et les stratégies de ce système,
- le point de vue informationnel permet la définition des informations traitées par les différentes ressources systèmes,
- le point de vue computationnel décrit la décomposition fonctionnelle d'un système et les interactions entre les interfaces des différents objets,
- le point de vue ingénierie décrit les moyens mis en œuvre pour que les objets du système interagissent et
- le point de vue technologique définit les technologies logicielles et matérielles utilisées.

La deuxième forme d'abstraction se définit au niveau de l'ensemble de termes permettant de maintenir la cohérence entre les modèles de spécification des différents points de vue. Le cadre permet de représenter et de décrire les différentes transformations correspondant aux différents actes de spécification, de structuration et de modélisation.

Un tel instrument va nous permettre de donner une description des différents phénomènes analysés suivant la terminologie du cadre ODP-RM et de la structurer suivant son architecture.

Le premier étudie la synergie entre une communauté du logiciel libre et une approche dirigée par les modèles. Elle consiste à observer deux projets sur les technologies éducatives normées gérés respectivement par le consortium IMS (Instructional Management Systems) et le comité IEEE (Institute of Electrical and Electronics Engineers). Cette analyse aborde quatre aspects : les limites des deux consortiums, la manière de transformer les modèles, les différentes perspectives d'évolution et leurs manières de négocier. L'objectif est d'effectuer une analyse critique des principaux projets initiaux sur les technologies éducatives normées. Pour chacun des aspects traités, nous mettons en œuvre le cadre ODP-RM en montrant ses capacités à les intégrer et à les décrire. Ces différents phénomènes observés nous amènent à proposer une structure organisationnelle à l'aide d'un point de vue, le point de vue métier, du domaine analysé.

Le deuxième analyse étudie la synergie entre une communauté du logiciel libre et un processus de développement logiciel. Le travail d'analyse effectué nous amène à instancier deux domaines complexes, la réingénierie du logiciel et les processus mettant en œuvre les technologies éducatives normées. Pour comprendre cette complexité, la démarche systémique nous demande d'observer le comportement de chacun de ces domaines et de chercher à les fédérer. Nous créons une instance du cadre ODP-RM de ces deux domaines afin d'observer les manques de chacun de ces travaux et de détailler ensuite certains comportements des processus identifiés dans le domaine analysé.

La dernière analyse se focalise sur la synergie entre une approche dirigée par les modèles et un processus de développement logiciel. L'objectif est de définir une ingénierie dans les EIAH selon une approche dirigée par les modèles. Pour cela, nous identifions trois communautés. Dans un premier temps, nous nous focalisons sur les choix techniques de ces communautés. Dans un deuxième temps, nous précisons les interactions entre les éléments représentés dans le modèle du point de vue métier afin d'expliciter l'activité du concepteur membre d'une communauté diffusant un EIAH. Nous utilisons les concepts de base décrivant l'acte de spécification où les concepteurs classifient les objets du système par des liens de composition et l'acte de modélisation permettant de définir, à différents niveaux d'abstraction, les modèles d'interaction entre les objets. De plus, nous utilisons le système de notation défini par le méta-modèle EDOC et diffusé par le comité OMG (Object Management Group) pour spécifier une séquence d'activités et une collaboration dans le point de vue métier proposé.

En conséquence, notre contribution vise à proposer trois instances du cadre ODP-RM sur un système de formation correspondant à nos trois analyses. Notre apport est d'identifier un cadre unique définissant un ensemble de concepts permettant de décrire et de partager les éléments méthodologiques manipulant les artefacts informatiques conçus, produits et réutilisés dans un domaine correspondant à notre vision de la conception des EIAH.

En proposant une structure organisationnelle du domaine considéré, en illustrant le travail de réingénierie d'un EIAH, et en représentant l'activité de spécification et de modélisation autour des objets diffusés par les communautés de logiciel libre, ce document présente les mises en application des principes sous-jacents du cadre ODP-RM et les éléments méthodologies définis dans la représentation du domaine considéré.

Cela nous a permis de présenter une analyse critique des projets à l'origine des propositions de standards sur les technologies éducatives et un nouvel angle de vue sur le travail de réingénierie et d'ingénierie d'un système de formation.

Trois perspectives sont envisagées sur la proposition d'un point de vue métier sur le domaine de réingénierie et d'ingénierie d'un système de formation. La première consiste à s'interroger sur le positionnement et les apports de cette proposition par rapport aux initiatives engagées dans les organismes de standardisation comme l'OMG, le W3C (World Wide Web Consortium) et l'IMS. La seconde se définit par la poursuite de l'analyse du méta-standard afin de déterminer ses limites à formaliser et à communiquer les observations sur différents aspects d'un système de formation. Et la troisième consiste à mener une analyse critique sur les langages de description d'architecture afin d'expliciter l'évolution des communautés du logiciel libre diffusant un EIAH.

CHAPITRE 1 : CONTEXTE GENERAL, PROBLEMATIQUE DE RECHERCHE ET APPROCHE METHODOLOGIQUE

1.1 Introduction

Les besoins des sciences de l'éducation, les interrogations sur l'utilisation des récentes propositions de normes sur les technologies éducatives et les principes de développement d'une communauté de pratique définissent les trois fondements théoriques du cadre général de cette thèse. A notre sens, ils définissent trois des points de vue du cadre théorique de la recherche dans la conception des EIAH présentés par (Derycke 2002; Tchounikine, Baker et al. 2004). Le travail consiste alors à effectuer un travail d'analyse sur la conception des EIAH où nous sommes amenés à considérer les points de vue de manière distincte dans un premier temps et complémentaire dans un second (Tchounikine, Baker et al. 2004). La Figure 2 montre que ces deux principes ont été respectés dans le travail d'analyse exposé dans ce mémoire. Les trois fondements théoriques sont présentés dans les parties suivantes de ce chapitre :

- le premier présente une des problématiques des sciences de l'éducation cherchant à produire des modèles à destination des concepteurs pour qu'ils puissent s'interroger sur leurs pratiques.
- le second présente le travail accompli par le chantier de définition de normes dans les technologies éducatives. Les concepteurs souhaitant utiliser ces outils sont à la recherche de nouveaux instruments capable de les aider à spécifier leurs pratiques dans ces modèles.
- le troisième concerne la définition d'une communauté de pratique dans une organisation selon Etienne Wenger (Wenger 1998). Les principes théoriques énoncés par l'auteur nous donnent les moyens d'identifier de telle communauté et d'inscrire cette culture dans notre contexte général.

Notre travail est alors de s'interroger sur les potentiels des relations entre ces trois fondements afin de définir la conception d'un système de formation au cœur duquel se positionnent les concepteurs.

Afin de mener à bien cette analyse, nous identifions trois outils (cf. Figure 2) comme les réponses à adresser aux concepteurs : le modèle LD (Learning Design) du consortium IMS (Instructional Management Systems) pour supporter une conception dirigée par les modèles, le langage de modélisation orienté objet UML (Unified Modelling Language) utilisé dans les organismes de standardisation pour représenter les modèles de spécification et la communauté OpenUSS (Open University Support System) prise comme l'un des exemples de communauté de logiciel libre diffusant

un EIAH. Les liens qui les mettent en relation présentent trois synergies. L'analyse de ces dernières est au cœur de la problématique que nous exposons à la fin de ce chapitre.

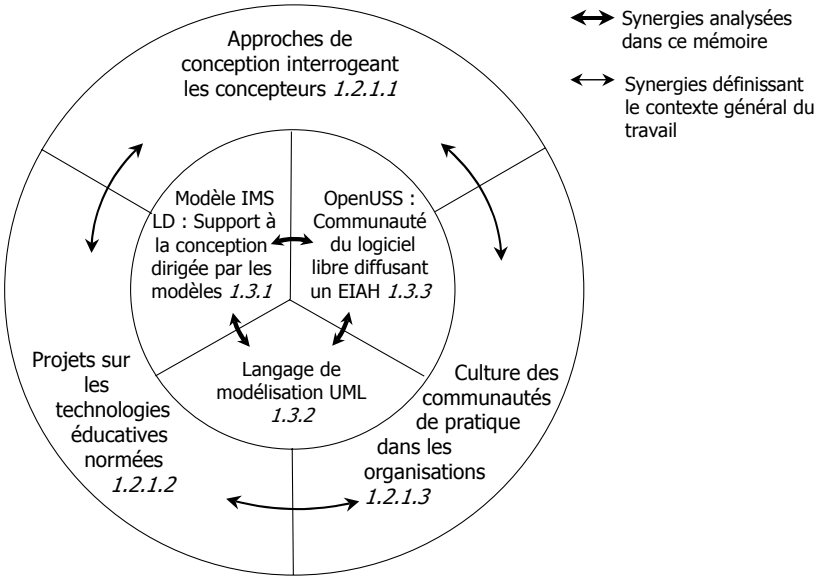


Figure 2 : Représentation des différentes synergies exposées dans ce mémoire

1.2 Contexte général

Après avoir présenté les trois fondements théoriques de notre problématique, nous effectuons une première analyse afin de montrer les liens qui existent entre eux. La finalité est d'exposer les potentiels du contexte choisi.

1.2.1 Fondements théoriques généraux

1.2.1.1 Approches de conception interrogeant les concepteurs

Dans une institution de formation, les concepteurs sont amenés à s'interroger sur leurs rôles, leurs méthodes et les stratégies mises en œuvre. Les réflexions menées par les sciences de l'éducation sur les mises en situation pédagogique et les mises en pratique des technologies de l'information et de la communication mettent à disposition des cadres théoriques afin de guider le concepteur. La qualité d'un dispositif se définit comme un ensemble d'instruments (Bruillard 1999) apte à observer et à comprendre l'accompagnement de l'apprenant dans sa session de formation.

Nous faisons référence à deux modèles issus des travaux sur les sciences de l'éducation : le triangle pédagogique de Houssaye (Houssaye 2000) et le modèle « Helices » (Linard 2001). Ces deux exemples ont montré pour l'un sa pertinence en faisant l'objet de nombreuses adaptations et pour l'autre d'être au cœur de l'actualité avec le chantier sur les technologies de l'information appliquées au domaine de la formation.

Le triangle pédagogique, un modèle référent pour la communauté éducative, cherche à identifier un modèle théorique pour retranscrire les pratiques pédagogiques. Trois processus, définis comme une structure stable, organisée et dynamique, se positionnent dans une situation pédagogique articulée autour de trois pôles (Savoir - Professeur - Elève).

- Le processus « enseigner » : Situation de l'enseignant dans laquelle l'élève est exclu. La structuration du contenu est pensée en minimisant l'impact des deux autres processus. Elle caractérise également un type culturel des pratiques d'enseignement.
- Le processus « apprendre » : Situation type de l'auto-apprentissage ou par objectif. L'enseignant est absent. Par contre, il est l'auteur de la planification pédagogique en amont de la situation.
- Le processus « former » : Situation type de travail de groupe, où l'enseignant joue alors le rôle de modérateur pour régler les conflits et redéfinir les rôles des élèves.

Ce cadre a d'ailleurs par la suite été adapté pour retranscrire le concept de distance et d'apprentissage collaboratif, comme par exemple la proposition de (Faerber 2003).

Le second modèle est le résultat des travaux de Linard qui, partant du constat des nombreux échecs dans les projets de formation en ligne de grande envergure, a identifié comme l'une des principales causes du dysfonctionnement, le manque de compétences des médiateurs [(Linard 1996), page 160]. Pour y répondre, elle propose un cadre qui superpose le modèle hiérarchique en trois niveaux du psychologue Leontiev et la définition cyclique d'une action humaine de Bruner. Ce modèle, intitulé « Helices », structure en 7 étapes le déroulement effectif d'un utilisateur effectuant un apprentissage sur un dispositif informatique.

1. Orientation motivée de l'intention (perception sélective initiale par le sujet d'un état précis de besoin ou manque d'un objet satisfaisant) ;
2. Formation d'une attention sélective et d'un but final par anticipation de l'avenir sous forme d'images de l'état désirable à atteindre ;
3. Elaboration de stratégies et plans d'action plus ou moins rationnels par rapport aux buts et sous-buts ;
4. Mobilisation des conditions opératoires et routines nécessaires à la réalisation des actions anticipées ;
5. Persistance du pilotage et de l'auto-correction jusqu'au jugement de fin de cycle par comparaison entre effets attendus et effets obtenus ;
6. Evaluation des résultats en termes de valeurs affectives (satisfaction de la réussite, déplaisir de l'échec) et cognitives (qualité, efficacité, coûts) ;
7. Mémorisation du parcours entier modifiant l'état et l'anticipation du sujet pour le cycle d'action suivant.

Un tel modèle est destiné au concepteur souhaitant se sensibiliser aux difficultés rencontrées par les utilisateurs d'une session distante de formation et lui permet de s'interroger sur ses intentions. D'ailleurs, dans [(Barbot et Camatarri

1999), page189], ce modèle psycho-pédagogique est repris pour être appliqué aux concepteurs. Pour [(Bélisle et Linard 1996), page 35], il est nécessaire « d'identifier de nouvelles compétences et de redéfinir les anciennes à la lumière des innovations produites par les technologies de l'information et de la communication, et d'élaborer un cadre théorique et organisationnel capable de motiver les formateurs suivant des parcours qui ne soient pas simplement informatifs ou comportementalistes, mais qui leur assurent une redéfinition existentielle de leur rôle ».

En présentant les deux exemples de modèles issus des travaux des sciences de l'éducation, nous mettons en avant que l'une des problématiques de cette communauté scientifique est de proposer aux concepteurs des cadres leur permettant de s'interroger sur leurs pratiques éducatives et d'amener à utiliser dans leurs discours à un ensemble de concepts pour supporter leurs échanges.

1.2.1.2 Projets sur les technologies éducatives normées

Dès le début des années 97[1], nous identifions une première tentative, américaine, d'initier un chantier sur la définition de normes sur les technologies éducatives. Aujourd'hui, elle produit et met à jour un ensemble de spécifications. Ces travaux sont le résultat de consensus obtenus entre les communautés scientifiques et industrielles. Actuellement, ils font l'objet de discussions tant sur le plan des pratiques de la formation à distance que sur leur potentiel à expliciter la créativité et l'innovation pédagogique.

Afin de mieux comprendre cette initiative, nous choisissons de retracer rapidement quelques éléments de l'historique des EIAH pour ensuite nous intéresser aux différents projets sur les technologies éducatives qui sont à l'origine des propositions de normes actuelles. Nous montrons leurs similitudes.

A la fin des années soixante, l'enseignement programmé se positionne à l'origine des EIAH. Il met en œuvre les théories béhavioristes de Skinner. En réaction à cette approche, le langage LOGO (Papert 1987) a été inventé par Seymour Papert et Marvin Minsky en 1966. Il permet à l'apprenant de programmer dans un langage clair et modulaire. Il s'inspire à la fois des procédures systémiques (où chaque cycle se positionne dans une boucle de rétroaction par effet visuel des actions produites par l'utilisateur) et constructivistes (où les mécanismes et processus permettant la construction de la réalité chez les sujets sont étudiés).

Les premières initiatives visant à créer des systèmes autonomes d'apprentissage datent du début des années soixante. Les Américains cherchent à typer la connaissance des différents items pédagogiques. Le résultat le plus

[1] Voir, http://edutool.com/ltsa/03/ltsa.html

notable est celui produit par Gagné. Il définit une typologie de capacités et de résultats de l'apprentissage d'un apprenant au cours d'une session de formation (Gagné 1992). Cette classification s'articule autour de cinq éléments :

- information verbale ;
- qualifications intellectuelles ;
- stratégies cognitives ;
- habiletés motrices ;
- attitudes.

Le système de formation doit être capable de s'adapter aux habiletés de l'apprenant. Il associe une typologie de stratégies déclinées en neuf catégories d'événements d'apprentissage. Pour chaque catégorie, le système identifie la condition nécessaire pour que l'apprentissage soit effectif ou non. Sur la base de ces travaux, (Merrill 1997) développe le « Component Display Theory (CDT) » qui définit une classification en deux dimensions d'une entité support à l'apprentissage: l'habileté et le type de contenu. Mais la structuration statique et limitée ne permet pas au système de prendre en charge de manière autonome la session d'apprentissage. Ces initiatives définissent l'une des principales dimensions de la conception des EIAH, celle cherchant à définir les mécanismes de résolution de problèmes et de gestion automatique des interactions humaines.

Ce bref rappel sur les EIAH est source d'enseignement quand nous reprenons la courte histoire des différents projets actuels sur les technologies éducatives normées. En 1988, l'industrie avionique américaine décide de spécifier les différents formalismes pour décrire les supports de formation. Neuf ans plus tard, un premier projet de standard se définit autour d'un consensus sur une architecture à quatre processus définissant le comportement d'un dispositif d'apprentissage. Ce standard, proposé par le comité IEEE (Institute of Electrical and Electronics Engineers), avait plusieurs objectifs. Le premier est de communiquer à l'industrie des plates-formes de formation les besoins de standards sur les technologies éducatives présentées par une telle architecture. Le second est de structurer et d'organiser une démarche visant à former les sous-groupes de travail du comité IEEE sur les différents aspects liés au fonctionnement interne de chaque processus et à définir le formalisme des données échangées. Comme le souligne (Blandin 2004), ce modèle de processus d'apprentissage ne définit que les interactions et les traitements en terme d'information et ne s'intéresse pas à leur sémantique pédagogique sous jacente. Il s'inscrit dans une mise en application directe de l'enseignement programmé.

Le consortium IMS (Instructional Management Systems) s'est formé à la même époque que le comité travaillant sur les technologies éducatives standards de l'IEEE. L'objectif est de définir un ensemble de spécifications s'adressant aux concepteurs et optimisant l'interopérabilité entre les différentes composantes d'un dispositif d'apprentissage. Un ensemble cohérent de modèles informationnels produits par le consortium est mis à la disposition des concepteurs pour

spécifier, selon des règles syntaxiques rigoureuses, le déroulement de sessions d'apprentissage. Nous décrivons ces modèles par la suite.

La première composante permet de décrire, de déterminer et d'échanger des contenus :

- « Meta-data » (Méta-données pédagogiques) : Structures descriptives définies facilitant la recherche et la diffusion de contenus.
- « Content Packaging » (Structuration de contenu) : Formalisme facilitant les échanges de contenu entre les différents systèmes d'apprentissage en définissant une structure type de l'information et des méta-descripteurs
- « Question & Test » (Questions et tests) : Structure décrivant des questions et des réponses ainsi que les mécanismes de mise à jour des résultats.
- « Digital Repositories » (Interopérabilité entre les répertoires numériques) : Recommandations rendant intéropérables la plupart des fonctionnalités de gestion de répertoires telles que la recherche et le stockage de contenu.

La deuxième composante définit les spécifications et les structures des contenus :

- « Simple Sequencing » (Simple séquencement) : Méthode définissant les séquences d'activités d'apprentissage en utilisant des règles. Ces dernières utilisent des instructions de branchement ou de flux en association avec les événements générés lors de l'utilisation des ressources par les apprenants.
- « Reusable Definition of Competency » (Compétences) : Modèle représentant les compétences associées à un contenu et à des apprenants.
- « Learning Design » (Unité d'apprentissage) : Modèles et langages décrivant les séquences pédagogiques et permettant un échange entre les systèmes de gestion d'apprentissage (ou LMS pour Learning Management Systems).
- « Accessibility » (Accessibilité) : Spécification définie pour développer les interfaces utilisateurs et le contenu accessible à l'apprenant en utilisant les systèmes d'accès alternatifs.

La troisième composante définit les formalismes supports à l'interopérabilité entre les systèmes :

- « Learner Information » (Structuration des informations sur l'apprenant) : Définition d'une structure pour organiser les informations concernant l'apprenant et pour partager entre les environnements d'apprentissage et les systèmes de formation.
- « Enterprise » (Métier) : Définition d'une structure pour optimiser le transfert des informations de l'organisation concernant les étudiants et les groupes d'apprenants entre les systèmes.

Pour la première fois, nous disposons d'un ensemble de concepts définis dans des modèles cohérents qui se positionnent eux-mêmes dans une organisation rationnelle. Contraint par ces structures syntaxiques, l'industrie de la formation à distance va produire et partager des contenus et des composantes de plates-formes de formation à distance. Les différents formalismes proposés par le consortium définissent les différents aspects d'un système de formation.

La recherche de matériaux pédagogiques dans des banques de ressources distribuées va ainsi augmenter en précision. La pertinence des objets trouvés par rapport au besoin exprimé dans une structure informationnelle est améliorée. En contre partie, un effort de description des matériaux doit être effectué lors de leur création avant leur stockage. La requête formulée est alors plus complexe qu'un simple mot-clé. Plusieurs interrogations sont posées aux concepteurs. La modularité des structures offre une grande souplesse en permettant différents assemblages. Les sémantiques sont alors définies dans la pratique de ces différents modèles. Les concepts orientés objet de composition et de décomposition ainsi que les notions d'encapsulation et d'abstraction se doivent d'être maîtrisés par les concepteurs et d'être utilisés pour expliciter leurs pratiques.

1.2.1.3 Culture des communautés de pratique dans les organisations

Le troisième fondement théorique qui nous permet de compléter la description du contexte général de notre problématique est centré sur l'évolution du discours scientifique sur la gestion de la connaissance. Nous nous intéressons plus spécifiquement au travail d'Etienne Wenger. Au début des années 80, il a défini un modèle dynamique de communication des connaissances pour un tuteur intelligent (catégorie de logiciels dotés de capacité à répondre de manière autonome aux solicitations de l'apprenant). Ensuite, il s'est consacré au phénomène de gestion des connaissances dans une organisation pour devenir l'un des référents sur le concept de communauté de pratique.

Vers la fin des années 80, (Wenger 1987) avait clairement identifié les difficultés de conception d'un système de formation. En proposant un modèle d'architecture représenté par la Figure 3, les différentes composantes et les différents flux permettaient d'établir un premier cadre explicitant les enjeux dans la conception d'un EIAH spécifique, un tuteur intelligent.

Ce premier travail lui a permis de dégager les deux points fondamentaux suivants (1) l'articulation entre les niveaux de connaissance et de diffusion des contenus enseignés et (2) les différences entre une approche mettant en œuvre les techniques de l'intelligence artificielle visant à coder l'expertise d'un concepteur et les mécanismes du développement logiciel contraint de mettre en œuvre des points de vue différents. Ces concepts sont introduits par l'auteur comme un contexte dans lequel le concepteur se place pour diagnostiquer un problème de conception, en considérant que le déplacement de ce point de vue change la vision du problème d'apprentissage [(Wenger 1987), page 355].

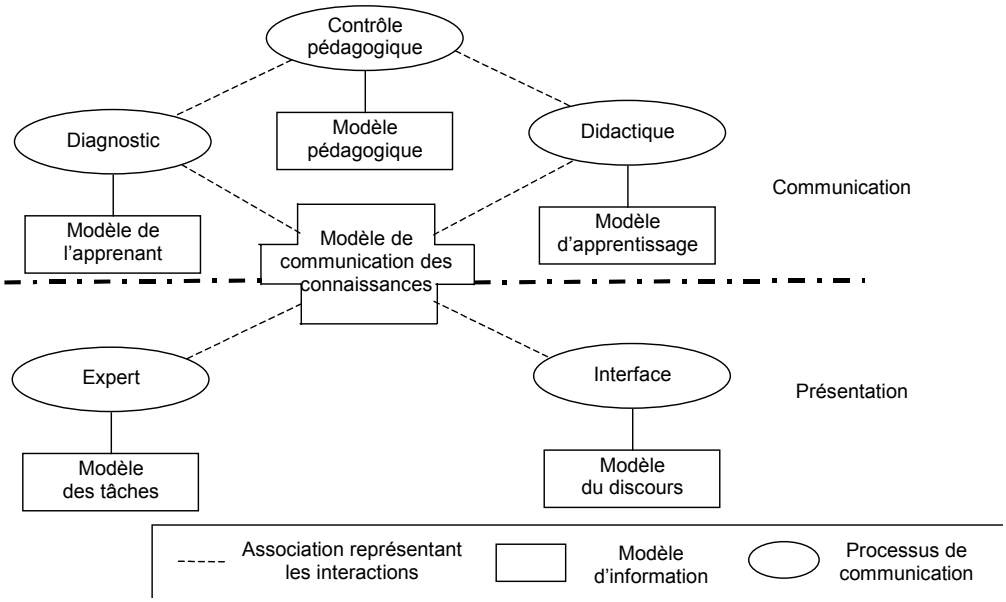


Figure 3 : Composantes d'une architecture pour la communication des connaissances, traduit de [(Wenger 1987), figure 19.3]

Précurseur sur ces premiers constats, les recherches poursuivies par l'auteur se sont focalisées sur l'étude du phénomène dit de « communauté de pratique ». L'enjeu se positionne alors au niveau de la cohérence d'un groupe d'individus qui est défini selon lui par trois dimensions : l'engagement mutuel, une entreprise commune et un répertoire partagé [(Wenger 1998), pages 72-73]. La première est le siège de négociation du sens entre les individus initiés par les actions du groupe. La seconde est le résultat d'un tel processus traduit sous la forme de règles ou d'objectifs. La troisième correspond aux ressources créées par la communauté ; elles incluent une terminologie, des outils, des procédures, des histoires ou des concepts.

Plus récemment dans [(Wenger, McDermott et al. 2002), pages 49-51], l'auteur expose les sept principes du développement d'une communauté de pratique. Nous les présentons ici :

1. « Design for evolution » : Ce principe met en avant que chaque élément de conception se doit d'être un catalyseur de l'évolution de la communauté. Le but de la conception est alors de ne pas imposer une structure d'organisation à la communauté mais de l'aider à se développer.
2. « Open a dialogue between inside and outside perspectives » : Ce principe, considérant qu'une conception s'appuie sur l'expérience collective des membres de la communauté, amène à s'interroger sur la structure de cette communauté et les moyens utilisés pour faire adhérer les experts extérieurs aux échanges de la communauté.
3. « Invite different levels of participation » : Ce principe amène à s'intérroger sur les types d'activité que nous pouvons identifier dans une communauté de pratique. Il permet de s'interroger sur les raisons les ayant conduit à un tel engagement. L'auteur nous amène à considérer trois niveaux de participation : (1) un premier groupe (de 10 à 15% du nombre des membres) se positionne au cœur de la communauté et coordonne la communauté, (2) un deuxième

groupe (de 15 à 20% du nombre des membres) a une implication moins intense dans les activités d'échanges et un troisième où les membres observent les interactions des deux groupes précédents. Le principe est qu'une communauté se doit de donner à chacun la possibilité de participer aux trois niveaux, et de s'y sentir membre à part entière.

4. « Develop both public and private community spaces » : La conception d'une communauté doit prendre en considération les événements publics, générés par tous les membres de la communauté où l'objet des échanges est souvent informel, et les événements privés, échangés entre deux membres d'une communauté partageant des intérêts communs. Le principe du développement d'une communauté est alors d'orchestrer les activités à la fois dans et entre les espaces publiques et privés. L'objectif est d'enrichir les événements et de renforcer les relations entre les membres.
5. « Focus on value » : Généralement, les membres participent à une communauté de manière volontaire. Chaque communauté a un impact qu'il est difficile de cerner. Au lieu de l'expliciter, une communauté a besoin de créer des événements, des activités et des interactions qui aident à qualifier la finalité à laquelle répond la communauté. Le principe est d'encourager les membres à expliciter la « valeur » de la communauté le plus tôt possible. Elle est abstraite et difficile à cerner au début. Par la suite, ces premières discussions aident la communauté à comprendre son impact réel.
6. « Combine familiarity and excitement » : Ce principe est d'encourager dans les communautés la nouveauté et l'excitation qui vient compléter la familiarité des activités journalières. Le principe est de considérer à la fois les activités routinières qui permettent de stabiliser les processus de conception et des évènements d'excitation qui fournissent un sens commun à l'aventure menée par la communauté.
7. « Create a rythm for the community » : Ce dernier principe est de qualifer une communauté par son rythme. Ce rythme est cadencé par les événements de la communauté, les activités autour du site Web, … Le principe est d'identifier le bon rythme à chaque étape du développement de la communauté.

L'industrie de la formation à distance a essuyé de nombreux échecs en voulant intégrer trop vite des technologies sans prendre le temps d'analyser leurs impacts et d'appréhender les contextes social et humain (Linard 1996; Linard 2001). Selon Etienne Wenger, la conception est alors vue comme un système support à la collaboration entre les membres. Les communautés de pratique nous enseignent que les concepteurs interagissent socialement mais sont également engagés dans une pratique commune et nous offrent un regard neuf sur les organisations et les méthodes de travail (Wenger 1987; Wenger 1998; Wenger, McDermott et al. 2002).

1.2.2 Relations existantes entre ces trois fondements théoriques

Cette partie effectue une première analyse des mises en relation entre les fondements théoriques venant d'être exposés :

- La première présente les liens existants entre les modèles éducatifs résultant des travaux des sciences de l'éducation et l'un des modèles de spécification diffusé par l'une des institutions de normalisation sur les technologies éducatives ;

- La seconde expose une réflexion sur l'articulation entre les modèles de spécification d'une session pédagogique et une communauté les mettant en pratique ;

- La troisième et dernière décrit les liens existants entre l'une des problématiques des sciences éducatives et une vision des phénomènes d'apprentissage au sein d'une communauté de pratique.

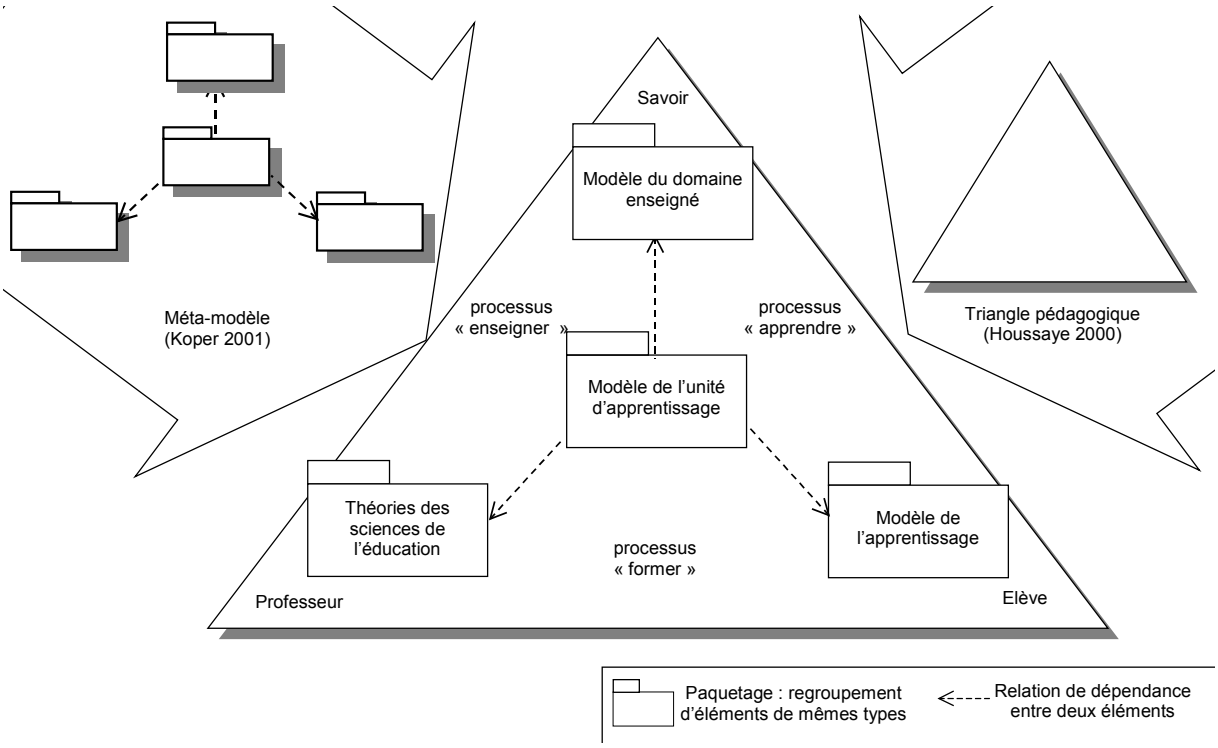


Figure 4 : Supperpostion du triangle pédagogique de (Houssaye 2000) et du méta-modèle de (Koper 2001)

Pour analyser la première relation, nous prenons le modèle de l'unité d'apprentissage[2] proposé par Koper (Koper 2001) qui est à l'origine du modèle « Learning Design » produit par le consortium IMS. L'auteur le positionne au cœur d'un processus d'apprentissage représenté par un méta-modèle (cf. Figure 4). Le modèle de l'unité d'apprentissage se décrit alors comme étant dépendant de trois autres modèles : celui représentant le domaine enseigné, le modèle d'apprentissage et l'ensemble des théories des sciences de l'éducation. Le concepteur amené à spécifier son unité d'apprentissage manipule des concepts impliquant d'autres concepts appartenant à des modèles différents. Koper nous

[2] Traduction de « Learning Design »

fait partager ainsi le raisonnement qui l'a conduit à la définition des concepts du modèle d'une unité d'apprentissage. Il présente ainsi les trois espaces de discours considérés dans un système de formation : celui des enseignants, des apprenants et des savoirs enseignés. L'activité du concepteur est de ramener son discours à une instance du modèle d'une unité d'apprentissage. Ce travail est à mettre en relation avec les travaux des sciences de l'éducation. En effet, nous associons respectivement le modèle du domaine enseigné, le modèle d'apprentissage et les théories des sciences de l'éducation aux trois sommets du triangle pédagogique (cf. Figure 4) définis dans (Houssaye 2000) : savoir, élève et professeur. Nous montrons ainsi que le modèle proposé par Koper s'adresse aux concepteurs et aux utilisateurs souhaitant expliciter les situations d'apprentissage qu'ils mettent en œuvre dans un système de formation.

L'analyse de la deuxième relation entre les travaux issus des projets sur les technologies éducatives normées et la notion de communauté de pratique nous amène à nous interroger sur les contextes d'utilisation des différents modèles de spécification. Dans une démarche de conception, le concepteur instancie les différents champs des méta-descripteurs et les différents concepts pour décrire son scénario pédagogique. En créant une telle instance, il s'interroge sur la cible présupposée de ces différentes productions. Par ailleurs, il se doit de considérer le développeur informatique chargé de mettre en œuvre ces différents modèles dans les différentes composantes d'un système de formation. Nous considérons alors que toute composante est produite par une communauté de pratique composée de membres actifs et engageant des négociations pour expliciter ses règles et ses objectifs (Wenger 1998). L'optimisation de la conception d'un système de formation passe auparavant par l'appréhension de sa structure tant sur le plan de l'ingénierie du développement que sur le plan de l'organisation sociale.

L'analyse de la dernière articulation que nous avons identifiée entre les trois fondements théoriques correspond aux liens existants entre les enseignants s'interrogeant sur leur pratique et la communauté de pratique au sein duquel est appréhendée les difficultés (Bertrand 2003) : (1) celles liées aux organisations ne disposant pas d'un cadre adéquat pour mutualiser les compétences pédagogiques, techniques et administratives, (2) celles liées aux capacités du concepteur [(Linard 1996), page 160] et (3) celles liées à la complexité technique des composantes des systèmes de formation. Les technologies éducatives normées ne suffisent pas. La mutualisation de leurs mises en pratiques individuelles et de leurs expérimentations est alors nécessaire pour stabiliser leurs utilisations. L'articulation entre la problématique des sciences éducatives cherchant à proposer des modèles éducatifs reconnus comme référents par les praticiens du terrain [(Houssaye 2000), page 24] et la perspective sociale de l'apprentissage insérée dans les pratiques collectives de conception au sein des communautés de pratique définie par (Wenger 1998), nous incite à chercher un formalisme. Ce dernier doit définir les différents cycles de conception ayant montré leurs utilités lors des mises en pratique en précisant dans un langage les différents choix et les outils mis en œuvre. Nous espérons ainsi initier une nouvelle communauté. Certains cycles observés alimenteront les discussions autour de ces méthodes de conception.

Nous venons d'exposer les articulations entre trois fondements théoriques : les modèles théoriques interrogeant les concepteurs pédagogiques, les projets sur les technologies éducatives normées et la culture des communautés de pratique dans les organisations. Nous pouvons conclure que la conception d'un système de formation mettant en œuvre les récentes propositions sur les technologies éducatives normées repose sur la compréhension et la communication entre les concepteurs que la communauté EIAH se doit d'instrumenter.

1.3 Analyse de trois outils

Pour mieux préciser le contexte, nous avons identifié trois outils qui, à notre sens, s'inscrivent dans une perspective instrumentale des trois fondements théoriques.

Nous les présentons selon la perspective développée dans (Rabardel 1995). L'analyse d'un outil consiste à identifier deux rôles pour son utilisation : le premier de nature « anthropocentrée » où « c'est l'homme et son activité qui sont placés au cœur du dispositif » [(Rabardel 1995), page 60], les techniques ne sont plus considérées comme une réponse au concepteur mais comme une aide dans son activité d'analyse, et le second de nature « techno-centrée » où les technologies informatiques sont utilisées comme moyen de production ou de recommandation de solutions [(Rabardel 1995), page 48].

Cherchant à définir une démarche de conception, nous sommes amenés à nous interroger à la fois sur les moyens pour former les concepteurs, sur la méthodologie à mettre en œuvre et sur les outils à utiliser. Dans l'objectif d'identifier les premiers éléments de réponse dans la définition d'une nouvelle démarche de conception, nous présentons les trois outils suivants :

- le modèle informationnel LD (Learning Design) du consortium IMS (Instructional Management Systems) Global Learning (Koper, Olivier et al. 2003b),
- le langage de modélisation UML (Unified Modelling Language) utilisé par les processus de développement orienté objet,
- et la communauté OpenUSS correspondant à une communauté de logiciel libre diffusant un EIAH.

En présentant les potentiels et les limites de ces outils, nous délimitons le contexte de notre travail de thèse.

1.3.1 Modèle IMS LD : Support à la conception dirigée par les modèles

Avant d'être des solutions techniques répondant aux préoccupations d'interopérabilité entre les systèmes de formation, les initiatives des projets sur les technologies éducatives normées se scindaient en deux courants différents. Le premier, à l'initiative des américains, est essentiellement centré sur la structuration et le référencement des contenus diffusés. Nous pouvons citer le standard LOM (Learning Object Metadata) (IEEE/LOM 2002) et des langages de

composition comme le modèle informationnel « Content packaging » de l'IMS (Smythe et Cooper 2003). Le second, à l'initiative européenne, est plus particulièrement sensibilisé à la définition d'un modèle conceptuel pour décrire le déroulement d'une unité d'apprentissage. L'étude sur les langages EMLs (Educational Modelling Language) (Rawlings, Rosmalen et al. 2002) effectue une première synthèse sur une sélection de six propositions de langages ouverts, issues de différents travaux européens. Actuellement, l'un d'eux est adopté comme l'un des projets de spécifications du consortium IMS. Ce dernier intitulé « Learning design » correspond à l'adaptation du modèle proposé par Rob Koper de l'université ouverte des Pays-Bas (Koper 2001) représenté par la Figure 5. La particularité du modèle d'origine a été de regrouper au sein d'un modèle unique les différents aspects du scénario pédagogique : les phases d'évaluation, d'adaptation de support hypermédia, de gestion des compétences de l'apprenant.

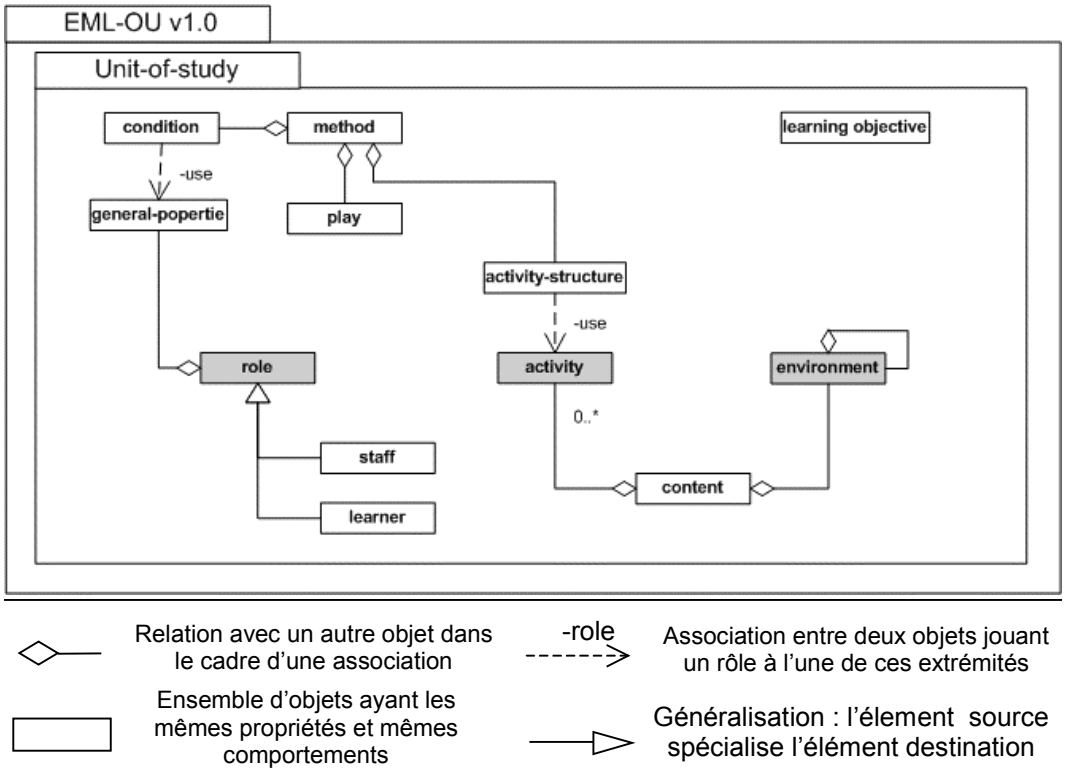


Figure 5 : Diagramme de classe UML représentant le modèle informationnel de l'« Open university » des Pays-Bas, adapté de [(Koper 2001), figure 5]

Dans ce modèle, le concepteur est amené à utiliser les concepts « role », « activity », « environment » pour décrire son scénario d'apprentissage, décrivant alors les flux de travail entre les activités des apprenants mais également entre les membres de l'équipe lors du déroulement de la session d'apprentissage.

Aujourd'hui, ce modèle a été adapté pour être intégré au processus de spécification du consortium IMS. Il se décline sous la forme de trois niveaux (A, B et C) afin de prendre en compte les difficultés de mise en œuvre par l'industrie des plates-formes de formation à distance. Le scénario correspondant à une instance du modèle de niveau A (cf. Figure 6) présente à l'apprenant différentes ressources pédagogiques sélectionnées au préalable par le concepteur.

La gestion événementielle est réduite au seul déclenchement de la diffusion d'une ressource à la fin de l'activité précédente. Le concepteur est alors contraint d'utiliser les concepts « role », « activity », « environment » et « outcome » pour spécifier les aspects statiques du scénario et le concept « method » pour spécifier les aspects dynamiques du scénario.

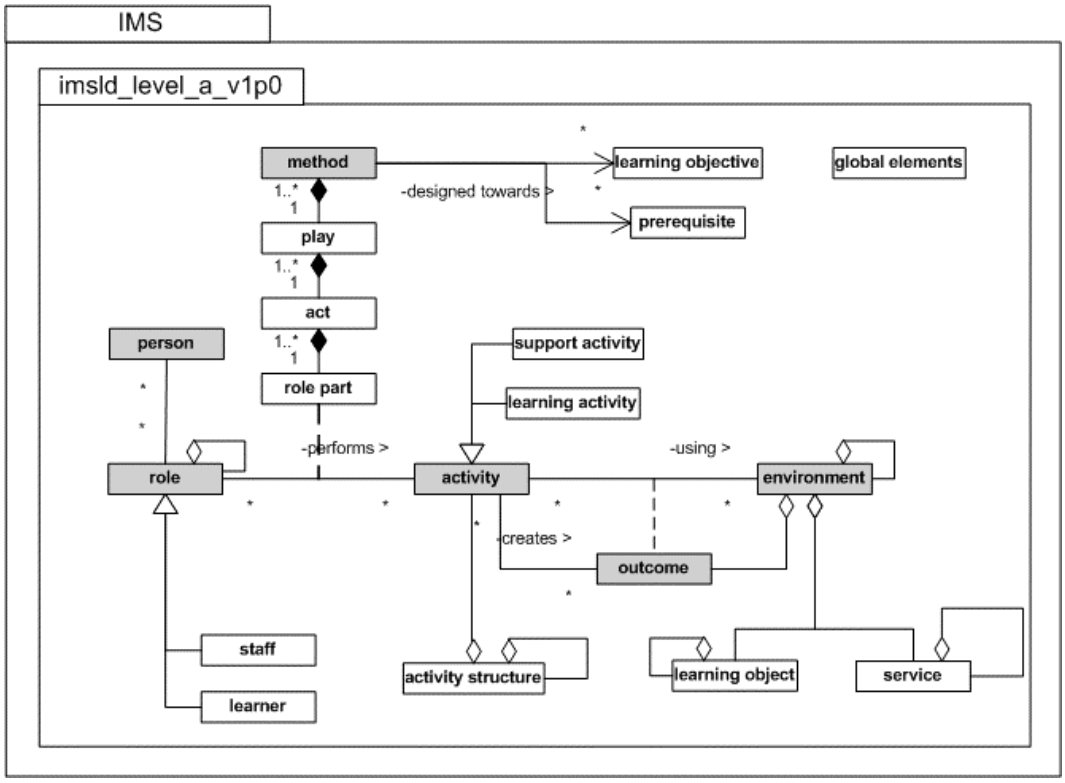


Figure 6 : Diagramme de classe UML représentant le modèle informationnel « learning design » de niveau A de l'IMS [(Koper, Olivier et al. 2003b), figure 3.1]

Les niveaux B et C définissent deux variantes du modèle précédent (cf. la Figure 7 et la Figure 8). Le niveau B ajoute deux nouveaux concepts « property » et « condition » pour prendre en compte le comportement d'un acteur tenant l'un des deux rôles apprenant ou équipe pédagogique. Il est alors possible de préciser les préférences, les connaissances et les caractéristiques de chaque acteur. Le niveau C gère des événements générés au niveau des ressources qui sont à destination des agents logiciels ou des acteurs lors du déroulement du scénario pédagogique.

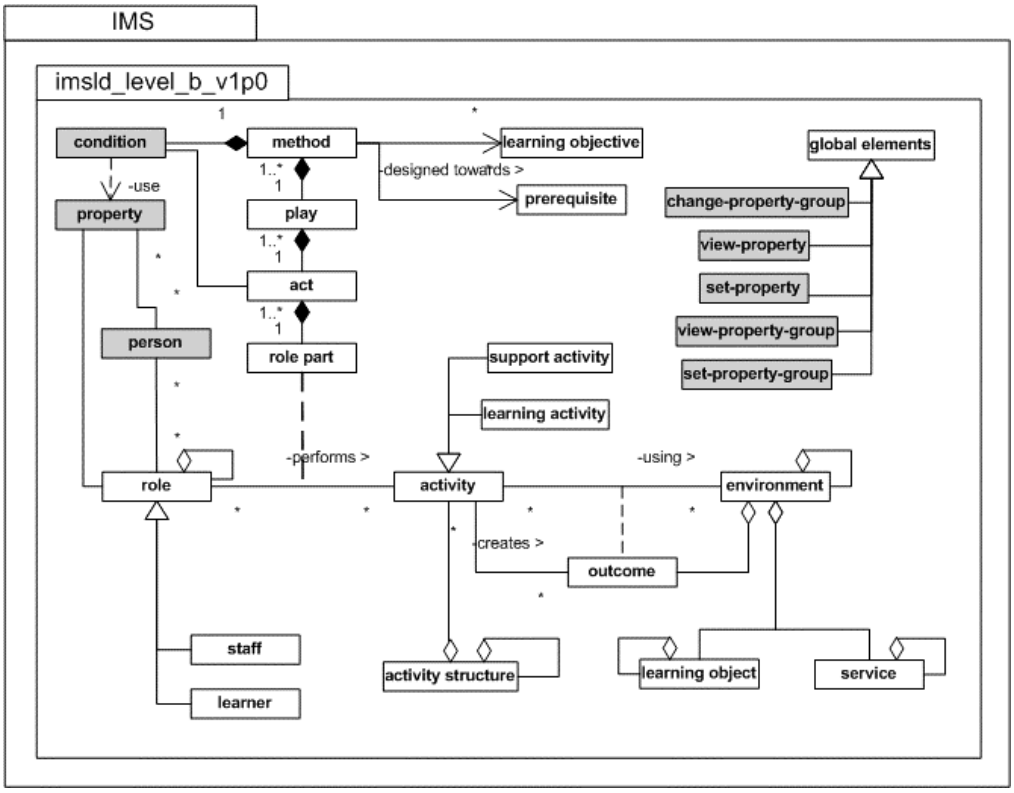


Figure 7 : Diagramme de classe UML représentant le modèle informationnel « learning design » de niveau B de l'IMS [(Koper, Olivier et al. 2003b), figure 3.2]

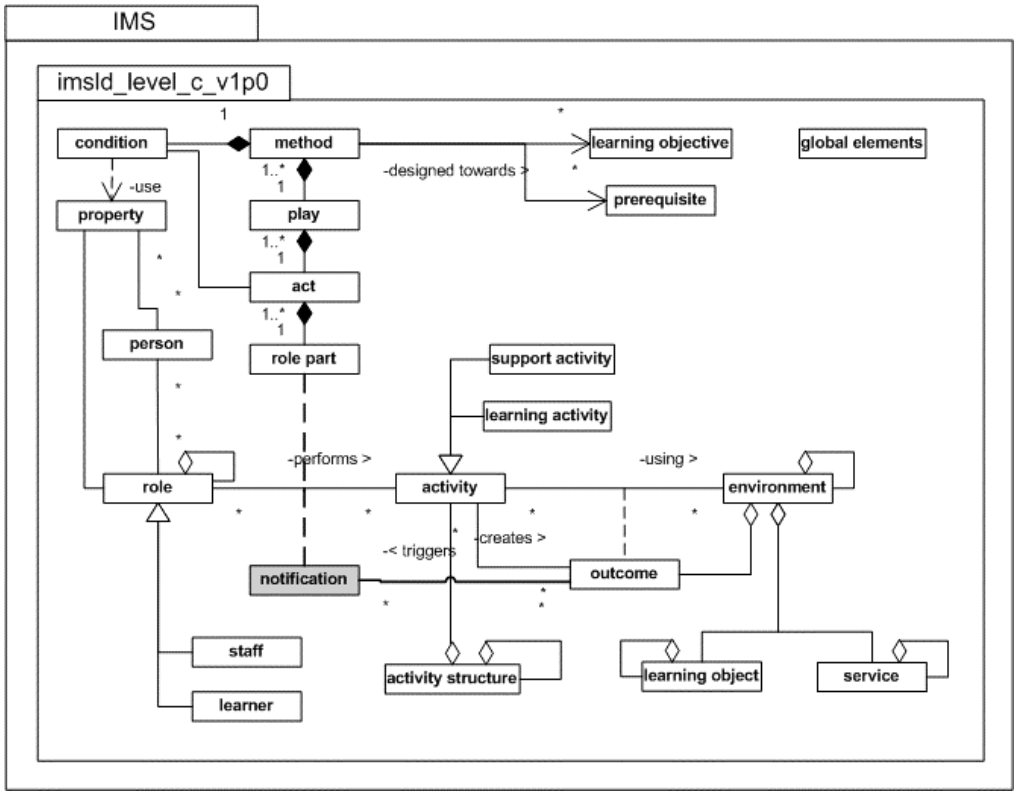


Figure 8 : Diagramme de classe UML représentant le modèle informationnel « learning design » de niveau C de l'IMS [(Koper, Olivier et al. 2003b), figure 3.3]

En présentant les quatre modèles précédents, nous pouvons constater qu'ils servent à supporter la spécification des scénarios d'apprentissage des concepteurs mais également à expliciter la complexité des décisions à prendre.

En effet, afin de prendre en compte la compatibilité entre les différentes évolutions des plates-formes d'apprentissage, le projet sur « learning design » du consortium IMS décline le modèle en trois versions A, B et C. Dans la communauté EIAH, cette déclinaison laisse place à des interprétations différentes de la part des concepteurs. Par exemple dans (Pernin et Lejeune 2004), le niveau A définit des scénarios « prescriptibles », le niveau B des scénarios « de personnalisation de l'apprentissage » et le niveau C des scénarios « dynamiques ».

De plus le modèle d'origine propose un ensemble de concepts pour spécifier la structuration d'un objet pédagogique [(Hermans, Koper et al. 2000), page 34]. Il permet par exemple une description détaillée d'un support textuel en précisant la structuration du texte, le type des éléments, les items importants, …. Ces derniers non seulement ne font plus partie des propositions actuelles mais ne sont intégrés dans aucun des projets du consortium IMS. Ce constat illustre une perte pour le concepteur souhaitant spécifier un tel aspect du contenu d'apprentissage.

L'auteur du principal modèle du scénario d'apprentissage, Rob Koper, précise bien que le premier travail a été de structurer dans un modèle cohérent les concepts pédagogiques. Un tel modèle définit alors un vocabulaire et un langage communs. En choisissant la technologie XML (eXtensible Markup Language), le consortium IMS montre sa volonté de s'inscrire dans la perspective définie par l'approche intitulée « web sémantique » (Berners-Lee, Hendler et al. 2001), centrée sur des problématiques d'interopérabilité.

En prenant cet exemple de modèle pouvant susciter un intérêt auprès des développeurs de plates-formes de formation à distance et s'adressant aux concepteurs souhaitant spécifier une unité d'apprentissage, nous avons mis en avant le fait que les projets sur les technologies éducatives normées mettent en œuvre les premiers outils permettant d'instrumenter les processus de conception dirigés par les modèles.

1.3.2 Langage de modélisation UML

En reprenant les deux chantiers entrepris par les deux organismes de normalisation sur les technologies éducatives, l'IEEE et l'IMS Global Learning, nous cherchons à identifier leurs potentiels et leurs limites dans la définition d'un processus de développement d'un système de formation. Ensuite, nous les comparons aux approches orientées objet de conception de logiciel autour du langage de modélisation UML.

Ces deux premiers chantiers sont américains mais leurs sources d'inspiration divergent. Ils fournissent chacuns des éléments de réponses pour définir une méthode de conception d'un EIAH. Le premier élément est de proposer un modèle d'architecture d'un système de formation et le deuxième un ensemble cohérent de modèles structurant des concepts informationnels.

Le modèle d'architecture (IEEE/LTSA 2003) (Learning Technology Systems Architecture) proposé par le comité IEEE reprend la méthode de décomposition fonctionnelle et les principes d'analyse et de conception structurée définis par (Yourdon 1989). La structure standard composée de processus, de flux et d'unités de stockage définit une vision homogène sur l'architecture d'un système de formation mettant en œuvre les technologies de l'information (cf. Figure 9).

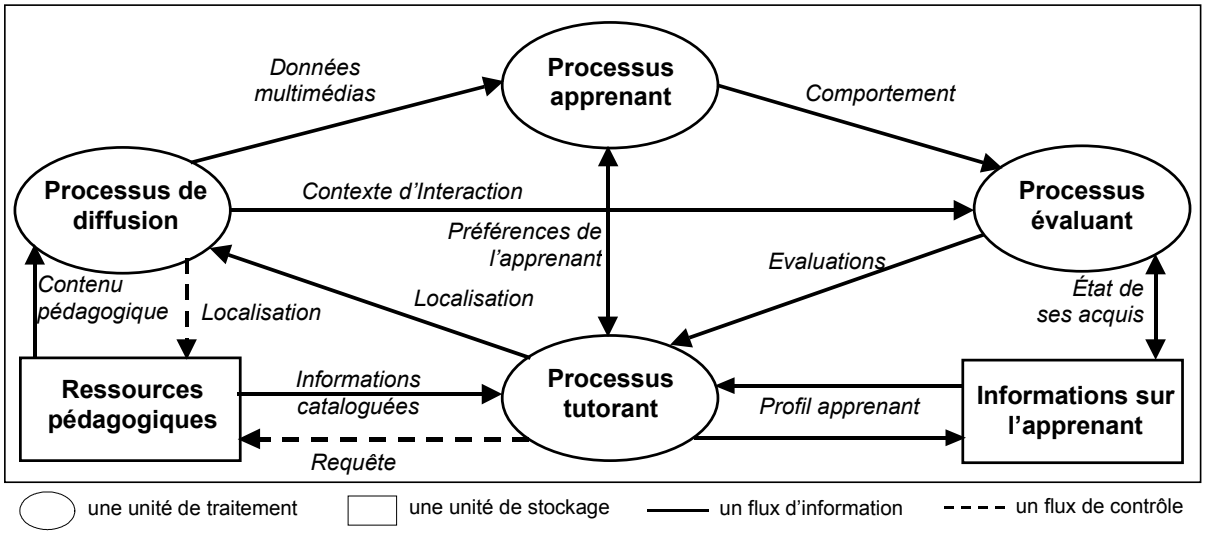


Figure 9 : Modèle à composants du système LTSA (IEEE/LTSA 2003)

Ce modèle permet d'identifier les différentes interfaces d'architecture à un haut niveau d'abstraction. Sa description est accompagnée d'un ensemble de perspectives définissant les spécificités technologiques des instances des composants du système LTSA. Ce standard n'a pas eu le succès escompté dans la communauté des technologies éducatives normées. Cependant, les différentes instances du modèle formalisent une première taxonomie de systèmes d'apprentissage au niveau des protocoles, des formats d'échange des données, des interfaces de programmation et des composants à base d'objets. Quant aux termes utilisés pour les identifier, ils correspondent au vocabulaire du discours des concepteurs des EIAH et à la terminologie définie par le cadre standard de description architecturale du comité IEEE (IEEE/ADS-IS 2000). Inscrit dans un tel cadre, le système LTSA ne se destine pas seulement à expliciter les détails techniques de mise en œuvre mais à servir de support d'échanges entre les différents concepteurs (cf. l'exemple représenté sur la Figure 10) .

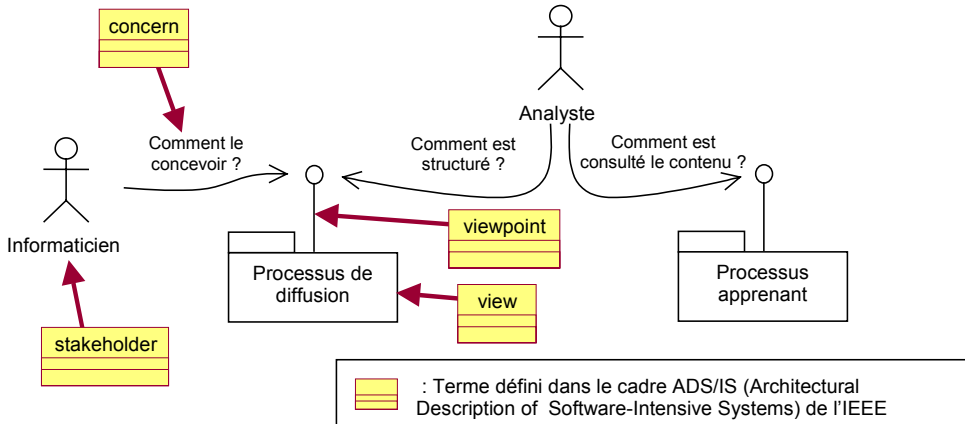


Figure 10 : Illustration d'une instance des concepts de description d'une architecture (IEEE/ADS-IS 2000) sur deux composants du système LTSA (IEEE/LTSA 2003)

Nous pouvons identifier dans cette proposition un premier instrument supportant la communication entre les concepteurs. En effet, l'adoption d'une vision architecturée d'un système de formation permet d'associer les termes utilisés par l'ingénierie éducative à une vision structurée représentant des contraintes sur les technologies mises en œuvre.

Pour le chantier entrepris par le consortium IMS, ce même constat a été observé sur plusieurs niveaux. Le premier définit l'organisation des différents groupes de travail. Chacun produit des spécifications prenant en charge un aspect particulier de l'interopérabilité d'un système distribué d'apprentissage. Le second définit le modèle de structure (cf. Figure 11) que le concepteur utilise pour spécifier l'organisation des différentes composantes d'une unité d'apprentissage.

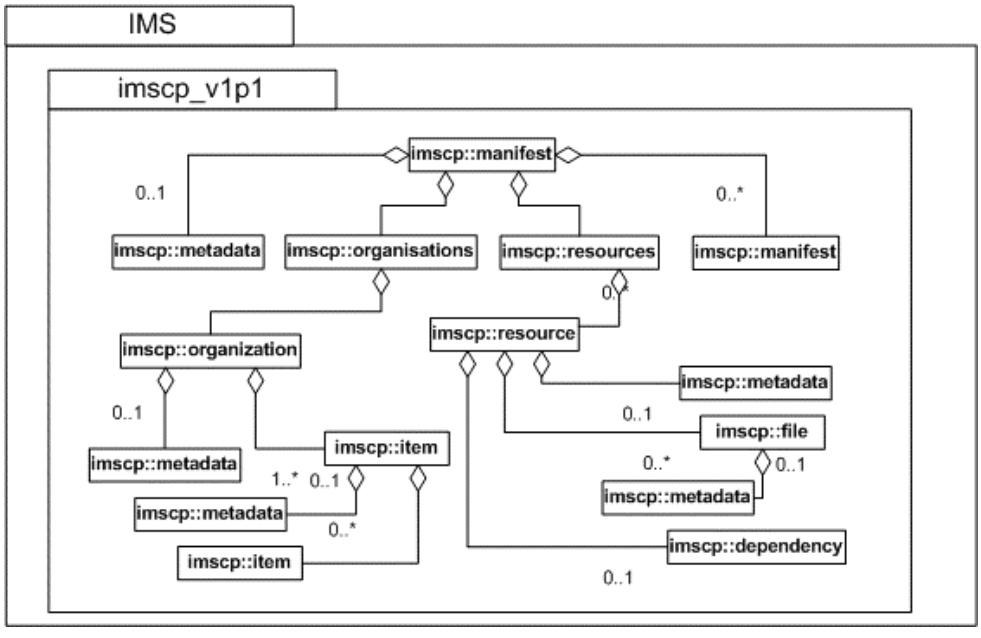


Figure 11 : Diagramme de classe UML représentant le modèle informationnel « content package » de l'IMS

Pour chacun des concepts informationnels « imscp:metadata » du modèle représenté sur la Figure 11, un méta-descripteur peut être associé. Nous mettons en évidence ici la relation existante entre la terminologie explicitant une sémantique pédagogique et une structure liée aux contraintes technologiques de l'architecture d'un système. L'auteur de ressources spécifiant une intention pédagogique se doit de respecter rigoureusement la syntaxe définie par le modèle utilisé et de comprendre le comportement du composant du système de formation mettant en œuvre la spécification.

L'organisation de ces documents, la structuration des différents processus de spécification du consortium IMS, les liens avec les projets comme LTSA de l'IEEE ou, moins abstrait, comme le cadre technique de référence SCORM (Sharable Content Object Reference Model) (ADL/SCORM 2004a) nous amène à nous interroger sur l'existence d'un processus de conception capable d'intégrer d'une manière globale ces différentes structures et d'identifier les relations entre ces différents points de vue.

L'ingénierie du développement logiciel orienté objet nous apporte plusieurs réponses. Elles sont principalement centrées sur le langage de modélisation UML. Ce langage (Booch, Rumbaugh et al. 2000) est le résultat d'une fusion de plusieurs approches. La méthode de conception proposée par Grady Booch, permet de modéliser les points de vue logique et physique d'un système. Le concept de package (élément d'organisation des modèles) a été repris dans UML. Ivar Jacobson issu d'un centre de développement d'Ericsson, en Suède, a défini la méthode OOSE (Object-Oriented Software Engineering) décrivant le cycle de développement. Cette méthodologie repose sur l'analyse des besoins des utilisateurs. James Rumbaugh issu du centre de R&D de General Electric a défini la méthode OMT (Object Modelling Technique) permettant de modéliser les points de vue statique, dynamique et fonctionnel d'un système. Les concepteurs représentent et communiquent de manière graphique ces différents aspects complémentaires. Ils prennent en compte les différents espaces du discours des développeurs d'un système orienté objet.

Nous présentons trois utilisations du langage UML dans la communauté de développement du génie logiciel orienté objet :

- Il sert à spécifier les vues sur l'architecture du système, (Kruchten 1995) : la vue des cas d'utilisation, la vue logique, la vue d'implantation, la vue du processus et la vue du déploiement. Ces visions permettent de guider les activités du développeur informatique d'un processus de développement et a permis par la suite d'unifier un ensemble de règles de bonnes pratiques d'un tel processus (Jacobson, Booch et al. 2000).
- Il sert à représenter les différents points de vue sur un processus unifié de développement logiciel orienté objet comme par exemple le processus 2TUP (Two Tracks Unified Process) proposé par Valtech (Roques et Vallée 2004). Les différents niveaux d'abstraction sur l'application et les règles de transformations sont ainsi spécifiés.

- Ils se positionnent au cœur du cadre MDA™ (Model Driven Architecture) défini par le groupe de standardisation OMG (Object Management Group). S'inscrivant dans une perspective à plus long terme, ce cadre générique s'adresse aux communautés industrielles et scientifiques du développement du logiciel orienté objet afin de fédérer des formalismes standards, des méthodes et des techniques de développement (OMG/MDA 2003).

Comme nous venons de le présenter, deux propositions viennent instrumenter le processus de développement d'un système de formation et répondent aux besoins de réutilisabilité et d'intéropérabilité. La première, en proposant des technologies éducatives standards, répond aux attentes des institutions et des entreprises informatiques produisant ou distribuant des logiciels ou des plates-formes pédagogiques et souhaitant avoir des garanties sur la pérennité des technologies pour utiliser des ressources déjà existantes et en créer de nouvelles. La seconde vise à réduire les coûts de développement en appliquant les éléments méthodologiques définis par la communauté de développement du génie logiciel orienté objet.

En considérant la complémentarité entre les différents modèles de spécification des technologies éducatives et le langage orienté objet de modélisation UML, il est alors pertinent de vouloir s'interroger sur la capacité du langage utilisé dans les processus de développement génie logiciel orienté objet à pouvoir décrire et partager les pratiques, les méthodes et les techniques du travail des différents concepteurs s'exprimant dans les modèles de spécification normés.

1.3.3 OpenUSS : Communauté du logiciel libre diffusant un EIAH

Nous détaillons dans ce paragraphe des différentes perspectives d'une communauté du logiciel libre diffusant un EIAH : OpenUSS (Open University Support system Application Programming Interface). Ce projet communautaire, initié en 2001 par (Dewanto 2002), est le premier projet en licence GPL (General Public License) d'un système d'administration centralisé visant à assister les universités et les enseignants. L'objectif de ce projet est d'établir une interface standard de développement qui peut être utilisée par les concepteurs des organismes de formation.

Cette communauté propose à la fois un site portail mettant à la disposition les fonctionnalités du logiciel et à la fois une diffusion sous la forme d'un logiciel libre, pouvant être installé, exécuté, copié, distribué, analysé et modifié. De part sa stabilité et de part le nombre régulier d'utilisateurs d'un tel outil, nous avons choisi OpenUSS comme représentant la communauté du logiciel libre diffusant un EIAH.

La communauté de logiciel libre OpenUSS ne se définit pas seulement au travers du logiciel diffusé mais par la philosophie de développement qui lui est propre. En effet, des réflexions sont actuellement menées afin d'analyser et de formaliser un certain nombre de caractéristiques d'une communauté du logiciel libre (Raymond 2001). Le rôle du concepteur informatique membre d'une telle communauté n'est plus de réaliser un programme en produisant le code

d'une application mais de communiquer en créant des instances de structures des données, en interconnectant des sous-systèmes et en déclenchant des services distants. Les membres d'une telle communauté jouent alors les rôles d'utilisateur, de développeur, de gestionnaire, de veille technologique ou de formateur.

Il est alors intéressant d'analyser une telle communauté pour identifier les éléments de compréhension permettant de mettre en relation les approches techniques et sociologiques. Un processus de développement est alors vu sous les angles de la communication entre pairs, des formalismes et des outils utilisés.

Face à la complexité de la conception des systèmes informatiques, les réflexions ne portent plus seulement sur les outils et les méthodes supportées par les environnements de communication mais également sur l'environnement social et humain dans lequel sont immergés les membres de la communauté du logiciel libre (Raymond 2001). La réponse apportée par cette communauté est de motiver l'implication d'un nombre de plus en plus important de participants au projet de développement informatique. Les notions au cœur d'une démarche projet d'aujourd'hui se définissent par le maintien de l'identité de chaque membre, du partage et de l'entraide. Les instruments utilisés sont avant tout des outils de communication. La variété des formalismes et des modalités de communication permet d'échanger directement entre les membres d'une telle communauté.

La prise de conscience de ces différentes dimensions est effective dans la conception d'un système informatique. Les nouvelles approches de développement se veulent opportunistes, dans le sens où une analyse du contexte dans lequel elles s'exécutent est à la fois descendante et ascendante (Raymond 2001) :

- elle exploite des informations décrivant les aspects organisationnels, technologiques et économiques.
- elle met en œuvre une approche par prototypage afin de comprendre et de répondre aux attentes des utilisateurs.

Différents rôles sont définis dans une communauté de logiciel libre. Dans (Nakakoji, Yamamoto et al. 2002), l'analyse de quatre projets a permis de répertorier huit rôles et de les représenter sous forme de strates circulaires.

- Meneur du projet : il est souvent la personne qui est à l'origine du projet. Il est le responsable pour la planification et les orientations du projet.
- Membre principal : il est responsable du guidage et de la coordination du développement du projet. Il est la personne qui est impliquée dans le projet depuis longtemps et a contribué de manière significative au développement et à l'évolution du système. Dans les communautés du logiciel libre, c'est un des acteurs dédié à la maintenance.
- Développeur actif : il intervient régulièrement pour créer de nouvelles fonctionnalités et corriger les erreurs de programmation.

- Développeur périphérique : il contribue occasionnellement au développement. Sa contribution est irrégulière et son implication n'est qu'épisodique.
- Correcteur d'erreur : il corrige les erreurs de programmation qui sont découvertes par lui-même ou répertoriées par les détecteurs d'erreur. Il intervient sur une partie du code source du système où l'erreur a été décelée.
- Détecteur d'erreur : il recherche et répertorie les erreurs, mais ne les corrige pas.
- Lecteur : C'est un utilisateur actif du système, il ne fait pas qu'utiliser le système, mais il essaye de comprendre comment il fonctionne en lisant le code source. Il joue de rôle de testeur d'un processus de développement logiciel classique.
- Utilisateur passif : Il utilise le logiciel comme un outil commercial. Son intérêt est de disposer d'un logiciel libre de bonne qualité et d'en changer quand il en éprouve le besoin.

A la différence du processus de développement logiciel classique, ce dernier rôle est pris en considération dans une communauté du logiciel libre. En effet, l'acte de téléchargement des productions est signifiant. Il marque un premier niveau d'adhésion et témoigne de l'intérêt porté à la communauté.

En intégrant une communauté du logiciel libre diffusant un EIAH dans le contexte de notre problématique, nous souhaitons pouvoir identifier des enjeux et des limites d'une « communautés de pratique » au sens de (Wenger 1998; Wenger, McDermott et al. 2002).

1.3.4 Synergies entre les trois outils dans une approche systémique

Les trois outils que nous venons de présenter nous permettent de dégager les trois champs définissant notre domaine d'analyse. Le premier montre que les différentes propositions des projets sur les technologies éducatives normées instrumentent les éléments méthodologiques d'une démarche dirigée par les modèles au sens où l'acte de modélisation se positionne au centre des méthodes de conception d'un système de formation. Le second montre qu'il est judicieux de considérer l'influence du langage de modélisation UML proposé par le groupe OMG. En effet, ce dernier est devenu l'outil standard manipulé dans la plupart des processus de conception génie logiciel s'étendant aux processus de conception d'un système d'information. A l'aide d'un tel outil, les concepteurs sont amenés à manipuler des modèles plus ou moins abstraits et à les transformer au cours des différentes phases d'ingénierie ou de réingénierie d'un système de formation. Le troisième champ montre que le nombre d'expériences réussies de projets de développement mettant en œuvre les principes d'une communauté de pratiques amène à s'interroger pour analyser leurs impacts sur la conception des composantes d'un système de formation.

Comme il est présenté dans (Tchounikine, Baker et al. 2004), une des contraintes de conception d'un EIAH est de penser non par une simple juxtaposition des disciplines mais d'expliciter leurs contributions respectives dans une

organisation. La finalité de cette dernière est de résoudre les problèmes posés par le système de formation développé. Suivant cette perspective, notre travail est d'observer et d'étudier les trois synergies suivantes :

- synergie entre une communauté du logiciel libre et une approche dirigée par les modèles ;
- synergie entre une communauté du logiciel libre et un processus de développement du logiciel ;
- synergie entre une approche dirigée par les modèles et un processus de développement du logiciel.

Afin d'effecuer ces analyses, nous mettons en œuvre une approche dite « systémique » afin d'appréhender un nouveau domaine d'analyse qui est alors considéré « par sa dynamique, sa complexité et sa totalité » [(De Rosnay 1975), page 119]. Nous considérons alors chacun des concepts fondamentaux de la systémique [(Durand 1979), pages 8-11] : l'organisation, la complexité et la rétroaction. Les synergies analysées nous permettent de représenter un domaine comme une structure organisée, complexe et possédant une grande variété d'interactions.

1.3.4.1 Synergie entre une communauté du logiciel libre et une approche dirigée par les modèles

Nous nous plaçons suivant le point de vue « organisation » défini selon les théories systémiques comme « l'agencement de relations entre composants ou individus qui produit une unité complexe ou système, dotée de qualités inconnues au niveau des composants ou des individus » [(Morin 1977), page 103].

L'objectif est d'identifier les propriétés de composition des éléments d'un système de formation. Cette vision se définit dans une démarche qui à la fois intégre les différents points de vue technologiques et met en évidence les facteurs créatifs et imaginatifs des différents concepteurs d'un tel système.

Notre vision de la conception d'un système de formation est celle d'un système complexe qui se compose d'un ensemble de sous-systèmes en interaction. Le diagramme de Gantt représenté dans la Figure 12 témoigne de notre interprétation de la nature dynamique de trois projets du consortium IMS proposant un modèle de spécification sur les technologies éducatives.

Les concepteurs développant des EIAH dans un esprit d'évolutivité se doivent de comprendre les aspects technologiques, d'expliciter leur analyse et de partager leur pratique. L'approche dirigée par les modèles et les communautés du logiciel libre nous offrent deux perspectives différentes de la collaboration et de la communication dans une structure organisationnelle de conception. La première, définie par les consensus autour des technologies éducatives cherche à la fois à promouvoir ces modèles (Koper 2001) et à établir de nouvelles perspectives (Koper 2004). La seconde se décrit comme une démarche de développement opportuniste et cherche à définir des règles au regard des expériences passées (Raymond 2001). Elle se crée sur la base du volontariat, de l'intérêt et de la sensibilité de l'individu, meneur du projet ou simple utilisateur.

Pour analyser l'articulation de ces deux perspectives au sein d'un même système et du point de vue organisationnel, nous avons fait le choix d'avoir une vision systémique. L'enjeu pour les technologies éducatives normées est de supporter les échanges entre les concepteurs d'environnement d'apprentissage (Hummel, Manderveld et al. 2004). Le choix d'un tel cadre théorique nous permet de rechercher et de présenter les structures, les règles et les regroupements. En effet, les projets du consortium IMS influencent actuellement les industries de conception de plates-formes de formation à distance. Le jeu des interfaces standards et des modules réutilisables vont encourager les initiatives de partage du code de ses composants.

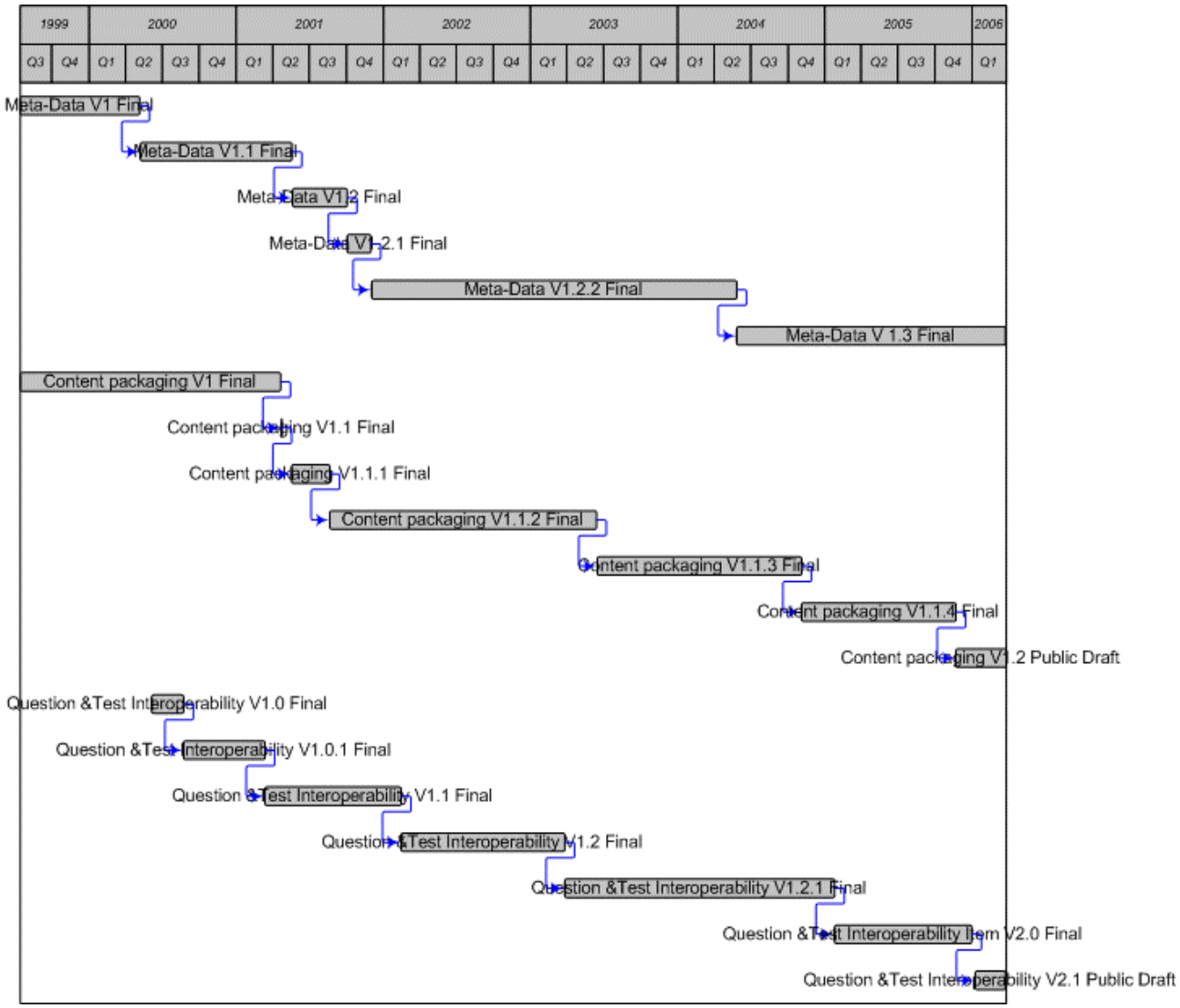


Figure 12 : Diagramme de Gantt de trois projets du consortium IMS

Or, ces formalismes ne suffisent pas. Les cas d'utilisation représentés dans les guides de mise en œuvre ou diffusés sur les sites de communautés du logiciel libre utilisent des formalismes supplémentaires comme par exemple le langage de spécification orienté objet UML. Les diagrammes de classes et d'interactions sont alors utilisés pour représenter respectivement les structures et les relations du modèle informationnel ou pour expliciter les comportements

des acteurs d'une session d'apprentissage. La synergie exposée nous interroge alors sur le rôle joué par ces différents formalismes pour comprendre et expliciter la structure et le fonctionnement dans l'organisation de la conception d'un système de formation. Centré sur les problématiques d'interopérabilité, les technologies éducatives définissent des langages informationnels permettant de décrire et de communiquer des scénarios *a priori*. Ils se destinent à être interprétés par les dispositifs de formation mais également à être utilisés par les concepteurs comme formalismes d'échange sur leurs pratiques.

La cybernétique, considérée ici comme la science de l'organisation dans une approche systémique [(Durand 1979), page 38], nous demande de rechercher un formalisme abstrait afin de représenter au mieux les phénomènes observés. L'enjeu est de favoriser la communication entre experts ou novices de disciplines différentes. La communauté du logiciel libre apporte une vision particulière au processus de développement. En effet, nous parlons non pas de processus « informatique » où seuls les outils sont pris en considération, mais d'un processus « organisationnel » (Debauche et Mégard 2004) intégrant les acteurs humains. Autant les outils supports à la communication se présentent comme des moyens pour optimiser le processus de conception, autant la communauté du logiciel libre crée une rupture dans les valeurs sociales des membres du projet de développement. Les rôles au sein de cette communauté sont alors remis en cause et induisent « des modifications qui touchent aux relations contractuelles, à l'économie, à l'évolution des métiers des informaticiens » [(Rastetter 2002), page 20]. Les expériences réussies sont communiquées par l'intermédiaire de règles (Raymond 2001) s'adressant directement aux concepteurs souhaitant initier ou maintenir de telles communautés.

Analyser l'impact de ces évolutions dans l'approche dirigée par les modèles mise en œuvre au sein des processus de standardisation sur les technologies éducatives des consortiums IMS et IEEE revient à expliciter les synergies existantes entre une communauté du logiciel libre et une approche dirigée par les modèles.

1.3.4.2 Synergie entre une communauté du logiciel libre et un processus de développement logiciel

La deuxième synergie se focalise sur les liens existants entre une communauté du logiciel libre et un processus de développement du logiciel. Nous utilisons alors le concept de « complexité » pour nous apporter l'éclairage suffisant dans l'analyse des spécificités de la synergie étudiée. Nous reprenons l'explication donnée par [(De Rosnay 1975), pages 101-105] qui a su vulgariser cette notion au travers d'un instrument : « le macroscope ». Un système « complexe » s'oppose à un système « simple » composé « d'éléments semblables, non organisés et présentant de faibles interactions » [(De Rosnay 1975), page 103]. Pour un système, qualifié de « complexe », il est nécessaire de s'inscrire dans une démarche de compréhension de son comportement. Une approche par simulation et dirigée par des modèles est un exemple illustrant ces principes.

Dans un processus de développement du logiciel, l'une des réponses pour concevoir un système « complexe » est de mettre en œuvre une approche par réutilisation de composants. Ces composants se situent à différents niveaux d'une structure guidant le déroulement du processus. Par exemple, le processus 2TUP (Roques et Vallée 2004) propose une architecture en Y dans lequel six types de composant sont définis. Ils représentent chacun un point de vue spécifique sur le processus de développement :

- le composant regroupant les cas d'utilisation : il constitue des modèles de spécification fonctionnelle ;
- le composant regroupant les modèles d'analyse : il organise les concepts liés à un domaine particulier ;
- le composant correspondant aux cadres génériques : il spécifie les contraintes techniques sur l'architecture logicielle ;
- le composant correspondant aux cadres techniques : il constitue un cadre réutilisable demandant à être adapté pour répondre aux besoins fonctionnels de l'application ;
- le composant correspondant aux librairies réutilisables de classes : il offre un ensemble de fonctionnalités que l'application utilise ;
- le composant à déployer : il correspond aux différents éléments nécessaires pour une exécution correcte de l'application.

Le travail du concepteur se situe au niveau des tâches d'interprétation et d'intégration de ces composants supportés par l'environnement de développement logiciel. Au cours des différentes étapes du processus, il se doit de produire un modèle cohérent en établissant des liens entre les structures d'un point de vue à l'autre. De plus l'intéropérabilité syntaxique des modèles produits est assurée par les différents langages de modélisation orientés objet adoptés par le groupe OMG.

Pour le processus de développement dans une communauté du logiciel libre, les réponses apportées pour gérer la conception d'un système « complexe » sont différentes. En effet, le concepteur est amené à réutiliser des composants et à favoriser le prototypage rapide. Il n'est plus contraint d'effectuer telle ou telle tâche mais prend part au développement en se proposant d'effectuer une première phase d'analyse des fonctionnalités pour les comparer à ses propres besoins ou, dans une deuxième phase, de comprendre la structure du code, pour identifier les décisions des concepteurs et prendre part aux évolutions.

Nous pouvons identifier, pour chacun des deux processus de développement, les deux mêmes réponses pour contrôler la complexité d'un système et apporter des solutions à la problématique de la réutilisabilité. La première est la représentation architecturale d'un système informatique (Garlan et Shaw 1994) et la seconde est le partage de consensus sur les pratiques de développement logiciel à l'aide de langages de patrons (Alexander, Ishikawa et al. 1977). En effet, elles se définissent comme des disciplines transversales aptes à supporter un aspect de la réutilisation. La première se positionne au cœur de plusieurs domaines dont les techniques d'extraction de structure à partir de code

existant, la définition de taxonomies à partir de styles d'architectures, l'identification de rôles dans un processus de développement suivant l'architecture, etc. La deuxième est de partager des solutions à des problèmes de conception récurrents. Organisés sous la forme de différentes structures et mis en relation par des liens sémantiques, les langages de patrons servent de guides pour la communauté de concepteurs. Ils sont utilisés lors des échanges de savoir-faire au sein d'une communauté et s'orientent sur la communication de modèles de bonne pratique. La communauté de développement du génie logiciel a largement adopté ces deux outils de manières conjointes. Le premier exemple est une approche dirigée par des patrons d'architecture définis par (Buschmann, Meunier et al. 1996). Leur classification reprend les structures d'architecture logicielle communément utilisées. Il existe des patrons dédiés plus spécifiquement à la conception orientée objet (Gamma, Helm et al. 1999) ou répondant à une volonté du concepteur de communiquer au travers du code à l'aide des techniques de « refactoring[3] » (Fowler, Beck et al. 1999).

En cherchant une démarche englobant à la fois les patrons orientés architecture logicielle et les différents cycles des réflexions menées par l'informaticien au cours du processus de développement, nous avons identifié les patrons de réingénierie du logiciel résultant du projet européen FAMOOS (Bär, Bauer et al. 1999). Ils mettent en œuvre des patrons au niveau de la structure à objets du système développé et au niveau des différentes phases du processus de réingénierie. Ce travail s'inscrit dans une volonté de décrire un ensemble de termes, supports d'échanges entre les différents acteurs d'un processus de réingénierie ; il s'adresse à une communauté élargie aux utilisateurs ou aux experts du domaine d'application souhaitant appréhender une telle culture. Nous identifions ici les mêmes préoccupations au niveau des membres d'une communauté du logiciel libre souhaitant faire adhérer de nouveaux membres en communiquant au travers du code des applications produites.

Les composantes d'un système informatique dédiées à la formation doivent laisser place au raisonnement de concepteurs ayant une expertise liée au domaine d'application du logiciel développé. En spécifiant une session de formation, le concepteur structure et crée des instances des différents modèles du consortium IMS. Plusieurs interprétations sont possibles pour une même spécification. Nous pensons qu'il est important d'identifier et de décrire des négociations pouvant être engagées entre les acteurs concepteurs experts du domaine de la formation ou les utilisateurs, et les développeurs familiarisés avec les principes d'ingénierie ou les technologies utilisées lors de la réalisation de tels systèmes. Nous faisons référence à plusieurs travaux de la communauté EIAH. Sur la base du modèle LTSA de l'IEEE, une première structure de patrons de conception est proposée par (Avgeriou, Papasalouros et al. 2003). Autour du modèle de spécification LD du consortium IMS, un composant Collage (Hernández-Leo, Villasclaras-Fernández et al. 2006) s'intégrant à l'éditeur de scénario pédagogique, diffusé par la communauté du logiciel libre

[3] Le « Refactoring » consiste à modifier le système logiciel sans altérer le comportement externe du code, tout en améliorant la structure interne.

Reload[4], est proposé. Ce composant gére un ensemble de patrons pour guider le travail du concepteur spécificant des situations d'apprentissage collaboratives. Nous pouvons également citer deux projets européens visant à fédérer un ensemble de patrons de conception : le projet e-LEN[5] et le projet « Design Patterns for recording and analyzing usage in learning systems[6] » (Choquet, Merceron et al. 2005).

Cherchant à identifier les liens existants entre une communauté du logiciel libre et un processus de développement logiciel, nous identifions la réingénierie comme notre discipline transversale dans cette analyse. De plus, elle nous interroge sur la pertinence d'une telle culture pour l'ingénierie des EIAH cherchant à fédérer des patrons supports à la négociation entre concepteurs. Il faut alors identifier le moyen capable à la fois d'intégrer les différentes tâches liées aux actes de réingénierie et d'analyser leurs potentiels appliqués dans un processus de développement logiciel mettant en œuvre les modèles de spécification issus des travaux sur les technologies éducatives normées.

1.3.4.3 Synergie entre une approche dirigée par les modèles et un processus de développement logiciel

La dernière synergie se définit au niveau de l'articulation entre une approche dirigée par les modèles avec un processus de développement de logiciel. Une analyse plus approfondie est présentée dans le chapitre 4 de ce mémoire, nous nous intéressons plus particulièrement à la notion de « rétroaction ». Dans (De Rosnay 1975), le principe de causalité linéaire est opposé au principe de causalité circulaire. Dans un cas nous sommes forcés de remonter dans le passé pour expliquer des faits observés. Dans l'autre, l'information est considérée comme une source de décision qui produit de nouvelles actions.

L'objectif est d'analyser dans un premier temps les propositions émanant du développement des logiciels orientés objet et dans un deuxième temps d'identifier les premiers éléments d'une approche dirigée par les modèles dans la conception d'un système de formation à distance.

Les processus de développement génie logiciel orienté objet mettent en œuvre un ensemble de bonnes pratiques. Tout projet est découpé en phases de courte durée (de 2 à 4 semaines), itératives et incrémentales qui permettent de suivre l'avancement global du processus. A la fin de chaque itération, une partie exécutable du système final est produite et permet d'identifier au plus tôt les risques majeurs du projet. La négociation entre les acteurs au cours de ces différents cycles s'effectue autour d'artefacts d'ingénierie correspondant aux informations créées, produites,

[4] Voir, http://www.reload.ac.uk/
[5] Voir, http://www2.tisip.no/E-LEN/
[6] Voir, http://www.noe-kaleidoscope.org/pub/activities/jeirp/activity.php?wp=33

modifiées ou utilisées par les acteurs du projet (Jacobson, Booch et al. 2000). Nous pouvons prendre comme exemple l'artefact de conception logiciel orienté objet correspondant à un diagramme spécifié avec le langage UML. Ce langage au cœur du cadre MDA™ visant à décrire une approche dirigée par les modèles afin de garantir une maintenance, une évolution et une réutilisation optimales des différents éléments du système développé (OMG/MDA 2003). La volonté est d'instrumenter les échanges et la communication d'artefact de conception logicielle. Cette perspective engage de nouveaux besoins de spécifications. Le groupe OMG a pour mission d'organiser des groupes de travail chargés de proposer de nouveaux formalismes. Son objectif est de répondre aux problématiques d'interopérabilité entre les environnements de développement génie logiciel. Il faut souligner que l'OMG a joué un rôle important dans le succès du formalisme UML devenu aujourd'hui le principal langage de spécification dans la conception orientée objet.

L'approche dirigée par les modèles s'inscrivant dans le domaine des pratiques pédagogiques, concerne l'utilisation des récentes propositions sur les technologies éducatives mises en œuvre par les outils utilisés en session de formation et comme formalisme d'échange des pratiques de formation. En étudiant les réactions des communautés scientifiques et en analysant le déroulement des processus visant à décrire les modèles informationnels, l'ingénierie des systèmes de formation est en train de définir les premiers éléments méthodologiques d'une démarche dirigée par les modèles. Les trois initiatives suivantes appliquent les prémisses d'une telle approche :

(1) au niveau du processus mis en œuvre par chacun des projets du consortium IMS (Friesen 2005).

(2) au niveau d'une démarche qualité spécifique au système de formation mettant en œuvre les technologies éducatives (Blandin 2005).

(3) au niveau d'une ressource pédagogique par l'intermédiaire du méta-descripteur « cycle de vie » (IEEE/LOM 2002).

Nous pensons qu'il est judicieux de s'interroger sur les relations pouvant exister entre une approche dirigée par les modèles et les processus de développement génie logiciel orienté objet dans le cadre d'un système de formation. Une telle perspective est à encourager au vu des différents efforts engagés par les différentes communautés entre les consortiums IMS et W3C (World Wide Web Consortium) (Koper 2004), le groupe OMG et le consortium W3C (OMG/MDA 2001a) et le consortium IMS et le groupe OMG (Koper, Olivier et al. 2003a). Par exemple, le modèle de méta-descripteur d'un objet pédagogique proposé par le consortium IMS utilise le système de notation de diagramme de classe UML spécifié par le groupe OMG.

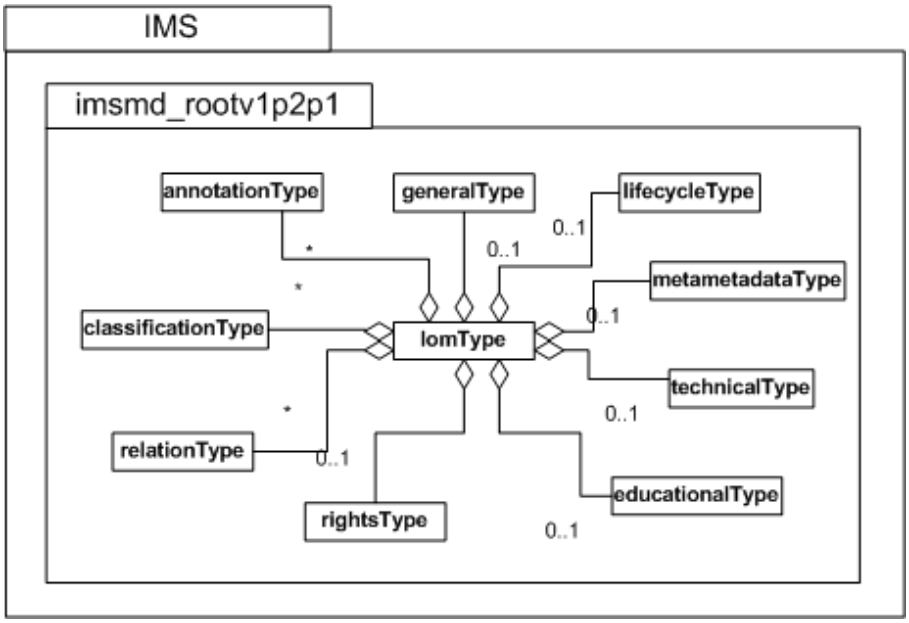


Figure 13 : Diagramme de classe UML représentant le modèle du « meta-descripteur » du consortium IMS

Dans la structure représentée par la Figure 13, nous identifions plusieurs éléments porteurs de sens différents :

- Le concept « lifecycleType » est le moyen d'action du concepteur pour préciser l'évolution de la sémantique pédagogique de la ressource dont il est l'auteur.
- La version du paquetage « imsmd_rootv1p2p1 » précise que l'artefact s'inscrit dans une logique d'adaptation et d'évolution dans un processus de développement.
- Le paquetage « imsmd_rootv1p2p1 » appartient à un ensemble plus général contenant différents paquetages. Chacun d'eux définit un espace de noms distinct. Ces espaces de noms définissent les différents aspects d'un système de formation à appréhender par le concepteur.

Les formalismes de représentation empruntent la notation des diagrammes orientés objet du langage UML. Ils doivent être interprétés par le concepteur.

Un tel modèle s'adresse autant au concepteur informaticien soucieux de concevoir des dispositifs informatiques capables d'interpréter de telles structures qu'au concepteur pédagogique contraint de créer, d'adapter et de stocker les matériaux pédagogiques. Un objet se décrit de manière intrinsèque par un modèle d'architecture (Roschelle, DiGiano et al. 1999; Verbert et Duval 2004) et s'intègre dans un scénario pédagogique (Koper 2001). Les différents concepteurs utilisant les technologies éducatives normées, sont appelés à réaliser les opérations d'instanciation, d'abstraction, d'affinement et d'encapsulation. Le contexte dans lequel ces opérations sont effectuées se doit d'être formalisé et partagé par une communauté de concepteurs la plus large possible.

1.4 Projet REDiM (Réingénierie des EIAH Dirigée par les Modèles)

L'analyse des trois synergies exposées dans la partie précédente engage trois nouvelles réflexions :

- sur l'analyse des processus de standardisation sur les technologies éducatives du consortium IMS et du comité IEEE ;
- sur l'impact des récents travaux de la réingénierie du logicielle sur la conception d'un système de formation ;
- et sur l'explicitation des actes de spécification et de modélisation des concepteurs d'un système de formation.

Le résultat de ces analyses constitue l'un des axes de travail du projet REDiM (Réingénierie des EIAH Dirigée par les Modèles) mené au LIUM (Laboratoire d'Informatique de l'Université du Maine).

Avant de présenter le projet REDiM, nous apportons une clarification terminologique sur les termes « réingénierie » et « dirigée par les modèles ».

La réingénierie[7] est définie dans (Chikofsky et Cross 1990) comme étant le processus « d'examen et d'altération d'un système afin de le reconstituer sous une nouvelle forme suivi de l'implantation de cette nouvelle forme ». Les auteurs suggèrent que les transformations apportées au système sont effectives dans et entre les niveaux d'abstraction correspondant aux trois étapes d'un cycle de développement : la capture des besoins, la conception et l'implantation.

L'activité de réingénierie des EIAH Dirigée par les Modèles requiert l'utilisation de modèles suffisamment élaborés qui permettent d'abord l'étude des détails d'une implantation et ensuite, l'application des correctifs ou des transformations nécessaires. Le processus de réingénierie mis en œuvre dans ce projet scientifique repose sur l'idée principale que la description formelle du point de vue du concepteur (i.e. le scénario prédictif), en termes de scénarios aide et dirige l'analyse des usages observés (i.e. les scénarios descriptifs). Ces résultats d'analyse peuvent alors être exploités pour guider la réingénierie de l'EIAH considéré, et plus largement, servir de guide lors de la conception d'un système de formation.

L'activité de réingénierie se définit en terme de méthodes, d'approches, de modèles de processus, etc…, le projet REDiM puise ses sources dans différentes propositions de normes internationales, de standards et de spécifications du domaine de la formation ainsi que dans les communautés les mettant en pratiques.

[7] Traduction de « reengineering »

Trois objectifs ont été définis. Le premier cherche à définir un modèle d'organisation du processus d'ingénierie et de réingénierie d'un EIAH (Corbière et Choquet 2004a) et un ensemble de termes à adresser aux concepteurs pour lui permettre d'expliciter les éléments méthodologiques d'un tel processus (Corbière et Choquet 2005). Le second se focalise plus spécifiquement sur le travail de négociation au sein d'un groupe de concepteurs pédagogiques. L'objectif est de définir des outils leur permettant de capitaliser et d'exploiter les connaissances susceptibles d'aider à l'ingénierie et la réingénierie du scénario pédagogique (El-Kechai et Choquet 2006). Le troisième cherche à définir les moyens d'observation d'une situation pédagogique afin d'adresser aux concepteurs des indicateurs lui faisant sens (Iksal, Barré et al. 2004; Randriamalaka et Iksal 2006).

Le travail présenté dans ce mémoire tente de répondre au premier objectif.

1.5 Problématique et méthodologie

1.5.1 Énoncé de la problématique

Aujourd'hui les systèmes d'information mettent à disposition des modèles conceptuels concernant l'algorithmique, les réseaux sémantiques, les hypertextes, les cadres et les scénarios de l'intelligence artificielle… Ils supportent une représentation du monde avec ses différentes facettes pour ensuite simuler, modéliser et expérimenter (Linard 1996). De tels modèles sont des images mentales supports à l'action et à la décision. Ces modèles sont organisés afin de ne présenter qu'un certain nombre de concepts simultanément.

A notre sens, l'un des enjeux pour la communauté de conception des EIAH est de définir les éléments d'une démarche méthodologique centrée sur l'utilisation des composants d'un système d'information. Cette communauté conçoit, produit et réutilise des artefacts informatiques. Nous nous interrogeons alors sur l'existence d'un cadre unificateur définissant un ensemble de concepts permettant de décrire et de partager avec un minimum d'ambiguïté les éléments méthodologiques manipulant ces artefacts.

Notre problématique consiste à rechercher un tel cadre, à démontrer sa pertinence sur différents exemples issus des pratiques de conception des EIAH et à proposer une représentation d'une communauté de réingénierie d'un système de formation.

1.5.2 Proposition : choix d'un cadre systémique

Le concept d'objet est prédominant dans la grande majorité des travaux sur les technologies éducatives et plus largement dans les systèmes d'information. En effet, le principal attrait d'une approche centrée objet est d'exploiter son potentiel de modularité au niveau des connaissances et au niveau de la perception des informations utilisées. La

maturité des différentes communautés scientifiques mettant en œuvre ce type d'approche argumente en faveur d'un tel choix.

Le cadre de référence du processus distribué ouvert de l'ISO et de l'IEC (International Electrotechnical Commission) se présente comme un méta-standard. La vision globale d'un système est explicitée par des mécanismes de transformations des différents points de vue. Ce cadre applique les théories systémiques (Wegmann et Naumenko 2001) qui ont pour rôle d'intégrer différentes disciplines telles que la théorie de l'information, la cybernétique et la théorie des systèmes. L'objectif est « d'identifier de nouvelles méthodes permettant de rassembler et d'organiser des connaissances en vue d'une plus grande efficacité dans l'action » (De Rosnay 1975).

Un tel standard a pour vertu d'adresser un ensemble de termes utilisés par les communautés de concepteurs soucieux de partager les pratiques. Un tel vocabulaire permet d'expliciter les invariants et les principes généraux, structuraux et fonctionnels d'un système informatique distribué. Ils correspondent aux artefacts émergeant des pratiques d'un système de formation. Ils sont alors plus facilement décrits et communicables.

L'ingénierie pédagogique mettant en œuvre les langages EMLs (Educational Modelling Language) (Rawlings, Rosmalen et al. 2002) utilise un ensemble de modèles informationnels cohérents. En particulier, le consortium IMS se présente comme une fédération de projets au sein d'une organisation structurée. Chacun d'eux produit un modèle définissant un aspect du système de formation à distance. Ce dernier permet de décrire, de déterminer ou d'échanger des contenus, de définir les styles d'interactions ou les formalismes supports à l'interopérabilité entre les systèmes. Au sein de chaque projet, la documentation détaille chacun des trois axes du modèle : l'axe fonctionnel (guide la mise en œuvre), l'axe statique (décrit la structure) et l'axe dynamique (définit son comportement). Le concepteur de ressources formalisant une intention pédagogique se doit de respecter rigoureusement la syntaxe définie par ces différents modèles.

L'ingénierie des systèmes d'information au travers du LOM (Learning Object Metadata) considère l'objet pédagogique comme « une entité, sur support numérique ou non, qui peut être utilisée, réutilisée ou référencée dans une activité de formation assistée par ordinateur » (IEEE/LOM 2002). Or, une démarche qualité ne peut pas réduire un objet à une entité réutilisable mais doit également la considérer comme étant amenée à évoluer en retour d'utilisation. Notre hypothèse est de considérer la structure interne de l'objet comme une source d'interrogation sur son comportement. Les concepteurs soucieux d'appréhender les usages des systèmes de formation à distance sont amenés à modéliser cette structure et les éléments qui la composent.

L'ingénierie à base de composants logiciels, encouragée par les pratiques des développeurs, positionne les artefacts architecturaux au cœur des problématiques de réutilisabilité (Bass, Clements et al. 2003). Elle constitue une première réponse au besoin d'élargir le potentiel d'adaptabilité des composantes d'un système afin de répondre aux contraintes de réingénierie des systèmes informatiques (Demeyer, Ducasse et al. 2003). L'initiative comme le projet

« Abstract Framework » (Smythe 2003) de l'IMS cherchant à effectuer une synthèse des propositions de cadres techniques, comme la spécification SCORM (Sharable Content Object Reference Model) d'ADLNet ou le standard LTSA de l'IEEE, permet d'optimiser et d'automatiser l'intégration de sondes logicielles visant à observer le comportement des composantes d'un EIAH.

L'ensemble de ces propositions de technologies éducatives permettent au concepteur de spécifier ses choix. De plus, ces différents points de vue techno-centrés viennent instrumenter la démarche cyclique dans laquelle les concepteurs s'expriment et où le comportement du système de formation est observé. Les concepteurs sont alors à la recherche d'un cadre leur permettant de comprendre les usages de ces technologies dans un processus global de conception, englobant les phases d'ingénierie et de réingénierie. Les éléments méthodologiques d'une telle démarche de développement d'un système de formation seront ainsi plus facilement identifiés. Notre proposition est d'identifier ce cadre systémique orienté objet et d'analyser son apport pour la communauté de conception d'EIAH. Différentes instances de ce méta-standard sur un système de formation sont présentées dans ce mémoire pour tenter de valider un tel choix.

1.5.3 Méthodologie : Mise en œuvre du cadre ISO/ODP-RM

1.5.3.1 Présentation du cadre ISO/ODP-RM

Le cadre ODP-RM se définit comme un cadre générique permettant de supporter le processus de modélisation d'un système complexe et distribué en demandant aux concepteurs d'instancier sur leur domaine un ensemble de concepts génériques (composition/décomposition d'objets, état et comportement d'un objet, points de vue et correspondance entre points de vue…). Ces concepts sont déclinés par rapport à trois actes de modélisation introduits par le cadre :

- la spécification du système, où les concepteurs raisonnent sur des compositions d'objets et les relations entre ces compositions et les amènent à formuler des exigences sur des langages de spécification ;
- la modélisation du système, qui définit, à différents niveaux d'abstraction, les modèles d'interaction entre objets ;
- la structuration du système, où les différentes structures à implanter dans le système sont définies.

Les concepts définis par le cadre ODP-RM sont regroupés sous la forme de trois ensembles décrivant (1) l'univers du discours, (2) les points de vue sur le système et (3) un méta-modèle regroupant les concepts liés au travail de spécification, de modélisation et de structuration. Le principal intérêt est de pouvoir décrire le contexte et les règles qui régissent les mécanismes de transformations au sein ou entre différents points de vue d'une structure d'un système de formation. Ces trois ensembles correspondent aux trois documents produis par le comité ISO (International Organization for Standardization) :

- le premier document « Overview » présente les différents objectifs du cadre et une première présentation des concepts clés. Ce document illustre l'utilisation des concepts de ce cadre pour répondre aux besoins de conformité par des exemples de développement de standards (ISO/IEC-10746-1 1998).
- le second document « Foundation » définit les concepts génériques liés aux actes de spécification, de modélisation et de structuration d'un système essentiellement informatique (ISO/IEC-10746-2 1996).
- le dernier document « Architecture » définit les différents points de vue sur un système, ainsi que les langages liés à ces points de vue (ISO/IEC-10746-3 1996).

ODP-RM propose cinq *points de vue* sur l'architecture d'un système :

- le *point de vue métier*[8] focalise sur les *objectifs*, le domaine d'application et les *stratégies* de ce système, à l'aide des concepts *rôle*, *communauté*, *processus*, *étape*, *objectif*, *artefact*, *acteur* et *ressource* ;
- le *point de vue informationnel* permet la définition des informations traitées par les différentes ressources systèmes, à l'aide des concepts *schéma dynamique*, *schéma statique* et *schéma invariant* ;
- le *point de vue computationnel* décrit la *décomposition* fonctionnelle d'un système et les *interactions* entre les *interfaces* des différents objets, à l'aide des concepts *signal*, *opération* et *flux* ;
- le *point de vue ingénierie* décrit les moyens mis en œuvre pour que les *objets* du système interagissent, à l'aide des concepts *grappe*, *capsule*, *noyau*, *nœud*, *canal*, *souche* et *éditeur de liens* ;
- le *point de vue technologique* définit les technologies logicielles et matérielles utilisées, à l'aide des concepts *standard implantable*, *implantation* et *informations supplémentaires sur l'exécution des mises à l'essai*.

Ces cinq *points de vue* définis par le cadre ODP-RM et représentés dans la Figure 14 guident le concepteur dans son interprétation de l'architecture du système. Ces *points de vue* sont les réponses de ce modèle de référence pour appréhender la complexité d'un système lors de sa spécification. En outre, dans une perspective de communication d'expertise pédagogique, (Wenger 1987) présentait un tel concept pour diagnostiquer un problème de conception. En changeant de point de vue, des comportements émergent. La tâche de diagnostic, l'analyse didactique d'une session d'apprenant et l'évolution d'une ressource d'apprentissage ou de ses cibles sont ainsi optimisées.

La figure suivante représente les différents actes de transformations du modèle du système au sens d'ODP-RM. Elle montre également que ce cadre n'impose aucun processus de développement où les passages d'un point de vue à un autre seraient imposés. De plus la notion d'abstraction permettant d'isoler la description des fonctions du système de la spécification des détails de mise en œuvre est associée à un ensemble de termes définis dans le document

[8] Tous les termes définis par le cadre ODP-RM sont représentés en italique.

« Foundation ». Cette terminologie est utile pour décrire le travail du concepteur chargé de maintenir la cohérence entre les modèles de spécification des différents *points de vue*.

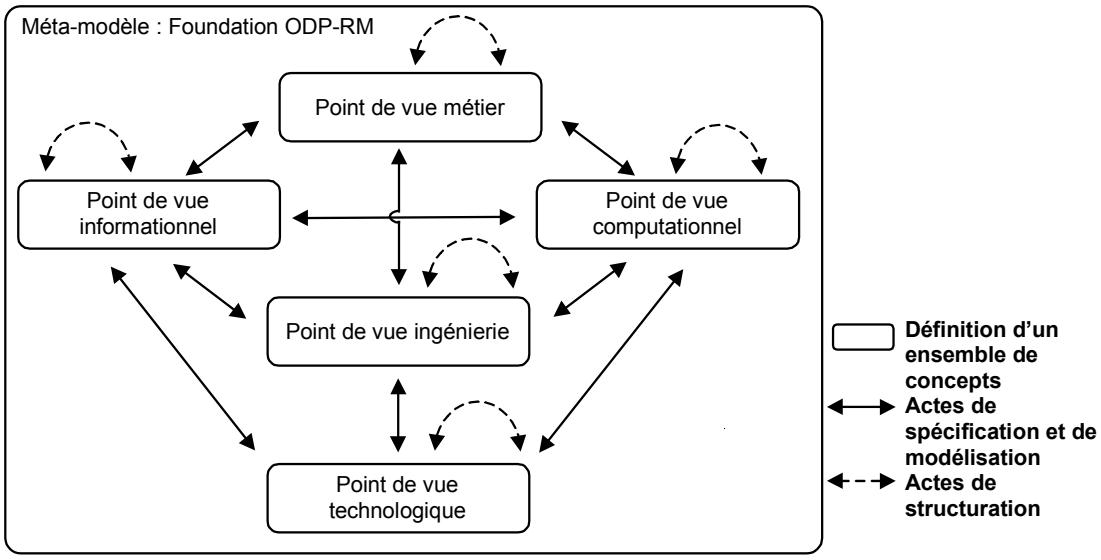


Figure 14 : Cinq points de vue inscrits dans le cadre ODP-RM

1.5.3.2 Mise en œuvre d'un tel cadre

Les projets de standards sur les technologies éducatives considèrent le système de formation comme une structure de traitement de l'information. Notre approche vise à explorer l'explicitation des activités de transformation des modèles, leur interprétation et l'observation des pratiques de la formation. En effet, la dualité de la participation et de la réification dans les communautés de pratique présentées par (Wenger 1998) explicite les relations existantes entre la participation des acteurs à la vie sociale et un processus de réification qui consiste à créer des points de vue autour desquels des négociations de sens peuvent se créer (Varela 1996).

Les standards sur les technologies éducatives et de manière plus globale les technologies des systèmes d'information offrent un ensemble d'outils pour dynamiser les cycles d'observation de retour des usages d'un dispositif de formation. Le cadre que nous avons choisi met à disposition une terminologie spécifique pour décrire et interpréter ces différentes observations. En effet, en considérant les projets sur les standards de technologies éducatives comme des instances de *point de vue*, l'ensemble des langages de spécification, de concepts et de règles du cadre ODP-RM permettent de formaliser les tâches de spécification et de modélisation des composantes d'un système de formation. Les mises en pratique de sessions de formation dans un tel cadre font émerger un ensemble de modèles destinés à être réutilisés.

Notre premier travail a été d'observer les évolutions à la fois des communautés du logiciel libre, diffusant des applications supports aux sessions de formation, et les institutions de normalisation sur les technologies éducatives.

Ensuite, nous avons effectué sur quatre années consécutives des mises à l'essai de sessions de formation sur le site de l'IUT de Laval. Elles nous ont amenées à effectuer trois analyses, correspondant aux trois chapitres suivants de ce document. Le cadre ODP-RM est utilisé comme outil pour guider ces trois analyses. Le chapitre 2 présente les concepts du cadre pouvant être utilisés pour formaliser les différents constats exposés. Le chapitre 3 utilise le cadre ODP-RM pour fédérer les travaux correspondant aux aspects de l'analyse présentée. Le chapitre 4 illustre la mise en œuvre du cadre ODP-RM dans le processus de développement logiciel. Trois instances d'un point de vue spécifique sur un système de formation ont été produites et présentées comme les résultats de nos travaux (cf. Figure 15).

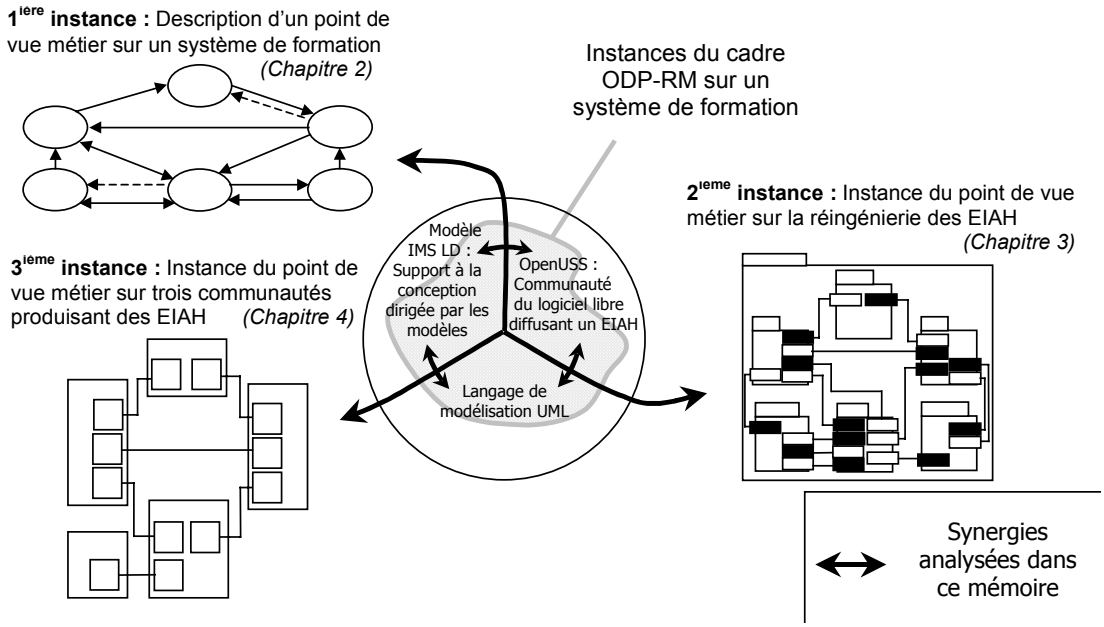


Figure 15 : Propositions de trois instances du cadre ODP-RM

CHAPITRE 2 : ANALYSE DE LA SYNERGIE ENTRE UNE COMMUNAUTE DU LOGICIEL LIBRE ET UNE APPROCHE DIRIGEE PAR LES MODELES

Dans le paragraphe 1.3.4.1 (page 29) du chapitre précédent, nous avons mis en évidence l'existence d'une synergie entre une communauté du logiciel libre et une approche dirigée par les modèles. Nous en avons conclu que chercher à la décrire consiste à analyser les réactions sur les technologies éducatives normées produites par les consortiums IMS et IEEE. Quatre aspects sont analysés et présentés dans ce chapitre :

- Le premier présente les limites des propositions émanant des deux consortiums. Nous considérons la structure à composants LTSA du comité IEEE et un point de vue sur l'organisation du consortium IMS. Ensuite, nous intégrons ces propositions dans le cadre ODP-RM. Le concepteur dispose alors de la terminologie du cadre pour formaliser les limites de ces deux propositions.

- Le deuxième aspect présente une vision élargie des deux propositions en observant les activités mettant en œuvre ces modèles dans les communautés (logiciel libre, scientifique, standardisation sur les technologies éducatives et sur les technologies de l'information). Nous cherchons à identifier une vision unifiée de ces différentes activités dans le cadre ODP-RM. Différents principes de conception d'un système de formation s'en dégagent par la suite.

- Le troisième présente les deux perspectives engagées à la suite de ces deux propositions (approche par réutilisation de composants ou gestion des cas pratiques). Nous étudions les communautés du logiciel libre pour constater que ces différentes perspectives sont déjà opérationnelles sur le terrain. Le cadre ODP-RM nous aide à formaliser le compromis adopté par ces communautés.

- Le dernier analyse la gestion des négociations identifiées dans chacune des propositions. Le modèle LTSA a été défini pour représenter les *processus* de négociation entre l'apprenant et les différents *processus* d'un système de formation. La vision proposée par le consortium IMS est de positionner la négociation au niveau de la gestion projet du consortium. Nous utilisons alors un nouveau *point de vue* défini par l'architecture du cadre ODP-RM sur un système de formation pour gérer les deux types de négociation.

Suivant la perspective d'organisation définie par les théories systémiques, ce chapitre s'interroge sur le potentiel du cadre ODP-RM à expliciter les nouveaux besoins liés à l'utilisation des modèles de spécification sur les technologies éducatives normées.

2.1 Analyse suivant une perspective organisationnelle d'un système de formation

Dans (Le Moigne 1990), l'auteur propose un modèle générique de l'action d'organiser dans un système. De manière transitive et récursive, la perspective sur l'organisation est décomposée en trois actions représentées sur la Figure 16 : se maintenir, se relier et se produire.

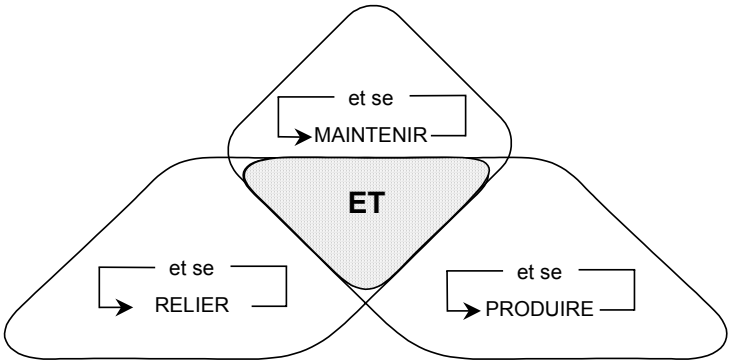


Figure 16 : Modèle générique de l'organisa(c)tion proposé dans [(Le Moigne 1990), page 75]

En s'interrogeant sur ces propriétés qualifiant un système de formation, nous sommes amenés à préciser l'organisation du système sur le domaine de la formation à distance.

Ainsi, l'analyse présentée cherche à identifier à la fois l'impact sur les technologies utilisées pour produire, transformer et faire évoluer un système de formation, et les nouveaux aspects sur une telle organisation. Le cadre ODP-RM est utilisé ici pour apporter de nouveaux concepts nous interrogeant sur leur pertinence dans l'analyse de la synergie entre une communauté du logiciel libre et une approche dirigée par les modèles.

2.2 Nouveaux regards sur les propositions des deux consortiums : IMS et IEEE

Chacun des consortiums produit des modèles. Nous analysons leurs limites, les principes de leurs transformations, leurs capacités à supporter l'innovation et la gestion des négociations. Pour cela, nous allons présenter

les réponses de chacun des consortiums et ensuite nous interroger sur la manière dont elles peuvent être prises en compte dans le cadre ODP-RM.

2.2.1 Les limites des propositions des deux consortiums

2.2.1.1 Les limites du modèle LTSA de l'IEEE

Deux propositions ont été définies comme standard au sein de groupe de travail sur les technologies éducatives normées LTSC (Learning Technology Standards Committee) de l'IEEE : LOM (IEEE/LOM 2002) et LTSA (IEEE/LTSA 2003). Chacune d'elles répond à des objectifs différents. Nous choisissons le modèle à processus LTSA qui positionne le concept d'organisation au cœur de ses préoccupations. De plus, ce modèle a été soumis au processus de standardisation internationale de l'ISO. Nous allons analyser cette proposition au regard des différentes réactions qu'elle a suscitées.

L'objectif du modèle LTSA est double. Le premier est de définir un cadre abstrait organisant et structurant les différentes architectures des dispositifs de formation existantes. Le second est de s'interroger sur les différents enjeux des futures technologies éducatives normées. En effet, en proposant un tel modèle, le comité LTSC souhaitait fédérer les besoins en standards et initier les sous-comités les prenant en charge (Farance 2003). Le standard LOM est un exemple de production de l'un d'eux.

Le modèle LTSA représenté sur la Figure 17 (page 48) est une structure composée de processus, d'unités de stockage et de flux. Ces concepts génériques de représentation sont ceux utilisés dans une démarche d'analyse structurée définie par (Yourdon 1989). Un tel modèle veut être un support de communication pour formaliser les intentions des futurs standards dans les technologies éducatives du comité IEEE. Les processus identifiés sont le siège de transformations des informations et les *flux* le moyen pour les véhiculer. L'une des particularités de ce modèle est de décrire sous la forme d'une boucle de rétroaction le comportement d'un dispositif de formation (Farance et Tonkel 1998). Une telle vision a eu pour conséquence de dissocier les composants effectuant les traitements et les données stockées et réutilisées.

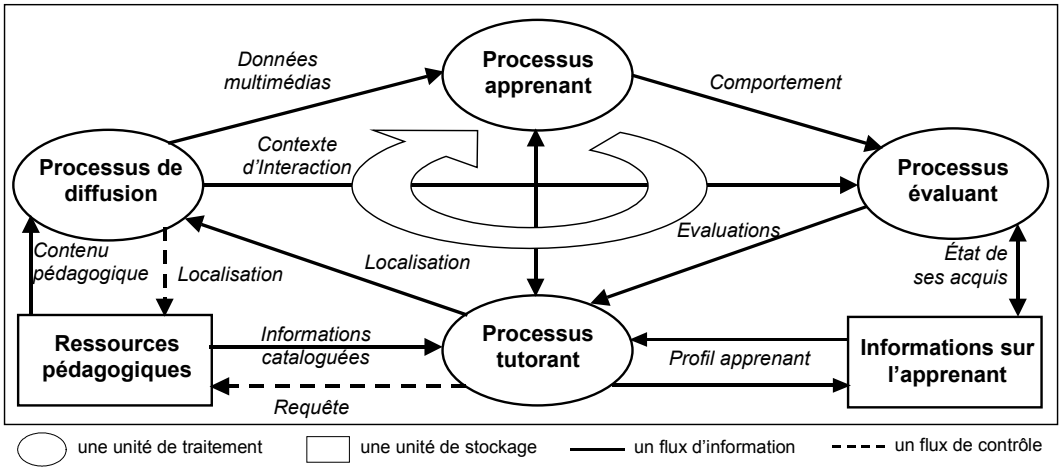


Figure 17 : Cycle du modèle à composants du système LTSA

Ce modèle a été promu au rang de standard au sein du comité IEEE et a été soumis au groupe de normalisation international SC36 géré par l'ISO. Cette proposition concerne deux des six thèmes du sous-groupe : le premier traite de l'assurance qualité et de l'identification d'un cadre descriptif et le second porte sur la gestion et la diffusion. Trois principales limites de ce modèle ont été relevées par (Blandin 2003; Lindner 2003) :

- il se centre uniquement sur les aspects techniques ;
- il se limite à deux unités de stockage (cf. Figure 17) ;
- et il interdit la création ou la mise à jour d'une ressource pédagogique.

Le modèle LTSA n'a pas eu le succès escompté pour la communauté des technologies éducatives normées. Mais ce travail est à considérer pour les critiques qu'il a suscitées. Celle que nous avons plus particulièrement relevée est de considérer une ressource pédagogique comme un *objet* finalisé et non comme un prototype susceptible d'évoluer à la suite d'utilisations (Lindner 2003; Corbière et Choquet 2004a).

La vertu du modèle LTSA n'est pas de proposer une structure universelle mais d'être un référent dans les discussions au sein des institutions de normalisation, scientifiques (Avgeriou, Papasalouros et al. 2003) et industrielles (Smythe 2003).

2.2.1.2 Les limites du processus de développement de chaque projet de spécification du consortium IMS

Nous présentons les différents objectifs des modèles produits par le consortium IMS pour nous intéresser par la suite aux différents modèles à processus qui définissent son organisation et plus spécifiquement à son processus de développement.

Le consortium IMS est une initiative cherchant à fédérer un ensemble de projets de normalisation sur les langages de spécification informationnelle traitant des différents aspects d'un système de formation. En les présentant ainsi sur un même plan, les différents concepteurs sont amenés à choisir, à comprendre et à utiliser ces différents modèles afin d'expliciter leurs décisions.

L'ensemble de ces modèles se définit comme étant indépendants d'un domaine de formation précis, d'une méthode pédagogique et d'un modèle d'apprentissage. Pour reprendre le cadre d'architecture d'un tuteur intelligent présenté sur la Figure 3, page 14, Etienne Wenger identifie en son cœur un « modèle de connaissances communicables » ayant le rôle d'organiser l'articulation entre les différentes expertises formalisées avec leur représentation en session d'apprentissage. Aujourd'hui, le consortium IMS a la volonté de faire jouer ce même rôle aux modèles de spécifications qu'ils diffusent librement aux concepteurs souhaitant les adopter. L'apport immédiat est de définir un vocabulaire commun entre le concepteur utilisant un tel modèle pour spécifier son scénario d'apprentissage, l'apprenant l'utilisant et le concepteur développant le composant logiciel l'interprétant.

Afin d'exposer les limites du processus de développement des projets du consortium IMS, il est nécessaire de disposer d'une vision globale de l'organisation du consortium IMS. Nous considérons alors les trois modèles à processus suivants :

- Le premier modèle décrit le processus cyclique de développement produisant, diffusant et gérant l'évolution des spécifications du consortium[9]. Seuls les membres de la communauté peuvent y participer (cf. Figure 18, figure 50).
- Le second définit le processus au cœur duquel deux acteurs principaux sont identifiés : le concepteur qui manipule les primitives des différents modèles pour spécifier la session d'apprentissage et l'apprenant qui la met en œuvre. Seule la contrainte d'interopérabilité syntaxique entre les systèmes de gestion de formation est prise en compte.
- Et le dernier définit la décomposition fonctionnelle des outils logiciels interprétant une telle spécification, l'un des trois documents décrivant chaque modèle de spécification lui est alors spécifiquement consacré. Son intitulé est « Best Practice and Implementation Guide ». Il s'adresse à la fois au concepteur souhaitant appréhender les exemples de mises en application du modèle et à l'informaticien cherchant les premières éléments méthodologiques de conception de ce composant.

[9] Voir, http://www.imsglobal.org/background.html

Le processus de développement mis en œuvre par chaque projet de consortium IMS, part d'un besoin d'interopérabilité exprimé par l'industrie de la formation. Chaque groupe de travail prenant en charge un projet décompose son processus en quatre étapes. Comme le présente la Figure 18, à chaque phase du développement plusieurs cycles itératifs sont effectués afin que chaque proposition développée soit implantée, testée et validée par les seuls membres du consortium.

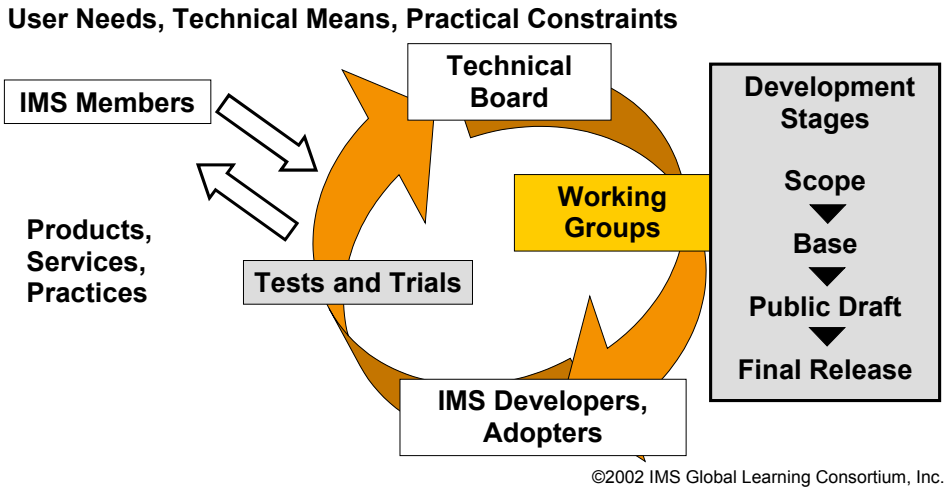


Figure 18 : Processus de développement de chaque projet de spécification de l'IMS

Les différents modèles diffusés par le consortium IMS formalisent uniquement les considérations techniques. Dans [(Rabardel 1995), page 60], l'auteur attire notre attention sur le danger de considérer ces modèles seulement aptes à décrire les logiques dites fonctionnelles. Il préconise de prendre en compte la perspective instrumentale de ces modèles. Ce sont leurs usages qui doivent être positionnés et décrits au cœur du modèle d'organisation d'un système de formation.

2.2.1.3 L'intégration de ces propositions dans le cadre ODP-RM : une réponse à ces limites

Les limites de ces deux propositions, le modèle LTSA du comité IEEE et les modèles de spécification du consortium IMS, sont similaires à celles définissant le travail de modélisation dit « Analytique ». Les phénomènes à modéliser par le concepteur se réduisent aux combinaisons d'*objets* constituant le système de formation. A contrario, la modélisation dite « systémique » cherche à modéliser l'action d'un système considéré alors comme complexe.

« La caractérisation d'une action ou d'une fonction peut se faire récursivement : elle passe commodément par la notion générale de processus. On définit un processus par son exercice et son résultat : il y a processus lorsqu'il y a, au fil du temps T, la modification de la position dans un référentiel "Espace-Forme", d'une collection de "Produits" quelconques identifiables par leur morphologie, par leur forme F donc. » [(Le Moigne 1990), page 46].

Les concepts d'*action*, de *comportement* et d'*état* du cadre ODP-RM sont alors utilisés pour décrire un tel processus. Ils correspondent à l'acte de modélisation, qui lui-même appartient aux catégories générales du cadre ODP-RM (ISO/IEC-10746-2 1996). Ainsi, suivant un tel cadre, le *rôle* donné aux propositions des différentes institutions de normalisation n'est pas de contraindre les futurs utilisateurs mais de supporter des négociations sur l'acte de spécification, de structuration ou de modélisation. Afin de formaliser ces actes, le cadre ODP-RM propose un cadre d'architecture structuré autour de cinq *points de vue* (cf. paragraphe 1.5.3.1). Il nous met à disposition les concepts de *point de vue* et permet de décrire les *langages* utilisés dans chacun d'eux.

En reprenant le modèle à processus LTSA du comité IEEE, nous pensons qu'il est nécessaire d'intégrer cette proposition dans le cadre ODP-RM. Nous identifions l'un des cinq *points de vue* qui exprime le mieux les préoccupations de ce modèle. Les objectifs exprimés dans (IEEE/LTSA 2003) et la méthodologie d'analyse utilisée (Yourdon 1989) correspondent aux préoccupations du *point de vue computationnel* qui cherchent à décomposer les fonctionnalités du système en proposant une structure à *objets* en *interaction*. En effet, l'objectif de la proposition LTSA est d'identifier ces différentes *interfaces* afin d'expliciter les besoins de standardisation (Farance 2003). Le concepteur peut alors formaliser les actes l'amenant à articuler cette proposition avec d'autres *points de vue* du cadre ODP-RM ou à créer des *instances* définies à partir d'un tel modèle.

Les trois modèles à processus que nous avons identifiés au sein du consortium IMS sont (1) le processus de développement identifiant les différents besoins formulés par l'industrie, produisant, adaptant et validant un document de spécification en interne et le diffusant (cf. Figure 18, page 50), (2) le processus guidant le concepteur dans la description de son scénario, et (3) le processus définissant le comportement de chaque composante d'interprétation du système. Nous nous intéressons au premier modèle, considéré comme le plus abstrait. Il correspond au processus de création de standard et partage les mêmes objectifs que le cadre ODP-RM (ISO/IEC-10746-1 1998) : définir des formalismes pour supporter l'interopérabilité syntaxique entre les systèmes distribués. Cela nous permet de déceler des différences dans leur intention. Pour le consortium IMS, les travaux s'inscrivent dans une vision restrictive et économique en réduisant la négociation aux seuls membres des différents projets. Les retours à la suite de leur diffusion ne viennent que compléter les bases de cas d'utilisation du consortium[10]. La vision du cadre ODP-RM sur la création d'un standard diffère. Elle se définit comme une coordination entre les concepteurs, intégrant des propositions d'un simple utilisateur ou de consortiums. En effet, au lieu de limiter le nombre de standards, la terminologie et la structure proposées par le cadre ODP-RM permettent de communiquer les réflexions des concepteurs sur des propositions *a priori* identiques et de disposer d'une terminologie appropriée pour les décrire.

[10] Voir, http://www.imsglobal.com/usescases et http://www.unfold-project.net:8085/UNFOLD/

2.2.2 Deux consortiums, deux manières de transformer les modèles

2.2.2.1 La communauté du logiciel libre : un regard supplementaire sur la transformation de modèle définie par LTSA

Nous allons nous intéresser plus particulièrement à la proposition LTSA du comité IEEE. Le modèle proposé est présenté comme un standard non pérenne et la documentation le décrivant expose différents principes de transformation du modèle à composants abstraits sur des dispositifs existants. Nous constatons que la communauté du logiciel libre adopte une approche similaire. La communauté OpenUSS à laquelle nous avons adhéré illustre ces différents principes de transformation.

Comme il est précisé dans le document de présentation de LTSA (IEEE/LTSA 2003), « ce standard fournit un cadre pour […] incorporer les évolutions techniques de 5 à 10 ans ». Le modèle abstrait LTSA a été approuvé comme standard par le comité IEEE. Afin de le valider, la documentation présente différentes *instances*. Elle montre ainsi la complexité d'un modèle de coordination et de pilotage d'un processus d'apprentissage.

En précisant dans l'annexe du document de présentation de LTSA que ce travail s'inscrit dans le cadre générique de description d'architecture (IEEE/ADS-IS 2000), les avantages d'une telle perspective sur un système de formation sont présentés dans [(Bass, Clements et al. 2003), page 26]. Les enjeux définis sont :

- la communication par des points de vue afin de créer des consensus, des négociations et une compréhension mutuelle des aspects techniques ;
- la compréhension des solutions techniques visant à explorer leurs avantages et leurs inconvénients ;
- et le partage de structure du modèle et des éléments qui le composent afin d'expliciter un nouveau besoin fonctionnel ou de nouveaux attributs qualitatifs.

En effet, le travail de transformation ayant produit différentes *instances* du modèle LTSA consiste à effectuer des regroupements et à leur attribuer un sens spécifique. Ce niveau sémantique est plutôt technique mais la terminologie utilisée pour décrire ces *instances*, est de nature à susciter de l'intérêt auprès des concepteurs d'un système de formation et d'initier de nouvelles transformations.

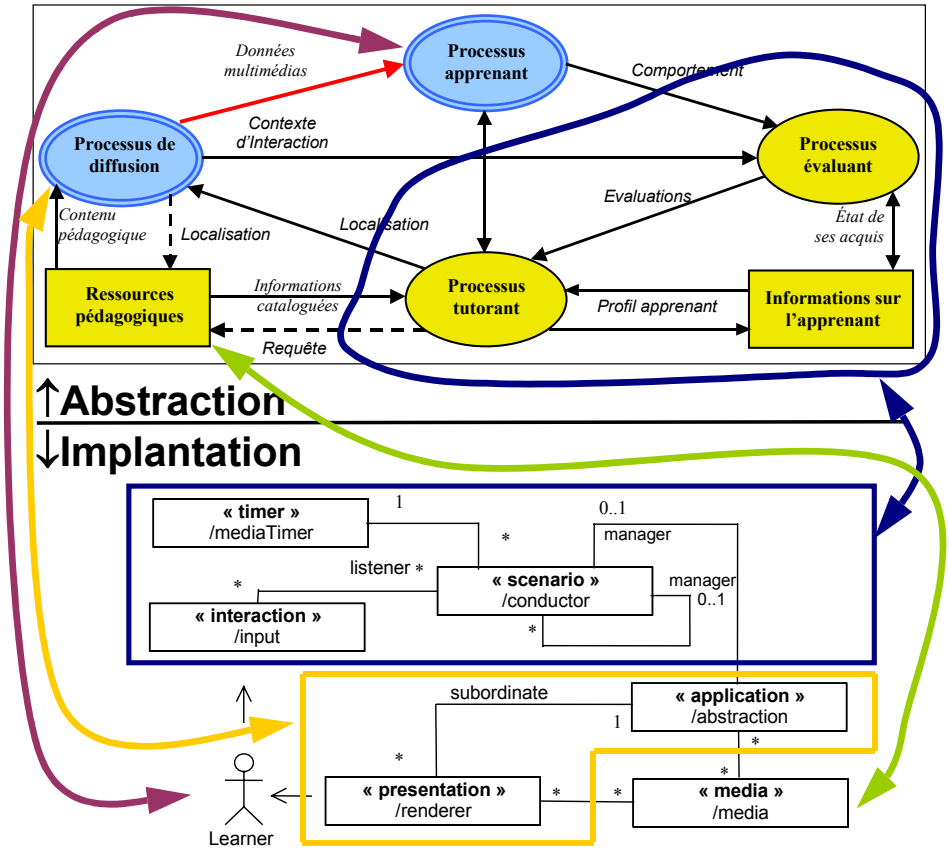


Figure 19 : Exemple d'une mise en relation des composants du modèle LTSA avec le profil ingénierie d'une application multimédia

L'exemple représenté sur la Figure 19 présente le profil d'ingénierie d'une application orientée *objet* dédiée au multimédia (Sauer et Engels 2001). Ce modèle est à mettre en correspondance avec l'*instance* du modèle à composants LTSA décrivant le cas spécifique de la diffusion d'une ressource multimédia. Un tel point de vue sensibilise le concepteur sur le fait que différents éléments informationnels (audio, vidéo, graphique, texte, etc …) sont gérés par le processus de diffusion et transmis au processus apprenant. Cette mise en relation engage les concepteurs à négocier sur un tel modèle.

De même, la communauté du logiciel libre illustre parfaitement ces propos. Le modèle logique représenté par la Figure 20 (page 54) présente les différents cadres techniques développés ou réutilisés par le projet communautaire OpenUSS (Open University Support System) produisant une plate-forme de formation à distance. Il affiche ainsi les composants techniques utilisés par cette communauté et structure l'ensemble des opérations effectuées par l'équipe de développement lors de la spécification de chacun d'eux (Johnson 1992). Dans l'esprit d'une communauté du logiciel libre, un tel diagramme permet de communiquer et d'initier des réflexions du *point de vue technologique* afin de gagner l'adhésion de nouveaux membres.

Nous pouvons également souligner que certaines initiatives issues des communautés du logiciel libre sont prises

en compte au sein même des réflexions menées par les consortiums à l'origine de ces propositions de technologies normées. L'exemple de la plate-forme de formation à distance OpenUSS (Grob, Bensberg et al. 2004) reconnue au même titre que le modèle LTSA comme une proposition influente du projet « Abstract Framework » du consortium IMS peut être cité.

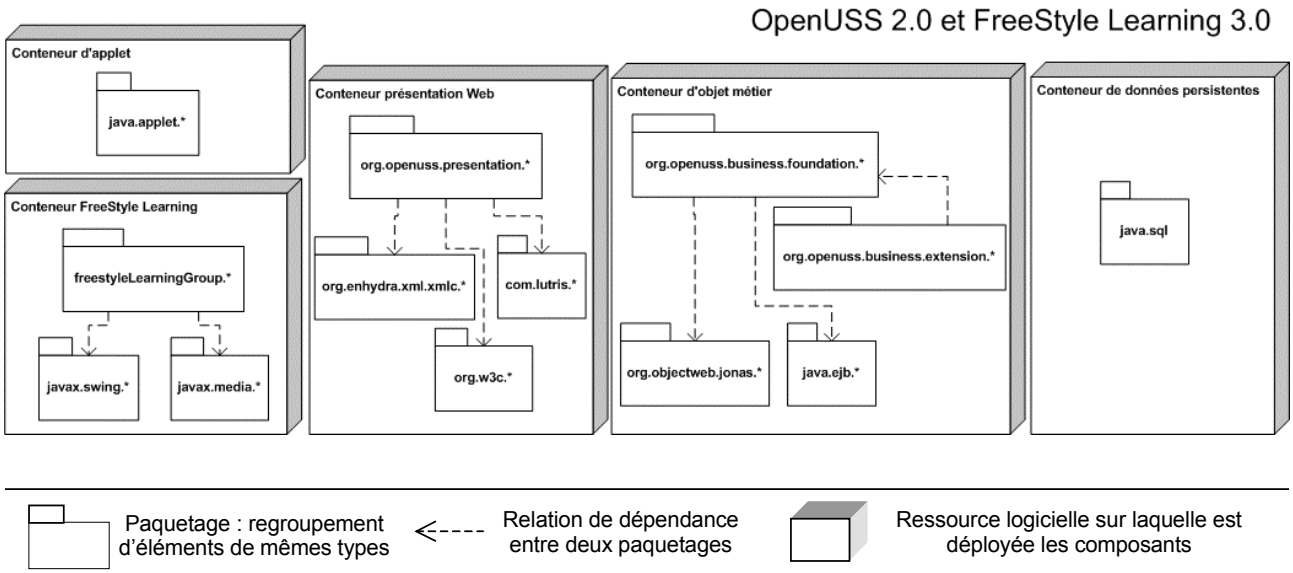


Figure 20 : Modèle logique de conception du projet de la plate-forme OpenUSS

Au niveau du modèle d'architecture abstrait proposé par le comité IEEE et du modèle d'architecture technique choisi par une communauté du logiciel libre, nous avons montré que la volonté n'est pas d'imposer une vision unique. Mais, ces modèles initient la communication entre concepteurs en définissant un vocabulaire commun et des structures communes ayant démontrés leur potentiel dans les récentes réflexions actuellement engagées sur ces technologies.

L'objectif de la proposition du comité IEEE est d'aboutir à une harmonisation des spécifications sur les architectures, les structures et les comportements des différents composants d'un système informatique supportant des formations à distance. Il rejoint les préoccupations des travaux actuels sur la spécification SCORM (ADL/SCORM 2004a) du réseau ADLNet et d'un des projets du consortium IMS « Abstract Framework » (Smythe 2003). Ces travaux s'adressent directement à l'industrie de la formation qui est en passe de créer un nouveau secteur économique. Les communautés du logiciel libre, en les utilisant et en les implantant dans le développement des plates-formes et des composants de formation dans un esprit d'ouverture, en font un autre usage. En effet, en donnant accès aux ressources du projet, le concepteur est amené à les utiliser afin de favoriser le prototypage ou la mise en œuvre rapide de sessions d'apprentissage. Le concepteur effectue des actes de transformation sur les modèles de ces ressources et a besoin de les communiquer pour engager des négociations.

2.2.2.2 Trois transformations observées du modèle IMS LD

En observant les évolutions des différents projets de spécification du consortium IMS, nous pouvons constater la création régulière de nouvelles versions ainsi que la définition de nouveaux projets. En observant un tel fait, nous nous intéressons aux moyens prévus par le consortium pour transformer ces modèles. Malheureusement en se cantonnant au seul processus de développement du consortium IMS, ces transformations sont principalement de nature technique. Nous cherchons alors à observer d'autres transformations car nous sommes conscients que les enjeux de création et d'innovation pédagogique sont bien réels. Nous présentons trois exemples de transformation.

Le premier nous vient des communautés de pratique scientifique qui utilisent et adaptent les formalismes diffusés par ce consortium. Elles induisent de nouvelles propositions. Nous pouvons en citer trois portant sur le modèle « Learning Design » du consortium IMS :

- le travail de (Hernández-Leo, Asensio-Pérez et al. 2004) qui propose l'adaptation d'un tel modèle (cf. Figure 21, page 55). Ils ont ajouté une nouvelle primitive « groupservice » afin que le concepteur puisse spécifier avec plus de précision l'élément « service » du modèle « Learning Design ».
- les études portant sur les pratiques de terrain qui constatent les limites de ce modèle (El-Kechaï et Choquet 2005; Ferraris, Lejeune et al. 2005; Nodenot 2005).
- le travail de (Barré et Choquet 2005), dans le cadre du projet REDiM, où les auteurs utilisent le schèma décrivant le modèle « Learning Design » afin de guider le concepteur souhaitant décrire les traces d'observation du *comportement* d'une session d'apprentissage.

Ces travaux montrent ainsi la nécessité de prendre en compte de nouvelles intentions non prévues par le modèle de spécification d'origine.

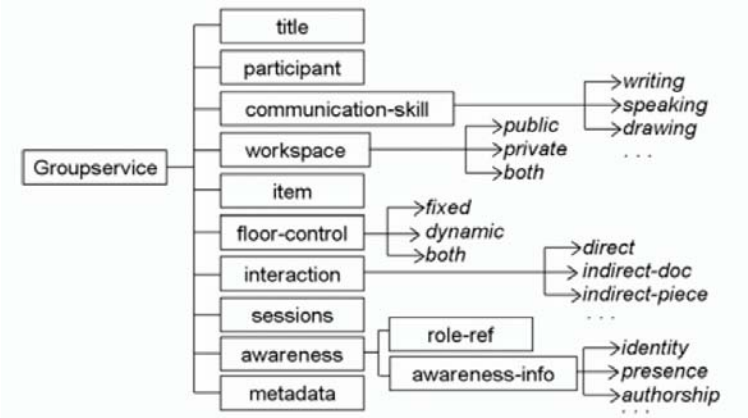


Figure 21 : Extension de l'élément « service » du schéma XML « Learning Design » (Hernández-Leo, Asensio-Pérez et al. 2004)

Le deuxième exemple reprend la volonté du consortium IMS de fédérer une organisation sur différents projets. Nous prenons l'exemple du projet de modèle informationnel « Simple Sequencing ». Ce dernier permet de préciser les séquencements des différents éléments qui composent un scénario pédagogique et les règles qui contrôlent son déroulement. Par la suite, ce modèle a été intégré dans la spécification SCORM du réseau ADLNet et a permis de révéler ses limites à spécifier le stockage et le partage de nouvelles données. En réponse le consortium IMS a produit une nouvelle spécification, « IMS Shareable State Persistence Information Model » (Jackl, Panar et al. 2004). La transformation présentée dans cet exemple nous montre que le consortium IMS est réactif aux besoins révélés par d'autres communautés mettant en œuvre ses modèles.

Le dernier exemple présente la réponse apportée par le consortium IMS afin de prévoir l'évolution du modèle de spécification « Learning Design » (Koper, Olivier et al. 2003b). En effet, afin de prendre en compte les difficultés techniques à interpréter un tel modèle pour un système de gestion d'apprentissage, trois déclinaisons du même modèle ont été prévues (niveau A, niveau B et niveau C). Derrières ces considérations technologiques des intentions pédagogiques existent. Elles doivent être formalisées et diffusées mais ne rentrent pas dans la préoccupation actuelle du consortium.

Nous venons de présenter trois exemples de transformation sur l'un des modèles proposé par le consortium IMS « Learning Design ». Ce modèle supporte l'activité des communautés scientifiques, la coopération entre les comités produisant des standards sur les technologies éducatives et l'activité liée au développement des composantes logicielles d'une plate-forme de formation à distance. Le consortium IMS apporte une réelle contribution en gérant et en diffusant librement des consensus sur les modèles de spécification mais il montre ses limites à fédérer les nombreuses transformations qu'elles suscitent sur le terrain.

2.2.2.3 ODP-RM : un cadre pour décrire les actes de transformation des concepteurs

Nous présentons les principes de transformation des modèles définis par le cadre ODP-RM. Ensuite, nous illustrons sur un exemple la capacité d'un tel cadre à expliciter les transformations des deux propositions sur les technologies éducatives normées et à fédérer les mises en pratique de composantes d'un système de formation développées par les communautés du logiciel libre.

Le cadre ODP-RM se présente en deux parties. La première, intitulée « Foundations » (ISO/IEC-10746-2 1996), dans laquelle sont définis les concepts liés aux tâches d'analyse du système. La seconde, intitulée « Architecture » (ISO/IEC-10746-3 1996), dans laquelle sont définis les moyens pour le spécifier. L'objectif du cadre ODP-RM est de manipuler différents aspects d'un système. Certains sont pertinents aux yeux des utilisateurs ou des concepteurs, d'autres aux deux. Des détails définissant les technologies utilisées jusqu'aux services rendus par le système, peuvent

être masqués. Les concepts et les directives fournis par le cadre ODP-RM permettent d'abstraire les comportements du système et ainsi de dégager les concepts clés. Ces derniers précisent des aspects particuliers ou décrivent les transformations les mettant en relation.

En conséquence, l'ensemble des concepts du cadre ODP-RM permet de décrire les activités de transformation des modèles. Les modèles diffusés par les deux consortiums sur les technologies éducatives présentent des aspects distincts et produisent des langages de spécification correspondant aux deux *points de vue* définis par le cadre ODP-RM : le *point de vue informationnel* et le *point de vue computationnel*.

Nous prenons un exemple tiré de nos mises à l'essai effectuées sur le site de l'IUT de Laval. L'objet est de décrire le travail de modélisation du concepteur souhaitant intégrer une vidéo dans son scénario pédagogique. Cet exemple implique trois des cinq *points de vue*. Le cadre permet de décrire les actes de modélisation du concepteur au sein de chacun d'eux. Contraindre le concepteur à formaliser ces différents actes lui permet de diffuser à la communauté le raisonnement qu'il a mené et d'engager des réflexions sur de nouvelles mises à l'essai ou sur le choix des modèles pour les expliciter.

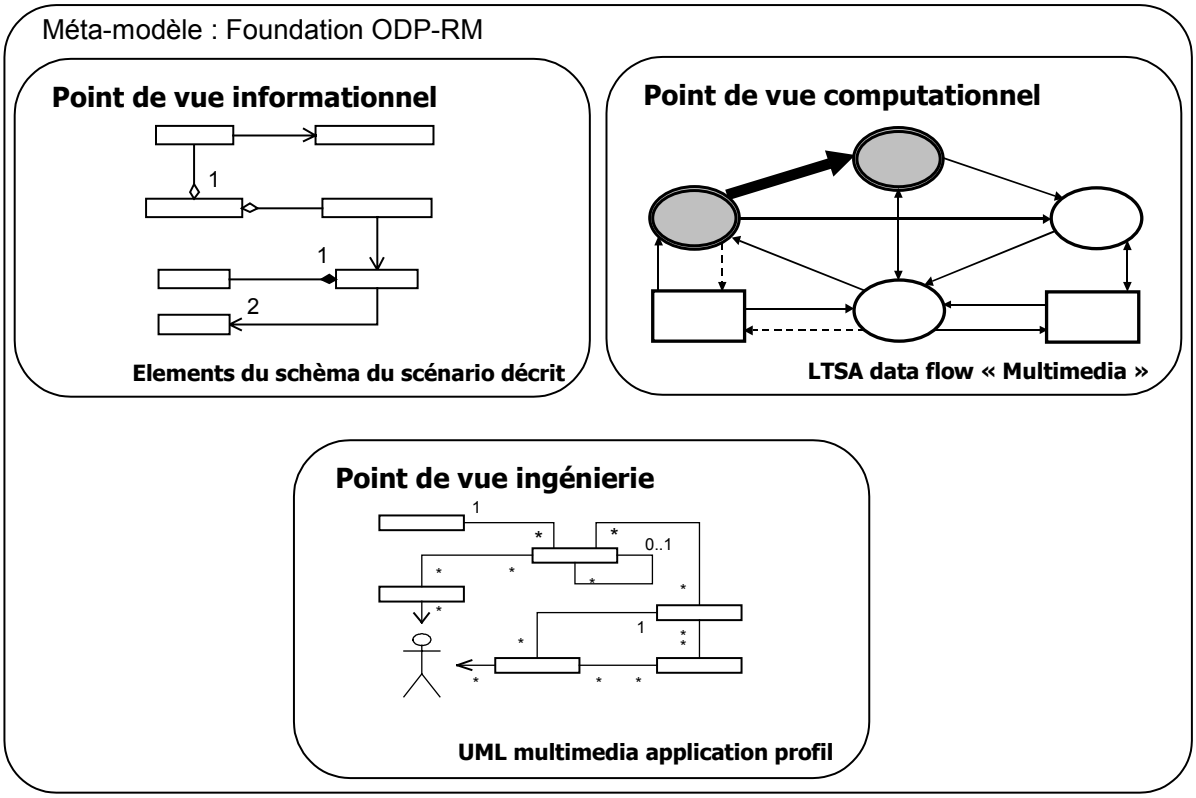


Figure 22 : Trois points de vue de l'intégration d'une vidéo dans un scénario

Nous considérons que toute démarche de conception est mise en œuvre dans le cadre ODP-RM. Chacune des mises à l'essai utilise les propositions sur les technologies éducatives normées et les composantes diffusées par la

communauté du logiciel libre. Le concepteur est amené à effectuer un travail de modélisation. Dans le cadre ODP-RM, il définit les sondes logicielles afin de tracer le *comportement* des composantes du système de formation. Le cas présenté ici illustre l'analyse du concepteur ayant selectionné le composant chargé de diffuser une ressource vidéo à l'apprenant. Les *traces* générées sont présentées au concepteur sous la forme de données statistiques.

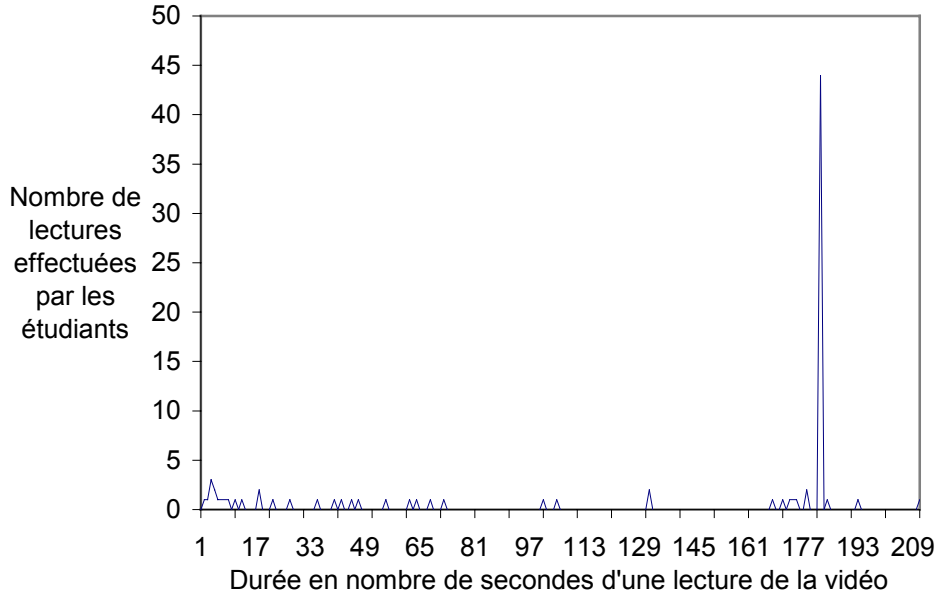


Figure 23 : Résultats statistiques sur la durée de visualisation par 50 lectures de vidéo effectuées par les étudiants de l'IUT de Laval

Une simple analyse statistique des *traces* montre que concepteur doit considérer la ressource comme un *objet* non sécable. En effet, sur 50 lectures d'une ressource vidéo d'une durée de 183 secondes, 45 lectures se sont effectuées sur la durée entière de la ressource vidéo. Guidé par ces *traces* en retour d'exploitation, le concepteur est alors amené à effectuer des actes de transformation sur différents modèles. Le nouveau modèle produit par le concepteur l'amène à structurer les différentes *instances* des modèles diffusés par le consortium IMS, le profil matériel décomposant les principaux éléments techniques du composant et le ou les *instances* du modèle LTSA correspondant à ce processus d'apprentissage. Comme le montre la Figure 22, page 57, l'architecture du cadre ODP-RM nous permet d'associer à chaque *point de vue* chacun des modèles manipulés par le concepteur. La mise en relation du *point de vue computationnel* avec le *point de vue ingénierie* permet de modéliser le comportement des *objets* tracés (cf. Figure 19, page 53). Deux *instances* du modèle LTSA sont alors considérées pour le même *point de vue computationnel* : la première considère l'environnement d'apprentissage comme un simple diffuseur de ressources et la deuxième considère l'environnement d'apprentissage comme un support aux activités de recherche d'information. Ainsi, le concepteur est amené à s'interroger sur le comportement de l'apprenant à considérer. Cela lui permet de modéliser avec plus de précision l'environnement d'apprentissage centré sur l'utilisation de la ressource vidéo de type multimédia. En effet, les

deux *instances* du modèle LTSA amènent respectivement à considérer la ressource visualisée soit comme un élément non sécable ou comme une composition d'éléments autour desquels sont révélés de nouveaux usages.

Cet exemple montre que l'usage des technologies éducatives normées est le siège de transformations engageant des négociations de nature « technocentrée » avec des enjeux pédagogiques sous-jacents. Le cadre ODP-RM se présente comme un instrument apte à formaliser les actes de transformation des modèles des concepteurs d'un système de formation.

2.2.3 Deux consortiums, deux perspectives d'évolution différentes

2.2.3.1 L'approche par réutilisation de composants : une perspective d'évolution du modèle LTSA

Afin de dégager une perspective d'évolution du modèle LTSA, nous cherchons à analyser les similitudes et les différences entre le modèle LTSA proposé par le comité IEEE et le projet d'une démarche qualité mettant en œuvre les technologies éducatives (Pawlowski 2002b). Nous identifions deux points communs entre ces deux approches.

Le premier point commun est la représentation cyclique des deux approches. La première proposition définit un cycle centré sur la ressource pédagogique, composé de quatre processus : diffusion, apprenant, tutorant et évaluant. La seconde propose un cadre d'analyse permettant de définir une approche qualité (Pawlowski 2002b) sur un processus global de formation (cf. Figure 24, page 60). Cette approche utilise le modèle de cycle de vie inspiré des cycles d'ingénierie du développement logiciel, en neuf phases : analyse, conception, développement, test, réalisation, usage, évaluation, validation et terminaison.

Le deuxième point commun est de présenter ces deux approches à un niveau d'abstraction suffisant pour engager l'adhésion des concepteurs. Le modèle LTSA est une architecture qui définit « l'organisation fondamentale d'un système représentée d'une part, par ses constituants, leur inter-relations, leurs relations avec l'environnement et d'autre part par les principes guidant sa conception et son évolution[11] » [(IEEE/ADS-IS 2000), partie 3.5]. L'utiliser pour analyser un système de formation existant amène le concepteur à reconnaître les processus et les interfaces de ce modèle référent. Un tel instrument supporte la communication dans une communauté de concepteurs. Par ailleurs, sur la base du travail de (Pawlowski 2002b), le sous-groupe SC36 de l'ISO a initié en septembre 2002 un nouveau thème cherchant à définir un cadre général de réflexion pour intégrer les différentes approches qualité (ISO/IEC-19796 2004) proposées par les communautés industrielles et scientifiques (cf. Figure 24, page 60). Ce cadre définit un modèle de cycle

[11] Traduction de « The fundamental organization of a system embodied in its components, their relationships to each

indépendant du domaine de la formation. Il montre très clairement le souhait de prendre en compte deux dimensions : (1) au niveau du processus de développement et (2) au niveau de son utilisation.

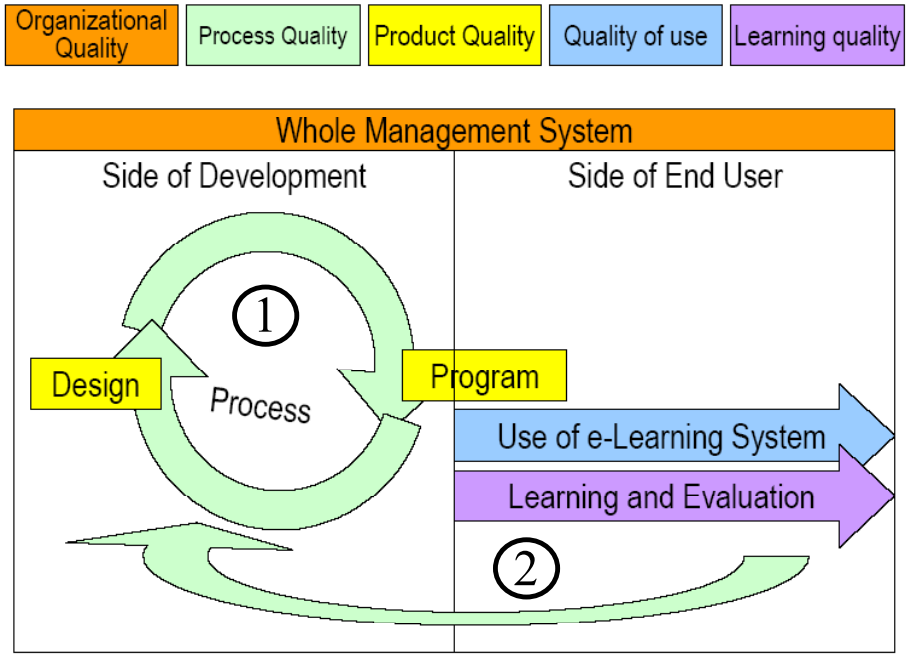


Figure 24 : Contribution du groupe japonais (Pawlowski 2005)

Cependant, les activités qui le composent sont liées aux spécificités du domaine de la formation. De plus, la vision de ces deux approches tend à réduire l'organisation des acteurs d'une formation à des comportements normés et réduit d'autant les perspectives d'évolution.

Une autre vision nous est offerte par les pratiques des architectures à composants logiciels dans un système d'information. Une première réflexion menée par le consortium IMS (Smythe 2003) aboutit à la proposition d'une première architecture abstraite d'un système de formation (cf. Figure 25, page 61). Dans (Barbier, Cauvet et al. 2004), les auteurs nous présentent l'intéret d'un tel travail. L'objectif est d'arriver à formaliser les composants architecturaux et leurs extensions pour un système d'information. Il permet ainsi de supporter et de partager les raisonnements des concepteurs sur les propriétés d'une telle architecture. Les auteurs entendent par propriété : les protocoles d'interaction, la conformité avec d'autres architectures, évolution, … [(Barbier, Cauvet et al. 2004), page 12]. De même dans [(Bass, Clements et al. 2003), page 12], le concepteur est alors placé au cœur d'un cycle rétroactif appelé « Architecture Business Cycle » où l'architecture du système est considérée comme l'un des premiers principes du développement d'un ou des systèmes. Chaque cycle se base sur l'identification pour une organisation des effets organisationnels et

---

other, and to the environment, and the principles guiding its design and evolution »

expérimentaux du développement d'une architecture et sur leurs réinvestissements dans les stratégies des métiers de l'évolution du projet.

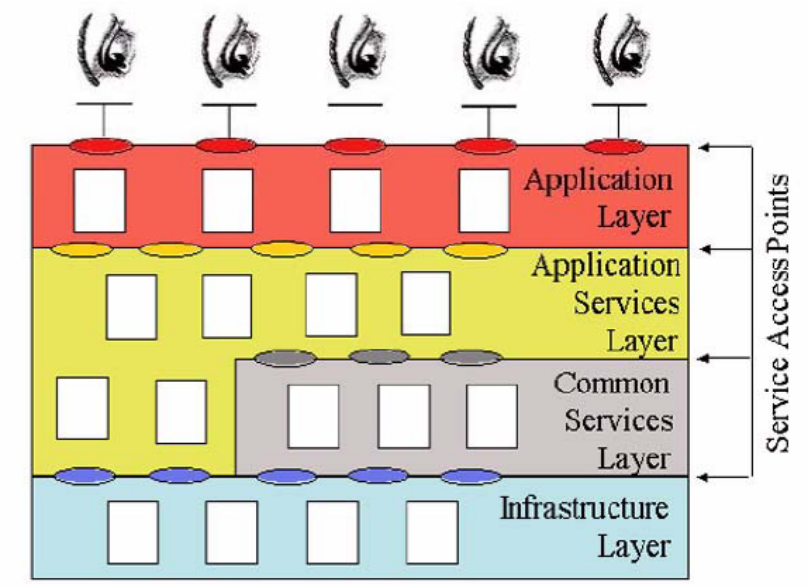


Figure 25 : Proposition d'un cadre abstrait sur l'architecture orientée service d'un système de formation [(Smythe 2003), figure 3.5]

En adoptant des systèmes basés sur des composants, le processus de conception veut être à la fois itératif, centré sur l'architecture et dirigé par la modélisation et la réutilisation. La structure d'un tel système est décrite par les différentes vues. De plus, les aspects statique et dynamique de ces structures logicielles spécifient les besoins auxquels répond le système. À partir d'une telle vision, le concepteur se focalise sur une partie du système en l'affinant et en créant ou en adaptant les composants qui répondent à ses exigences fonctionnelles.

En exemple, nous présentons le projet OpenUSS. Son objectif est de mettre à disposition un système d'administration centralisé visant à assister les universités, les différents acteurs et de produire une *interface* de référence pour l'intégration de nouvelles fonctionnalités. Une approche par conception orientée composants est alors adoptée. Deux types de composants sont alors définis : le composant de base intitulé « Foundation » et les composants assurant les services intitulés « Extension Component ». De plus la particularité de cette plate-forme est d'intégrer des logiciels produits par des communautés du logiciel libre sous la forme de composant respectant les directives du modèle de développement intitulé EJOSA[12] (Enterprise Java Open Source Architecture).

[12] Voir, http://ejosa.sourceforge.net/

Ce dernier exemple illustre parfaitement notre perspective d'évolution initiée par le standard IEEE : LTSA où l'architecture est vue comme « l'organisation d'un système représenté d'une part, par ses constituants, leurs interrelations, leurs relations avec l'environnement et d'autre part par les principes guidant sa conception et son évolution[13] » [(IEEE/ADS-IS 2000), partie 3.5].

2.2.3.2 La gestion des cas pratiques : une perspective d'évolution des modèles IMS

Les modèles de spécification produits par le consortium IMS offrent un potentiel de réflexivité en supportant les actes de modélisation et de simulation. Mais cela ne suffit pas pour garantir une approche globale sur un système de formation. Les concepteurs, conscients de ces limites, s'orientent vers des techniques et des formalismes permettant de représenter et de partager leurs raisonnements (Koper 2004). En effet, les modèles produits actuellement par le consortium IMS définissent uniquement la syntaxe des documents que les concepteurs pédagogiques sont amenés à respecter. La perspective envisagée est de considérer l'apprenant en auto-formation et de l'assister par un réseau d'agents logiciels dans lesquels est intégré le raisonnement formalisé du concepteur. Cette perspective applique ainsi le modèle technique défini par le web sémantique (Berners-Lee, Hendler et al. 2001).

Elle décompose la démarche en deux processus. Le premier, qualifié de bas niveau, supporte les tâches de description, de publication et de recherche. Le deuxième, qualifié de haut niveau, supporte les tâches d'annotation, de certification, de validation,. Plusieurs technologies viennent instrumenter un tel modèle. La mise en œuvre de la spécification RDF (Resource Description Framework) du W3C est l'un des éléments supportant le haut niveau du processus. Cet outil vient en complément des schémas XML produits par le consortium IMS. RDF permet de définir des schémas communs pour formaliser un raisonnement. Par la suite, il est utilisé par des protocoles de recherche de ressources pédagogiques. En reprenant la proposition d'un premier schéma RDF de l'IMS[14], nous présentons notre réflexion sur les potentiels et les limites d'une telle perspective.

[13] Traduction de « The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution ».
[14] Voir, http://www.imsglobal.org/rdf/

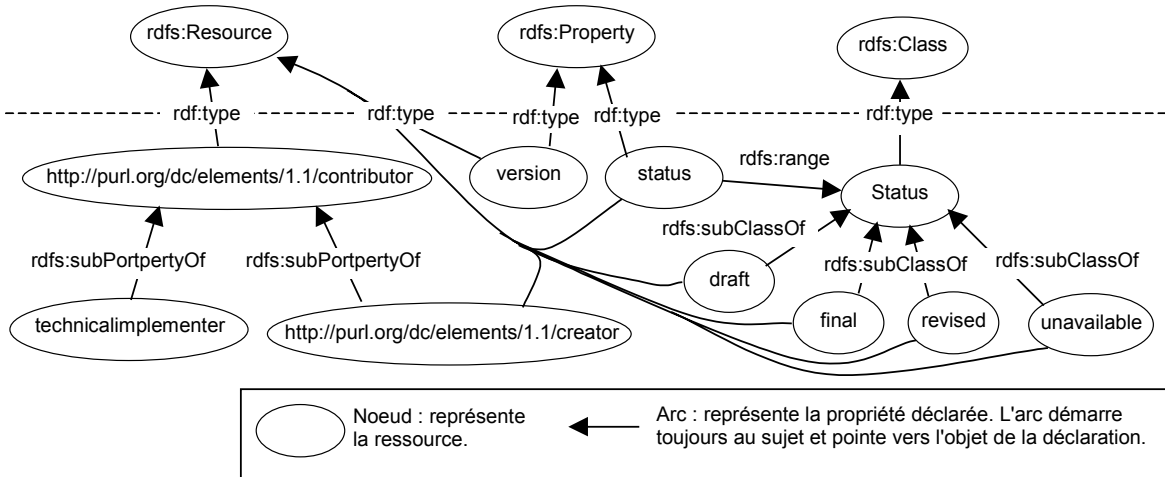


Figure 26 : Extrait d'une représentation en graphe du schéma RDF pour le méta descripteur « cycle de vie »[15]

La figure ci-dessus représente l'extrait du schéma RDF décrivant le méta-donnée LOM « cycle de vie ». Les primitives du langage RDF sont utilisées pour représenter les types et les relations entre les *objets* de ce graphe de connaissances. Ce dernier montre ainsi le travail des groupes de normalisation sur les adaptations du schéma intitulé « Dublin-Core » et explicite l'ajout d'un ensemble de nouvelles propriétés. Cette description permet ainsi de communiquer le raisonnement partiel de l'adoption d'un consensus autour du méta-descripteur LOM.

En complément de la spécification RDF, le W3C préconise d'utiliser OWL (Web Ontology Language)[16] pour représenter le sens d'un ensemble de termes et les relations qui existent entre ces termes. Ce dernier se décline en trois sous langages graduant les niveaux d'expressivité : OWL Lite, OWL DL et OWL Full. Mais leurs mises en application restent un thème de recherche scientifique en soi. Nous pouvons prendre l'exemple du travail présenté dans (Amorim, Lama et al. 2006) sur le schèma XML du modèle de spécification IMS LD. Les auteurs présentent le langage OWL comme une des solutions aux limites d'expressivité de la sémantique du langage IMS LD et comme une alternative aux initiatives visant à simplement opérationnaliser ce langage.

De plus, nous pouvons constater qu'aucun des projets menés au sein du consortium IMS ne visent à définir des consensus sur les ontologies. Par ailleurs, seul le modèle sur les méta-descripteurs possède un schéma RDF. Le consortium IMS se restreint alors au premier niveau du modèle technique du web sémantique et montre ainsi sa faible implication à engager des consensus autour de modèles sémantiques. Pour cela, le consortium a choisi le langage XML pour formaliser ces différentes propositions. Les caractéristiques lexicales et syntaxiques représentées dans un tel schéma sont utilisées pour valider la conformité des différentes *instances*. De plus, le schéma XML répond aux

[15] Voir, http://www.imsglobal.org/rdf/
[16] Voir, http://www.w3.org/TR/2004/REC-owl-features-20040210/

exigences de réutilisation et de modularité intégrées dans les contraintes dans tout projet informatique orienté *objet*. En effet, la technologie des schémas XML mettant en œuvre des mécanismes d'héritage et de surcharge instrumente les activités de transformation.

Par ailleurs, les propositions actuelles de schémas XML du consortium sont utilisées dans différents projets les mettant en pratique. Ces derniers mettent en accès libre un nombre de plus en plus important de spécifications. Cette philosophie permet de supporter deux activités du concepteur. La première met en œuvre les tâches liées aux logiques de description du concepteur effectuant une recherche dans les spécifications existantes et la seconde concerne plus spécifiquement le travail du concepteur les utilisant. Dans un cas, l'objectif est de mettre en correspondance un descripteur renseigné par le concepteur avec le descripteur décrivant chacune des spécifications disponibles. Celles identifiées sont alors présentées au concepteur qui effectue ensuite son choix. Dans le deuxième cas, l'utilisation d'une spécification soulève des interrogations de la part du concepteur sur les spécificités de sa mise en œuvre. Dans les deux cas, le site portail de la communauté diffuse non seulement les spécifications mais également un ensemble d'informations afin de spécifier leurs utilisations. Nous pouvons citer l'exemple du projet européen UNFOLD[17]. Son objectif est de fédérer les meilleurs pratiques dans le domaine de l'ingénierie pédagogique et de mettre en œuvre les modèles de spécification du consortium IMS. Pour cela, il gère et rend l'accès libre à une base d'*instances* des modèles de spécification que tout concepteur peut alors consulter ou alimenter en soumettant de nouvelles.

2.2.3.3 La gestion des services d'un système distribué : la perspective d'évolution du cadre ODP-RM

Dans l'ingénierie de la formation, l'utilisation des technologies éducatives standards se définit comme un vecteur de pratiques communautaires définissant de nouveaux enjeux. Deux rôles peuvent être identifiés dans leur utilisation, le premier de nature « anthropocentrée » et le second de nature « technocentrée » (Rabardel 1995). Le rôle « anthropocentrée » correspond aux concepteurs d'environnements de formations souhaitant prendre part aux innovations pédagogiques. L'enseignant, les institutions de formation, les membres d'une communauté du logiciel libre, les communautés de scientifiques jouent ce rôle (Rabardel 1995). Le rôle « technocentrée » s'inscrit dans une perspective économique de la formation à distance, où les technologies sont considérées comme de simples médiateurs de contenu. Notre approche ne veut pas opposer ces deux rôles mais proposer un cadre permettant d'articuler ces perspectives considérées comme complémentaires.

[17] Voir, http://www.unfold-project.net/

Même si le cadre ODP-RM n'a pas pour ambition de couvrir tous les rôles et les enjeux liés à l'usage des technologies éducatives, il se présente comme un cadre formel complet et consistant pour la spécification des services rendus par le système de formation. Ce dernier est alors vu comme un ensemble de composants distribués supportant ces différents services que l'utilisateur ou le concepteur peut rechercher et solliciter [(ISO/IEC-10746-1 1998), partie 6.2]. Le cadre ODP-RM définit un jeu de masques pour rechercher les services, un ensemble de fonctionnalités pour les gérer et un ensemble d'outils pour les décrire. L'ensemble des services se décompose ainsi en trois éléments dans le cadre ODP-RM : les *transparences*, les *fonctions* et les *points de vue*.

Les *transparences* fournissent une vue uniforme simplifiée d'un système distribué pour l'utilisateur et le concepteur. Chacune des *transparences* est mise en œuvre à l'aide d'une ou plusieurs *fonctions*. Ces *transparences* permettent de masquer les détails liés aux mécanismes d'accès, des échecs, de localisation, de migration, de relocation, de réplication, de persistance ou de transaction. Elle fournit une portabilité dans les environnements hétérogènes à travers des standards de travail et d'*interfaces*. Les *points de vue* ODP-RM définissent un cadre structurant la démarche de spécification d'un système distribué. L'acteur, l'utilisateur sollicitant un service ou le concepteur sélectionnant un service au travers d'un tel cadre, est amené à manipuler des modèles. En effet, les actes de modélisation définis par les concepts du cadre ODP-RM sont induits dans sa recherche de services.

Ce concept s'illustre parfaitement dans les processus de développement des communautés du logiciel libre. Par exemple, le projet communautaire de plates-formes supports aux activités collaboratives de formations FLE (Futur Learning Environment) (Leinonen, Virtanen et al. 2002) diffuse sur son site plusieurs aspects sur l'application développée. L'un d'eux détaille la structure à *objets* stockée dans la base de données. Deux *points de vue* du cadre ODP-RM sont utilisés pour représenter l'acte de modélisation effectué par le développeur membre de cette communauté. Deux *points de vue* sont présentés au concepteur amené à sélectionner cet environnement (cf. Figure 27). L'un correspond aux concepts de *point de vue informationnel* (à gauche) et l'autre aux concepts de *point de vue technologique* (à droite). Tous deux appartiennent au cadre architectural défini dans (ISO/IEC-10746-3 1996). En conséquence, cette figure représente le modèle négocié entre le concepteur spécifiant son scénario à l'aide du langage LD de l'IMS et le développeur effectuant les tâches d'ingénierie du logiciel.

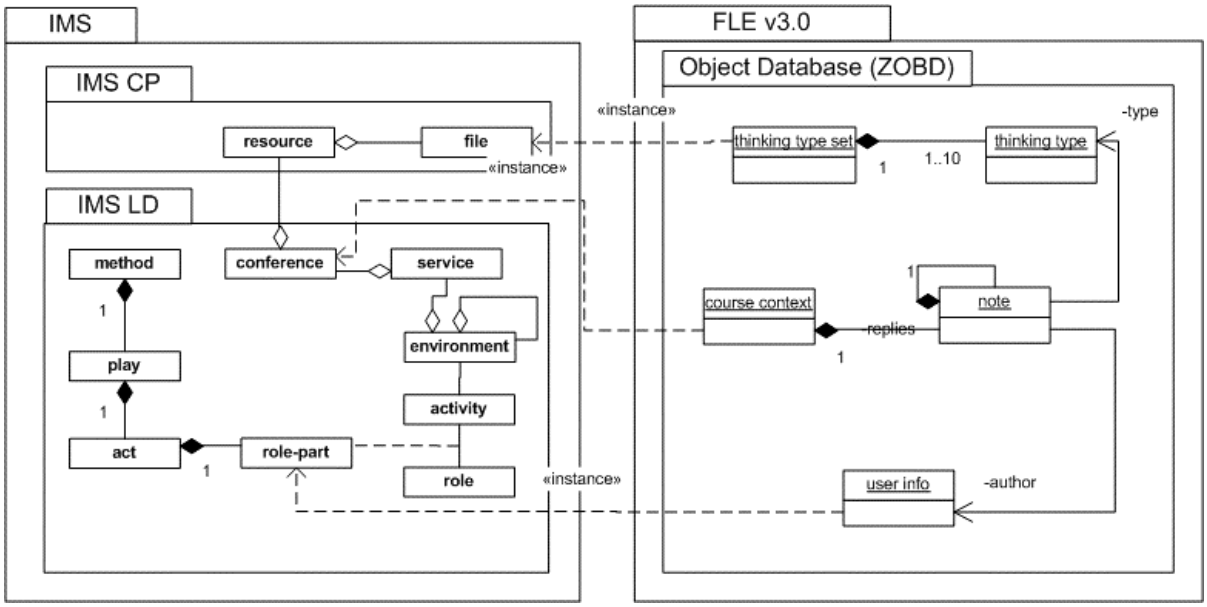


Figure 27 : Diagramme UML diffusé sur le site Web de l'environnement FLE[18] *représentant le point de vue informationnel (à gauche) et le point de vue technologique (à droite)*

La structure du modèle et les relations entre ces différentes composantes définissent un ensemble de règles de cohérence. Des faits ou des *états* observables des *objets* (ISO/IEC-10746-2 1996) peuvent être spécifiés. En l'occurrence, l'une des tâches du développeur informatique est d'intégrer les sondes logicielles permettant de capter les *états* des *objets* liés au *comportement* du modèle. Dans l'exemple ci-dessus, les *objets* ayant une *instance* dans le *point de vue informationnel* seront observés. Tout accès à l'un des *objets* appartenant au *point de vue technologique* génère une *trace* caractérisée par une note, son auteur, le type de pensée et le contexte du cours.

Les analyses sur ces différentes *traces* suite à l'utilisation du composant par les étudiants ont pour objectif de présenter des informations faisant sens au concepteur pédagogique. De nouvelles interrogations peuvent porter sur la négociation autour d'un nouveau modèle. Ce dernier peut être enrichi par des *objets* ou des *points de vue* supplémentaires explicitant par exemple les différents événements de nature technologique modifiant la structure ou les éléments du modèle.

Cet exemple montre que les concepteurs travaillent bien au niveau des fonctions, au sens ODP-RM, permettant de gérer la persistance de l'information tout en cachant les fonctionnalités du système liées à la gestion des échecs, la localisation et la description.

[18] Voir, http://fle3.uiah.fi/uml/

En préconisant le cadre ODP-RM, nous apportons les premiers éléments de réponses aux besoins de communications et d'échanges sur les différents modèles des composantes d'un système de formation. En effet, la dimension qualité d'ODP-RM se définit par le retour quantitatif des usages observés mais également par la recherche d'indicateurs qualitatifs manquants [(ISO/IEC-10746-1 1998), partie 9.1]. L'ensemble des concepts de ce cadre ODP-RM organisé en deux grandes catégories permet de décrire chacun de ces cycles. La première catégorie définit les concepts liés aux propriétés des systèmes et des *objets* (ISO/IEC-10746-2 1996). Un *contrat* spécifie le *rôle* d'un *objet*, les paramètres de *qualité de service*, les indicateurs en terme de validité temporelle ou comportementale d'un *objet*. Une qualité de service exprime des besoins en terme de qualité sur le comportement du système ou d'un *objet*. Un *contrat* avec son *environnement* spécifie les contraintes d'utilisation et de gestion. La seconde catégorie définit les concepts spécifiques aux cinq *points de vue* d'une vision architecturale d'un système. Cette catégorie reprend les concepts présentés dans le document (ISO/IEC-10746-3 1996). En utilisant un langage et des règles structurelles spécifiques à chaque *point de vue*, le concepteur exprime les différents contrats sur le système. Le respect ou non de ces différentes règles peut être observé lors des mises en situation des composantes d'un système de formation au sein de la communauté les ayant explicitées ou par une autre communauté.

2.2.4 Deux consortiums, deux manières de négocier

2.2.4.1 La communauté du logiciel libre : une nouvelle manière de négocier dans un processus de développement induit par LTSA

Le comité LTSC de l'IEEE, au début des années 97, a fait le choix de définir ses différents groupes de travail autour d'une architecture correspondant au standard LTSA. L'approche par composant choisie par ce comité a permis de proposer une décomposition fonctionnelle résultant d'une analyse structurée d'un système de formation. Les échanges d'information entre les différentes composantes du modèle s'effectuent au travers des *interfaces* de chaque composant. Elles établissent les points critiques d'interopérabilité sur lesquels le comité souhaite engager ou identifier des groupes de travail pour établir des consensus sur les technologies utilisées dans le domaine éducatif. Aujourd'hui, les différentes *instances* du modèle spécifient les limites approximatives des standards et des modèles de spécifications existants (Farance 2003). En proposant un seul modèle pour représenter les activités de standardisation, la volonté est de considérer le besoin en consensus sur les modèles spécifiant le comportement interne des composants comme non prioritaire au sens du modèle LTSA (Farance 2003).

Nous présentons sur la Figure 28 les *instances* du modèle LTSA correspondant aux « stakeholder [19]» intitulés « Evaluation », « Behavior », « Interaction Context » et « Assessment » [(IEEE/LTSA 2003), parties 5.5, 5.4, 5.19 et 5.10]. Elles sont à mettre en relation avec le projet de standardisation du groupe QTI (Question-Test Interoperability) du consortium IMS.

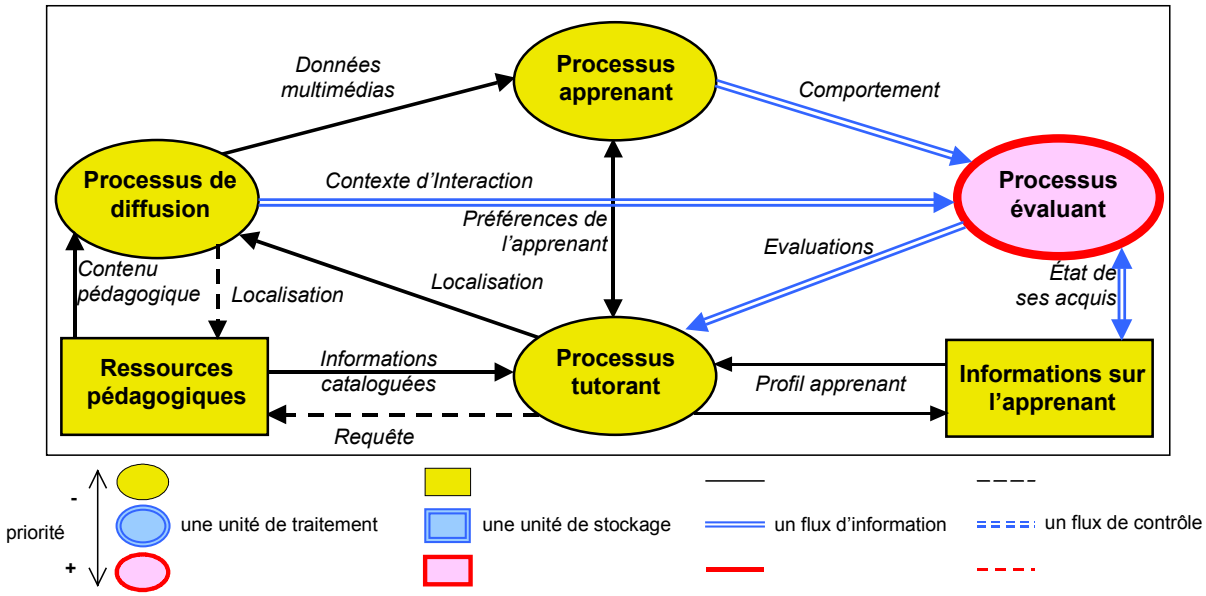


Figure 28 : Instance du modèle à composants LTSA (IEEE/LTSA 2003)

Le travail mené par l'IEEE et ayant produit comme résultat le standard LTSA (IEEE/LTSA 2003) s'adresse plus particulièrement à l'industrie informatique chargée de concevoir des plates-formes de formation à distance. Ce modèle veut influencer les concepteurs dans leur travail de structuration du système développé. L'objectif n'est pas de leur imposer une architecture mais de les sensibiliser sur les différentes négociations menées actuellement par les différents consortiums. Il s'agit de leur permettre d'intégrer plus rapidement les modèles finalisés résultant de ces consensus.

Les différents principes présentés définissent le processus de standardisation adopté par le comité LTSC de l'IEEE. Ils sont appliqués dans l'ingénierie centrée sur l'architecture des systèmes informatiques. Une structure générique est adoptée tout au long du cycle de développement et permet aux concepteurs d'engager des négociations au niveau de l'expression des besoins (IEEE/ADS-IS 2000). Ce principe est actuellement mis en œuvre dans les processus industriels, par exemple dans 2TUP (Two Tracks Unified Process) (Roques et Vallée 2004) où l'objectif est d'engager le plus rapidement possible des négociations pour identifier les risques sur l'incapacité de l'architecture technique à répondre aux contraintes opérationnelles ou sur l'inadéquation du développement aux besoins des

[19] Désigne l'individu ou l'organisme qui est partie prenante avec un système [(IEEE/ADS-IS 2000), partie 3.8]

utilisateurs. Mais contrairement au modèle LTSA, l'architecture est également utilisée pour définir la structure et la nature des organisations dans un processus de développement, lors des échanges d'expériences et de la réutilisation d'environnements techniques (Bass, Clements et al. 2003). La problématique de l'ingénierie centrée sur l'architecture est alors de rechercher des formalismes pour décrire les cycles de développement selon les perspectives technologique et métier (Bass, Buhman et al. 2001; Barbier, Cauvet et al. 2004; Oussalah, Sadou et al. 2005). L'enjeu de l'architecture décrivant le processus « métier » est de préciser les propriétés des systèmes développés et de permettre la définition des stratégies des futurs projets (Bass, Clements et al. 2003).

Les travaux d'analyse actuellement menés sur les communautés de développement du logiciel libre apportent quelques réponses en identifiant les propriétés sur les démarches des concepteurs centrées sur la communication entre les membres. Des règles émergeant des pratiques sur l'organisation des communautés du logiciel libre sont formalisées en tenant compte du contexte socio-économique spécifique (Raymond 2001). Les contraintes sur les ressources humaines ou sur la planification d'un projet du logiciel libre sont pratiquement absentes. Un membre d'une telle communauté se prête à un jeu où seules les attitudes, les intentions, les habiletés sont prises en compte. Les règles et les propriétés qui les régissent sont alors explicitées dans les rôles négociés pour chacun des membres d'une communauté.

L'objectif du modèle LTSA a été atteint en définissant, au début des années 97 et à un niveau générique suffisant, les « stakeholders » correspondant aux projets actuels de normalisation sur les technologies éducatives. Cependant, ce niveau d'abstraction constitue un frein car seule est considérée l'activité des organismes de standardisation. Aujourd'hui, les principes d'ingénierie centrée sur l'architecture du modèle LTSA se doivent d'être transposés pour analyser les communautés du logiciel libre, où l'architecture joue le même rôle de sensibilisation afin de gagner l'adhésion d'éventuels membres.

L'analyse présentée montre qu'il est nécessaire de disposer non pas d'un niveau d'abstraction sur l'architecture mais d'un *point de* vue supplémentaire. Il répond au besoin de formaliser les négociations engagées entre les concepteurs et leurs rôles joués sur la spécification de nouvelle fonctionnalité en analysant les expériences menées par la commnauté du logiciel libre.

2.2.4.2 Des similitudes entre les modèles IMS SS et IMS LD : nécessité d'engager une négociation

L'ensemble des modèles du consortium IMS se définit comme étant capable de décrire à la fois les différents comportements d'un système de formation et de répondre aux besoins des concepteurs. La structuration des projets de ce consortium et des modèles informationnels impose des choix aux concepteurs les manipulant. Ils suscitent de l'intérêt auprès des apprenants utilisateurs des différentes mises en œuvre de ces modèles, auprès des concepteurs amenés à

choisir l'un d'eux pour décrire sa séquence pédagogique et auprès des informaticiens développant les composants intégrant de telles *interfaces*.

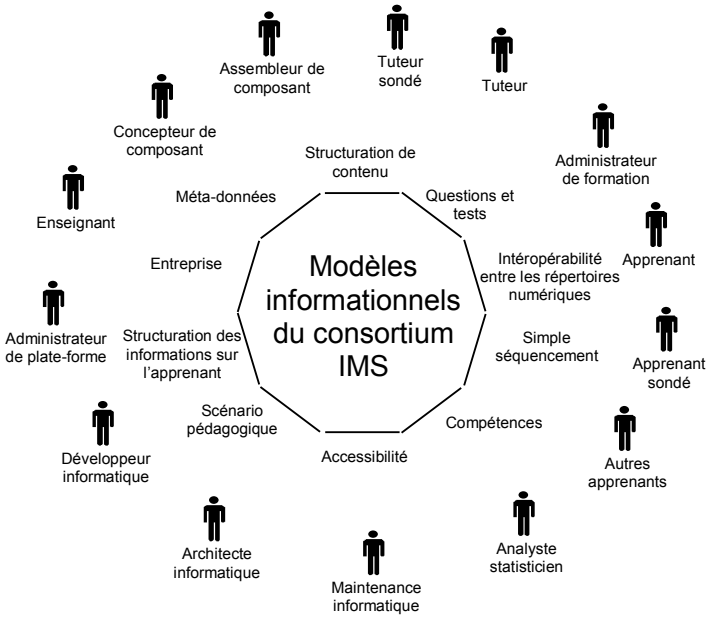


Figure 29 : Ensemble des modèles du consortium IMS et des acteurs d'un système de formation

L'ambition est de répondre aux différents besoins des acteurs d'un système de formation. Ces modèles ne sont qu'au stade de propositions mais ont été produits dans un processus de standardisation visant à les promouvoir en tant que normes internationales. L'évolution des propositions va amener à supprimer ou à préciser les détails de chaque spécification en cours d'élaboration afin de respecter la syntaxe des modèles préconisés par les institutions de normalisation internationales. Or, nous considérons que ces détails engagent les concepteurs utilisant les modèles proposés par le consortium IMS dans de nouvelles négociations les amenant à partager et à communiquer leurs pratiques.

Par exemple, en analysant la documentation des deux projets appartenant au même consortium, IMS, nous identifions des directives différentes sur l'utilisation du même modèle « Simple Sequencing ». Ces dernières font l'objet de négociations entre les concepteurs mettant en œuvre un tel modèle.

La première destine ce modèle à la spécification des conditions et des actions d'une séquence pédagogique et suggère son utilisation à la description des règles de synchronisation des activités de collaboration entre les acteurs d'une telle séquence [(Norton et Panar 2003), page 5].

La deuxième utilisation de ce modèle est de spécifier le déroulement d'une séquence pour un seul acteur, un apprenant par exemple [(Koper, Olivier et al. 2003a), page 129]. Cette précision est formulée par les membres du projet ayant produit le modèle « Learning Design » permettant de décrire l'exécution en parallèle de plusieurs séquences,

l'association d'un rôle à une structure d'activités et l'utilisation de propriétés pour modifier le déroulement d'une séquence.

Ces deux préconisations sont différentes pour un même modèle. Par ailleurs dans [(Tattersall, Manderveld et al. 2003), page 6], où cette ambiguïté a également été relevée, les auteurs préconisent comme solution d'adopter des *points de vue* différents sur l'approche d'une séquence d'apprentissage. Deux perspectives sur l'utilisation du modèle « Simple Sequencing » sont alors considérées. La première décrit le *comportement* générique du composant d'un système de formation alors que la deuxième décrit celui d'un apprenant n'étant engagé dans aucune collaboration. Nous distinguons bien une première préoccupation centrée sur la technique utilisée pour concevoir le système et une deuxième sur l'acteur humain du système.

Ce constat, au sein d'un même consortium, montre qu'il est nécessaire d'instrumenter les communautés mettant en pratique ces modèles. En effet, l'un des objectifs atteints du consortium consiste à diffuser largement des modèles de spécification permettant de décrire les différents aspects d'un système de formation. Cette structuration est utile mais insuffisante pour une ingénierie des EIAH. Il est alors nécessaire de rechercher de nouveaux formalismes à adresser aux concepteurs pour supporter les négociations sur les utilisations de ces différents modèles.

2.2.4.3 ODP-RM : un cadre pour décrire les actes de négociation

Dans les deux sous-parties précédentes, nous avons présenté les différents besoins de gérer les négociations soit au niveau des récentes alternatives des processus de développement définies par les communautés du logiciel libre soit au niveau des ambiguïtés sur les directives d'utilisation préconisées sur les modèles de spécification issues de même consortium. En choisissant le cadre ODP-RM, nous nous donnons les moyens de les expliciter et de les communiquer à une communauté plus large de concepteurs. En effet, ce cadre distingue les différents *points de vue* permettant au concepteur d'exprimer aussi bien des propriétés sur les *ressources* que les directives identifiant le service rendu par un *objet* du système.

En proposant un *point de vue* spécifique intitulé *métier*, ce cadre instrumente les actes de modélisation du concepteur en lui permettant de formaliser la gestion séparée des sous-systèmes qui partagent l'information, la composition ou la décomposition des éléments et des règles qui les gouvernent. Les différents concepts de ce *point de vue* permettent d'une part de définir un *langage* indépendant des choix sur les technologies et d'autre part de pouvoir formaliser les négociations émergeant du *comportement* d'une *communauté métier*. Quatre grandes catégories de concepts sont alors définies par le langage associé à ce point de vue (ISO/IEC-15414 2002) :

- *Communauté* : Ce concept permet de décrire la fédération des *objets métier*, caractérisés par un *objectif* distinct et de spécifier ainsi un nouvel *objectif*. Ce dernier se doit d'être décrit afin de préciser les futurs *états*

de la *communauté* pouvant être quantifiés. Le concepteur utilisant ce concept pour spécifier un tel *point de vue* est amené à identifier et à spécifier les négociations pouvant être engagées.

- *Comportement* : Ce concept décrit un *objet* appartenant au *point de vue* spécifique *métier* du cadre ODP-RM. Il distingue les *objets* appartenant à une même *communauté*.
- *Stratégie* : Ce concept permet de spécifier le *comportement* d'un *objet métier*. La fédération des directives implique leur interconnexion et facilite ainsi le travail de description.
- *Responsabilité* : Ce concept permet de décrire l'*action* d'un *objet métier* en précisant ses intentions.

Afin d'illustrer un tel potentiel, nous prenons l'exemple de deux projets issus de communautés de développement du logiciel libre dans le domaine de la formation à distance. Nous contraindre à les positionner dans *le point de vue métier* du cadre ODP-RM a pour conséquence d'expliciter à la fois des objectifs communs et des liens dans lesquels peuvent être engagées des négociations.

La premiere communauté diffuse une plate-forme de formation à distance pour supporter des sessions d'apprentissage collaboratives (Leinonen, Virtanen et al. 2002). La seconde propose des directives de développement liées à l'ingénierie des composants appliquée au domaine de la formation (Grob, Bensberg et al. 2004). Ces différentes propositions sont adressées aux concepteurs en présentant des similitudes et des différences. Par exemple toutes les deux manipulent le même concept d'acteur apprenant et mettent en œuvre les technologies orientées objet pour gérer le stockage. Cependant des différences existent dans la manière de gérer ce concept par le dispositif informatique et dans le processus de développement mis en œuvre par les membres de ces deux communautés.

Nous en présentons une analyse rapide. Dans un premier temps, nous nous situons au niveau des productions diffusées. Les différents projets du consortium IMS nous présentent différents aspects à appréhender sur un système de formation. En effet, la première communauté diffuse une plate-forme mettant en application le modèle de spécification d'une unité d'apprentissage (Koper, Olivier et al. 2003b) et la deuxième communauté diffuse un profil technique de composant logiciel d'un système de formation cité comme l'un des référents dans le projet du consortium IMS « Abstract Framework » (Smythe, Collier et al. 2002). Le concept d'*acteur* apprenant est alors considéré soit comme le *rôle* spécifié dans le déroulement d'une unité d'apprentissage soit comme un *objet implanté* dans un composant technique du gestionnaire de formation. Dans un deuxième temps, nous nous situons au niveau du processus de conception. Pour la première communauté, le concepteur utilisant la plate-forme collaborative travaille sur la configuration et l'analyse en retour de formation. Il est amené à s'intérroger sur les détails du déroulement de l'unité d'apprentissage. Dans la deuxième communauté, le concepteur se doit de comprendre le cadre technique et ses possibilités d'extension. Il est par conséquent amené à s'interroger sur le *comportement* global du gestionnaire.

Les *objets* manipulés par le concepteur appartiennent à des *points de vue* différents et le domaine d'expertise n'est pas le même. Cela implique la nécessité d'identifier le contexte d'utilisation des différents concepteurs utilisant des production de communauté du logiciel libre. A l'aide des concepts de *communauté*, de *comportement*, de *stratégie* ou de *responsabilité*, le *point de vue métier* du cadre ODP-RM se présente comme l'un des instruments capables de décrire un des premiers langages de spécification du *point de vue métier* d'un système de formation.

2.3 Apport de l'instance ODP-RM : Proposition d'un point de vue métier

2.3.1 Vers un point de vue métier de l'organisation d'un système de formation

L'analyse de la synergie entre une communauté du logiciel libre et une approche dirigée par les modèles montre à la fois l'impact des propositions des technologies éducatives normées des deux communautés et le potentiel du cadre ODP-RM à gérer chacune des perspectives soulevées. Quatre aspects sont analysés :

- les limites des propositions des deux consortiums ;
- les actes de transformations des modèles explicités par les deux consortiums ;
- les perspectives d'évolution définies par les deux consortiums ;
- et les actes de négociation des concepteurs.

Par conséquent, l'ingénierie de la formation est vue comme un processus dans lequel le travail d'observation et d'évaluation des mises en pratique de *ressources* d'un système de formation est effectué. L'objectif est d'interroger les méthodes d'ingénierie et d'apporter de nouveaux éléments critiques. Cette perspective répond aux besoins de penser de façon large et d'intégrer les nouveaux systèmes de formation (Moreau et Majada 2002) où l'objectif est d'engager des modifications et des adaptations touchant à l'organisation ou aux compétences des acteurs (Le Boterf 2001).

Nous nous référons également aux théories liées au développement des organisations apprenantes [(Wenger 1998), pages 241-262] où l'auteur propose une réflexion en distinguant deux aspects : l'organisation qui apprend, qualifiée d'« institutionnelle » et l'organisation sur les « pratiques ». La réflexion menée par l'auteur produit les quatre principes suivants :

- La participation et la réification : il faut être amené à s'interroger sur la nature des pratiques auxquelles les outils mis à la disposition d'une « institution » sont destinés.

- Le conçu et l'émergeant : il faut faire en sorte que la structure conçue par l'institution reste minimale pour assurer la cohérence de l'organisation mais laisser assez d'espace aux pratiques émergeantes.

- Le local et le global : il s'agit de permettre la connexion des compétences entre pratiques par la création d'interfaces, que ce soit des objets frontières ou des rôles interfaces permettant d'assurer le lien entre différents niveaux de pratique.

- L'identification et la négociation : il s'agit de faire référence aux capacités de l'individu à contribuer aux communautés, de prendre des responsabilités et d'exercer une influence sur le développement d'une organisation.

Les technologies éducatives proposées par les différentes « institutions » de normalisation sont des outils aptes à spécifier les sessions d'apprentissage et la structure d'une ressource pédagogique. En s'interrogeant sur l'articulation entre la communauté du logiciel libre et une approche dirigée par les modèles, nous cherchons à formaliser un modèle d'organisation afin de coordonner les diverses interactions et de coordonner des communautés de pratique auxquelles nous avons adhérées au cours des mises à l'essai effectuées pendant quatre années.

Nous allons nous intéresser plus spécifiquement à l'apport du *point de vue métier* du cadre ODP-RM à une *communauté* de concepteurs d'un système de formation. En effet, ce dernier dispose d'un *langage* spécifique en définissant des termes et des règles de structuration. L'objectif est de spécifier la structure et le *comportement* du système afin de capter les contraintes dues à son *environnement* (ISO/IEC-15414 2002). Ces dernières sont liées à des aspects techniques et/ou d'organisation sociale et métier.

2.3.2 Première proposition du point de vue métier sur un système de formation

Nous faisons le choix de représenter le *point de vue métier* en appliquant la technique de modélisation par *flux*. Cette dernière permet à l'aide d'un système de notation de représenter sur un seul schéma les traitements, les unités de stockage, les *flux* d'information et les *flux* de contrôle du système analysé. L'enjeu est d'amener les concepteurs à décomposer les différents éléments du système considéré.

Cette vision générale d'un système d'information modélise les valeurs traitées, les informations, les contraintes et les dépendances fonctionnelles. Nous utilisons le système de notation issu de la méthode d'analyse et de conception des systèmes d'information proposée par (Yourdon 1989). En effet, le cadre ODP-RM ne propose pas de formalisme mais définit un ensemble de termes. Le langage spécifiant le *point de vue métier* doit répondre à un ensemble de règles structurelles et reprendre les concepts de *communauté*, de *comportement*, de *stratégie* et de *responsabilité* (ISO/IEC-10746-3 1996; ISO/IEC-15414 2002). L'enjeu est de formaliser par la suite les négociations entre les *objets métier*, les mises en relation entre les domaines, ou avec d'autres *points de vue* du cadre ODP-RM.

Cette première description d'un système de formation est incomplète, comme d'ailleurs tout modèle inscrit dans le cadre ODP-RM. Ce principe met en application la notion de « stakeholder » utilisée par l'architecture logicielle (Bass, Clements et al. 2003) où l'utilisateur, le concepteur informaticien ou pédagogue sont alors concernés par les propriétés de la structure du système (Kilof 2004). Cette première *instance* du *point de vue métier* nous permet de décrire une première vision structurée du comportement d'un système de formation. Le *point de vue métier* est décrit par les trois éléments suivants :

- La première *instance* du *point de vue métier* est représentée par un diagramme structuré composé de plusieurs unités de traitement et de *flux* de données ;
- Une description des différents *flux* ;
- Une séquence du déroulement d'un cycle représenté par le modèle.

Un diagramme de *flux* de données est composé de plusieurs unités de traitement qui, au sens de (Yourdon 1989), représente un composant transformant les *flux* d'information en entrée en *flux* de sortie. Dans la terminologie du cadre ODP-RM, nous parlons d'*objet métier* qui compose systématiquement une *communauté métier*. Dans le modèle que nous proposons, les six *objets métier* constituent la *communauté* de réingénierie d'un système formation. Pour chaque unité de traitement, nous utilisons le terme de *processus*, qui selon la terminologie du *point de vue métier*, définit un ensemble d'*actions* décrites de manière prédictive, et permet d'atteindre différents *objectifs* [(ISO/IEC-15414 2002), sous-partie 6.3.5].

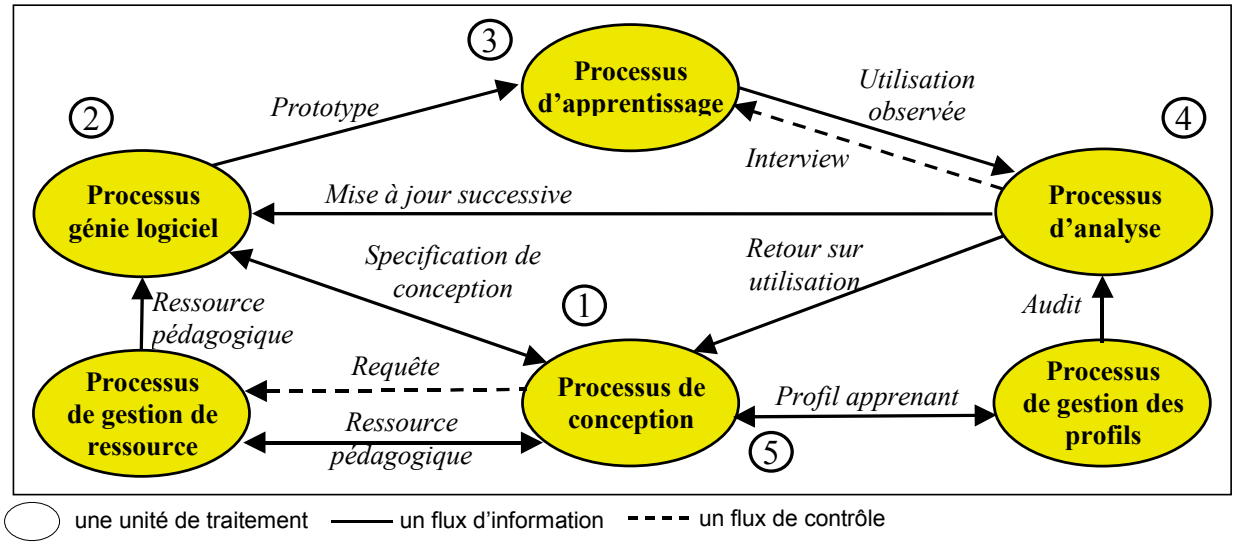


Figure 30 : Première instance du point de vue métier sur la communauté de réingénierie d'un système de formation [adaptée de (Corbière et Choquet 2004a)]

Dans la terminologie du cadre ODP-RM (ISO/IEC-15414 2002), il est préconisé d'utiliser le terme *rôle d'interface*. Nous l'utilisons dans les chapitres suivants (cf. la sous-partie 3.3.3, page 84, et la partie 4.3, page 105). Pour l'instant, nous reprenons le terme de *flux* d'information de [(Yourdon 1989), page 143] qui définit de manière large tout type d'information échangée entre deux ou plusieurs *processus*. Nous les décrivons ci-dessous.

- **Utilisation observée** : Le *processus* d'apprentissage produit des observables lors de l'exploitation des *ressources* pédagogiques. Ces informations sont des *traces* numériques, le résultat des interviews des acteurs ou des échanges entre *acteurs* lors des sessions d'apprentissage.
- **Retour sur utilisation** : Le *processus* d'analyse expertise la *ressource* en négociant au travers de ce *flux* les éventuels problèmes liés à la spécification de la session d'apprentissage. Ces retours sensibilisent les *acteurs* du *processus* de conception sur les différences entre l'utilisation prescrite et l'analyse des retours observés.
- **Mise à jour successive** : Le *processus* d'analyse soulève des problèmes qui peuvent relever d'une simple mise à jour liée aux contenus jusqu'à la modification de la structure de l'environnement d'apprentissage (Duqesnoy 2002). Le *processus* génie logiciel peut être amené à automatiser ces modifications.
- **Profil apprenant** : Le *processus* de conception accède aux informations concernant les apprenants. Son appartenance à un groupe et ses réactions suite à la session d'apprentissage sont deux exemples d'informations caractérisant son profil.
- **Spécification de la conception** : Le *processus* de conception engage avec le *processus* génie logiciel des négociations visant à spécifier les séquences d'activité, les rôles et les environnements pédagogiques utilisés. A la suite de ces *interactions*, des sondes logicielles sont définies et intégrées dans le dispositif d'apprentissage par le *processus* génie logiciel.
- **Ressource pédagogique** : Le *processus* d'apprentissage utilise des *ressources* pédagogiques qui supportent des rétroactions au sein du *processus* de conception, du *processus* génie logiciel, du *processus* d'apprentissage ou du *processus* d'analyse. Ces *ressources*, intitulées « objets », peuvent être réutilisées, ou servir de base à de nouveaux *objets*, en les assemblant ou en les décomposant.
- **Audit** : Le *processus* d'apprentissage dans lequel se déroulent les différentes sessions d'apprentissage effectuées par l'apprenant génère un historique de *traces* pour chacun des apprenants.
- **Prototype** : Les *processus* de conception, génie logiciel, d'apprentissage et d'analyse supportent une conception cyclique et en conséquence utilisent des *objets* d'apprentissage considérés comme des prototypes.

Contrairement aux *flux* d'informations qui véhiculent soit des valeurs discrètes, soit le plus souvent des valeurs continues, un *flux* de contrôle échange exclusivement des valeurs discrètes et formalise ainsi les contraintes temps-réel de certains aspects du système :

- **Interview** : Le *processus* d'analyse expertise l'usage du scénario et des *ressources* au regard des observables. Ce *flux* lui permet de compléter les informations obtenues afin de guider son travail d'analyse.
- **Requête** : Le *processus* de conception effectue des requêtes afin de rechercher les *ressources* répondant aux besoins spécifiés.

S'agissant d'une *instance* du *point de vue métier* du cadre ODP-RM, nous reprenons dans sa description le concept de *comportement* en se référant aux *étapes* numérotées de la Figure 30, page 75. Le modèle présenté se définit par un ensemble d'*actions* (ISO/IEC-10746-2 1996). Le *processus* de conception (1) prend en charge la spécification du cursus en terme d'unité d'apprentissage (Koper, Olivier et al. 2003b) et d'*objets* d'apprentissages (IEEE/LOM 2002).

En conséquence, la spécification des différents environnements informatiques supportant la session d'apprentissage et les procédés d'observation de son déroulement peuvent faire l'objet de négociations entre le *processus* de conception et le *processus* génie logiciel (2). En effet, ce dernier raisonne sur les descriptions techniques et pédagogiques définies par les concepteurs. Elles spécifient l'ensemble des *ressources* mises en œuvre dans le *processus* d'apprentissage. L'intégration de mécanismes d'observation du déroulement de la session est l'une des tâches confiées à ce *processus*. Afin de s'inscrire dans le contexte de réutilisation promu par les technologies éducatives normées et d'optimiser les phases de rétroconception et de réingénierie, les *ressources* utilisées sont sélectionnées en fonction du respect des contraintes de qualité dans sa mise en œuvre.

La phase d'exploitation des *ressources* du cycle global de formation est mise en œuvre par le *processus* d'apprentissage (3) et sa décomposition correspond au modèle à composants LTSA.

Le *processus* d'analyse (4) compare la description des scénarii *a priori* corrélés avec l'usage observé par le *processus* d'apprentissage et le *comportement* des apprenants. Des différences peuvent être relevées automatiquement par un composant spécifique, intégré au dispositif d'apprentissage ou à la suite d'un travail d'analyse mettant en relation des observables avec des profils d'usages existants. Deux types d'échanges peuvent être envisagés. Le premier, à destination du *processus* génie logiciel, vise à modifier la sensibilité de certains capteurs ou à supprimer des incohérences sur les informations observées. Le second, à destination du *processus* de conception, soulève des problèmes critiques rencontrés par certains apprenants. Dans ce cas, tous les éléments permettant au *processus* de conception d'appréhender ces difficultés doivent être transmis. Ils correspondent principalement aux informations définissant le contexte de l'événement observé.

L'une des tâches du *processus* de conception (5), et en particulier les experts pédagogiques, ergonomiques et didactiques est alors d'effectuer une évaluation des anomalies et des validations constatées en retour de formation. Différentes opérations peuvent alors être effectuées. Des précisions peuvent être apportées par les experts pour affiner

le modèle pédagogique en modifiant le scénario spécifié pour répondre aux difficultés constatées ou en aidant le système à mieux interpréter les données observées liées au *comportement* de la session d'apprentissage.

2.4 Synthèse du chapitre

Ce chapitre effectue une première analyse de deux comités de normalisation sur les technologies éducatives, en prenant en compte les réactions des communautés scientifiques et les choix des développeurs les mettant en œuvre. L'émergence des premiers consensus autour des technologies éducatives normées proposées par les deux consortiums IEEE et IMS engagent de nombreuses réflexions. Elles portent sur les limites des propositions actuelles, les mécanismes de transformation des modèles de spécification proposés, leurs perspectives d'évolution et les actes de négociation des concepteurs. Sur chacun des aspects, nous présentons les potentiels du cadre ODP-RM à formaliser ces différentes réflexions. De tels constats nous amènent à proposer un *point de vue* supplémentaire, le *point de vue métier* du cadre ODP-RM. Cette vision est le résultat d'une analyse pragmatique d'un système de formation au vu des analyses effectuées sur les différentes propositions sur les technologies éducatives normées et le niveau d'abstraction offert par le cadre ODP-RM. Nous proposons et adressons une *instance* de ce *point de vue* à la communauté de concepteurs d'un EIAH.

CHAPITRE 3 : ANALYSE DE LA SYNERGIE ENTRE UNE COMMUNAUTE DU LOGICIEL LIBRE ET UN PROCESSUS DE DEVELOPPEMENT LOGICIEL

Ce chapitre présente une analyse guidée par les concepts définis par le cadre ODP-RM. Cette réflexion se fonde d'une part sur l'articulation entre une communauté du logiciel libre et un processus de développement logiciel et, d'autre part, sur les problématiques de réingénierie d'un système de formation. La notion de complexité, l'un des fondements des théories systémiques, nous sert d'éclairage théorique dans cette analyse. Elle est définie dans la première partie de ce chapitre.

Dans la deuxième partie, nous précisons la problématique liée à la définition d'une réingénierie des EIAH. Ensuite, nous y répondons en réinvestissant les résultats des travaux sur la réingénierie du logiciel et les modèles de spécification sur les technologies éducatives normées. Nous utilisons le cadre ODP-RM afin de fédérer ces différents travaux.

Une partie spécifique illustre ensuite l'*instance* du cadre ODP-RM proposée par deux cas d'usage de réingénierie d'un dispositif de formation : le premier décrit une phase de rétroconception impliquant un concepteur pédagogue et un développeur informaticien ; le second porte sur les échanges entre un concepteur pédagogue et un analyste des usages, lors de la réingénierie du système. Nous concluons par une discussion sur les perspectives de cette analyse.

3.1 Analyse suivant la notion de complexité d'un système de formation

Cette notion de complexité qualifie un système qui répond aux axiomes suivants [(De Rosnay 1975), pages 103-104]

- Un système complexe est constitué par une grande variété de composants ou d'éléments possédant des fonctions spécialisées ;
- Ces éléments sont organisés en niveaux hiérarchiques internes ;
- Les différents niveaux et éléments individuels sont reliés par une grande variété de liaisons. Il en résulte une haute densité d'interconnexions ;
- Les interactions entre les éléments d'un système complexe sont d'un type particulier. On dit que ces *interactions* sont non linéaires.

En mettant en évidence les caractéristiques qui opposent une approche analytique à une approche systémique, nous disposons de différents aspects nous permettant de mieux comprendre et d'illustrer les liens existants entre la philosophie adoptée par la communauté du logiciel libre et le processus de développement de logiciels orientés objet.

Le travail présenté dans ce chapitre est une analyse des synergies entre ces deux communautés. Chacune d'elle participe ou bénéficie aux évolutions de l'autre. L'objet est de mettre en évidence les mécanismes de transformations sur des modèles ayant à l'origine une sémantique pédagogique ou technologique. Ces derniers utilisent des approches, des formalismes et des expertises différents suivant les deux communautés référentes. L'instrumentation des mécanismes de transformation entre et au sein des différents niveaux d'abstraction est l'un des enjeux de la conception des EIAH (Tchounikine, Baker et al. 2004). Notre travail a été d'analyser ces deux communautés et de montrer les potentiels du cadre ODP-RM à expliciter et à formaliser les liens les mettant en relation. Deux exemples viennent illustrer l'utilisation de ce cadre.

3.2 Réingénierie dans les EIAH

L'activité de conception des systèmes de formation nécessite de communiquer et de partager des informations concernant les usages de ces systèmes. Les divers projets de la communauté travaillant sur les futures normes des technologies éducatives mettent à disposition différents langages de spécification permettant de décrire le déroulement d'une session de formation et de communiquer dans un format interopérable. La limite de ces technologies est de réduire la spécification à une simple prescription du comportement au dispositif d'apprentissage. Or, la démarche de conception d'un EIAH (Bruillard et Vivet 1994; Tchounikine, Baker et al. 2004) ne peut pas être réduite simplement à la spécification du dispositif, elle doit également instrumenter les tâches d'observation et de modélisation des sessions d'apprentissage. Notre objectif est donc de réinvestir les travaux sur les technologies éducatives normées dans le support à la communication entre concepteurs dans un cadre de réingénierie dans les EIAH.

Les propositions récentes de normes et de standards dans le domaine des technologies éducatives introduisent de nouvelles pratiques et méthodologies d'ingénierie des EIAH, sans toutefois proposer aux concepteurs un ensemble de concepts permettant d'expliciter son organisation et l'ingénierie mise en œuvre, de définir leur activité dans ce cadre et de s'y référer pour communiquer avec les autres acteurs d'un système de formation.

Des éléments de réponse sont pourtant apportés par la communauté du génie logiciel orienté objet dans ses travaux sur la réingénierie du logiciel. De manière pragmatique tout d'abord, l'expérience de gestion de communautés de conception participative, accumulée par les mouvements du logiciel libre, s'est traduite par la mise au point d'outils supports effectifs de communication et par la diffusion des différents artefacts de conception. D'un point de vue scientifique ensuite, les résultats de recherche du domaine proposent un ensemble d'éléments méthodologiques ainsi

qu'un vocabulaire commun à la réingénierie des systèmes. Ils sont centrés sur les problématiques d'interopérabilité et de réutilisabilité. (Chikofsky et Cross 1990), par exemple, ont finalisé une première terminologie pour définir les tâches de rétroconception[20] et de réingénierie[21] d'un logiciel. Plus récemment, et sur la base de cette taxonomie, un ensemble de langages de patrons a été produit à la suite du projet européen ESPRIT 21975 (Bär, Bauer et al. 1999) définissant à la fois des éléments méthodologiques et un vocabulaire commun.

3.3 Utilisation du cadre ODP-RM pour définir la réingénierie dans les EIAH

3.3.1 Instanciation d'ODP-RM sur la réingénierie du logiciel

ODP-RM se définit comme un cadre générique permettant de supporter le processus de modélisation d'un système complexe en demandant aux concepteurs d'instancier sur leurs domaines un ensemble de concepts génériques (*composition* / *décomposition* d'*objets*, *état* et *comportement* d'un *objet*, *points de vue* et correspondance entre *points de vue*…). Ces concepts sont déclinés par rapport aux trois actes de modélisation introduits par le cadre :

- la spécification du système, où les concepteurs classifient les objets du système par des liens de *composition* ;
- la modélisation du système, qui définit à différents niveaux d'abstraction les modèles d'*interactions* entre objets ;
- la structuration du système, où les différentes structures à *implanter* dans le système sont définies.

L'objectif du cadre ODP-RM est de favoriser l'émergence d'un consensus (un standard) dans une communauté de concepteurs donnée. Une spécification architecturée définie dans le cadre ODP-RM permet de focaliser l'attention des acteurs du développement d'un système à l'aide de *points de vue*. Ces derniers sont l'une des réponses de ce modèle de référence pour appréhender la complexité d'un système lors de sa spécification.

Le projet européen ESPRIT 21975 (Bär, Bauer et al. 1999) a défini un ensemble de patrons pour la réingénierie du logiciel. Ces patrons sont structurés en deux familles principales :

- les patrons orientés processus, qui permettent de guider les différents acteurs du processus de réingénierie ;
- les patrons orientés structure, qui permettent de partager et de réutiliser une solution technique éprouvée.

[20] L'objectif de cette tâche est « d'analyser un système afin d'identifier ses composantes et ses relations dans l'objectif de créer des représentations avec des formalismes variés ou à des niveaux d'abstraction différents » (Chikofsky 1990).
[21] Se définit comme une tâche « d'examen et d'altération d'un système afin de le reconstituer sous une nouvelle forme suivie de l'implantation de cette nouvelle forme » Ibid.

Ces différents patrons sont décrits dans des langages spécifiques, dont l'objet est de définir un ensemble de termes adoptés de manière consensuelle par la communauté des concepteurs. Notamment, le langage de patrons orientés processus propose une définition partagée des différentes tâches intervenant dans le cycle de réingénierie. On peut, par exemple, citer les expressions : « spéculer autour de la conception » et « capturer les détails du modèle » employées pour nommer les patrons décrivant deux tâches de la phase de rétroconception.

Dans ces travaux, les actes de spécification, modélisation et structuration sont instanciés par les phases d'ingénierie, de rétroconception et de restructuration du cycle de réingénierie du logiciel. Les patrons orientés processus, parce qu'ils s'adressent aux acteurs de la réingénierie en décrivant et en guidant leurs différentes tâches, constituent l'instanciation du *point de vue métier* d'ODP-RM sur la réingénierie du logiciel. Les patrons orientés structure instancient les *points de vue ingénierie*, *computationnel* et *technologique* en décrivant une solution technique de réingénierie. Le *point de vue informationnel* d'ODP-RM n'est pas pertinent dans cet exemple, puisque le propos même des patrons pour la réingénierie du logiciel est d'être indépendant des modèles informationnels gérés par l'application.

ODP-RM se présente alors comme un cadre générique capable de mettre en relation ces résultats sur la réingénierie du logiciel avec les modèles de spécification issus des travaux sur les technologies éducatives normées, constituant une instanciation du *point de vue informationnel*.

3.3.2 Instanciation d'ODP-RM sur le processus de développement mettant en œuvre les technologies éducatives normées

Les trois propositions majeures de normes et standards des technologies éducatives que nous retiendrons sont :

- Le standard de description LOM (Learning Object Metadata) du comité IEEE. Ce standard permet de décrire une ressource pédagogique afin qu'elle puisse être référencée dans une banque distribuée de *ressources*, et réutilisée dans une session d'apprentissage. Il se présente sous la forme d'un méta-descripteur permettant de donner une sémantique pédagogique à une *ressource* distribuée.
- Les modèles informationnels du consortium IMS permettent de spécifier les différents aspects du déroulement d'une session d'apprentissage. Les formalismes utilisés imposent une rigueur syntaxique aux concepteurs.
- Les spécifications techniques de SCORM (Sharable Content Object Reference Model) d'ADLNet et de « Abstract Framework » de l'IMS permettent aux concepteurs de spécifier à un niveau abstrait, indépendant de toute technologie, le système qu'ils souhaitent mettre en place.

Le concepteur exploitant ces différents travaux met en œuvre les actes de modélisation, de spécification et de structuration définis par ODP-RM. Un travail de modélisation va produire une *instance* de description LOM pour partager, en la caractérisant, une ressource pédagogique avec la communauté de conception. Une spécification, utilisant un ou

plusieurs *langages* proposés par le consortium IMS, décrit le déroulement d'une session de formation. Une structuration au niveau du contenu exploité en sessions de formation ou au niveau des composantes logicielles aide le concepteur à améliorer la modularité et la réutilisation de son système de formation.

Les différents chantiers entrepris par les comités de standardisation, les consortiums, les communautés scientifiques et industrielles créent une grande synergie au sein de l'ingénierie normée des EIAH. Tous cherchent à définir des formalismes traitant d'un aspect spécifique d'un système de formation tout en affirmant garantir une neutralité pédagogique par rapport au contexte des situations modélisées. Cette notion est à rapprocher du concept de *point de vue* du cadre ODP-RM. Ce dernier permet de structurer dans un même cadre différentes considérations sur un système. En analysant les différentes propositions de technologies éducatives normées, nous identifions une *instance* des différents *points de vue* d'une description de l'architecture d'un système suivant ODP-RM.

- Les EMLs (Educational Modelling Languages) (Rawlings, Rosmalen et al. 2002; Barré 2004), en particulier « Learning Design » de l'IMS, définissent une *instance* du *point de vue informationnel*. Ces langages permettent de spécifier un scénario pédagogique en mettant en relation des concepts informationnels. Les contraintes associées sont définies par les modèles informationnels, comme par exemple ceux diffusés par le consortium IMS.
- Les propositions SCORM d'ADLNet et « Abstract Framework » de l'IMS décrivent le *point de vue ingénierie* en effectuant une synthèse sur la spécification du cadre technique d'un système de formation à distance. Le consensus adopté dans le projet de l'IMS fédère à la fois des projets de la communauté du logiciel libre (Grob, Bensberg et al. 2004), des spécifications produites par des industriels[22] et des modèles standardisés (IEEE/LTSA 2003). Le développeur doit ici s'employer à respecter les *interfaces* et le *comportement* des différentes composantes d'un système de formation.
- L'un des trois documents produit par les projets du consortium IMS, intitulé « Best Practice and Implementation Guide », décrit la décomposition fonctionnelle du système de formation en explicitant les liaisons, les *interfaces* et les *interactions* entre les objets. Plus ancienne, la proposition LTSA (Learning Technology Systems Architecture) de l'IEEE vise à la définition d'un modèle abstrait explicitant les composantes fonctionnelles et les différents *flux* d'un processus cyclique centré apprenant. L'ensemble de ces différents travaux contribue à définir le *point de vue computationnel*.

Instancier le cadre ODP-RM sur les technologies éducatives normées comme nous venons de le faire, permet de faire de ressortir les manques actuels de ces travaux. L'objectif initial des technologies éducatives normées est de

[22] Voir, http://www.sun.com/products-n-solutions/edu/whitepapers/pdf/framework.pdf

répondre aux problématiques d'interopérabilité entre les dispositifs de formation et de réutilisabilité des productions. Comme une conséquence, elles instrumentent essentiellement l'acte de spécification du concepteur en négligeant la dimension itérative de la conception des composantes d'un système de formation. Notamment, l'instanciation d'ODP-RM sur l'ingénierie normée des EIAH nous montre clairement l'absence d'une réflexion méta sur le cycle de vie du système de formation, sur les *langages* employés dans chaque *point de vue*, et sur les interrelations entre les *activités* de chaque *acteur* du processus. C'est pour ces raisons que nous utilisons le point de vue métier du cadre ODP-RM pour représenter le domaine de la réingénierie dans les EIAH.

3.3.3 Point de vue métier d'ODP-RM instancié sur la réingénierie dans les EIAH

Une approche qualité au sens d'ODP-RM porte sur l'organisation des acteurs, des documents et des étapes du cycle de vie. La qualité se définit par les retours quantitatifs des usages observés mais également par la recherche d'indicateurs qualitatifs manquants [(ISO/IEC-10746-1 1998), partie 9.1].

L'*instance* du cadre ODP-RM sur l'ingénierie normée des EIAH montre l'absence du *point de vue métier*. Or, ce dernier permet d'expliciter dans un cadre unificateur les dimensions organisationnelles liées aux itérations d'une démarche qualité et aux intentions des *acteurs* de l'*instance* d'un tel *point de vue*. De plus, les aspects *métiers* propres au système de formation ne sont pas pris en compte actuellement dans les différents projets sur les technologies éducatives normées. Le travail présenté dans (Pawlowski 2002a) en constitue une tentative mais se limite à synthétiser les différents projets d'assurance qualité des systèmes de formation. De plus, mené indépendamment des projets sur les technologies éducatives, ce travail ne vise pas à définir un cadre unificateur gérant les aspects qualitatifs. Les travaux présentés dans (Oubahssi, Grandbastien et al. 2004) cherchent quant à eux à définir un nouveau méta-descripteur pour permettre aux auteurs de décrire tout type d'*objet* pédagogique durant toutes les phases du processus de conception, d'utilisation et d'analyse d'un système de FOAD. Les concepts liés au *point de vue métier* ne sont cependant pas pris en compte et ne s'inscrivent pas dans les perspectives décrites par les auteurs. Seules les *instances* sur les *points de vue computationnel* et *informationnel* sur une plate-forme de formation à distance sont actuellement prises en compte.

Nous proposons d'utiliser le cadre ODP-RM pour fédérer les contraintes d'un cycle de réingénierie mettant en pratique les technologies éducatives normées. Dans ODP-RM, tout *objet* est considéré comme actif et est le siège de décisions. Ses *actions* internes ou ses *interactions* avec son *environnement* se doivent d'être modélisées. Un *objet métier* définit l'entité de base d'une spécification *métier* pour accomplir au moins un *rôle métier*. Un ensemble de concepts est fourni pour décrire le modèle *métier* d'un système de formation. Nous reprenons la terminologie décrite dans (ISO/IEC-15414 2002) pour spécifier un tel *point de vue* :

- *Rôle* : Définit le premier niveau d'abstraction de l'*interface* du *comportement* d'un *objet métier* ou d'une *communauté* [(ISO/IEC-15414 2002), sous-partie 6.3.4].
- *Communauté* : Regroupe un ensemble d'objets *métier* ayant des *objectifs* communs [(ISO/IEC-15414 2002), partie 6.2].
- *Processus* : Définit un ensemble d'*étapes* agencées de manière prédéfinie et conduisant à un *objectif* [(ISO/IEC-15414 2002), sous-partie 6.3.5].
- *Etape* : Définit l'abstraction d'une *action* dans un *processus* [(ISO/IEC-15414 2002), sous-partie 6.3.6].
- *Objectif* : Qualifie une *communauté* en précisant ses intentions [(ISO/IEC-15414 2002), sous-partie 6.2.1].
- *Artefact* : Qualifie un *objet métier* référencé dans une *action* [(ISO/IEC-15414 2002), sous-partie 6.3.2].
- *Acteur* : Qualifie un *objet métier* participant à une *action* [(ISO/IEC-15414 2002), sous-partie 6.3.1].
- *Ressource* : Qualifie un *objet métier* nécessaire à son *comportement* [(ISO/IEC-15414 2002), sous-partie 6.3.3].

La proposition faite ici vise à inscrire à moyen terme les technologies éducatives dans la problématique de réingénierie dans les EIAH. Le modèle présenté sur la Figure 31 reprend les conventions graphiques du profil UML pour ECA (Entreprise Collaboration Architecture) présentées par le groupe OMG (OMG/ECA 2004). Il est destiné à supporter un processus de développement dirigé par les modèles utilisant le cadre ODP-RM (Nagase, Hashimoto et al. 2004). Nous reprenons les règles de transcription graphique associées aux termes du cadre ODP-RM. Dans la sous-partie 4.3.4 du chapitre suivant, page 130, nous présentons ce méta-modèle.

En appliquant la terminologie et les règles du *langage* de spécifications sur le *point de vue métier*, nous proposons de spécifier notre vision d'une *communauté métier* de réingénierie d'un système de formation. En reprenant la première *instance* du *point de vue métier* présentée dans la partie 2.3 du chapitre précédent, nous décrivons les six *processus métier* de cette nouvelle vision du *point de vue métier*.

- Le *processus* de conception produit des spécifications en étant guidé par les structures des principaux modèles informationnels. Il les adapte en fonction des ressources pédagogiques existantes, des différents profils d'apprenants et du retour d'analyse sur le *comportement* du *processus* d'apprentissage.
- Le *processus* génie logiciel réutilise les différentes composantes de l'environnement de formation. Il met en œuvre une démarche de réingénierie sur ces composantes. La négociation avec le *processus* de conception permet de préciser les observables et d'intégrer les sondes logicielles.
- Le *processus* d'apprentissage supporte le déroulement du scénario. Sa spécification repose sur la définition des *activités* effectuées par chacun des *acteurs* de la session d'apprentissage.

- Le *processus* d'analyse met en œuvre les techniques d'analyse des utilisations, automatiques ou non. En négociation avec le *processus* génie logiciel, de nouvelles combinaisons d'observables peuvent être établies. Des retours d'analyse structurés sont présentés au *processus* de conception.
- Le *processus* de gestion des ressources administre les *ressources métier* de type *objet* pédagogique.
- Le *processus* de gestion des profils apprenants administre les *ressources métier* des *processus* d'analyse et de conception du cycle de réingénierie.

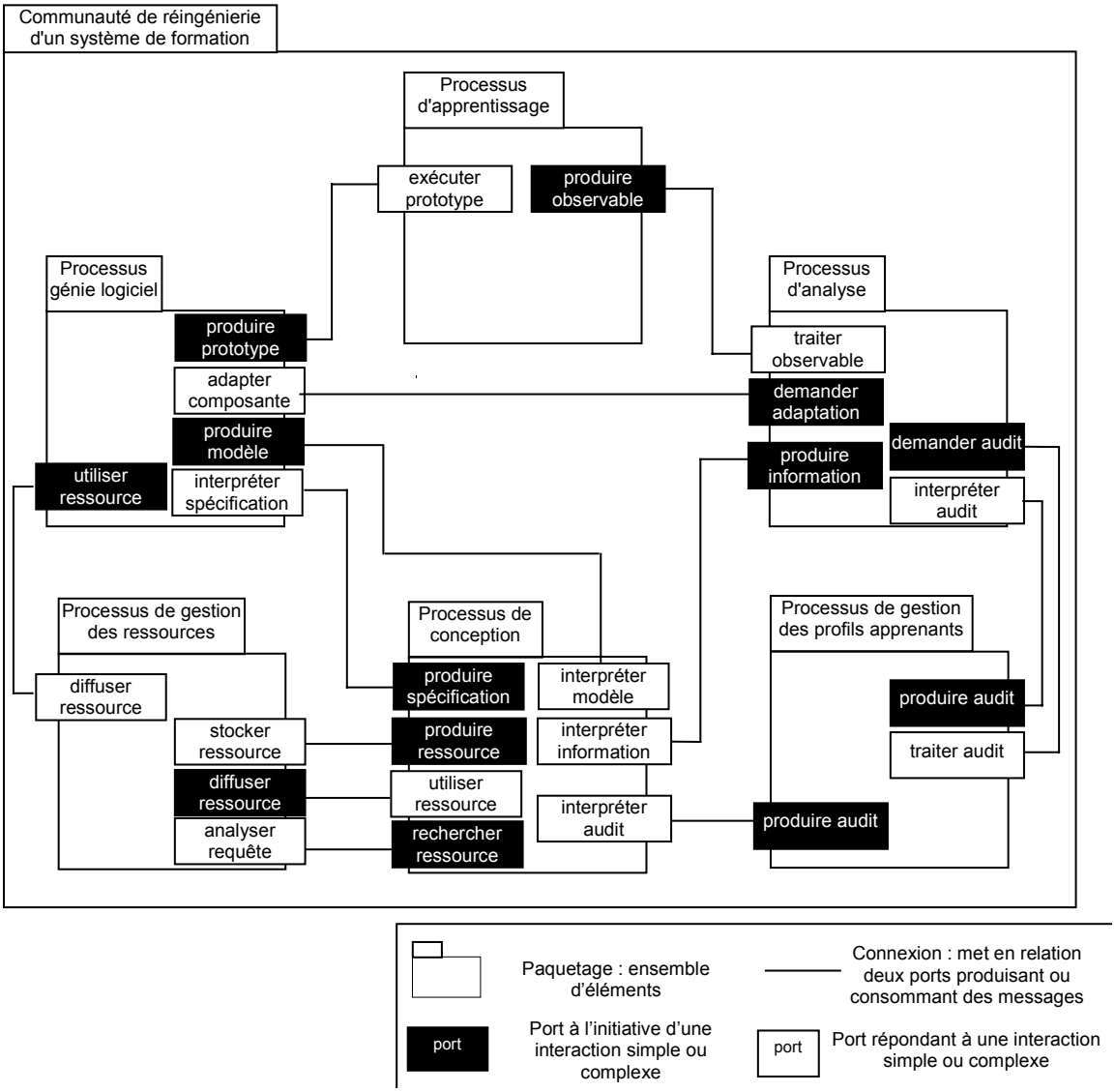


Figure 31 : Diagramme de paquetage UML représentant le processus de réingénierie d'un EIAH (OMG/ECA 2004)

Chaque *processus* prend part à l'*action* liée à la démarche de réingénierie d'un système de formation. En effectuant des regroupements, ils peuvent être considérés de manière plus générale comme des *communautés* d'objets *métier*. Cette nouvelle considération permet de s'interroger sur les *objectifs*, les *rôles* et les *processus* de ces différentes *communautés*.

3.4 Exemples

3.4.1 Présentation du contexte général

Les deux exemples présentés sont extraits des mises à l'essai effectuées sur le site de l'IUT de Laval. Au cours de ces quatre dernières années, une cinquantaine d'étudiants a participé à chacune d'elles. Nous disposons d'un contexte d'usage des outils produits et diffusés par les communautés du logiciel libre. Nous avons ainsi un cadre d'analyse afin de comprendre et de formaliser les premiers éléments méthodologiques d'une démarche de réingénierie d'un système de formation.

La session d'apprentissage forme les apprenants aux principes de fonctionnement d'un serveur HTTP. Elle appartient au module de la formation « Systèmes et réseaux de communication » du programme pédagogique du département Services et Réseaux de Communication. Le scénario, organisé sous la forme de plusieurs activités, laisse libre l'apprenant dans son parcours.

- Introduire l'unité d'apprentissage : Sous la forme d'une vidéo, les principes généraux sur le serveur web sont exposés. Cette activité a pour objectif de présenter l'ensemble des concepts nécessaires à la réalisation de l'unité d'enseignement pour inciter et motiver l'apprenant.
- Approfondir les concepts : Cette activité présente un contenu textuel et quelques schémas décrivant les principaux protocoles du web et les outils de programmation à utiliser.
- Synthétiser les concepts : Deux diaporamas sont présentés : l'un exposant les concepts liés au serveur web, l'autre décrivant l'exécution, étape par étape, du code Java d'un serveur.
- S'exercer pour comprendre : L'étudiant est amené à résoudre un ensemble de problèmes portant sur la programmation d'un serveur HTTP.
- S'auto-évaluer : Deux QCM permettent à l'apprenant de s'évaluer.
- Apprendre en faisant : L'apprenant met en pratique l'ensemble des connaissances du domaine et modifie, étape par étape, jusqu'à sa réalisation fonctionnelle, le code d'un serveur web.

Le *point de vue métier* du système de formation considéré se définit comme étant composé de six *processus* : apprentissage, analyse, conception, génie logiciel, gestion des *ressources* et gestion des profils apprenants. Au sein de chacun d'eux sont réutilisés les composantes logicielles et opérationnelles issues des communautés ouvertes utilisant les technologies informatiques orientées objets. Le tableau suivant établit la liste des communautés de développement auxquelles nous avons adhéré pour instancier les six *processus* mis en œuvre dans nos mises à l'essai.

Processus	***Composantes logicielles***
Processus de conception	Reload[23], Mozilla[24], OpenOffice[25], OpenUSS[26], FreeStyle Learning[27] et Futur Learning Environment[28]
Processus d'apprentissage	Mozilla, OpenUSS, FreeStyle Learning et Futur Learning Environment
Processus génie logiciel	Eclipse[29], AndroMDA[30] et Poseidon CE[31]
Processus d'analyse	Weka[32] et WUM[33]
Processus de gestion de ressources	GForge[34] et CVS[35]
Processus de gestion des profils apprenants	OpenLDAP[36]

Tableau 1 : Composantes logicielles utilisées

Le premier exemple présenté traite plus spécifiquement de la rétroconception et le second de la réingénierie sur un système de formation.

3.4.2 Exemple 1 : Rétroconception sur un EIAH

3.4.2.1 Cinq étapes du cas illustré

Partant de la définition de la rétroconception logicielle au sens de [(Chikofsky et Cross 1990), page 15] où la finalité de la tâche du concepteur est « d'analyser un système afin d'identifier ses composantes et ses relations dans l'objectif de créer des représentations avec des formalismes variés ou à des niveaux d'abstraction différents », la rétroconception d'un EIAH dans le cadre ODP-RM se définit par les actes de mise en cohérence des différents modèles appartenant à des *points de vue* différents et amène le concepteur à spécifier avec plus de détails le système de formation. Nous considérons que le premier échange engagé entre le *processus* de conception et le *processus* génie logiciel du *point de vue métier* correspond à la première des cinq *étapes* de l'exemple présenté (cf. Figure 32).

[23] Voir, http://www.reload.ac.uk/
[24] Voir, http://www.mozilla.org
[25] Voir, http://www.openoffice.org/
[26] Voir, http://openuss.sourceforge.net/
[27] Voir, http://sourceforge.net/project/showfiles.php?group_id=6259&package_id=110628
[28] Voir, http://fle3.uiah.fi/
[29] Voir, http://www.eclipse.org/
[30] Voir, http://www.andromda.org/
[31] Voir, http://gentleware.com/
[32] Voir, http://www.cs.waikato.ac.nz/ml/weka/
[33] Voir, http://hypknowsys.sourceforge.net/wiki/The_Web_Utilization_Miner_WUM
[34] Voir, http://gforge.org
[35] Voir, http://ximbiot.com/cvs/
[36] Voir, http://www.openldap.org/

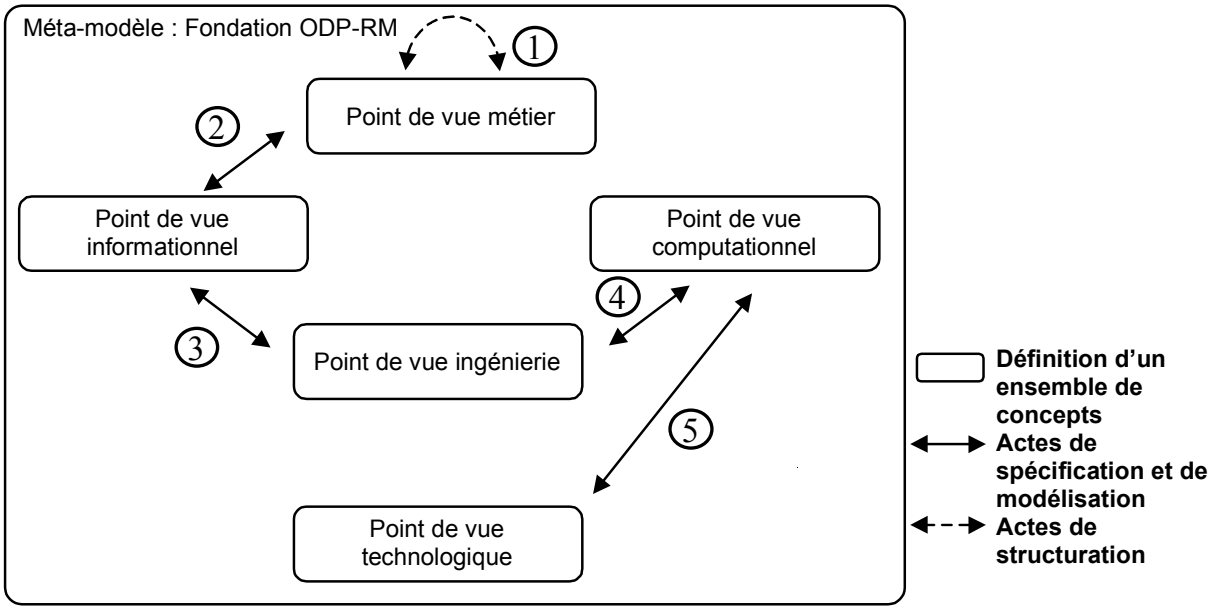


Figure 32 : Représentation des cinq étapes de l'exemple 1

3.4.2.2 Etape 1 : Point de vue métier sur la rétroconception sur un EIAH

La rétroconception d'un EIAH est représentée au niveau du *point de vue métier* par un dialogue entre les deux objets *métier* : le *processus* génie logiciel et le *processus* de conception (cf. Figure 33). Le regroupement de ces deux *processus* permet de spécifier une nouvelle *communauté* et permet de formaliser un premier modèle de cette négociation.

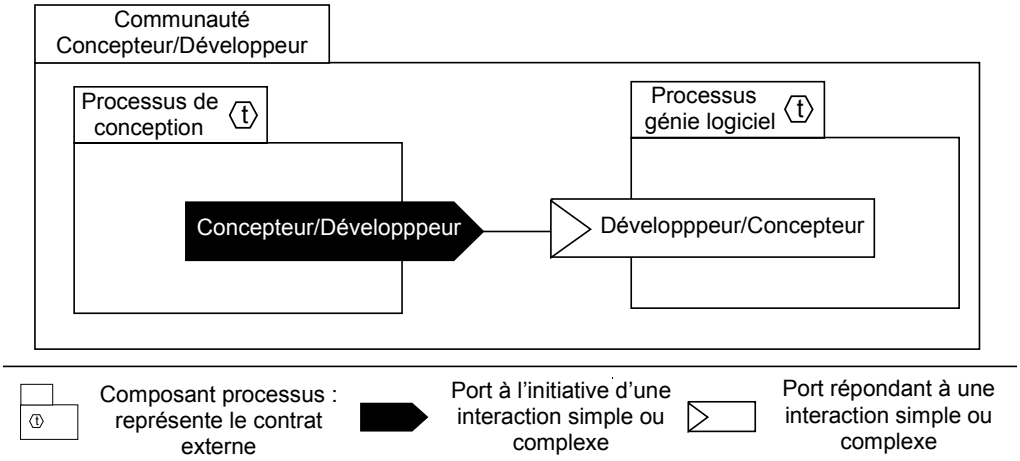


Figure 33 : Instance du modèle métier suivant le système de notation ECA (OMG/ECA 2004)

3.4.2.3 Etape 2 : Point de vue informationnel sur la rétroconception sur un EIAH

Le *rôle* du *processus* de conception, initiateur du dialogue, est de produire une spécification en créant une *instance* du *point de vue informationnel*. Nous utilisons les différents schémas *informationnels* diffusés par le consortium IMS. La Figure 34 est extraite du scénario pédagogique mis en œuvre lors de nos mises à l'essai. Elle représente l'environnement sélectionné par le concepteur et composé de deux ressources. L'objet « knowledge-type » est la page « author_text1.html » spécifiée à l'aide du langage HTML (Hypertext Markup Language) du consortium W3C et définit le contenu pédagogique. L'objet « tool-object » est le composant développé en technologie Java « manager.jar » qui gère la diffusion de la page. Le concepteur en spécifiant un contenu pédagogique (« knowledge-object ») explicite son souhait d'y associer un objet support (« tool-object ») (Koper 2001).

Description d'un environnement composé de deux ressources

```
<imsld:environment identifier="ENV-textStudy-e1">
  <imsld:learning-object identifier="LO-textStudy-e1" type="knowledge-object">
    <imsld:item identifier="ITEM-textStudy-e1" identiofierref="RES-textStudy-e1" />
  </imsld:learning-object>
  <imsld:learning-object identifier="LO-textStudyManagerFSL" type="tool-object">
    <imsld:item identifier="ITEM-textStudyManager" identiofierref="RES-textStudyFSL" />
  </imsld:learning-object>
</imsld:environment>
```

Description du lien entre la ressource et l'objet spécifique.

```
...
<imscp:resource identifier="RES-textStudy-e1" adlcp:scormtype="asset" type="webcontent"
href="/textStudy/author_text1.html">
  <imscp:file href="/textStudy/author_text1.html" />
</imscp:resource>
...
<imscp:resource identifier="RES-textStudyFSL" adlcp:scormtype="sco" type="">
  <imscp:file href="/learningUnitViewManagers/textStudy/manager.jar" />
</imscp:resource>
```

Figure 34 : Extrait de la spécification du scénario d'apprentissage

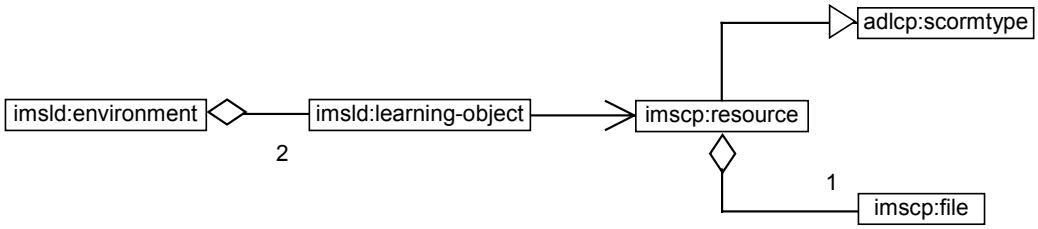
3.4.2.4 Etape 3 : Point de vue ingénierie sur la rétroconception sur un EIAH

La transmission de la spécification, émise par le *processus* de conception, met en œuvre une tâche de modélisation effectuée par le *processus* génie logiciel. Un travail préliminaire est effectué sur la spécification. En reprenant les directives du cadre ODP-RM, les *états* des différents objets d'une spécification *informationnelle* doivent être identifiés. Il est demandé d'identifier les trois *schémas* suivants [(ISO/IEC-10746-3 1996), partie 6.1] :

- *Schéma statique*, définissant l'*état* des objets *informationnels* à un instant donné ;
- *Schéma invariant*, représentant les relations entre les objets *informationnels* ;
- *Schéma dynamique*, explicitant le changement d'*état* des objets *informationnels*.

Le *schéma statique* de l'exemple présenté sur la Figure 35 associe différents éléments des modèles de spécification IMS-LD, IMS-CP et ADL-CP. Les objets *informationnel* imscp:file, imscp:resource, imsld:item,

imsld:environment, imsld:learning-object et adlcp:scormtype sont regroupés au sein d'une seule et même représentation. Le *schéma invariant* définit les liens entre ces différents objets. Il est représenté ici dans le formalisme graphique du diagramme de classe UML.



Ce schéma décrit les relations existantes entre les différents objets informationnels. L'objet **imsld:environment** est composé de deux objets **imsld:learning-object**. Chacun des objets **imsld:learning-object** fait référence à un objet **imscp:resource**. L'objet **imscp:resource** est de type **adlcp:scormtype** et est composé d'un objet **imscp:file**.

Figure 35 : Représentation du schéma des invariants par un diagramme classe UML

Néanmoins, aucun *schéma dynamique* n'existe, ce qui implique, pour le *processus* génie logiciel, d'engager une démarche de modélisation pour identifier de nouveaux objets *informationnel*, permettant l'établissement du *schéma dynamique*. Il s'agit ici de spécifier le *comportement* de ces différents objets.

3.4.2.5 Etape 4 : Point de vue computationnel sur la rétroconception sur un EIAH

Afin de spécifier le comportement dynamique, des détails liés à la conception vont être extraits. Le patron « Capturer les détails du modèle » (Demeyer, Ducasse et al. 2003) assiste une telle démarche de rétroconception du logiciel. Inscrit dans le *point de vue computationnel*, le *processus* génie logiciel cherche à obtenir une *décomposition* des fonctionnalités du système. L'objet SCORM de type « asset » (par exemple, une page au format HTML) est non sécable. Ce type d'objet sera simplement fourni à l'apprenant, aucun retour n'étant explicitement défini. Cependant, un objet SCORM de type « SCO » lui est associé. Il détient en son sein une structure d'objets décrivant son fonctionnement. Le cadre ODP-RM propose trois types d'*interfaces* supportant les interactions entre ces objets : l'*interface d'opérations*, l'*interface de flux* et l'*interface de signaux*.

Le paradigme de programmation événementielle utilisé par la technologie Java de Sun Microsystems pour concevoir l'interface utilisateur graphique (Loy, Eckstein et al. 2002) et par l'objet SCORM est à mettre en relation avec le concept de *signaux* du *point de vue computationnel* du cadre ODP-RM.

3.4.2.6 Etape 5 : Point de vue technologique sur la rétroconception sur un EIAH

Dans la perspective *computationnelle*, le *processus* génie logiciel doit identifier l'*état* des objets spécifiés par le concepteur pédagogique. Cette démarche consiste du *point de vue technologique* à intégrer le code correspondant aux sondes logicielles permettant de tracer l'*état* des différents objets *technologique* correspondants aux objets *informationnel*.

En reprenant le paradigme événementiel, il est alors nécessaire d'identifier les différents *objets* jouant les *rôles* d'initiateur ou de répondeur dans les *signaux* échangés. Les patrons de conception d'architecture utilisés pour concevoir l'interface utilisateur sont maintenant largement adoptés par la communauté de développement. L'un d'eux, le paradigme MVC (Modèle/Vue/Contrôle) (Buschmann, Meunier et al. 1996), est implanté dans la librairie java « swing/JFC »[37] (Java Foundation Classes) développée par Sun Microsystems. Chaque composante appliquant ce paradigme est un objet à observer.

La Figure 36 représente le résultat de la modélisation des deux ressources spécifiées par le concepteur pédagogique : la page HTML « author_text1.html » et le composant Java « manager.jar ». Le travail accompli a été guidé par le paradigme d'architecture logicielle MVC. La ressource au format HTML est gérée par l'objet « HTMLDocument » appartenant à la librairie « swing/JFC » qui applique rigoureusement la syntaxe définie par le schéma XML du format HTML du consortium W3C. La rétroconception sur le code du composant « manager.jar », diffusé par la communauté « FreeStyle Learning », montre que l'objet « FLGTestStudyElementsContentPanel » utilise l'objet JScrollPane de la librairie « swing/JFC ». Cet objet se décompose en trois objets « JTextComponent », « JViewPort » et « JScrollBar » représentant respectivement : le modèle, la vue et le contrôleur.

Deux nouveaux éléments sont ajoutés au modèle. L'objet « FLGHtmlObserver » contenant le code de la sonde logicielle. Un évènement, représenté par l'objet « AdjustmentListener », est déclenché à chaque changement de la barre de défilement de l'interface graphique diffusant la ressource HTML. L'objet « FLGHtmlObserver » capte alors l'*état* de l'environnement correspondant au contenu de la page réellement affiché sur l'interface. Deux nouvelles balises viennent étendre le schéma d'une page HTML afin de marquer dans le modèle de la page captée la zone affichée.

[37] Voir, http://java.sun.com/products/jfc

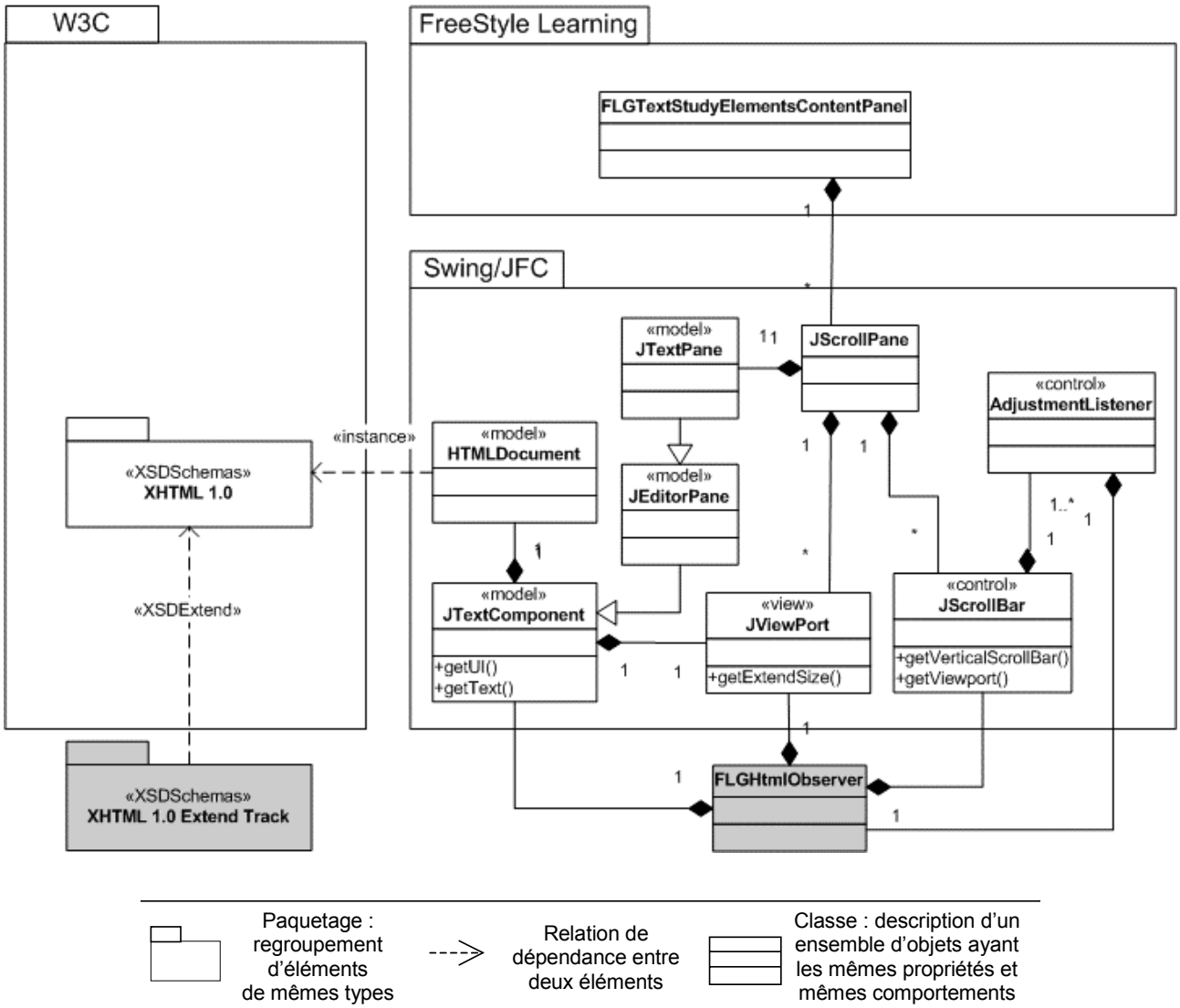


Figure 36 : Diagramme de classe UML obtenu suite au travail de rétroconception et intégrant la sonde logicielle

3.4.2.7 Bilan

Ce cas d'usage nous montre un exemple de tâche de rétroconception d'un EIAH initiée par le *processus* de conception et implique le *processus* génie logiciel. L'essentiel du raisonnement est effectué par ce *processus* qui cherche à identifier le *schéma dynamique* dans la spécification informationnelle fournie par le concepteur pédagogique. Une sonde est alors intégrée, d'un *point de vue technologique*, pour tracer le *comportement* lié à la diffusion d'un contenu pédagogique. Cette rétroconception s'inscrit dans un cycle où la finalité est d'amener les concepteurs à négocier sur (1) le scénario pédagogique par la prise en compte de la sémantique des *signaux* générés par les sondes, (2) l'objet pédagogique et la restructuration de son contenu et (3) la réingénierie logicielle de l'objet technique support à la diffusion afin de mettre à disposition de la *communauté* de concepteurs un *objet* disposant d'une *interface* permettant l'analyse de son utilisation.

D'un *point de vue métier*, cet exemple a produit un modèle qui est représenté sur la Figure 36. Cette production sert à apporter des précisions sur la description de l'*artefact métier* et à initier de nouvelles négociations entre les autres objets de la *communauté* de réingénierie d'un système de formation (cf. Figure 31, page 86).

3.4.3 Exemple 2 : Réingénierie sur un EIAH

3.4.3.1 Cycle de collaboration du cas illustré

Selon la définition de la réingénierie du logiciel au sens de [(Chikofsky et Cross 1990), page 15] où la finalité est d'effectuer les tâches « d'examen et d'altération d'un système afin de le reconstituer sous une nouvelle forme suivie de l'implantation de cette nouvelle forme », le travail de spécification des besoins, des solutions de conception et de la mise en œuvre du système s'inscrit dans une démarche cyclique et dirigée par des modèles. La réingénierie des EIAH se définit par le souhait du concepteur d'être guidé dans les changements à effectuer sur le modèle du système de formation, dans ses actes de spécification et de modélisation. Les concepts du cadre ODP-RM permettent d'expliciter ces modifications ainsi que les décisions de réingénierie prises.

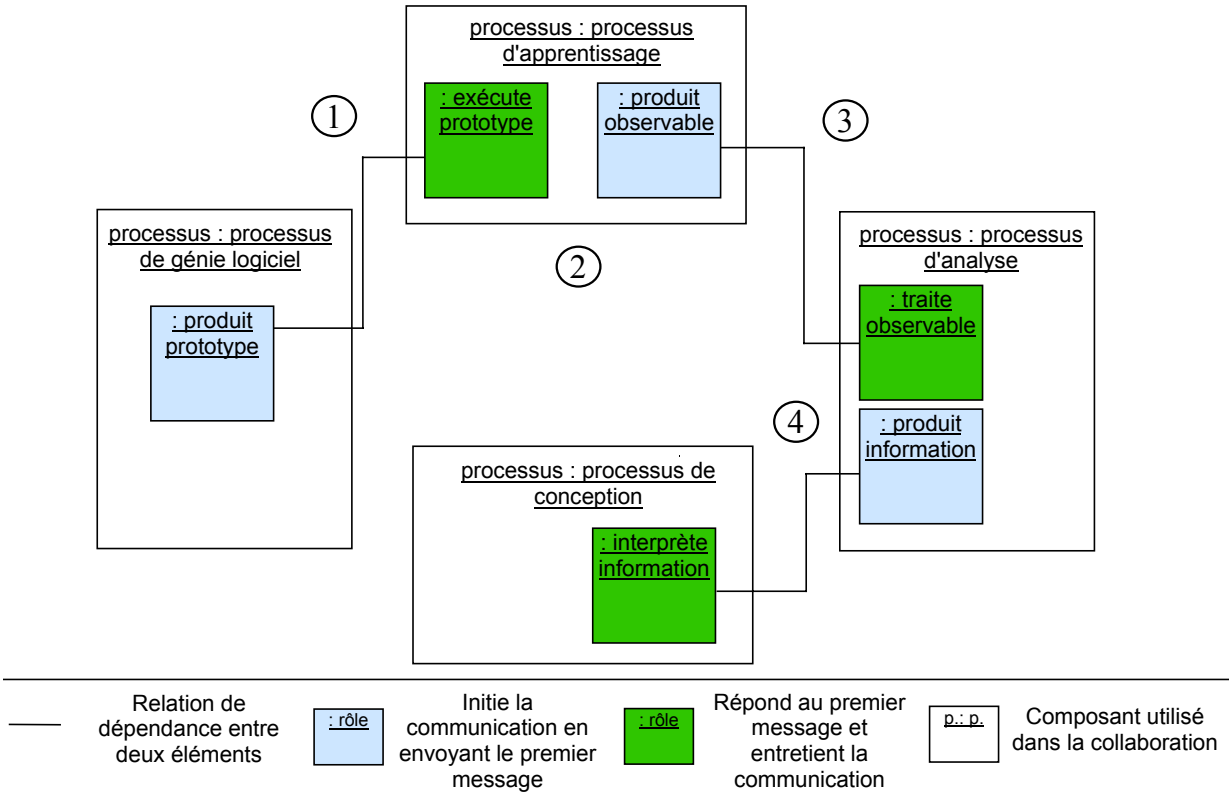


Figure 37 : Diagramme de collaboration UML (OMG/ECA 2004) du deuxième cas d'usage

Le cycle d'une *communauté* de réingénierie d'un système illustré par l'exemple se décompose en plusieurs étapes présentées sur le diagramme de collaboration de la Figure 37. Dans [(ISO/IEC-10746-2 1996), sous-partie 7.8.1],

le concept de *rôle* est défini comme une abstraction du *comportement* d'une *communauté*. Une *communauté* de réingénierie des systèmes de formation se définit par les différents *rôles* joués par chaque *processus métier*. L'exemple présenté utilise ce concept pour décrire un cas d'usage sur la réingénierie d'un EIAH.

3.4.3.2 Etape 1 : Rôle du processus génie logiciel sur la réingénierie d'un EIAH

Toute tâche de réingénierie induit par définition une tâche de rétroconception. Dans le cas d'usage présenté, elle est prise en charge par le *processus* génie logiciel appliquant le patron de rétroconception du logiciel « Spéculer autour de la conception » [(Demeyer, Ducasse et al. 2003), pages 76-84]. Des sondes sont intégrées afin de produire des observables liés aux décisions prises du *point de vue ingénierie*.

Dans l'exemple présenté, l'outil de déploiement utilisé est « JavaWebStart[38] » de Sun Microsystems. Suivant la terminologie du cadre ODP-RM, les objets *ingénieries* sont gérés par une entité intitulée *grappe*. Dans l'exemple présenté, le composant « learningUnitViewAPI.jar » correspond à cet objet et regroupe l'ensemble des objets du gestionnaire de diffusion des unités d'apprentissage « FreeStyle Learning » (Brocke 2001).

```
<jnlp spec="0.2 1.0"
      codebase="http://172.30.8.9/jnlp"
      href="fsl.jnlp">
   <information>
      …
      <description>FreeStyle Learning</description>
      …
   </information>
   <security>
     <all-permissions/>
   </security>
   <resources>
      …
      <jar href="learningUnitViewAPI.jar"/>
      …
      <jar href="learningUnitViewManagers/intro/manager.jar"/>
      <jar href="learningUnitViewManagers/textStudy/manager.jar"/>
      <jar href="learningUnitViewManagers/slideShow/manager.jar"/>
      <jar href="learningUnitViewManagers/caseStudy/manager.jar"/>
      <jar href="learningUnitViewManagers/learningByDoing/manager.jar"/>
      <jar href="learningUnitViewManagers/checkUp/manager.jar"/>
   </resources>
   <resources os="Windows">
      <nativelib href="ext-libs/jmfDll.jar"/>
      <nativelib href="Win32RegKey.jar"/>
   </resources>
```

Objet "Grappe"

Objets "Ingénieries"

Figure 38 : Extrait du script de déploiement du dispositif « FreeStyle Learning »

Les différentes composantes s'intègrent à ce gestionnaire et représentent les objets *ingénierie*. Chacun d'eux

supporte des activités différentes : visualiser une vidéo, consulter un support textuel, consulter un diaporama, résoudre des problèmes, répondre à des questions à choix multiples et pratiquer sur un outil logiciel. La *décomposition* des différentes composantes du système déployé sur le poste client (côté apprenant), est spécifié dans le script de déploiement (cf. Figure 38).

3.4.3.3 Etape 2 : Rôle du processus d'apprentissage sur la réingénierie d'un EIAH

Les précédentes considérations du *point de vue ingénierie* sont corrélées avec les spécifications du *point de vue computationnel* définies au niveau client par SCORM d'ADLNet et coté serveur par « Abstract Framework » de l'IMS. Dans le cas d'usage présenté, il s'agit des composants déployés sur le site client. En conséquence, la spécification SCORM explicite les différents *états* et les transitions des objets *ingénierie*. Chaque *état* de ces objets est tracé en respectant la terminologie de la spécification technique SCORM (cf. Figure 39).

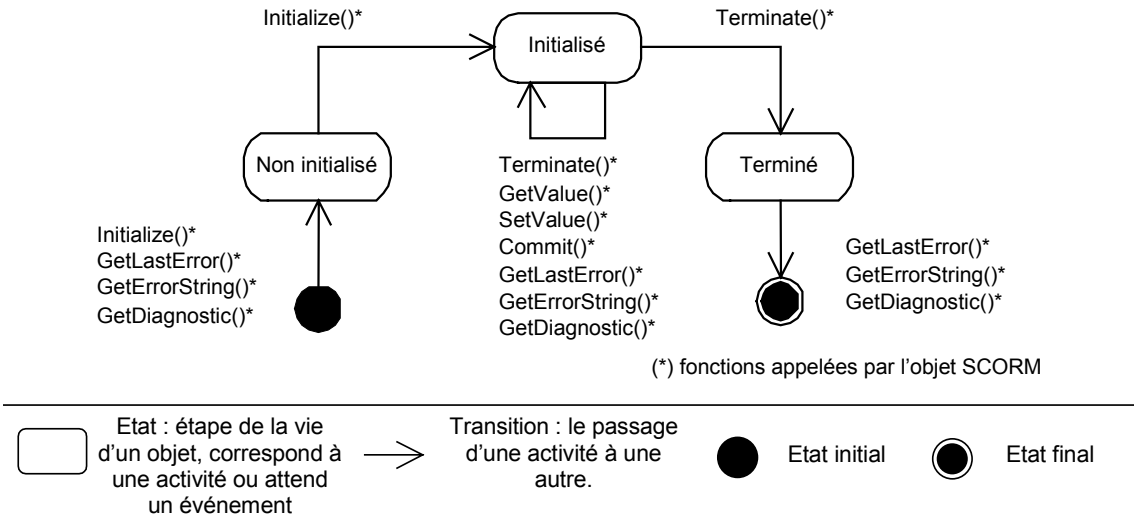


Figure 39 : Diagramme état-transition UML d'un objet SCORM [(ADL/SCORM 2004b), figure 3.1.6a]

3.4.3.4 Etape 3 : Rôle du processus d'analyse sur la réingénierie d'un EIAH

En entrée, le *processus métier* d'analyse reçoit les différents observables générés par le dispositif d'apprentissage. Chaque observable est caractérisé par un identifiant de session, la date de la capture, le paquetage auquel appartient l'objet *ingénierie* et l'intitulé de l'événement capté (cf. Figure 40).

[38] Voir, http://www.jcp.org/en/jsr/detail?id=56

```
…
<observer id="1139401753483" role="learner">
  <date>
    <element name="Format" value="ISO 8601"/>
    <element name="Value" value="31/03/2004:10:45:42 +0100"/>
  </date>
  <event>
    <element name="Package" value="freestyleLearning.learningUnitViewManagers.checkUp"/>
    <element name="Event" value="Initialize()"/>
  </event>
</observer>
…
<observer id="1139401753483" role="learner">
  <date>
    <element name="Format" value="ISO 8601"/>
    <element name="Value" value="31/03/2004:10:56:21 +0100"/>
  </date>
  <event>
    <element name="Package" value="freestyleLearning.learningUnitViewManagers.checkUp"/>
    <element name="Event" value="Terminate()"/>
  </event>
</observer>
…
```

Figure 40 : Exemple de trace générée par le composant « FreeStyle Learning »

Dans cet exemple, le *processus* d'analyse du *point de vue métier* effectue des opérations statistiques sur les observables. L'objectif n'est pas de valider des hypothèses mais d'identifier de nouvelles pistes. Cette fouille de données cherche à structurer le raisonnement dans un modèle logique à base d'arbres ou de règles (Lefébure et Venturi 2001). Il s'agit de permettre la communication autour de ces structures afin de faire émerger leurs sémantiques dans le domaine de la formation à distance. Ce *processus métier* s'intègre parfaitement dans le cadre ODP-RM qui partage le même objectif d'émergence de sens. Plus concrètement ; dans notre exemple le *processus* d'analyse cherche à identifier des règles associatives à partir du temps passé sur chaque objet et de l'enchaînement de ces derniers. Cependant, la recherche des associations de manière aléatoire est difficile à mettre en œuvre. L'analyste utilise alors des outils l'assistant dans son travail. Nous en citons deux en exemple : l'application « Weka[39] » (Waikato Environment for Knowledge Analysis) mettant à disposition un ensemble de fonctionnalités de fouille de données et l'application « WUM[40] » (Web Utilization Miner) effectuant des analyses sur les données résultant des consultations des ressources sur un serveur Web. Le *langage* utilisé est celui du *point de vue ingénierie*. Il permet de spécifier les différents *comportements* des différents composants.

[39] Voir, http://www.cs.waikato.ac.nz/ml/weka/
[40] Voir, http://munch.wiwi.hu-berlin.de/hypknowsys/wum/

3.4.3.5 Etape 4 : Rôle du processus de conception sur la réingénierie d'un EIAH

A l'initiative de l'analyste, une négociation est effectuée entre le *processus* de *conception* et le *processus* d'*analyse*. La finalité est de mettre en correspondance les *instances* des concepts du *point de vue ingénierie* avec les concepts du *point de vue informationnel*. Dans l'exemple du cas d'usage présenté, le concept de *grappe* (regroupement d'objets *ingénieries*) est à mettre en relation avec les objets *informationnel* décrivant la structure des activités du scénario pédagogique. La Figure 41 montre l'évolution du *point de vue ingénierie*, en corrélation avec le *point de vue informationnel*.

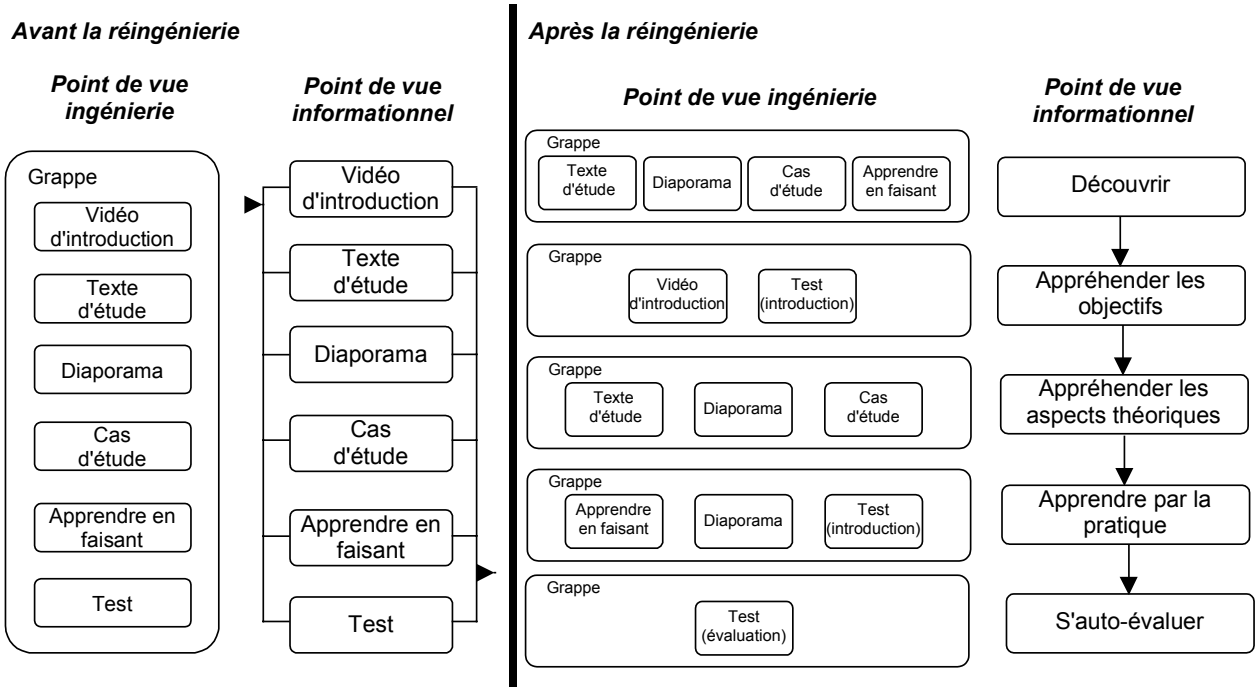


Figure 41 : Diagrammes état-transition UML des objets ingénierie et informationnel

Ces mises en correspondance entre les *points de vue* produisent des règles de conformité, par exemple le regroupement des *objets ingénierie* « vidéo d'introduction » et « Test (introduction) » correspond à l'*objet informationnel* « Appréhender les objectifs ».

3.4.3.6 Bilan

La tâche de réingénierie des EIAH illustrée dans ce cas d'usage fait intervenir les différents *rôles* des *processus métier* d'une « communauté de réingénierie d'un système de formation ». Le travail d'articulation des *points de vue* est au cœur du cas d'usage présenté. Le fait d'expliciter les correspondances entre les différents *points de vue* met en évidence des propositions à adresser aux différentes communautés :

- d'un *point de vue ingénierie*, une spécification de cinq nouveaux gestionnaires de *grappe* peut être adressée à la communauté « FreeStyle Learning ». Ils correspondent aux activités pédagogiques « Découvrir », « Appréhender les objectifs », « Appréhender les aspects théoriques », « Apprendre par la pratique » et « S'auto-évaluer ».
- le cas présenté peut être proposé comme un exemple d'*instance* du modèle « Learning design » du consortium IMS. Il viendrait compléter la base de cas gérée par le projet européen UNFOLD[41].

Dans le cadre ODP-RM, l'ensemble des mises en correspondance entre les *points de vue*, produit des règles de conformité spécifiant le cycle de réingénierie. Intégrées dans les fonctionnalités de la nouvelle version d'un environnement d'apprentissage ou dans une base de cas pratiques, elles s'adressent aux concepteurs de nouvelles suggestions. Derrière cette opportunité se définit la démarche qualité d'un système au sens d'ODP-RM [(ISO/IEC-10746-1 1998), partie 9.1].

3.5 Synthèse du chapitre

Dans ce chapitre, nous avons présenté un cadre permettant d'expliciter les mécanismes de transformations sur des modèles contenant des concepts liés au domaine de la formation et/ou aux aspects technologiques. Ces derniers utilisent des formalismes et des expertises différents suivant les communautés concernées. L'instrumentation des mécanismes de transformations entre – et au sein – des différents niveaux d'abstraction et l'explicitation des liens de composition et de décomposition définissent l'un des enjeux de la conception des EIAH (Corbière et Choquet 2005).

En inscrivant les propositions de technologies éducatives normées dans le cadre ODP-RM et en démontrant son potentiel à fédérer les expériences de formation, nous souhaitons sensibiliser les concepteurs à s'approprier un tel cadre. En effet, une mise à l'essai effectuée par un enseignant, une institution de formation ou une équipe de recherche se doit d'être mise à disposition de la communauté. La communauté du logiciel libre illustre parfaitement les perspectives de nos travaux car elle a su promouvoir à la fois une philosophie de conception (Raymond 2001), des outils [(Bar et Fogel 2003), pages 203-223] et des supports de communication[42]. Le concepteur ou l'utilisateur qui en est membre est pris dans une synergie où les ressources produites sont considérées comme des prototypes, par nature évolutives. L'analyse des liens existants entre la philosophie adoptée par la communauté du logiciel libre et le processus de développement de logiciels orientés objet démontre qu'il y a là une vraie dynamique de structuration d'une communauté. L'adoption du cadre proposé permet de comprendre et d'illustrer un telle synergie.

Voir, http://www.unfold-project.net/

[42] Site portail offrant un ensemble de services pour supporter les communautés du logiciel libre. Voir, http://www.sourceforge.net

CHAPITRE 4 : ANALYSE DE LA SYNERGIE ENTRE UNE APPROCHE DIRIGÉE PAR LES MODELES ET UN PROCESSUS DE DÉVELOPPEMENT LOGICIEL

Ce chapitre présente une analyse guidée par le cadre ODP-RM de la synergie entre une approche dirigée par les modèles et un processus de développement du logiciel orienté objet. Après avoir présenté les différents éléments théoriques auxquels nous faisons référence dans ce chapitre, nous exposons une synthèse de l'évolution des processus de développement. Elle nous amène à considérer dans notre analyse les productions de trois communautés du logiciel libre. Nous utilisons le cadre ODP-RM pour comprendre, exposer et formaliser l'activité de spécification, de structuration et de modélisation des concepteurs de ces trois communautés. Pour finir, nous présentons deux exemples de réflexion sur le travail des concepteurs de deux communautés. Cette analyse permet à la fois d'illustrer l'utilisation du cadre ODP-RM et de réifier le *point de vue métier* d'un système de formation en intégrant le travail d'analyse des deux composants produits par ces communautés de pratique.

4.1 Analyse suivant la notion de rétroaction d'un système de formation

L'analyse présentée dans ce chapitre se positionne suivant le concept théorique de « rétroaction » caractérisant la démarche systémique où on ne considère plus une relation entre deux éléments A et B comme une simple relation mais comme une double action de A vers B et de B vers A (Durand 1979). On parle alors de boucle de rétro-action, dont l'objectif est d'informer le concepteur afin qu'il puisse prendre connaissance d'une cause et agir en conséquence. Ce concept nous amène à nous intéresser à la finalité du système de formation et non à définir *a priori* les objectifs auxquels il doit répondre. La modélisation d'un système ne consiste pas à formaliser le problème déjà posé mais à formuler le ou les problèmes qu'il s'avère pertinent de résoudre : il faut apprendre à résoudre le problème qui demande d'abord à poser ce problème [(Le Moigne 1990), page 66]. Ce principe a également été développé dans les théories des communautés de pratique (Wenger 1998), les théories de l'instrumentation (Rabardel 1995) et l'éducation et les technologies de l'information et de la communication (Linard 2002).

Le concepteur d'un système de formation est alors amené à choisir deux alternatives : soit il contrôle toutes les propriétés, applique les modèles proposés par les consortiums de standardisation et ne prend pas en compte les autres acteurs, soit il considère l'outil d'apprentissage comme un système qui est amené à être adapté et à évoluer dans une communauté d'acteurs qui est socialement située (Linard 2002). Dans notre approche, nous n'envisageons que la

deuxième alternative. En effet, le concepteur se doit de répondre aux besoins de l'apprenant (Linard 2001), d'engager des négociations sur les termes utilisés ou les expériences passées et de mettre en œuvre les trois fonctions d'une approche instrumentale : analyser, concevoir et former. L'approche psycho-cognitive de (Rabardel 1995) considère que tout artefact amène les acteurs à se former et à pratiquer. Ce même auteur définit la notion d'artefact en donnant la possibilité d'analyser de manière différente les relations qui lient le sujet à l'objet. Ceci inclut les objets matériels mais également les objets symboliques. De plus pour les communautés de pratique théorisées par [(Wenger 1998), pages 57-62], cette notion est reprise sous la dénomination de « réification » que l'auteur articule avec la « participation ». Toute communauté de pratique produit des abstractions, des outils, des symboles, des termes et des concepts qui se projettent sur la réalité. Un processus mettant en œuvre une réification inclut la conception, la représentation, le nommage, l'encodage et la description, aussi bien que la perception, l'interprétation, l'utilisation, la réutilisation, le décodage et le reclassement.

La réification dans un processus de développement d'un système de formation met en œuvre des outils, des formalismes et des acteurs. La question est d'identifier les « schèmes de participation », les artefacts articulant les outils et les personnes, et le moment où ils se produisent.

4.2 Evolution du processus de développement logiciel

Afin d'identifier les outils, les formalismes et les acteurs d'un processus de développement, nous exposons les différentes contraintes considérées dans un processus génie logiciel pour présenter ensuite la perspective d'évolution dans laquelle les concepteurs sont positionnés au cœur de la démarche de conception.

4.2.1 Processus de développement génie logiciel

Quatre dimensions sont couramment adoptées pour définir un processus de développement logiciel.

La première concerne la gestion du développement exprimée en terme de coût et de délai. Les travaux de (Jacobson, Booch et al. 2000) définissent un ensemble d'activités organisées en phases clairement définies pour transformer en produits les besoins exprimés par les utilisateurs. Les acteurs considérés dans un tel processus ont des « rôles » définissant leurs responsabilités et leurs engagements sur les activités au début de chaque projet.

La seconde se focalise sur la mise en commun d'un ensemble de termes à partager entre les experts exprimant leurs besoins, les ergonomes produisant leurs prototypes, l'architecte informatique ayant partitionné le système et les composants utilisés de manière interne en conception. La méthode proposée par (D'Souza et Wills 1999) définit un processus de développement et « décrit comment les artefacts peuvent être structurés, exploités et déduits sur un projet

particulier ». Un tel processus s'inscrit dans une architecture dans laquelle les artefacts sont articulés dans différentes vues. Le processus 2TUP défini par (Roques et Vallée 2004) répond à ces préoccupations.

La troisième est liée au langage utilisé. Le formalisme UML (OMG/UML 2005) est adopté par la plupart des communautés de développement logiciel orienté objet. Le groupe OMG est chargé de diffuser et de faire évoluer les spécifications de ce langage. Un ensemble de formalismes standardisés a été produit autour d'UML (XMI, MOF, …) pour répondre à la fois au problème d'interopérabilité entre les plates-formes de développement logiciel et au besoin d'étendre le langage UML. Il a, par ailleurs, adopté des spécifications sur les plates-formes techniques (CORBA, J2EE/EJB, …) à destination des processus de développement de l'industrie centrées sur l'architecture.

La quatrième dimension est actuellement initiée par ce même comité qui adopte une architecture, intitulée MDA™, dont l'objectif est de définir les problématiques d'interopérabilité entre les aspects techniques et les aspects liés à des domaines spécifiques. L'objectif à moyen terme est d'y répondre en proposant de nouveaux langages grâce, par exemple, à l'appel à propositions sur la définition d'un formalisme de transformations QVT RFP (OMG/QVT 2003). Ce nouveau langage est au cœur des récentes problématiques scientifiques sur l'ingénierie dirigée par les modèles (Bézivin 2004). En effet, la volonté est d'inscrire cette démarche dans un système de communication où les négociations révèlent de nouveaux besoins sur la formalisation des spécifications techniques mais également les liaisons existantes entre des domaines particuliers.

En considérant ces quatre dimensions comme complémentaires dans un processus de développement logiciel, le concepteur est amené à manipuler à la fois des règles de bonnes pratiques, des principes pour favoriser la réutilisabilité et des formalismes pour communiquer. Mais cette vision est trop restrictive car elle vise avant tout à définir un ensemble limité de langages et elle s'adresse aux concepteurs maîtrisant leurs sémantiques et les environnements de développement les mettant en œuvre.

4.2.2 Alternative aux processus de développement génie logiciel

Une alternative aux processus de développement génie logiciel intitulée « processus de développement agile » (Cockburn 2001) se positionne suivant une culture organisationnelle (Highsmith 2002). Dans (Highsmith 2000), l'auteur met en opposition le cycle de vie statique « planification/conception/réalisation » avec le cycle de vie dynamique « spéculation/collaboration/apprentissage » (cf. Figure 42)

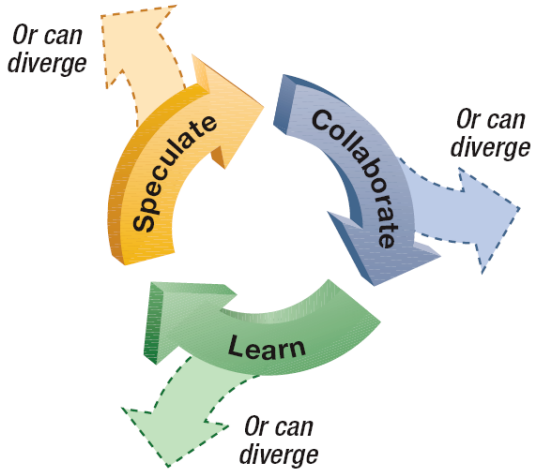


Figure 42 : Cycle de vie adaptatif (Highsmith 2000)

Une méthode de développement qualifiée d'« agile » répond aux principes suivants (Cockburn 2001) :

- Les individus et les interactions priment sur les processus et les outils : la méthode prend en compte les compétences des individus et les synergies les mettant en relation.
- Le travail des concepteurs prime sur la quantité de documentation : la méthode privilégie la compréhension du travail au travers des réalisations des concepteurs et minimise la documentation rédigée au cours du processus. La qualité du travail des acteurs tant au niveau des phases d'analyse, de conception que du codage est au cœur d'une telle méthodologie.
- La collaboration des acteurs prime sur la négociation de contrats : cette méthode ne veut pas imposer une démarche aux concepteurs mais les amener à considérer leurs expertises *métier* en engageant des actes de collaboration.
- La prise en compte du changement prime sur le suivi d'un plan : cette méthode se veut être adaptative et non prédictive.

C'est bien du point de vue de l'activité des concepteurs que nous souhaitons analyser la mise en relation entre une approche dirigée par les modèles et un processus génie logiciel orienté objet. En choisissant d'analyser les composantes d'un système de formation diffusées par les communautés du logiciel libre, nous voulons considérer l'impact de cette nouvelle alternative sur le processus de développement d'un système de formation. Les modèles de spécification sur les technologies éducatives sont également considérés comme influents dans les activités des concepteurs intervenant dans le cycle de vie du développement.

4.3 Utilisation du cadre ODP-RM pour définir une ingénierie des EIAH selon une approche dirigée par les modèles

Les deux productions mises à disposition par des communautés du logiciel libre et les différents modèles de spécification sur les technologies éducatives ne doivent pas être simplement présentées comme de simples outils mais doivent amener les concepteurs à comprendre et à formaliser les activités de conception qu'ils supportent. En effet, les actes d'expérimentation, d'analyse, d'adaptation et de collaboration des concepteurs dans un système de formation ne sont pas assez explicités et se doivent d'être communiqués.

Les concepts génériques définis par le cadre ODP-RM (ISO/IEC-10746-2 1996) et le système de notation défini par le profil UML EDOC (OMG/ECA 2004) du groupe OMG (présenté dans la sous-partie 4.3.4, page 130) sont les moyens que nous avons choisis pour analyser l'utilisation des modèles de spécification sur les technologies éducatives et les outils produits par ces communautés. Intégrer ces modèles et ces productions dans le cadre ODP-RM nous permet de rendre explicite le travail de modélisation et de spécification des concepteurs à une large communauté jusque là réduite aux seuls membres d'une communauté de développement du logiciel libre et d'une institution définissant les futurs standards sur les technologies éducatives. A notre sens, le travail présenté définit une ingénierie des EIAH selon une approche dirigée par les modèles.

Cette sous-partie commence par présenter le travail de modélisation au sens du cadre ODP-RM sur les outils diffusés par trois communautés. La notion d'*objet* de chacune est analysée et présentée. L'objectif est de nous permettre de comprendre le processus de développement appliqué, d'expliciter et de formaliser leur organisation. Pour y répondre, nous présentons deux exemples. Le premier décrit le travail de modélisation et le second un travail de spécification du concepteur. A la suite de chacun de ces deux exemples, un travail de spécification du *point de vue métier* est effectué et présenté. Nous utilisons dans un premier temps les termes généraux du cadre ODP-RM sur les actes de spécification et de modélisation (ISO/IEC-10746-2 1996) pour ensuite utiliser une terminologie plus spécifique, correspondant au langage de spécification du *point de vue métier* du cadre ODP-RM, le système de notation défini par le profil UML EDOC (OMG/ECA 2004).

4.3.1 Utilisation du cadre ODP-RM pour modéliser les objets de trois communautés du logiciel libre

La notion d'*objet* dans le cadre ODP-RM a une signification particulière. Elle est caractérisée par le *comportement* et l'*état* de l'*objet*. En considérant les *actions* internes à l'*objet* ou les *interactions* avec son *environnement*, tout objet dans le cadre ODP-RM se doit d'être modélisé afin de préciser les *règles* d'usages des ressources et les limites des boucles de rétro-action avec leur *environnement*. D'ailleurs un ensemble d'*actions* associé

à des contraintes est décrit à l'aide d'un concept particulier le *comportement*. La description obtenue correspond au travail de modélisation de tout objet dans le cadre ODP-RM.

Nous l'avons effectué afin d'expliciter dans cette sous-partie le sens donné à un *objet* dans chacune des trois communautés choisies. Pour cela, nous nous plaçons suivant le niveau d'abstraction défini par le cadre ODP-RM, où les actes de spécification et de modélisation autour de ces *objets* peuvent être formalisés et illustrer les premiers éléments d'une démarche de conception d'un système de formation dirigée par les modèles. Ce travail a consisté à identifier les *interfaces* et les *états* des objets qui permettent par la suite de modéliser leurs *comportements* et à décrire les *traces* à observer leurs *interactions*. Pour cela, nous avons analysé soit le code de l'application, soit les cadres techniques référents, soit les modèles de spécification standards mis en œuvre.

4.3.1.1 Présentation de trois outils pour un système de formation

Nous avons adhéré à trois communautés du logiciel libre. Elles diffusent trois outils :

- une plate-forme de formation à distance « Open University Support System » (OpenUSS).
- un gestionnaire de diffusion de contenu « FreeStyle Learning » (FSL).
- un dispositif support aux activités collaboratives « Future Learning Environment » (FLE)

Elles se présentent clairement comme des communautés au sein desquelles de nombreux échanges sont engagés entre les concepteurs et les utilisateurs. Ces derniers touchent à la définition et à l'utilisation des modèles proposés par les comités travaillant sur la spécification de modèles pédagogiques standards. Pour exemple, la plate-forme « Open University Support System » (OpenUSS) est citée comme référent dans le travail visant à définir un cadre technique de l'architecture de plate-forme mené au sein du consortium IMS (Smythe 2003). Depuis la version 2.0, cette plate-forme intègre une mise en œuvre de la spécification SCORM d'ADLNet (ADL/SCORM 2004a). La communauté « FreeStyle Learning » (FSL) (Brocke 2001) propose un composant de la plate-forme OpenUSS prenant en charge la gestion du contenu centrée apprenant. Certains formalismes proposés par le consortium IMS peuvent être utilisés pour spécifier les logiques fonctionnelles des différents composants du gestionnaire. La communauté « Future Learning Environment » (FLE) (Leinonen, Virtanen et al. 2002) propose une plate-forme mettant en œuvre des théories constructivistes et intègre de manière explicite le formalisme « Learning Design » du consortium IMS.

De même, dans chacune des communautés choisies, des décisions de conception sont communiquées dans la documentation, dans la structure du code ou dans le code source de l'application. Le dispositif FLE s'appuie sur la plate-forme ZOPE[43] (Deckmyn et Grizel 2003) qui propose un ensemble de technologies qui réalise par assemblage de

[43] Voir, http://www.zope.org/Documentation/Books/ZopeBook/

composants des applications collaboratives accessibles depuis le web. OpenUSS[44] et FSL mettent en œuvre de manière explicite une approche par composant logiciel. Ce dernier se définit comme une entitée encapsulée et autonome de déploiement (Szyperski 2003). Les modèles, qui correspondent à des patrons, à des standards ou à des cadres techniques, sont mis en œuvre dans ces applications et explicitent ainsi les choix des concepteurs.

Cette présentation, techno-centrée, nous amène à détailler cette première synthèse afin d'expliciter les actes de structuration, de modélisation et de spécification accomplis par les concepteurs.

4.3.1.2 Modélisation de l'objet pour la communauté OpenUSS

Afin de guider notre interprétation de la notion d'*objet* pour cette communauté, nous avons identifié le modèle de spécification de composants coté serveur choisi : EJB (Entreprise Java Bean) de Sun Microsystems. Un composant respectant ce standard optimise sa portabilité vers un serveur d'application adoptant ce même modèle. L'indépendance vis à vis du système prenant en charge son exécution lui assure une certaine pérennité dans le temps vis à vis de l'évolution des technologies des systèmes.

De plus, ce standard décrit le comportement de l'*objet*. Les logiques liées à un domaine métier se distinguent des logiques liées à la gestion des ressources système de nommage, de communication, de transaction, de persistance, de concurrence et de sécurité.

Les recommandations de l'éditeur Lutris, auteur initial de la plate-forme Enhydra[45] choisie comme environnement support par la communauté OpenUSS, définit l'*objet* comme une brique logicielle de base pour l'assemblage d'applications qui touche à un domaine ou à un *processus métier*. Il contient différentes logiques en intégrant toutes les stratégies, les algorithmes décisionnels et les manipulations des données. Le cadre technique de cette plate-forme définit trois types d'*objets* :

- **Objet "Presentation"** : L'*objet* « Presentation » contient la logique de présentation des informations. Il intègre le modèle de navigation de l'utilisateur. Il implante l'*interface* « HttpPresentation » du serveur d'application Enhydra.

- **Objet "Workflow"** : L'*objet* « Workflow » ou « Helper » est utilisé pour gérer les interactions entre les *objets* et représenter une tâche ou un processus. Il implante l'*interface* « session » de la spécification standard EJB de Sun Microsystems.

[44] Voir, http://www.campussource.de/software/openuss/doku/OpenUSS-Developer.pdf
[45] Voir, http://enhydra.objectweb.org/doc/EnhydraApp.html

- **Objet "Business"** : L'*objet* « Business » représente une entité manipulée par le système de gestion d'apprentissage. Il implante l'*interface* « entity » de la spécification standard EJB de Sun Microsystems.

Pour intégrer un *objet* dans la plate-forme OpenUSS, le concepteur est amené à effectuer la distinction entre les logiques de présentation, des tâches ou des entités. Cette contrainte apporte par la suite une souplesse dans l'analyse du *comportement* des *objets*. Les *objets* « Business » sont utilisés pour convertir des données en *objets* ou pour décrire les concepts *métier* exprimables sous la forme de noms mais ils ne sont pas adaptés pour représenter un *processus* ou une tâche. De même, les *objets* « Presentation » sont utilisés pour abstraire les modèles de présentation et pour s'adapter aux différents types de logiciel client.

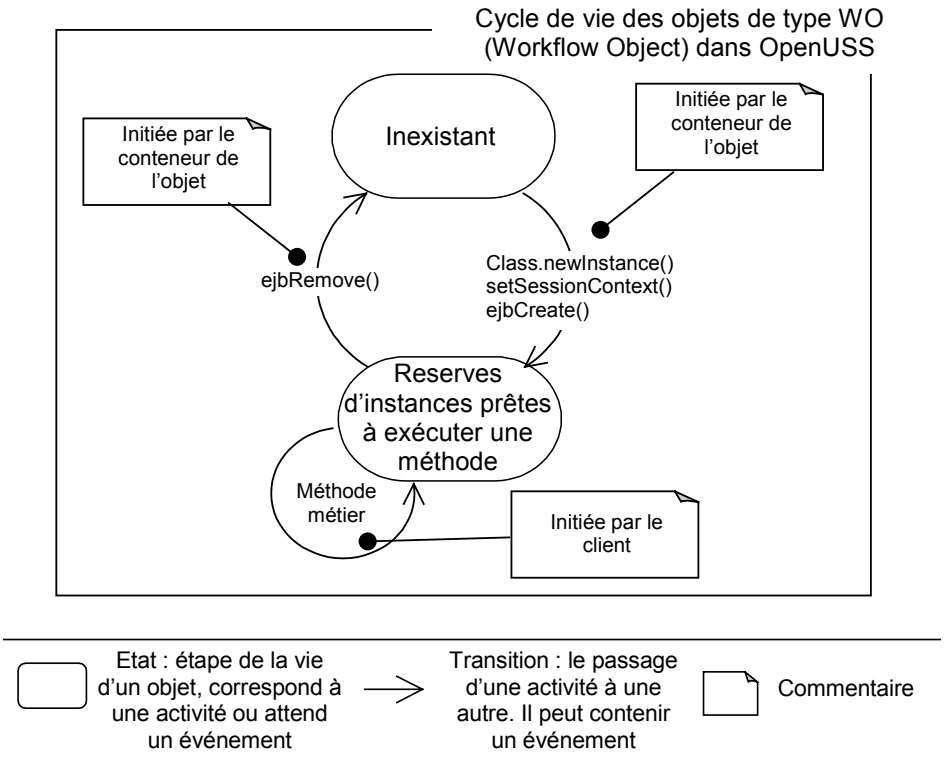


Figure 43 : Diagramme état-transition UML de l'objet considéré pour OpenUSS [(Monson-Haefel 2002), figure 12-1]

En conséquence, seul l'*objet* « Workflow » intègre la logique à considérer. La spécification reprend alors les *états* des *interfaces* opérationnels de ce type d'*objets* et permet par la suite d'observer le *comportement* de l'*objet* en traçant les appels des méthodes de son cycle de vie (cf. Figure 43). Ces observations donnent lieu par la suite à des actes d'assemblage, de modification, d'extension afin de prendre en compte les nouveaux phénomènes observés. D'ailleurs, la plate-forme Enhydra met à disposition une bibliothèque de gestion de *traces* du serveur d'application : la librairie

logicielle « monolog[46] ». Un tel outil sépare l'insertion d'une *trace*, de sa destination, de son affichage et du comportement de son gestionnaire. En effet, un concepteur souhaitant intégrer le code d'une sonde définit l'information à générer soit un message en précisant une simple chaîne de caractères, soit en faisant appel à un *objet*, en précisant par exemple l'*état* de la pile des différents appels de méthodes correspondant au contexte d'exécution.

Nous pouvons constater que les termes utilisés reprennent les concepts d'un langage de spécification du *point de vue computationnel* du cadre ODP-RM. Le *comportement* observé consiste à tracer les différents appels des méthodes de l'*interface* de l'*objet* et offre également la possibilité d'observer le fonctionnement de certains services du conteneur de l'*objet*. Focalisée sur ce seul *point de vue*, l'analyse se limite à tracer le nombre d'appels et leur déroulement temporel. L'intérêt immédiat est de produire des informations que le concepteur ayant le rôle d'administrateur du serveur est seul à pouvoir interpréter et analyser pour comprendre le comportement des services de base de la plate-forme. Cependant, ces observables peuvent mettre en avant des informations pertinentes pour les différents *acteurs* du *point de vue métier* d'un système de formation.

Les séquences des appels tracés s'adressent aux nouveaux *acteurs* : analyste de données, concepteur pédagogique et apprenant qui souhaitent les interpréter. Contextualiser et formaliser le travail de ces *acteurs*, favorise l'adhésion de nouveaux membres ou la création de nouvelle *communauté*. Ces différents aspects peuvent être spécifiés dans le cadre ODP-RM.

4.3.1.3 Modélisation de l'objet pour la communauté FSL

L'outil FSL se présente comme un gestionnaire de contenus d'apprentissage. Il joue la fonction d'intégrateur et de médiateur de ressources pédagogiques. La conception mise en œuvre par la communauté repose sur une grande modularité de l'architecture de l'application.

En effet, le paradigme MVC (Modèle/Vue/Contrôlleur) choisi par les développeurs de cette communauté permet la gestion des *interactions* entre le dispositif et les deux *acteurs* considérés : l'apprenant et le concepteur pédagogique. Ce paradigme d'architecture appréhende l'*interface* en distinguant trois éléments (Krasner 1988) :

- Le **modèle** spécifie la structure et les *comportements* que l'on veut représenter.
- La **vue** représente l'*interface* de sortie qui sert à représenter extérieurement cette structure et à animer ses *comportements*.

[46] Voir, http://monolog.objectweb.org/

- Le **contrôleur** représente l'*interface* d'entrée qui permet d'interagir avec le modèle ou avec sa vue.

A chaque vue est associée un contrôleur spécifique, ce qui permet de gérer des événements de façon contextuelle et en fonction de la vue. Guidé par ce paradigme, le travail de modélisation sur le dispositif FSL a permis d'extraire la structure des composants du gestionnaire FSL. Elle est représentée sur la Figure 44 à l'aide des différents paquetages du composant gérant la diffusion des ressources pédagogiques. Comme il est précisé dans le patron de conception[47] MVC [(Buschmann, Meunier et al. 1996), pages 125-144], il est conseillé au développeur d'appliquer le patron de conception orientée *objet* Observateur [(Gamma, Helm et al. 1999), pages 343-355]. Deux types d'*objet* doivent être créés : le sujet et l'observateur. Les *objets* observateurs délèguent la responsabilité du contrôle d'un événement à un *objet* central, le sujet. Ce dernier connaît les *objets* observateurs car ils s'enregistrent auprès de lui. Il doit les avertir dès qu'un événement a lieu.

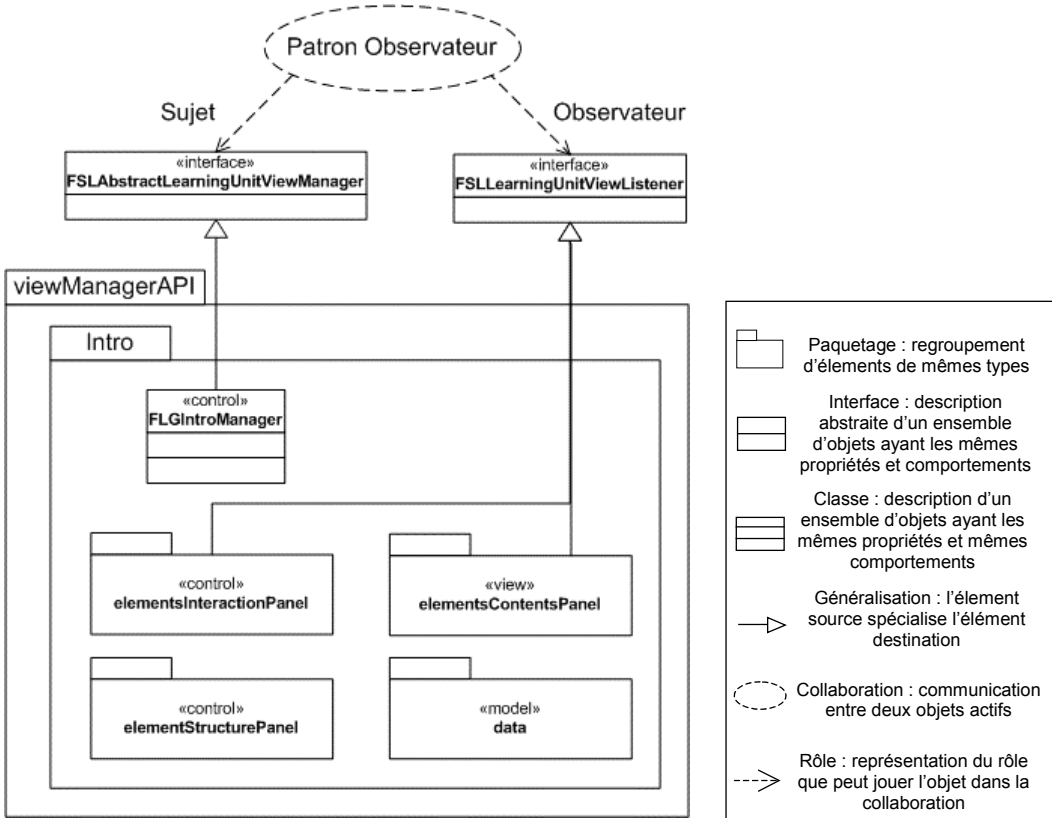


Figure 44 : Décomposition de chacun des composants FSL en objets selon le patron « Observateur »

L'*objet* au sens d'ODP-RM correspond alors à la mise en œuvre de l'*objet* sujet. En effet, le *rôle* de cet *objet* est central car il a la charge de capter tous les évènements et de les signaler aux observateurs correspondants. L'*objet*

[47] Un patron de conception, ou « design pattern », améliore la communication et l'apprentissage en adoptant une terminologie et une approche communes au sein d'une équipe de développement.

intitulé « FSLAbstractLearningUnitViewManager » joue ce *rôle* dans le gestionnaire FSL. L'analyse du code de ce dernier nous permet d'extraire le modèle comportemental des composants gérés. Nous le représentons sur la Figure 45 sous la forme d'un diagramme état-transition UML. Le nom des différentes *actions* gérées par le composant FSL reprend l'*interface* de l'*objet* « FSLLearningUnitViewListener » qui joue le *rôle* de l'observateur.

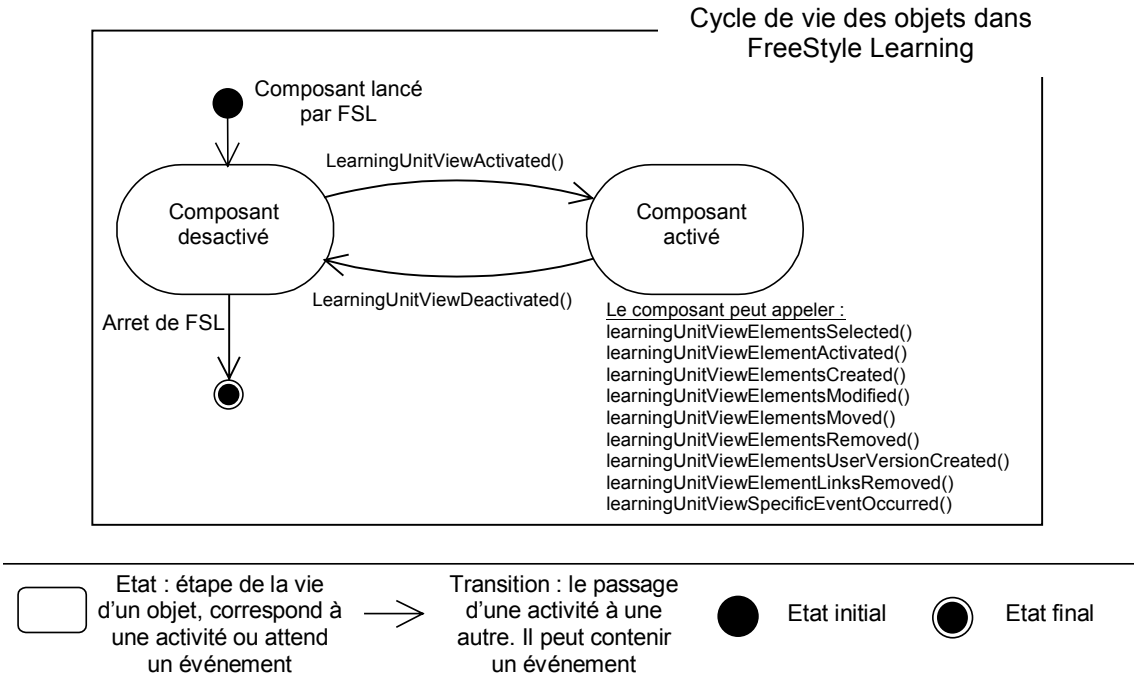


Figure 45 : Diagramme état-transition UML de l'objet considéré pour FSL

Un tel modèle décrit le mécanisme des *interactions* entre les *objets* et reprend ainsi la terminologie du *point de vue ingénierie* du cadre ODP-RM. Par ailleurs, ce modèle est très similaire à la spécification du cycle de vie d'un *objet* SCORM définie par le consortium ADLNet (cf. Figure 39, page 96). Dans le document diffusé par ce consortium, le modèle permet de spécifier trois aspects, le premier porte sur l'environnement d'exécution, le second porte sur la structure des contenus diffusés et le dernier sur la description d'un tel *objet*.

Ces deux derniers aspects ne sont pas abordés par l'*objet* considéré. Cependant, l'observation de son *comportement* amène le concepteur du système de formation à s'interroger tant sur la structure interne de l'environnement d'exécution que sur le contenu des ressources diffusées. L'architecture du cadre ODP-RM instrumente une telle modélisation en gérant l'articulation entre ces trois *points de vue* complémentaires.

4.3.1.4 Modélisation de l'objet pour la communauté FLE

Dans le cas de la communauté FLE, les concepteurs ont choisi un système de gestion de contenu, la plate-forme ZOPE. Ce type de plate-forme est dédié à supporter des applications informatiques d'un type particulier. En effet, l'un

des concepts principaux est de mettre à disposition un ensemble de ressources systèmes pour gérer le cycle de vie d'un contenu informationnel. Par exemple la plate-forme choisie par la communauté FLE offre une gestion transparente de l'historique des *états* successifs des différents *objets*. Ces derniers stockés dans la base de données ZOPE permettent par la suite de rétablir les *états* successifs des *objets* en retour d'utilisation.

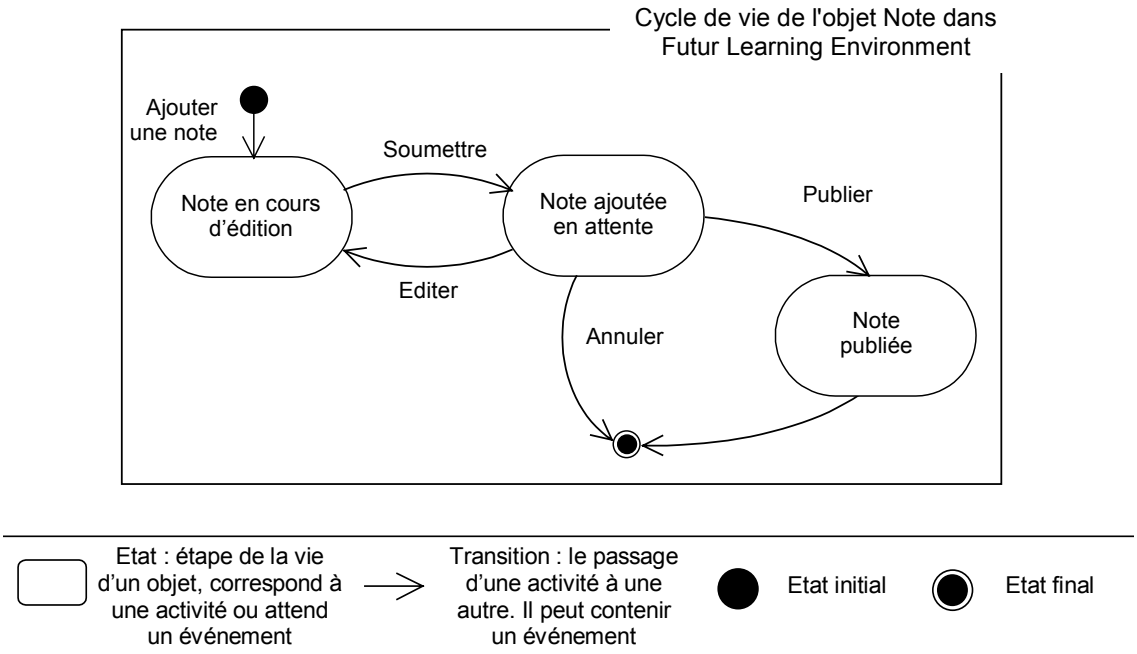


Figure 46 : Diagramme état-transition UML de l'objet considéré pour FLE

En conséquence, dans le code de l'application FLE nous considérons un *objet* s'il applique un tel cycle de vie de gestion d'un contenu. Par exemple, l'*objet* « note » est alors identifié dans le code de l'application FLE. La représentation de son cycle de vie (cf. Figure 46) est partagée par tous les développeurs souhaitant comprendre, utiliser ou étendre l'outil FLE. De plus, cet environnement donne la possibilité[48] d'importer ou d'exporter une unité d'apprentissage décrite dans un langage de spécification se rapprochant du modèle *informationnel* « Learning Design » du consortium IMS. En conséquence, les principales ressources du socle fournies par la plate-forme ZOPE permettent d'opérationnaliser l'articulation entre le *point de vue informationnel* et le *point de vue technologique* du cadre ODP-RM (cf. Figure 27, page 66).

Par ailleurs, les autres concepts spécifiques à un système de gestion de contenu sont à rapprocher des fonctionnalités d'administration définies par le cadre ODP-RM. En effet, la sécurité, la gestion des *rôles*, des *responsabilités* engagent des enjeux directement liés aux propriétés des *objets* manipulés dans le *point de vue métier*

[48] Voir, http://fle3.uiah.fi/

d'un système de formation. Les règles définissant les structures des contenus et les droits de chacun des acteurs du système de gestion de contenu peuvent être spécifiées en terme de *stratégie* au niveau du *point de vue métier*.

4.3.1.5 Bilan

Les trois communautés, OpenUSS, FSL et FLE apportent trois modèles différents pour la notion d'*objet* dans un système de formation. Le cadre unificateur ODP-RM nous permet de les considérer comme complémentaires.

Nous avons illustré le travail de modélisation des outils diffusés par trois communautés. En effet, elles communiquent avant tout sur les fonctionnalités de nature technique des dispositifs développés. Le cadre ODP-RM nous met à disposition une architecture de *points de vue* pour analyser, modéliser et comprendre le *comportement* de ces *objets*.

4.3.2 Exemple 1 : Utilisation des concepts de modélisation du cadre ODP-RM sur le modèle de l'objet de la communauté FSL

Nous proposons d'analyser le déroulement d'une activité de modélisation des concepteurs, présenté dans (Corbière et Choquet 2004b). Ces derniers mettent en œuvre à la fois les modèles proposés par les consortiums travaillant sur les technologies éducatives et les modèles adoptés par les développeurs des communautés du logiciel libre produisant les objets utilisés. Les concepteurs manipulent dans cet exemple la notion d'*objet* de la communauté « FreeStyle Learning ».

4.3.2.1 Etapes et concepts du cas illustré

Nous nous positionnons selon le *point de vue métier*, présenté dans la sous-partie 3.3.3 (page 86), pour représenter les *interactions* ayant eu lieu entre les différents *objets métier* (cf. Figure 47). La première étape (1) illustre la sélection d'une *ressource* effectuée par le *processus* de conception, où la primitive du langage utilisé par le concepteur pédagogique est directement liée aux choix des auteurs de la *ressource* utilisée. L'enjeu est de sensibiliser le concepteur pédagogique sur les différentes structures internes des *ressources* répondant à une même description pédagogique, du *point de vue informationnel*. La deuxième étape (2) décrit le contexte d'utilisation de la *ressource* sélectionnée. Des points de liaisons existent entre les *points de vue informationnel* et *computationnel*. L'enjeu est d'utiliser des *artefacts*, produits par d'autres communautés, comme support à la négociation entre le *processus* d'analyse et le *processus* de génie logiciel. La dernière étape (3) cherche un formalisme de représentation des *états* de l'*objet* modélisé pour faire sens au concepteur pédagogique.

La situation d'apprentissage dont est extrait l'exemple présenté a été mise en œuvre sur le site de l'IUT de Laval à quatre reprises. A chaque fois, une cinquantaine d'étudiants mettent en œuvre l'unité d'apprentissage proposée, d'une

durée de deux heures. Elle vise à former les étudiants aux architectures logicielles des services sur réseau. Six types de contenu sont diffusés par le gestionnaire FSL : une vidéo présentant les principes du fonctionnement d'un serveur Web, des supports textuels et illustrés présentant les principaux protocoles du Web, des diaporamas détaillant les principes présentés sur la vidéo, une série de questions/réponses, une série de QCMs et trois activités amenant l'apprenant étape par étape à réaliser le code d'un serveur Web.

Dans le cadre ODP-RM, l'acte de modélisation consiste pour les concepteurs à manipuler trois concepts qui correspondent aux trois étapes présentées du cas illustré :

- *Interface* : Définit l'abstraction du *comportement* d'un *objet* qui identifie un sous-ensemble des *interactions* entre les *objets* et les cache à tous les autres [(ISO/IEC-10746-2 1996), partie 8.4].
- *Comportement* : Définit un ensemble d'*actions* avec un ensemble de contraintes exprimées à l'aide du langage de spécification choisi [(ISO/IEC-10746-2 1996), partie 8.6].
- *Etat* : Représente à un instant donné, la situation de l'*objet* permettant de modéliser l'ensemble des *actions* auxquelles l'*objet* participe [(ISO/IEC-10746-2 1996), partie 8.7]. Une *action* appartient à un ensemble modélisant les *interactions* et *actions* internes de l'*objet* [(ISO/IEC-10746-2 1996), partie 8.3].

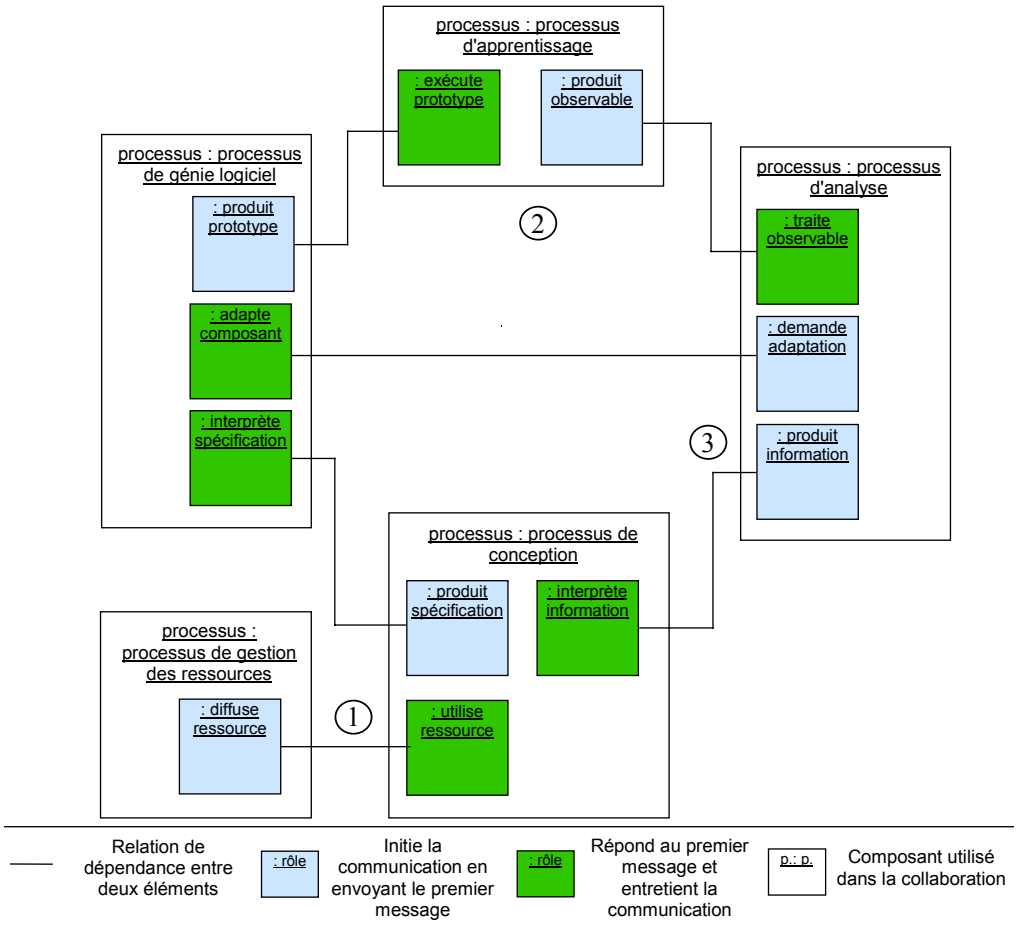


Figure 47 : Diagramme de collaboration UML de l'exemple 1

4.3.2.2 Etape 1 : Utilisation du concept d'interface pour modéliser

La première activité du concepteur appartenant au *processus* de conception est de créer une nouvelle *instance* d'une description en utilisant deux *langages* de spécification du *point de vue informationnel*, IMS-LD (IMS Learning Design) et IMS-CP (IMS Content Packaging). Le concepteur réutilise trois *ressources*: la première en choisissant le gestionnaire diffuseur de contenu « FreeStyle Learning », représenté par le composant « learningUnitViewAPI.jar » ayant le rôle d'*interface*, la seconde en identifiant la composante diffuseur spécifique au type du média choisi, le composant « manager.jar », et la dernière correspondant à la *ressource* vidéo, « author_video1.mpg ». Deux *points de vue* sont alors utilisés pour représenter cette description : le *point de vue informationnel* (cf. Figure 48) et le *point de vue computationnel* (cf. Figure 49). Ce dernier *point de vue* représente le *signal* « END_OF_VIDEO_REACHED » échangé entre les deux composants logiciels suivants :

- learningUnitViewAPI.jar qui réalise l'*interface* fournie par le gestionnaire
- et manager.jar qui en dépend.

Description du type et de l'identifiant de la propriété

```
<imsld:locpers-property identifier="END_OF_VIDEO_REACHED">
  <imsld:datatype datatype="boolean"/>
</imsld:locpers-property>
…
<imsld:learning-activity identifier="LA-video">
  <imsld:environment-ref ref="ENV-video"/>
  <imsld:on-completion>
    <imsld:change-property-value property-ref="END_OF_VIDEO_REACHED"
      property-value="TRUE"/>
  </imsld:on-completion>
  …
</imsld:learning-activity>
…
<imsld:environment identifier="ENV-video">
  <imsld:learning-object identifier="LO-video" type="knowledge-object">
    <imsld:item identifier="ITEM-video" identiofierref="RES-video"/>
  </imsld:learning-object>
  <imsld:learning-object identifier="LO-componentVideoFSL" type="tool-object">
    <imsld:item identifier="ITEM-VideoManagerFSL"
      identiofierref="RES-componentVideoFSL"/>
  </imsld:learning-object>
</imsld:environment>
…
<imscp:resources>
  <imscp:resource identifier="RES-video" type="">
    <imscp:file href="/learningUnits/intro/author_video1.mpg"/>
  </imscp:resource>
  <imscp:resource identifier="RES-componentVideoFSL" type="">
    <imscp:file href="/learningUnitViewManagers/intro/manager.jar"/>
    <imscp:file href="/learningUnitViewAPI.jar"/>
  </imscp:resource>
…
</imscp:resources>
```

Description de l'action à effectuer quand l'activité se termine

L'environnement est composé de deux objets d'apprentissage

Chaque ressource est associée à un ou des composants logiciels

Figure 48 : Extrait du scénario d'apprentissage

Pour simplifier leurs représentations sur la Figure 49, seul l'intitulé de l'événement échangé entre ces deux composants est représenté. En réalité, la méthode « learningUnitViewSpecificEventOccurred » appartenant à l'*interface*

du gestionnaire de diffusion est invoquée. Elle transmet en paramètre l'*objet* « FSLLearningUnitViewEvent » correspondant au *signal* « END_OF_VIDEO_REACHED ».

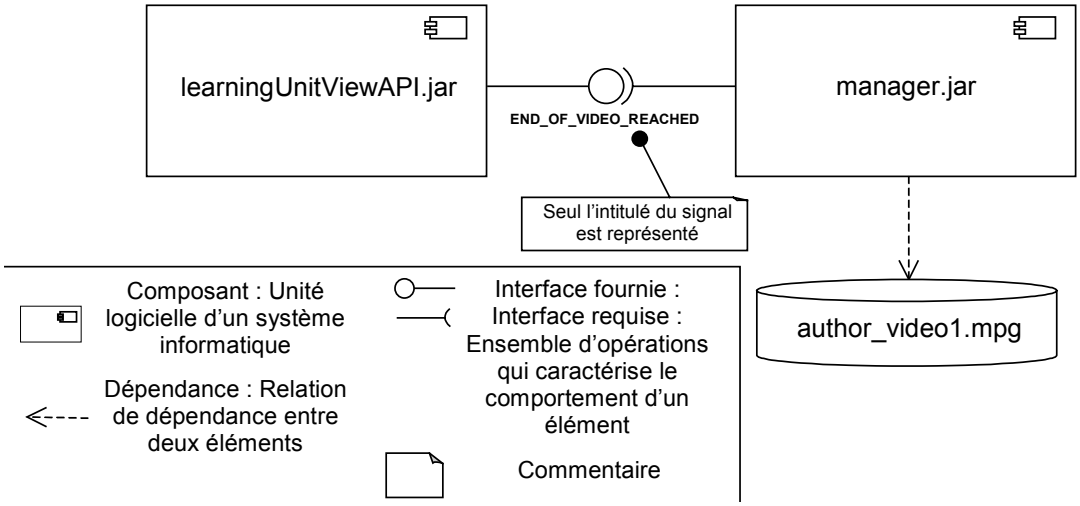


Figure 49 : Diagramme de composants UML

Cette mise en perspective identifie le *signal* « END_OF_VIDEO_REACHED » présenté sur l'*interface* du composant comme un artefact logiciel sélectionné par le concepteur. L'activation de cet événement met à jour la valeur de la propriété définie dans la spécification du scénario d'apprentissage. Cette *action* est décrite par le concepteur. Elle est à mettre en correspondance avec la décomposition fonctionnelle du *point de vue computationnel* (cf. Figure 49).

4.3.2.3 Etape 2 : Utilisation du concept de comportement pour modéliser

Dans le *processus* d'analyse, les opérations statistiques effectuées sur les sessions d'utilisation montrent que le *signal* « END_OF_VIDEO_REACHED » a été rarement activé. Cependant cette information n'est pas suffisante pour analyser ce *comportement*. Un dialogue s'engage entre le *processus* génie logiciel et le *processus* d'analyse. Pour supporter une telle négociation, il est nécessaire de faire appel à un *point de vue* supplémentaire pour supporter cet échange. Il est nécessaire de rechercher le modèle générique appliqué lors de la conception du composant chargé de contrôler la diffusion de la *ressource* vidéo. Le *processus* génie logiciel choisit alors le profil UML d'une application multimédia proposé par (Engels et Sauer 2002). Ce profil technique représenté sur la Figure 50 présente les principaux stéréotypes du modèle recherché et guide le travail de modélisation du concepteur.

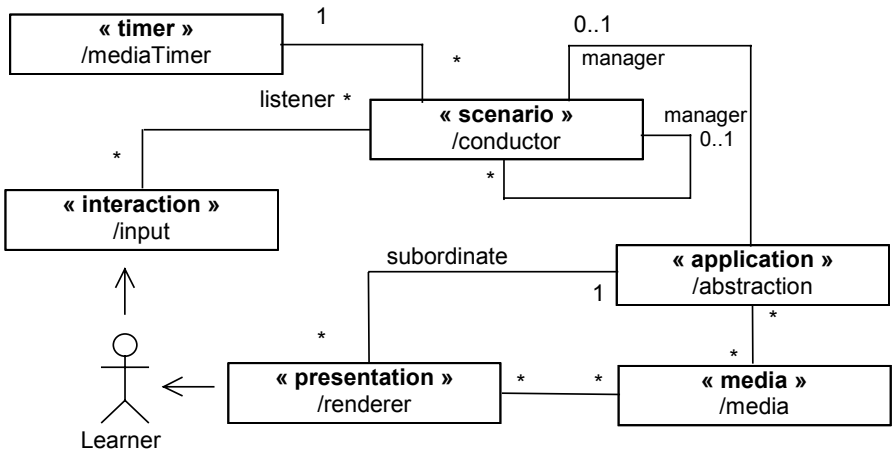


Figure 50 : Diagramme de classe UML représentant le Profil pour les applications multimédias [(Sauer et Engels 2001), figure 3]

L'informaticien appartenant au *processus* génie logiciel identifie les stéréotypes dans le code des composants logiciels du gestionnaire FSL. En résultat, deux diagrammes UML (cf. Figure 51 et Figure 52) sont obtenus et présentés au concepteur.

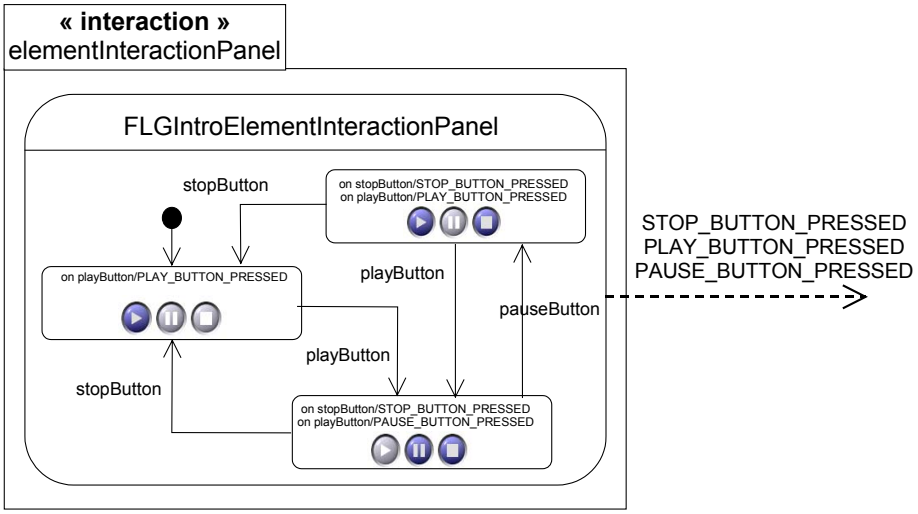


Figure 51 : Diagramme état-transition UML

La Figure 51 présente, par un diagramme d'état-transition UML, le *comportement* de l'objet en modélisant les *interactions* entre les différents *états* générant les différents *signaux*.

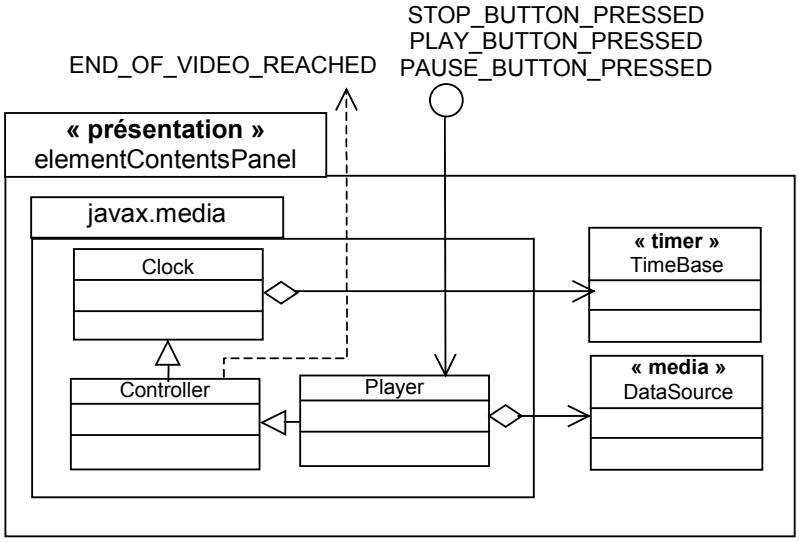


Figure 52 : Diagramme de classe UML[49]

La Figure 52 représente le diagramme de classe UML extrait du document technique sur la librairie « Java media framework[50] ». L'analyste constate que le composant « présentation » est un gestionnaire de diffusion intégrant un temporisateur. Dans ce cas, l'*état* de la classe « player » correspond à l'*état* de la classe « timer ». L'analyste demande à l'informaticien d'intégrer une sonde logicielle qui *trace* chaque *signal* et le type avec la valeur du temporisateur donnée par la classe « timer ». A la prochaine utilisation, le *processus* d'analyse utilise le diagramme d'état-transition UML pour représenter de manière significative le *comportement* de l'*objet* au concepteur. Ce dernier est amené à analyser le faible taux d'activation des *signaux* « END_OF_VIDEO_REACHED ». Les changements des *états* correspondant aux *signaux* « STOP_BUTTON_PRESSED », « PLAY_BUTTON_PRESSED » et « PAUSE_BUTTON_PRESSED » peuvent être également tracés.

4.3.2.4 Etape 3 : Utilisation du concept d'état pour modéliser

Nous venons de montrer que la mise en relation entre deux *instances* du *point de vue informationnel* et du *point de vue computationnel* engage l'informaticien et l'analyste dans une négociation afin de répondre au besoin de consistance défini par le cadre ODP-RM. Pour y répondre, le cadre propose un ensemble de concepts, définis dans le document « Foundation », permettant de spécifier l'entité d'un modèle représentant aussi bien un *objet* qu'une *action*.

Dans le cas d'étude présenté, la mise en correspondance des *points de vue informationnel*, *computationnel*, *ingénierie* et *technologique* explicite un nouveau *comportement*. Un des concepts de modélisation est de proposer un

[49] Voir, http://java.sun.com/products/java-media/jmf/2.1.1/guide/JMFArchitecture.html (figure 2.9)
[50] Voir, http://java.sun.com/products/java-media/jmf

nouveau formalisme pour représenter l'*état* des *objets* gérant la visualisation d'une *ressource* vidéo. Pour représenter cette nouvelle information, le *processus* d'analyse propose le *langage* SMIL (Synchronized Multimedia Integration Language). C'est un format standard du W3C pour spécifier les présentations multimédias interactives. Ce langage conçu à l'origine pour supporter des actes de spécification, est utilisé ici pour supporter les activités de modélisation. En effet, en retour, le concepteur du *processus* de conception à l'aide d'un lecteur SMIL peut voir comment la *ressource* vidéo a été visualisée par chaque apprenant. Ce formalisme peut également être utilisé pour représenter les résultats statistiques des utilisations effectuées par le *processus* d'analyse.

Ainsi, l'*état* de l'*objet* est représenté dans un *modèle* issu de la communauté travaillant sur la représentation multimédia. Dans l'illustration de la Figure 53, une partie du schéma XML du langage SMIL est utilisée pour décrire la structure, le contenu et la sémantique des *actions* observées en retour de sessions d'apprentissage.

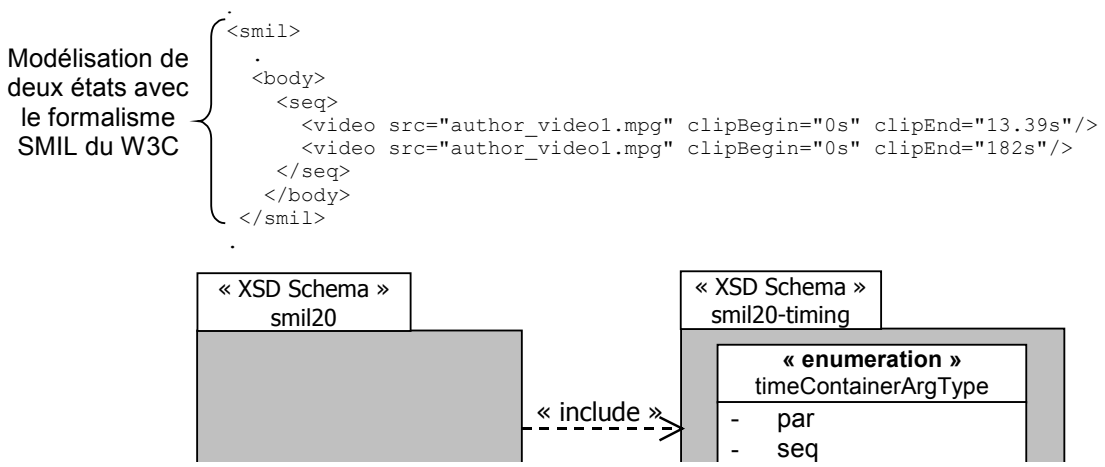


Figure 53 : Description SMIL générée par la sonde logicielle

Ce formalisme est alors une réponse au besoin de modéliser les *états* des *objets* à la suite de la mise en correspondance *des points de vue informationnel* et *computationnel*.

4.3.2.5 Bilan

L'acte de modélisation est défini par un ensemble de concepts dans le cadre ODP-RM. Nous avons présenté dans cet exemple un cas illustrant la démarche des concepteurs appartenant aux différents *processus* d'un cycle d'une ingénierie d'un EIAH : le *processus* de conception, le *processus* génie logiciel, le *processus* d'apprentissage et le *processus* d'analyse. Selon une approche dirigée par les modèles, trois concepts du cadre ODP-RM guident la séquence des *étapes* de travail de collaboration entre ces différents *processus*.

4.3.3 Exemple 2 : Utilisation des concepts de spécification du cadre ODP-RM sur le modèle de l'objet de la communauté OpenUSS

Nous proposons d'analyser le déroulement d'une activité de spécification du concepteur informatique. Ce dernier met en œuvre à la fois les modèles adoptés dans les développeurs d'une communauté du logiciel libre produisant les objets utilisés et les langages de spécification de scénario pédagogique diffusés par le consortium IMS. Nous reprenons dans cet exemple la notion d'*objet* de la communauté « OpenUSS ».

4.3.3.1 Phases et concepts du cas illustré

Dans le cadre ODP-RM, la spécification d'un système demande aux concepteurs de vérifier en permanence les relations entre les spécifications et leurs *implantations* [(ISO/IEC-10746-1 1998), partie 9.1]. Le processus de développement, dans le cadre ODP-RM, consiste à transformer le discours des concepteurs en une spécification à l'aide des différents *langages* mis à leur disposition. Chaque spécification est alors responsable de la génération d'un certain nombre de sous-spécifications en effectuant des « transformations », soit en produisant une spécification ayant la même signification mais dans un *langage* différent, soit en produisant une spécification en décrivant de nouveaux détails. Dans un cas, il s'agit de chercher à déterminer l'ensemble des éléments qui n'ont pas changé, dans l'autre de produire de nouvelles spécifications en détaillant plus précisément les anciennes.

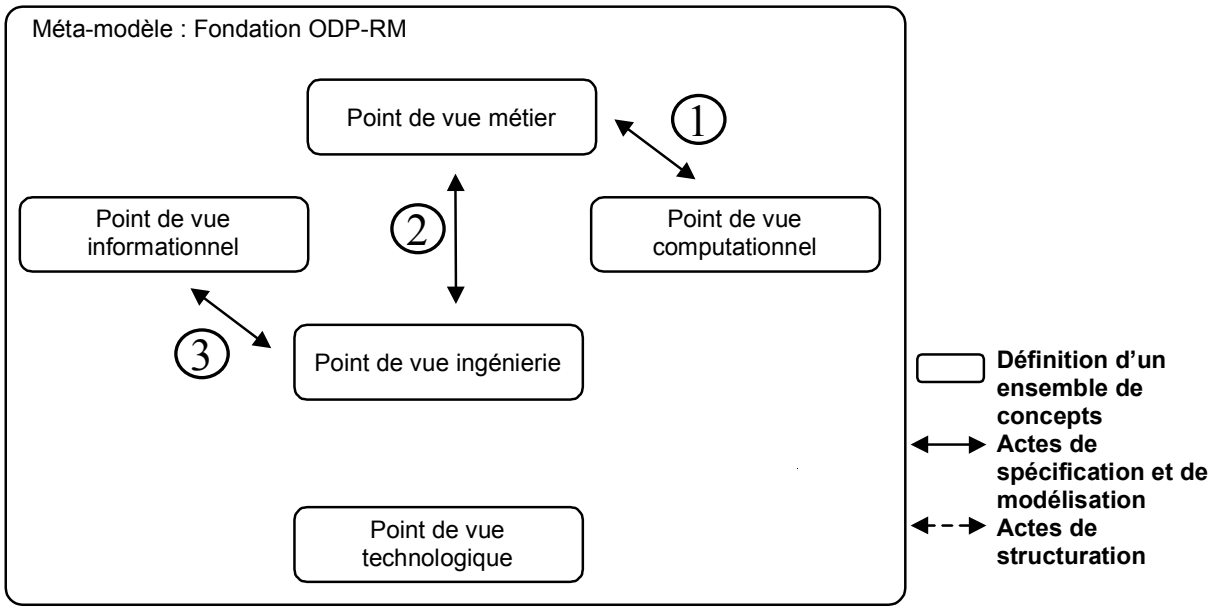


Figure 54 : Représentation des trois phases de l'exemple 2

Dans l'exemple présenté, nous reprenons la notion d'*objet* de la communauté OpenUSS. Comme nous l'avons montré dans le paragraphe 4.3.1.2, l'*objet* se définit à l'aide d'un *objet computationnel* sur lequel les différents *acteurs métier* vont interagir.

Cet exemple illustre l'activité de spécification du concepteur informatique, *acteur* du *processus* génie logiciel. Elle se définit en trois phases où différentes *ressources* diffusées par la communauté OpenUSS sont utilisées. La première (1) correspond à un travail d'analyse de la documentation produite par la communauté. La deuxième (2) est d'amener le concepteur informatique à comprendre les choix de conception des membres de la communauté choisie. La dernière (3) phase décrit le travail de spécification où le concepteur informatique cherche à communiquer l'interprétation d'un modèle en s'exprimant dans l'un des langages de spécification du concepteur pédagogique.

Cet exemple présente l'utilisation de trois concepts du cadre ODP-RM (ISO/IEC-10746-2 1996) qui correspondent respectivement aux trois phases présentées :

- *Décomposition* : Spécifie les *objets* ou les *comportements* constituant un *objet* ou un *comportement* donné [(ISO/IEC-10746-2 1996), partie 9.3].
- *Gabarit* : Spécifie des caractéristiques communes d'un ensemble d'*objets* à un niveau de détail suffisant pour être *instanciées* [(ISO/IEC-10746-2 1996), partie 9.11].
- *Compatibilité comportementale* : Définit l'*environnement* dans lequel deux *objets* sont considérés comme équivalents [(ISO/IEC-10746-2 1996), partie 9.4].

4.3.3.2 Phase 1 : Utilisation du concept de décomposition pour spécifier

Cette première phase est effectuée par le *processus* génie logiciel du *point de vue métier* (cf. Figure 31, page 86). Le concepteur informatique consulte la documentation mise à disposition par la communauté OpenUSS. Cette dernière produit une plate-forme suivant un processus de développement centré sur l'architecture et basé sur des composants. Une grande rigueur est mise en œuvre à chaque phase de son développement. Notamment, le modèle d'architecture d'OpenUSS représente les liens complexes entre ses parties tout en donnant la possibilité à d'autres concepteurs d'en garder la maîtrise. De plus, cette communauté diffuse une documentation sur les différents modèles utilisés tout au long des phases de développement. L'un d'eux présente le diagramme des composants du système de formation OpenUSS (cf. Figure 55).

Les liens de dépendances empruntés au système de notation du langage UML rendent explicite au concepteur informatique la position centrale du composant « foundation ». La spécification de la décomposition en *objets* de la plate-forme de formation constitue la première phase du travail de spécification effectuée par l'*acteur* du *processus* génie logiciel.

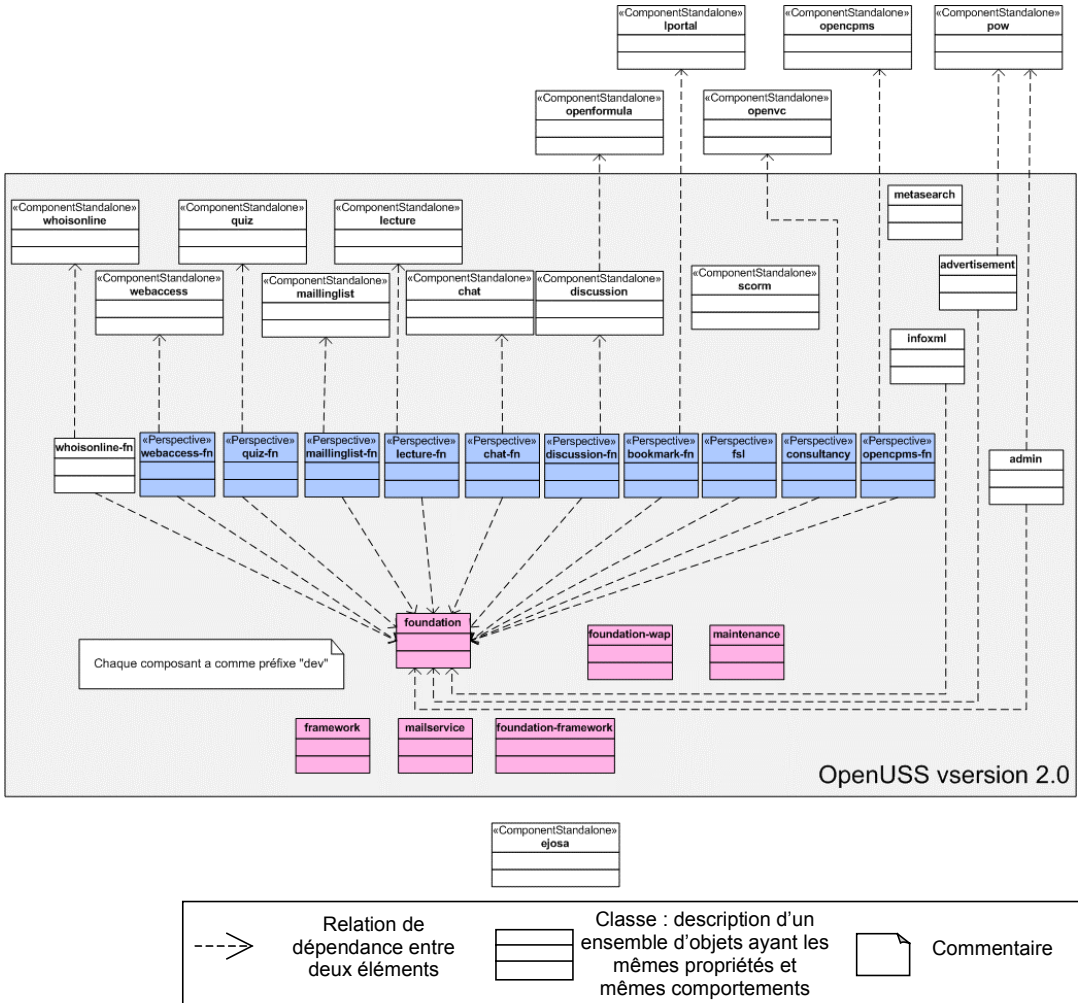


Figure 55 : Diagramme UML de classe des composants d'OpenUSS[51]

Cependant, le diagramme représenté sur la Figure 55 se limite à spécifier la *décomposition* fonctionnelle de l'application OpenUSS et ne correspond qu'au seul *point de vue computationnel* du cadre ODP-RM. Le travail de spécification dans le cadre ODP-RM consiste pour le concepteur informatique à décrire avec plus de détails les composants de la plate-forme afin d'identifier des *comportements* susceptibles d'engager de nouveaux échanges entre les différents *processus* du *point de vue métier*.

Le dispositif technique, sur lequel sont déployés ces composants, est la plate-forme compatible avec la technologie des composants EJB de Sun Microsystems. Elle est développée par la communauté JOnAS (Java Open Application Server). Chaque composant est spécifié à l'aide des modèles de descripteur de déploiement diffusés par Sun Microsystems[52]. Deux nouvelles *ressources* diffusées par la communauté OpenUSS sont alors utilisées : le fichier

[51] Voir, http://cvs.sourceforge.net/viewcvs.py/openuss/openuss/design-general/model/src/
[52] Voir, http://java.sun.com/ (http://java.sun.com/dtd/application_1_3.dtd et http://java.sun.com/j2ee/dtds/ejb-jar_2_0.dtd)

de déploiement des composants de la plate forme OpenUSS (cf. Figure 56, page 123) et le fichier de description des objets associé à chaque composant (cf. Figure 57, page 124).

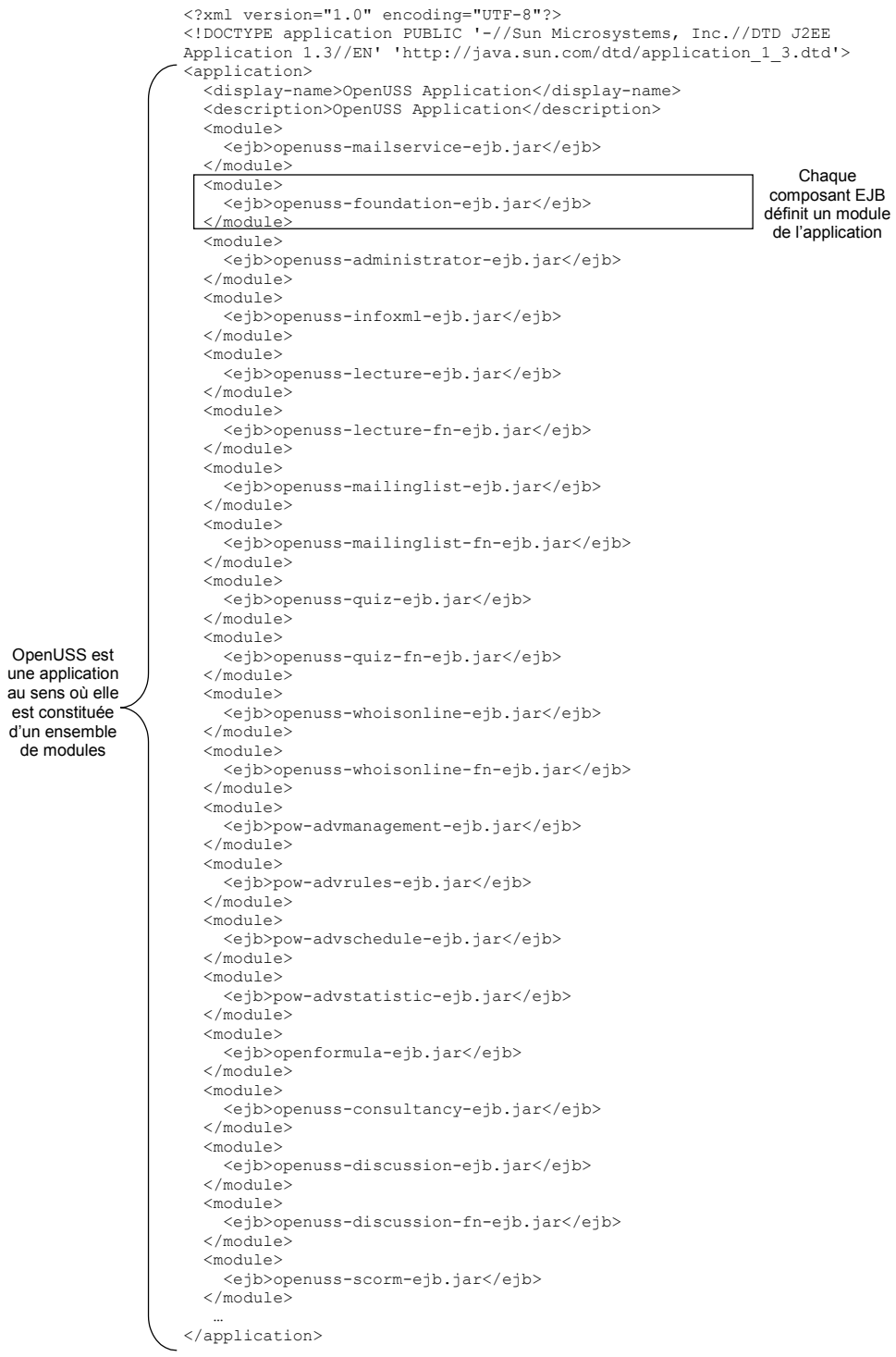


Figure 56 : Extrait du fichier de déploiement des composants EJB de la plate-forme OpenUSS[53]

[53] Voir, http://cvs.sourceforge.net/viewcvs.py/openuss/openuss/dev-all/resource/application/jonas/application.xml

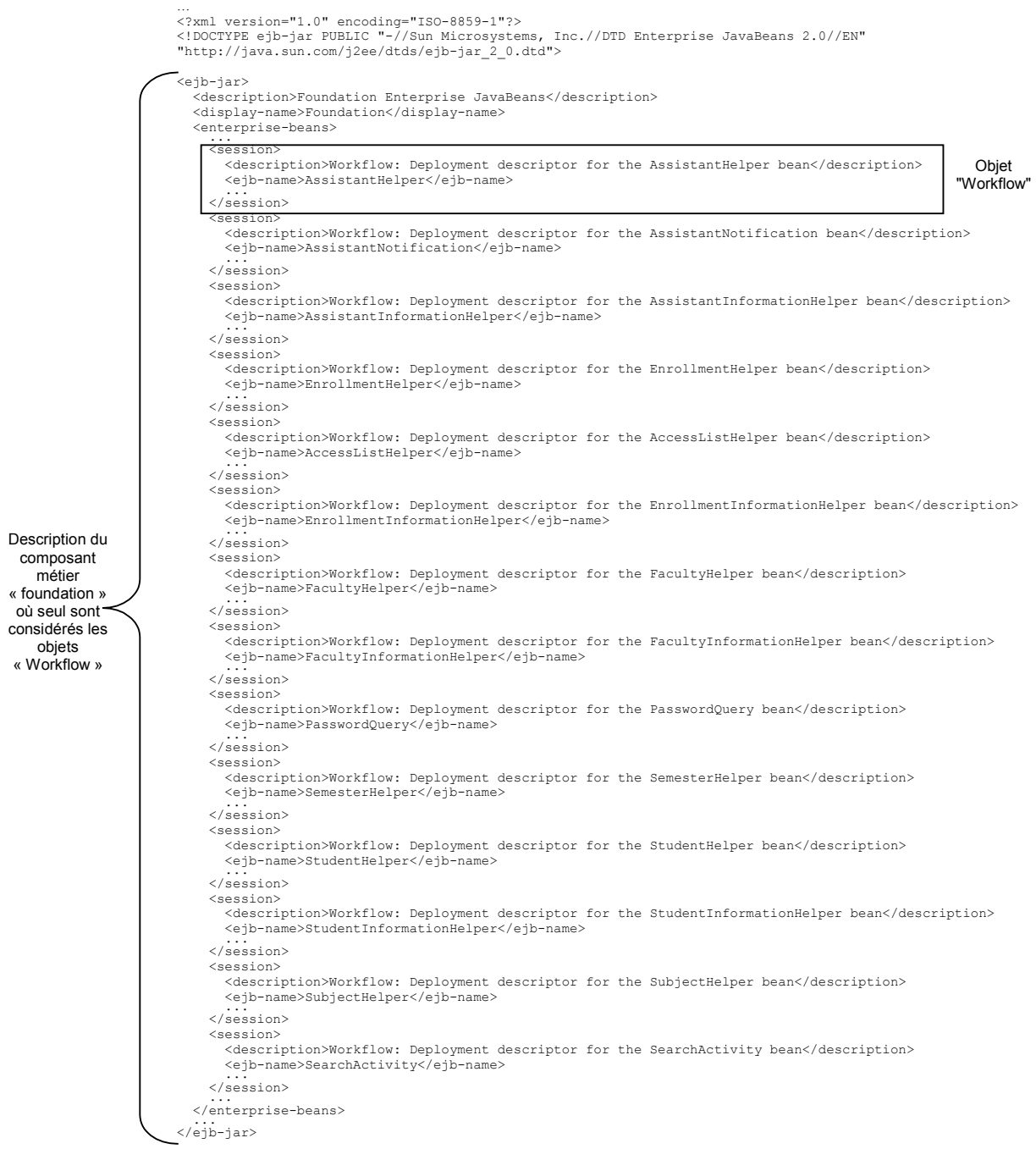


Figure 57 : Extrait du fichier spécifiant la décomposition en objets du composant « foundation[54] »

Dans le paragraphe 4.3.1.2, page 107, nous avons constaté que la notion d'*objet* pour cette communauté ne permet de spécifier que les aspects définis par le *point de vue computationnel*. Ces deux nouvelles *ressources* amènent le concepteur informatique à spécifier cet *objet* du *point de vue ingénierie*. De ce dernier, chaque *objet* est considéré

[54] Voir, http://cvs.sourceforge.net/viewcvs.py/openuss/openuss/dev-foundation/business/resource/jonas/ejb-jar.xml

comme une unité pouvant être activée, désactivée, contrôlée ou réutilisée par la plate-forme d'exécution. Identifié comme central dans le *point de vue computationnel*, le composant « foundation » est alors présenté au concepteur informatique comme une décomposition d'*objets ingénierie* (cf. Figure 57, page 124) correspondant aux *objets* « workflow » de la communauté OpenUSS.

4.3.3.3 Phase 2 : Utilisation du concept de gabarit pour spécifier

La spécification de la *décomposition* du composant « foundation », représentée sur la Figure 57, n'est pas suffisante. Elle se réduit à donner une liste d'*objets* considérés du *point de vue ingénierie*. Le concepteur informatique cherche alors à identifier et à modéliser les *interactions* qui peuvent exister entre ces *objets*. Or, les développeurs de la communauté OpenUSS ayant produit les *objets ingénierie* appliquent une démarche de conception spécifique qui est définie par le projet communautaire EJOSA[55]. Cette initiative a pour objectif de définir un *gabarit* à destination des développeurs informatiques désirant *instancier* une architecture J2EE (Java 2 Entreprise Edition) sur des composantes logicielles produites uniquement par des communautés du logiciel libre. Ce *gabarit* impose des règles de nommage pour les fichiers et une structure arborescente pour les organiser (cf. Figure 58).

Figure 58 : Structure extraite de la documentation version 2.3 du gabarit EJOSA

[55] Voir, http://ejosa.sourceforge.net/

Tout développeur se doit d'appliquer ce *gabarit* pour intégrer un nouveau composant dans la plate-forme OpenUSS. De plus, ce *gabarit* spécifie un domaine particulier, celui de l'ingénierie à base de composants (Roques et Vallée 2004). Pour chaque composant, il demande aux concepteurs de placer la structure de classes, produite à la suite du travail d'analyse du développement, dans le répertoire « spécification ». Les répertoires « presentation », « business » et « data » contiennent les *instances* de cette structure et le code des différents classes en tenant compte des contraintes des technologies utilisées. L'objectif de cette transformation est d'associer les moyens nécessaires pour faire interagir les *objets* issus du modèle de « spécification » avec son environnement d'exécution.

Dans le cadre ODP-RM, la structure de classes *implantée* dans le répertoire « spécification » correspond au *point de vue informationnel* qui définit les différents *objets* traités par le système de formation. Cette structure est représentée par le diagramme suivant (cf. Figure 59). Elle est obtenue à la suite du travail de rétroconception effectué par le concepteur informatique sur le code des fichiers sources stokés dans le répertoire « spécification[56] ».

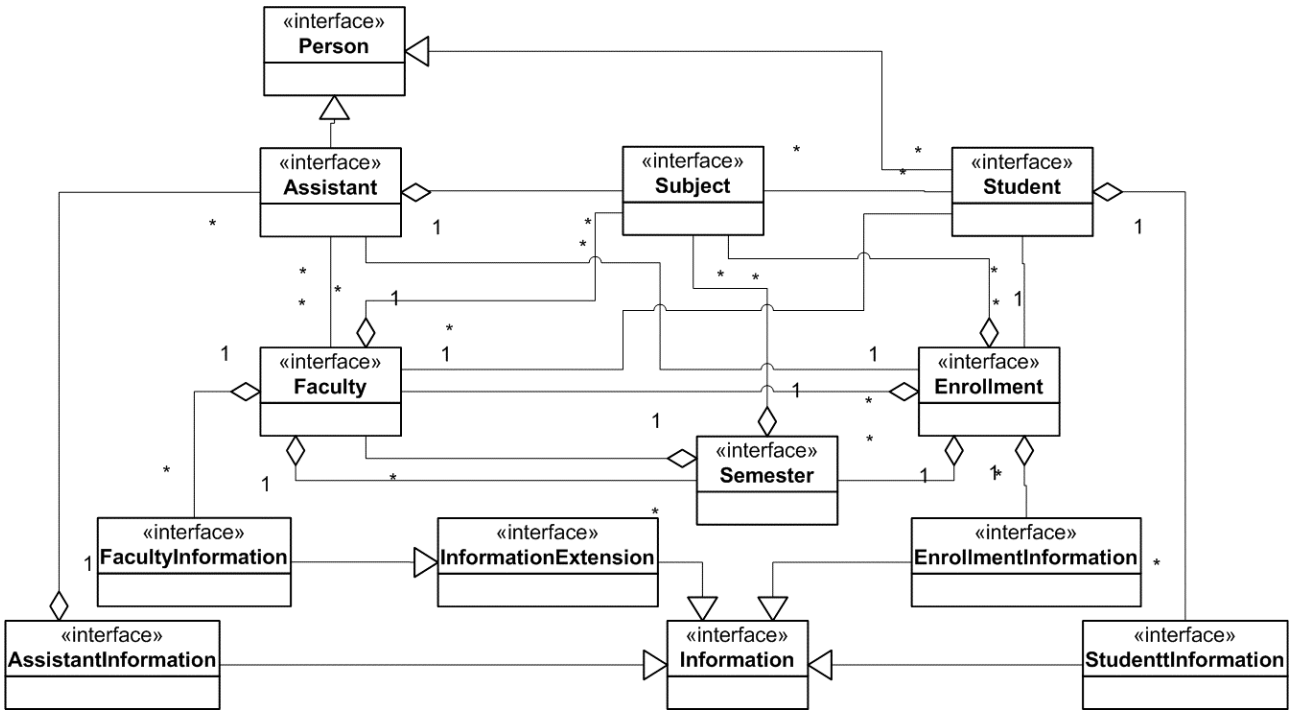


Figure 59 : Diagramme de classe UML du modèle de « spécification » du composant « Foundation »

[56] Voir, http://cvs.sourceforge.neviewcvs.py/openuss/openuss/dev-foundation/specification/src/

4.3.3.4 Phase 3 : Utilisation du concept de compatibilité comportementale pour spécifier

Afin de communiquer aux concepteurs du *point de vue métier* l'interprétation du modèle conceptuel produit par l'équipe de développement de la communauté OpenUSS, le travail du concepteur informatique est alors d'identifier le modèle *informationnel* diffusé par le consortium IMS qui se rapproche le plus du modèle analysé. Son activité est alors de mettre en adéquation la structure des classes correspondant au modèle de « spécification » du composant « foundation » de la plate-forme OpenUSS avec l'un des modèles de spécification produit et diffusé par le consortium IMS. Guidé par le cadre ODP-RM, le concepteur informatique spécifie les *objets ingénierie* ayant un *comportement compatible* avec les concepts du modèle IMS LD (cf. Tableau 2) ce qui l'amène à mettre relation des modèles *a priori* distincts.

Tableau 2 : Mise en correspondance des objets du composant « Foundation » avec les concepts du modèle de spécification

Objets identifiés dans le modèle de « specification » du composant « Foundation »	***Concepts du modèle de spécification IMS « Learning Design »***
Enrollment	Environment
Subject	activity structure
Semester	learning activity
Person	Role
Assistant	Staff
Student	Learner

De plus, pour valider cette *compatibilité comportementale*, le concepteur informatique doit spécifier les liens mettant en relation les *objets informationnel* et leurs *comportements*. Trois schémas se doivent d'être définis pour représenter le *point de vue informationnel* dans le cadre ODP-RM : le *schéma statique*, le *schéma dynamique* et le *schéma invariant* [(ISO/IEC-10746-1 1998), partie 8.3].

- *Schéma statique*, exprime des relations entre les *objets informationnel* valides à un instant donné.
- *Schéma invariant*, exprime les relations entre les *objets informationnel* qui sont toujours vraies pour tout *comportement* valide du système.
- *Schéma dynamique*, exprime les relations entre les *objets informationnel* applicables à un contexte particulier.

En reprenant les aspects statiques du schéma XML du modèle IMS LD (cf. Figure 60) comme le *schéma invariant*, le concepteur informatique considère que le *comportement* de l'*objet informationnel* « role » est lié au *comportement* des *objets informationnel* « activity structure » et « environment ».

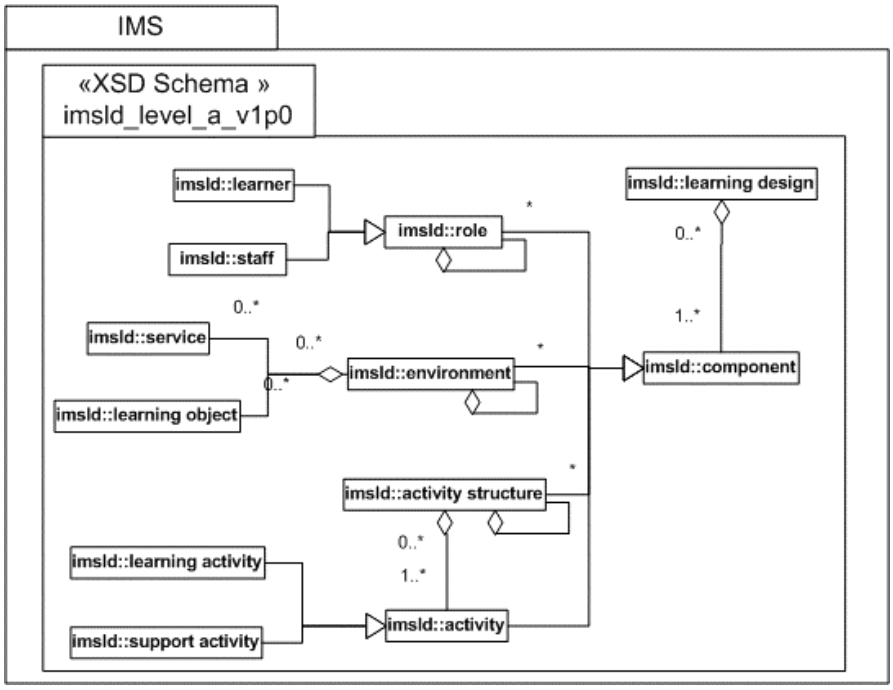


Figure 60 : Représentation à l'aide d'un diagramme UML de l'aspect statique d'IMS LD [(Koper, Olivier et al. 2003b), figure 2.1]

De plus, la seconde composante du schéma XML définit des *objets informationnels* complémentaires « method », « play », « act », « role part », « condition » et « property » représentés sur la Figure 61. Comme le précise la documentation du modèle de spécification IMS LD (Koper, Olivier et al. 2003b), ces concepts servent à spécifier les aspects dynamiques du scénario et correspondent au *schéma statique* du *point de vue informationnel* du cadre ODP-RM considéré par le concepteur informatique.

En s'appuyant sur le modèle de cycle de vie de l'*objet* considéré dans la communauté OpenUSS (cf. Figure 43, page 108), le concepteur informatique dont le travail de spécification consiste à intégrer des sondes logicielles est alors amené à mettre en relation ce dernier avec les six concepts du *schéma statique* du *point de vue informationnel*. Lors du fonctionnement de la plate-forme OpenUSS, l'appel des méthodes invoquées sur l'un des objets « Enrollment », « Subject » ou « Semester » se doit d'être tracé. Les concepts spécifiant l'aspect dynamique du modèle IMS LD sont alors utilisés pour décrire les *traces* générées par la sonde logicielle. Sur la Figure 62, nous illustrons le cas où un étudiant invoque l'*objet informationnel* « Enrollment » identifié dans le modèle de « specification » du composant « Foundation ». La *trace* générée est représentée par le concept « environment » du modèle de spécification d'unité d'apprentissage. L'information indiquant le nom de la méthode invoquée sur l'*interface* de l'*objet*, ici

« findEnrollmentById », est spécifiée. De plus, les dépendances avec les *objets informationnel* « Student » et « Subject » sont également représentées dans la *trace* générée conformément au *schéma statique* du modèle de spécification IMS LD.

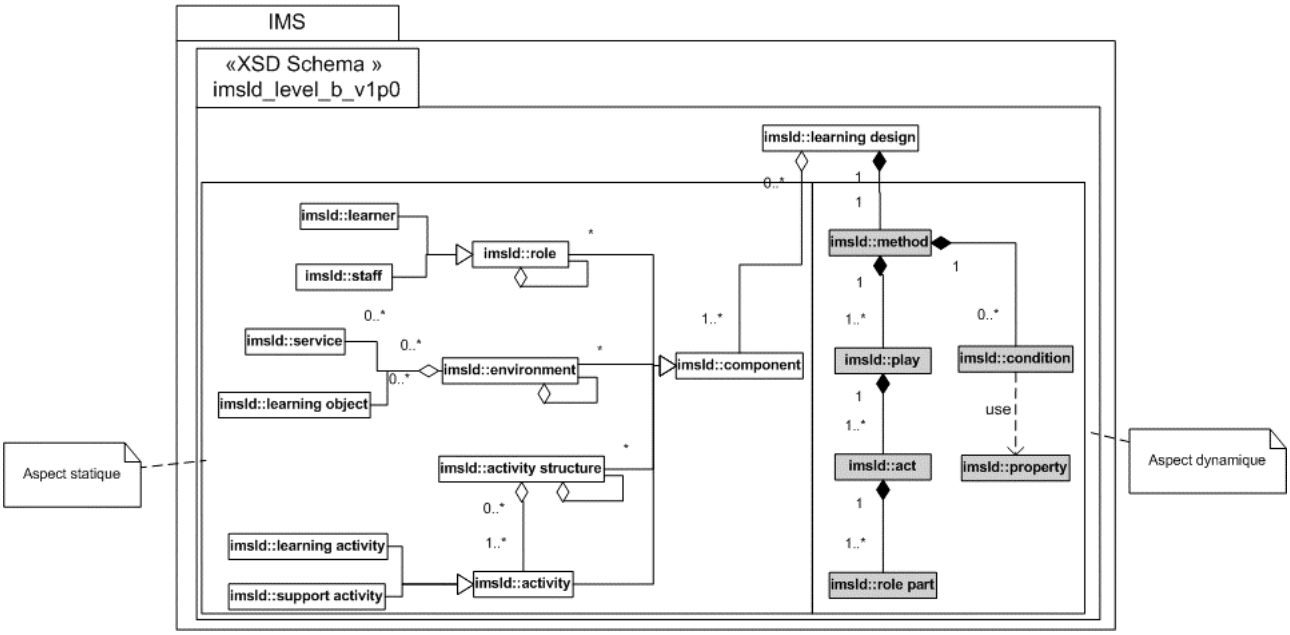


Figure 61 : Représentation à l'aide d'un diagramme UML des aspects statique et dynamique d'IMS LD [(Koper, Olivier et al. 2003b), figure 2.1]

```
...
<imsld:method>
   <imsld:play identifier="PLAY-intro">
      <imsld:act identifier="ACT-intro">
         <imsld:role-part identifier="RP-intro">
            <imsld:role-ref ref="1082988145588"/>
            <imsld:structure-activity-ref ref="1082988145705"/>
         </imsld:role-part>
      </imsld:act>
   </imsld:play>
   <imsld:conditions>
      <imsld:if>
         <imsld:and>
            <imsld:is>
                  <imsld:property-ref ref="id-dateTrack"/>
                  <!-- date et heure au format ISO 8601 -->
                  <imsld:property-value>2005-12-07T12:45:30</imsld:property-value>
            </imsld:is>
            <imsld:is>
                  <imsld:property-ref ref="id-methodeTrack"/>
                  <imsld:property-value>findEnrollmentById</imsld:property-value>
            </imsld:is>
         </imsld:and>
      </imsld:if>
      <imsld:then>
         <imsld:show>
            <imsld:environment-ref ref="1032988145101"/>
         </imsld:show>
      </imsld:then>
   </imsld:conditions>
</imsld:method>
...
```

Identifiant de l'objet « **Student** » sur OpenUSS

Identifiant de l'objet « **Subject** » sur OpenUSS

Le nom de la méthode de l'objet « **Enrollment** » et la date de son appel sont décrits sous la forme d'une régle conditionnelle

Identifiant de l'objet « **Enrollment** » sur OpenUSS

Figure 62 : Exemple de traces générées conformément au schéma statique considéré par le concepteur informatique

La définition d'une sonde logicielle est au cœur de l'exemple présenté. Le concepteur informatique explicte son travail de spécification des *traces* générées par les *objets* diffusés par une communauté du logiciel libre OpenUSS. Il initie ainsi :

- d'éventuelles négociations avec le concepteur pédagogique et l'analyste sur les *traces* captées, correspondant respectivement à l'acteur du *processus* de conception et à l'*acteur* du *processus* d'analyse.
- la création de nouvelles fonctionnalités de la plate-forme OpenUSS en intégrant un interpréteur du langage de spécification IMS LD.
- l'utilisation d'un formalisme conçu pour décrire *a priori* une session d'apprentissage et utilisé ici comme support d'une description *à posteriori*.

4.3.3.5 Bilan

L'acte de spécification est clairement défini par un ensemble de concepts dans le cadre ODP-RM. Nous avons présenté dans cet exemple un cas illustrant la démarche d'un acteur appartenant au *processus* génie logiciel dans le cycle d'une ingénierie d'un EIAH. Trois concepts du cadre ODP-RM guident le travail de spécification d'un objet diffusé par la communauté OpenUSS selon une approche dirigée par les modèles : *décomposition*, *gabarit* et *compatibilité comportementale*.

En effet, cette communauté ne se restreint pas à diffuser une plate-forme mais cherche à faire adhérer de nouveaux membres. Nous pouvons citer en exemple le travail de Ingo Düppe proposant un nouveau composant OpenCPMS (Open Controlling Performance Measurement System) chargé d'observer le comportement des composants du système de la plate-forme OpenUSS. Ou encore, le travail de (Iksal et Choquet 2005) proposant un nouveau langage UTL (Usage Tracking Language) qui permet de spécifier la sémantique des *traces* générées en session d'apprentissage. Les auteurs présentent une décription de la sémantique liée à l'utilisation d'une partie du modèle IMS LD comme modèle de représentation de *traces*. Arriver à expliciter et à décrire la logique du concepteur informatique exposée dans cet exemple permet de mieux comprendre ces récentes initiatives.

4.3.4 Utilisation du profil UML EDOC comme langage de spécification du point de vue métier

Ce chapitre s'intéresse à mettre en œuvre des propositions faisant la promotion de l'évolution conjointe entre la proposition du cadre ODP-RM de l'ISO et le langage de modélisation UML promu par le consortium OMG. Tous les deux instrumentent le travail de spécification des concepteurs. Très tôt, des propositions cherchant à rapprocher ces deux standards (Blanc, Gervais et al. 1999) ont émergé. Aujourd'hui, plusieurs réponses sont disponibles. Nous avons retenu celle de l'OMG qui propose un profil UML spécifique. Elle définit un système de notation permettant de représenter les

composants d'un système au sens d'ODP-RM. Dans cette partie, nous ne cherchons pas à valider un tel outil mais à montrer son potentiel à spécifier les deux exemples présentés.

4.3.4.1 Spécifier à l'aide du profil UML EDOC

L'interopérabilité dans un système de formation est l'une des principales problématiques liées à la spécification d'objets pédagogiques (Hummel, Manderveld et al. 2004). En choisissant le système de notation défini par le profil UML EDOC (Entreprise Distributed Object Computing) et standardisé par le groupe OMG, nous identifions un moyen complémentaire pour spécifier dans un formalisme interopérable l'aspect *métier* d'un système de formation. Ainsi, nous pouvons communiquer des structures organisationnelles et les pratiques métiers des communautés analysées selon le cadre ODP-RM.

La spécification du *point de vue métier* se définit à travers une vision fonctionnelle d'un système de formation qui permet d'exposer les *comportements* et les entités du domaine et les règles et les *stratégies* qui les mettent en relation. Avec cette proposition de standard, l'OMG définit une première *instance* du cadre ODP-RM de l'ISO et l'un des premiers éléments méthodologiques instrumentant le projet sur l'approche MDA^{TM} (OMG/MDA 2001b).

Le profil UML EDOC proposé par le groupe OMG est un méta-modèle qui étend la sémantique du langage de spécification orienté objet UML pour spécifier les composants d'un système. Ce méta-modèle se décompose en quatre modèles :

- le modèle « Component Collaboration Architecture » (CCA) permet de spécifier les différents niveaux de granularité, la structure et le *comportement* de ces différents composants ;
- le modèle « Entities Model » permet de décrire les concepts d'un domaine d'application ;
- le modèle « Events Model » permet d'apporter des précisions sur un système dirigé par des événements ;
- le modèle à « Business Process Model » spécifie l'architecture de collaboration des composants (CCA) ; il permet de spécifier le *comportement* du système dans un contexte métier précis lié à un domaine.

Ce méta-modèle se définit comme un langage de spécification et reprend les concepts de spécification du cadre ODP-RM. En effet, les quatre modèles présentés fournissent un système de notation pour spécifier les trois *points de vue métier*, *informationnel* et *computationnel* (cf. Figure 63).

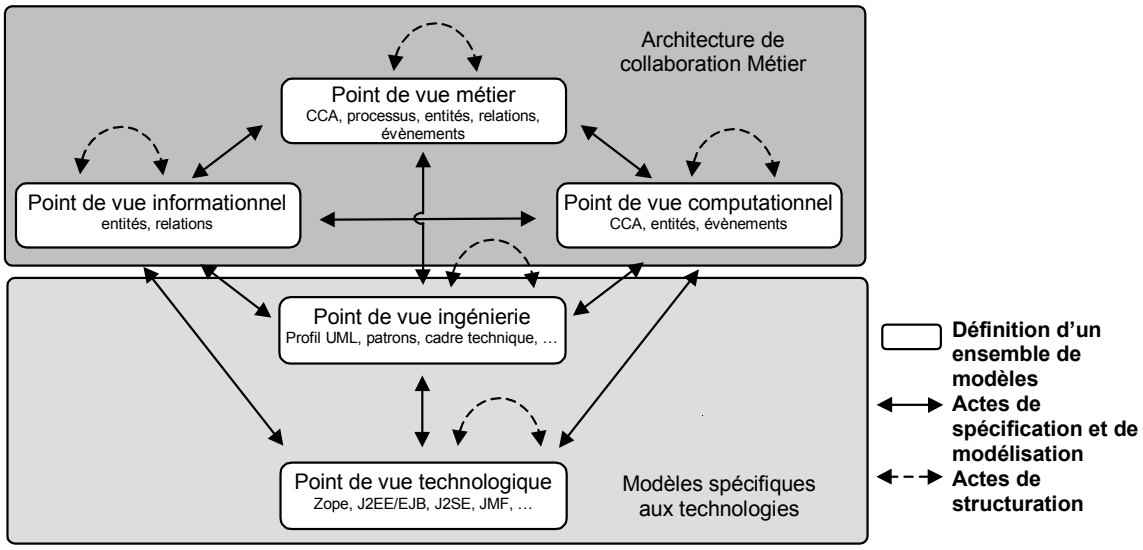


Figure 63 : Eléments du profil ECA par rapport aux points de vue ODP-RM, adaptée de la figure 2-1 (OMG/ECA 2004)

Nous voulons proposer un tel instrument aux concepteurs des EIAH comme un outil permettant de produire des spécifications partielles d'une *communauté* de réingénierie d'un système de formation. Les deux exemples suivants illustrent l'utilisation de ce formalisme. L'objectif est de spécifier les *activités* des concepteurs des deux exemples présentés.

4.3.4.2 Utilisation du profil UML EDOC pour spécifier le point de vue métier

Dans le *point de vue métier*, les modèles *métier* cibles et les besoins du système sont modélisés. Le concept principal de ce *point de vue* est le concept de *communauté*. Par définition, il désigne le regroupement d'un ensemble d'*objets métier* ayant des *objectifs* communs (ISO/IEC-15414 2002). La spécification du *processus métier* consiste à instancier les éléments du modèle CCA pour spécifier le *comportement* du système lié à un domaine spécifique. La finalité d'une telle *instance* est de représenter et de communiquer l'organisation des *rôles* des objets dans une chaîne de traitement de l'information [(OMG/ECA 2004), partie 2.4]. Les concepts principaux du *point de vue métier* du cadre ODP-RM sont mis en correspondance avec les entités du modèle CCA. Nous en présentons quatre :

- Le concept de *communauté métier* est modélisé par un assemblage de composants, désigné par l'entité « ProcessComponent », associés aux entitées « Composition » et « Choreography ». Une « Choreography » permet de spécifier la manière dont les messages sont échangés dans une « Composition » de plusieurs composants. « CommunityProcess » est l'entité de plus haut niveau dans la *composition*. Elle permet de désigner les composants qui travaillent ensemble.
- Le *processus métier* est spécifié comme l'entité « ProcessComponent ». Elle représente un *processus* actif de traitement du *point de vue métier*.

- Les *comportements* correspondent aux *interactions* dans lesquelles peuvent participer les *objets métiers*. Ils sont spécifiés par l'association des éléments « Port » et « Protocol ». Chaque « Port » permet de connecter deux composants en collaboration qui mettent en œuvre le même « Protocol ».
- Le *rôle* est représenté par le concept « PortConnector ». Il définit les points de connexion entre les composants et détaille chacun des ports mis en œuvre dans une *composition* donnée.

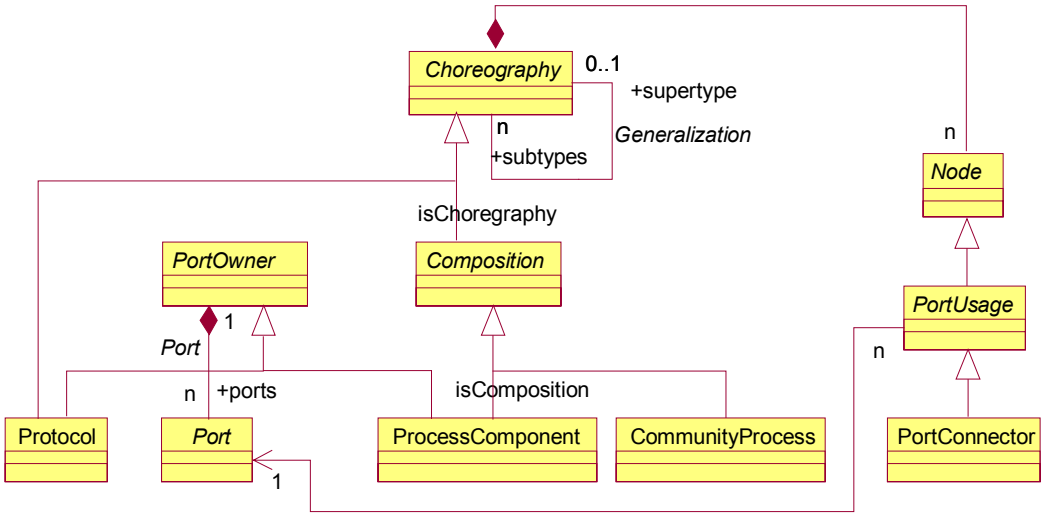


Figure 64 : Eléments extraits de la structure du méta-modèle du profil UML [(OMG/ECA 2004), figure 3-4]

La Figure 65 reprend notre proposition de *point de vue métier* présenté dans la sous-partie 3.3.3, page 84. Nous avons utilisé différents éléments du méta-modèle du profil UML EDOC auxquels l'*instance* présentée fait référence.

L'avantage du profil UML EDOC est de couvrir toutes les phases d'un *processus* de conception, de l'analyse à la maintenance en passant par la conception. Les productions des projets de communauté du logiciel libre et les propositions de modèles standards pour les technologies éducatives présentent une réelle opportunité pour spécifier et communiquer de nouveaux *comportements* de la *communauté* de réingénierie d'un système de formation. En effet, nous cherchons à réifier un tel modèle sur deux exemples où les logiques de transformations entre une démarche orientée produit et une démarche orientée processus peuvent être spécifiées et communiquées. Cette problématique est au cœur du travail mené actuellement par le groupe de l'ISO cherchant un modèle de processus qualité d'un système de formation (Blandin 2005; Corbière 2005).

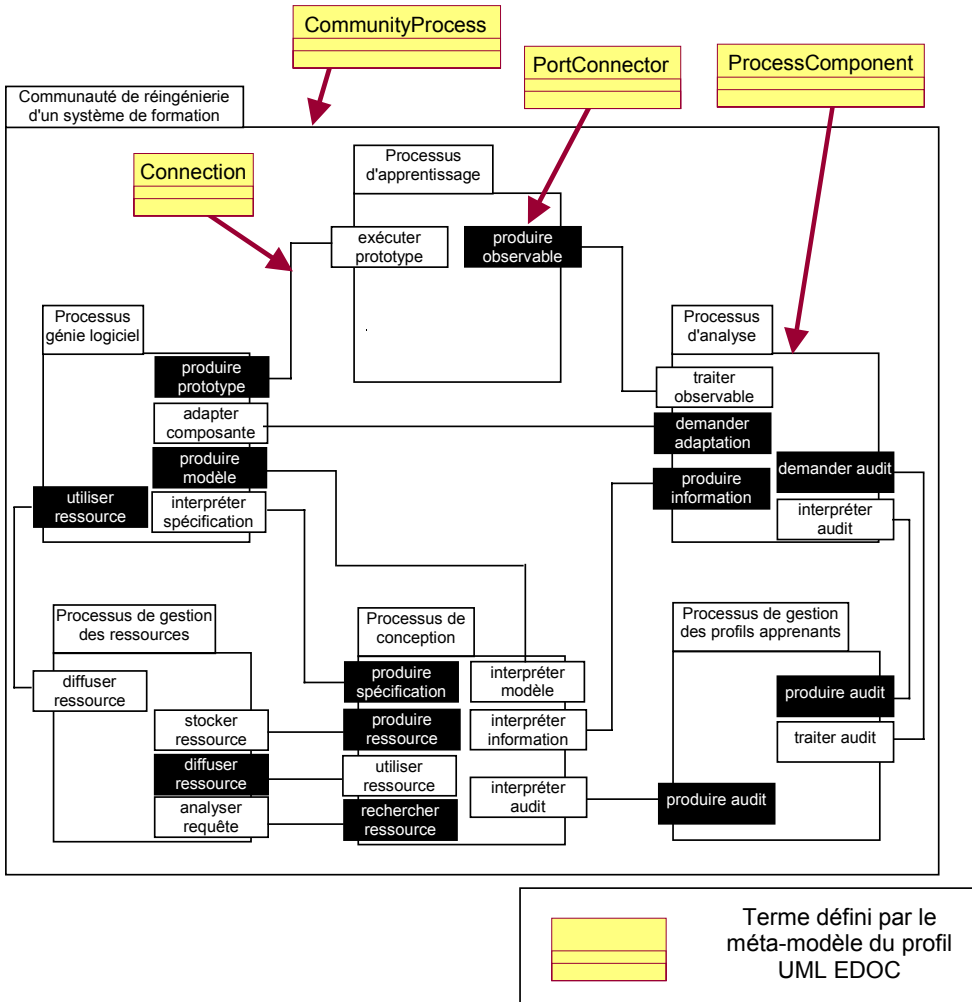


Figure 65 : Instanciation des entités du modèle CCA sur la proposition du point de vue métier d'un système de formation

4.3.4.3 Utilisation du profil UML EDOC pour spécifier les trois étapes de l'exemple 1

Les techniques d'observation des utilisations et les techniques de modélisation se doivent d'être spécifiées afin d'être communiquées à la communauté des concepteurs. Les trois phases présentées dans l'exemple 1 permettent d'identifier les premiers éléments méthodologiques d'un processus de développement dirigé par les modèles. Ces éléments peuvent être alors formalisés et partagés par les communautés de pratique. Pour cela, nous utilisons les éléments du profil UML EDOC pour spécifier les *activités* des différents *acteurs* au niveau de l'*instance* de la *communauté métier* de réingénierie d'un système de formation représentée par la Figure 66.

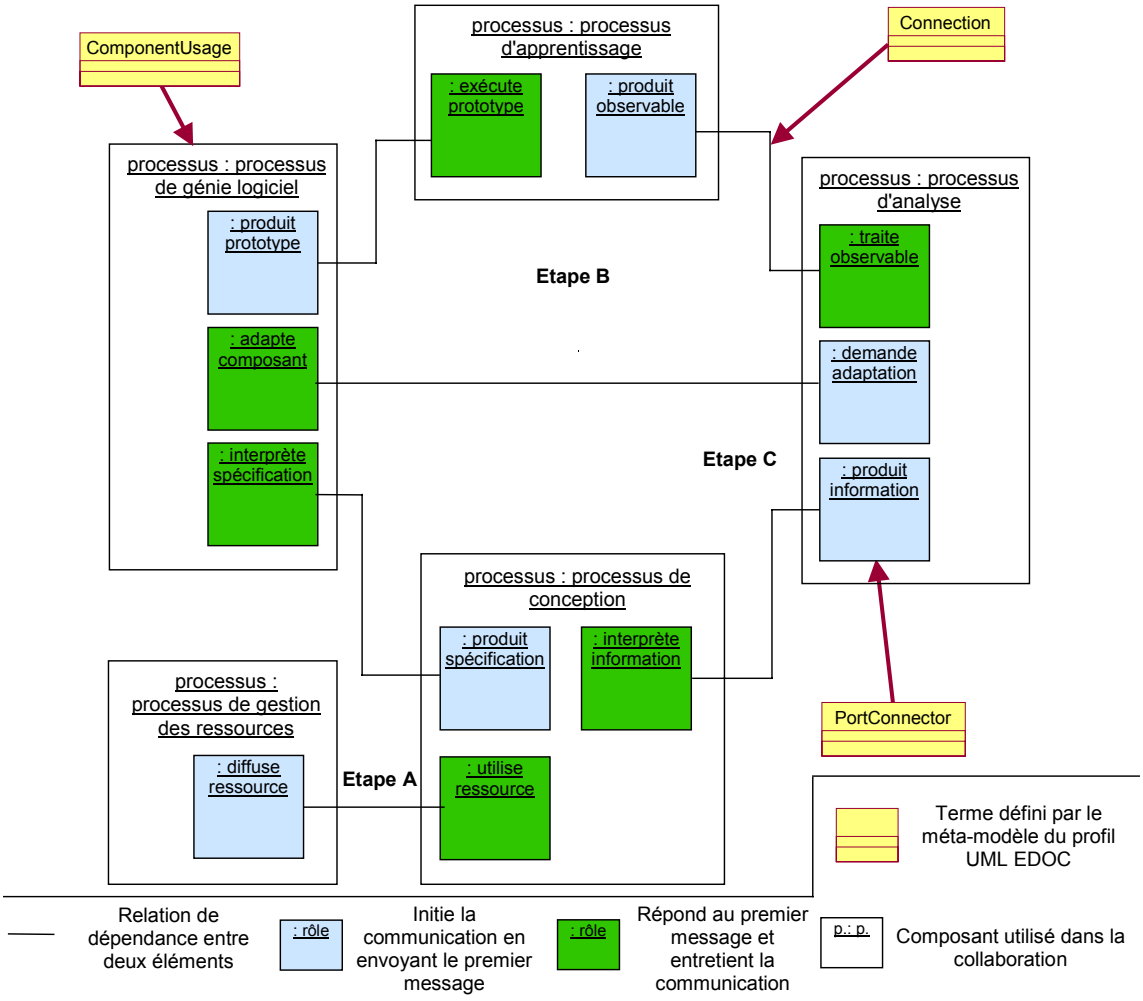


Figure 66 : Diagramme de collaboration UML de l'exemple 1

Nous cherchons maintenant à spécifier l'utilisation des différents *processus métier* de cet exemple. Pour cela, nous utilisons l'entité « ComponentUsage » du méta-modèle pour représenter chacun des composants utilisés. Ces derniers contiennent un ensemble d'*activités*. Chaque *activité* représente l'unité de travail à accomplir pour que le *processus* soit effectif. Pour représenter la collaboration entre les composants, nous utilisons également les concepts « PortConnector » et « Connection » du méta-modèle CCA (cf. Figure 67).

L'exemple illustré sur la Figure 66 représente les phases de modélisation sur les *ressources* diffusées par le communauté « FreeStyle Learning ». Elles montrent la collaboration entre les *processus métier* dans lesquels les concepteurs sont amenés à modéliser l'*interface*, le *comportement* et l'*état* des *objets* utilisés.

Les concepts du méta-modèle du profil UML EDOC nous permettent de décrire avec plus de précision l'utilisation de ces composants. L'*instance* de « CompoundTask » permet de désigner un regroupement de plusieurs *activités* (cf. Figure 68). Chacune des *activités* représente le *comportement* partiel d'un composant et correspond à l'entité

« Activity ». Le méta-modèle UML EDOC nous permet d'expliciter les conditions dans lesquelles se déroule une *activité*. Une *instance* de « ProcessRole » est alors créée. Les informations utilisées, les *acteurs* identifiés ou les *ressources* utiles sont alors spécifiées respectivement par les concepts « Performer », « Artefact » et « ResponsabilityParty ».

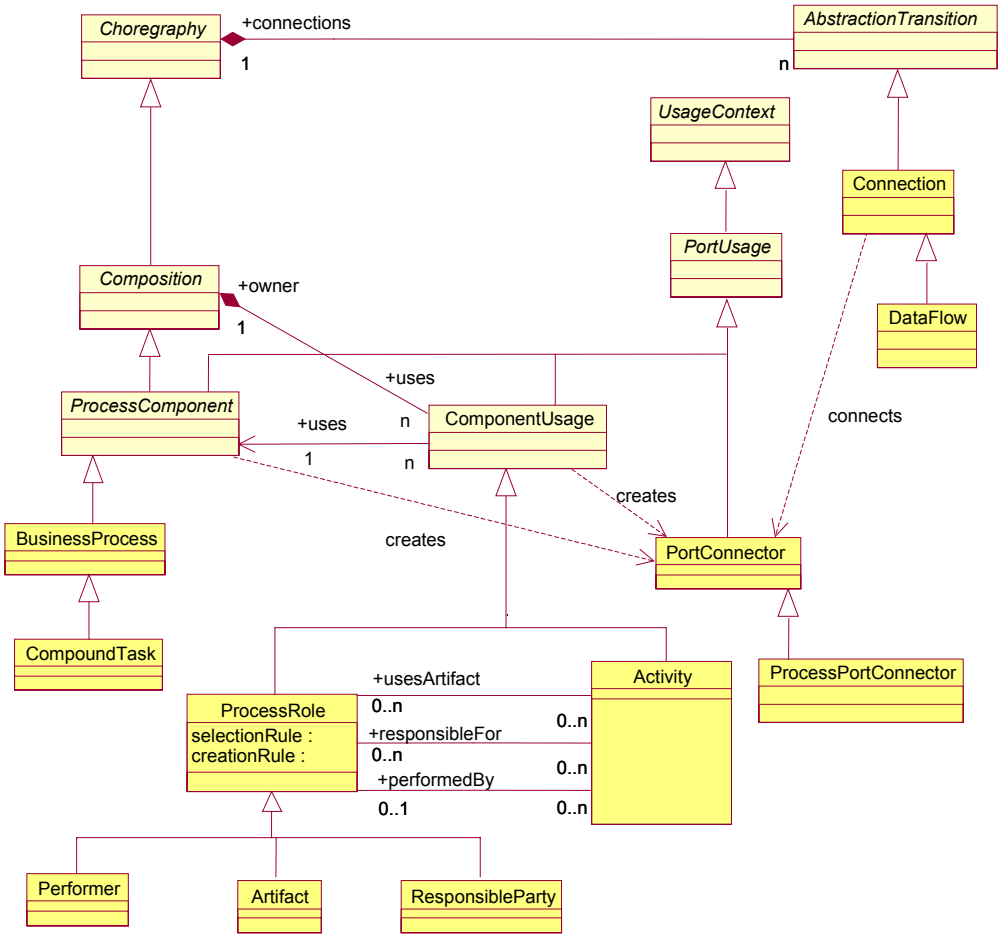


Figure 67 : Eléments extraits de la structure du méta-modèle du profil UML [(OMG/ECA 2004), figures 3-6 et 3-48]

Les trois *étapes* de l'exemple 1 sont décrites par les *activités* suivantes :

A. celle où le concepteur intervient lors de la spécification du scénario d'apprentissage ;

B. celle de l'informaticien qui, à l'aide d'un profil d'architecture, identifie le *comportement* à observer ;

C. et celle où l'analyste utilise un formalisme pour représenter les *états* de l'*objet* lors de son utilisation.

Le *processus* de négociation représenté par la Figure 68 décrit le regroupement de ces trois activités. Le concept « ProcessPortConnector » permet de décrire les logiques d'enchaînement entre ces *activités*. Ces logiques intègrent les contraintes temporelles ou de dépendance de données.

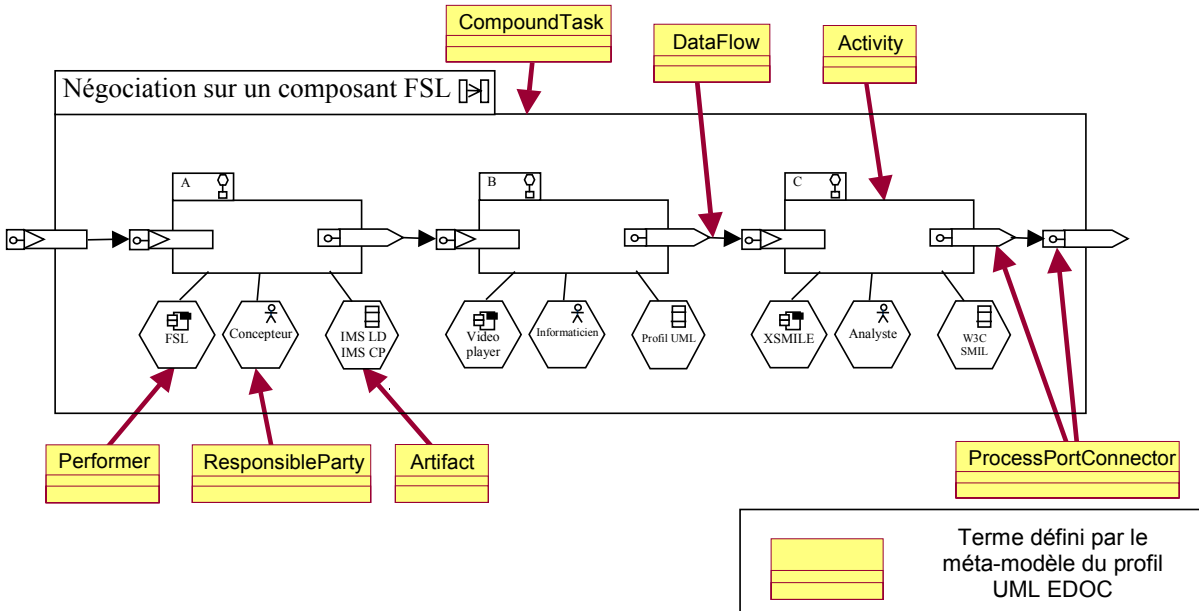


Figure 68 : Trois étapes de modélisation d'un EIAH présentées dans l'exemple 1

Cet exemple nous montre le potentiel du système de notation du profil UML EDOC à spécifier un *comportement* spécifique de notre *point de vue métier*. L'exemple présente le *processus* où trois *acteurs* (le concepteur pédagogique, l'informaticien et l'analyste) sont amenés à négocier sur les primitives des modèles de spécification du scénario d'apprentissage, du profil technique du composant et du formalisme pour représenter les *traces* analysées. Nous avons utilisé le système de notation du profil UML EDOC pour représenter la séquence des activités de cet exemple au cours de laquelle plusieurs acteurs interviennent et utilisent différents *artefacts*. Cette spécification détaille de manière partielle le *comportement* d'une *communauté métier* de réingénierie d'un système de formation.

4.3.4.4 Utilisation du profil UML EDOC pour spécifier la collaboration de l'exemple 2

Nous souhaitons maintenant utiliser le profil UML EDOC comme système de notation pour formaliser l'*activité* de spécification décomposée en trois phases de l'exemple 2.

Dans les première et deuxième phases, le concepteur informatique manipule plusieurs sources d'information diffusées par les membres de la communauté OpenUSS. Ce concepteur effectue alors un travail de spécification en étant guidé par les choix de conception pour développer les composants (cadres techniques, interface de programmation, guide de développement, …). Le concepteur informatique se doit de comprendre dans un premier temps les spécifications techniques, et dans un deuxième temps, d'utiliser et de spécifier les *objets* diffusés par la communauté OpenUSS. L'objectif, induit par le cadre ODP-RM, est de spécifier les observables des *objets* utilisés. Ce

concepteur a besoin de se placer selon le *point de vue ingénierie* où les logiques des développeurs de la communauté sont explicitées.

La troisième phase consiste pour le concepteur informatique à mettre en correspondance les concepts du modèle de spécification diffusés par la communauté du logiciel libre avec les items d'un des modèles de spécification diffusés par le consortium IMS.

Ces trois phases nous permettent de préciser un *comportement* particulier d'une *communauté métier* au sens du cadre ODP-RM. En effet, l'exemple décrit les *interactions* entre le *processus* de gestion de *ressource* et le *processus* du génie logiciel. L'activité de spécification présentée dans l'exemple 2 peut se décrire selon le profil UML EDOC par une collaboration entre deux « ComponentUsage » (cf. Figure 69). La diffusion de *ressources* par le *processus* de gestion des *ressources* et l'utilisation de ces *ressources* par le *processus* génie logiciel sont spécifiées par des échanges. Ces derniers sont représentés par l'entité « Connection » qui permet de spécifier la logique des échanges des messages produits et/ou consommés dans la collaboration.

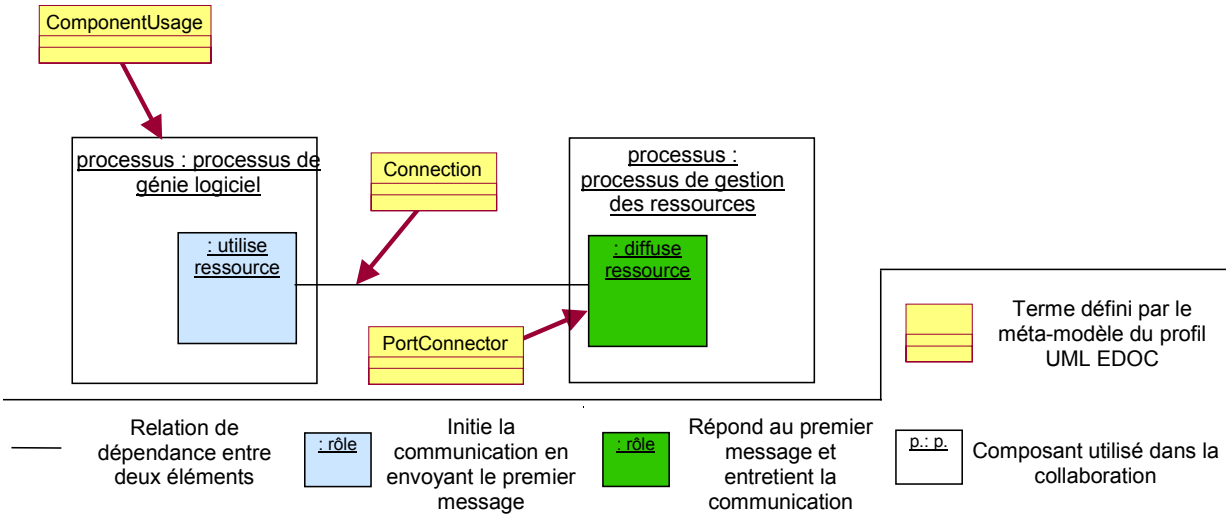


Figure 69 : Diagramme de collaboration entre le processus génie logiciel et le processus de gestion de ressources

Dans l'exemple 2, le concepteur informatique utilise des *ressources* diffusées par la communauté OpenUSS, ce qui l'amène à exploiter trois types de documents :

- Un modèle technique ;
- Une structure des sources ;
- Un modèle de spécification.

Ces échanges peuvent être décrites à l'aide de l'entité « Protocol » qui spécifie la communication entre deux composants. Le *comportement* des ports de chaque composant considéré dans la collaboration est défini par un *flux* de données correspondant à l'entité « FlowPort » du profil UML EDOC qui correspond à la forme la plus élémentaire d'un

port produisant et consommant des données de type simple. Afin de préciser le *comportement* du protocole complexe entre les deux composants, un graphe d'état est défini à l'aide des entités « PseudoState », « Transition » et « PortActivity », spécifiant respectivement les différents *états* d'un « Protocol », les enchaînements entre les *états*, et l'*activité* de chaque port. De plus, un schéma conceptuel peut être associé en utilisant le concept « CompositeData » permettant de spécifier le type de donnée manipulé dans la collaboration.

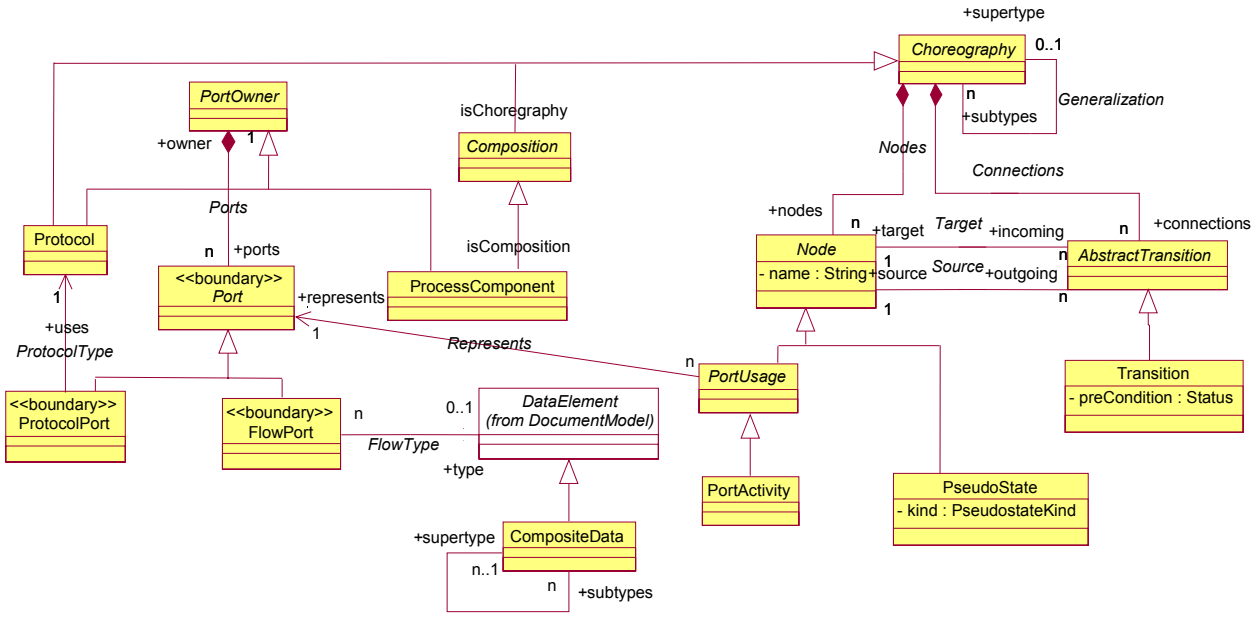


Figure 70 : Eléments extraits de la structure du méta-modèle du profil UML [(OMG/ECA 2004), figures 3-5 et 3-7]

Trois vues complémentaires (la décomposition du composant lié au protocole spécifié, le diagramme des *états* de ce protocole et le diagramme de classe définissant la structure du protocole) nous permettent d'illustrer sur la Figure 71 l'utilisation du profil UML pour spécifier le *comportement* lié à l'utilisation des différentes *ressources* mises à la disposition par la communauté OpenUSS.

L'exemple présenté utilisant le formalisme UML EDOC nous permet d'expliciter le besoin, pour le *processus* génie logiciel, d'accéder à ces trois types de documents et, du côté du *processus* de gestion des *ressources*, de diffuser ces différents documents.

Dans cet exemple, le système de notation du profil UML EDOC nous permet de formaliser certains aspects de la collaboration entre deux *processus* du *point de vue métier* où les *ressources* utilisées sont produites par des consortiums institutionnels comme IMS ou des communautés du logiciel libre comme OpenUSS. Nous pouvons envisager de communiquer cette spécification au consortium IMS afin de les sensibiliser sur l'existence d'outils aptes à spécifier la collaboration dans une communauté de concepteurs d'un système de formation et par conséquent d'engager un

nouveau projet sur cette problématique ou de communiquer à la communauté OpenUSS un modèle d'utilisation des *ressources* produites et diffusées.

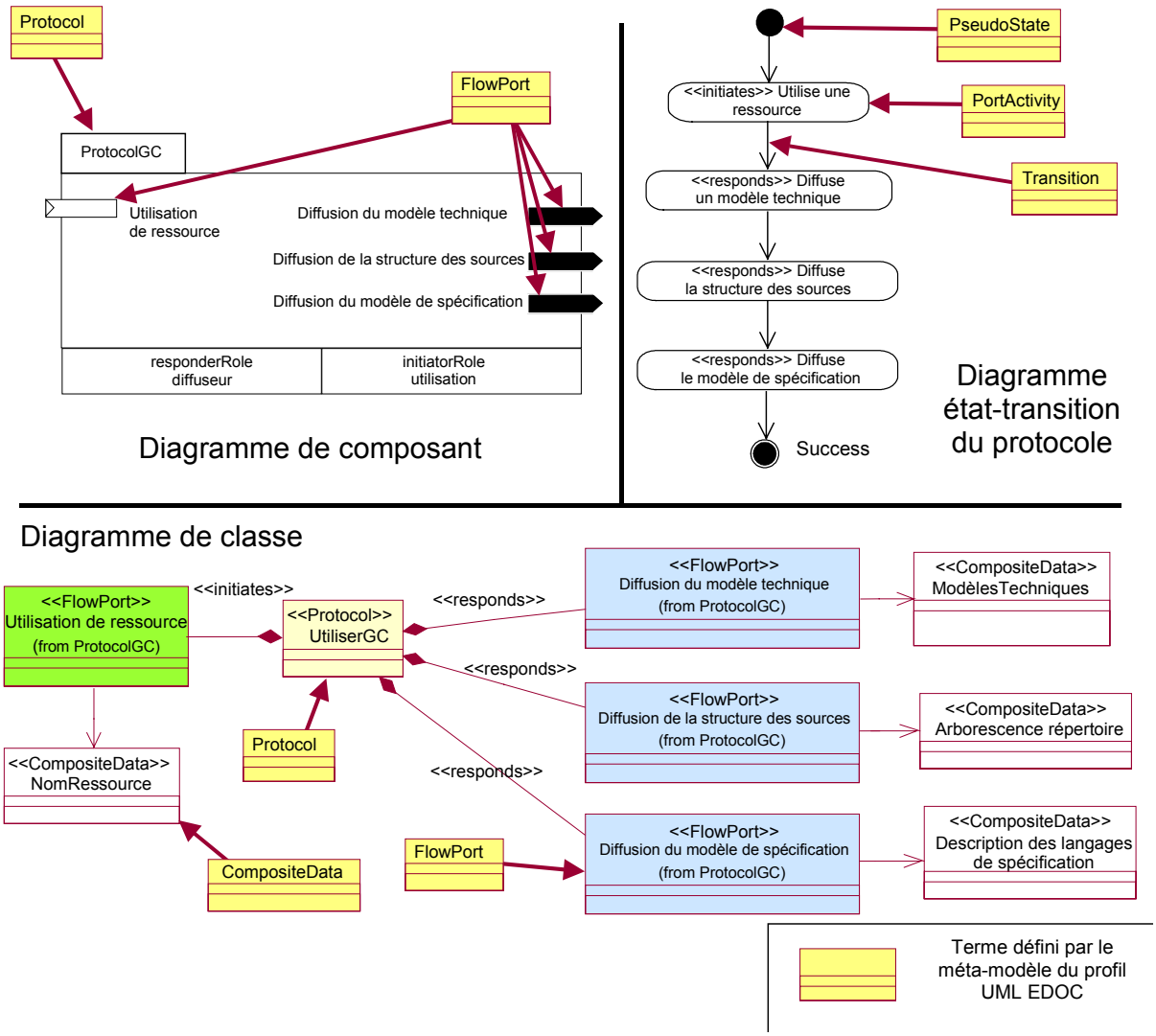


Figure 71 : Trois aspects du protocole définissant la collaboration entre les processus génie logiciel et de gestion de ressources de l'exemple 2

4.3.4.5 Bilan

Les deux exemples présentés spécifient le détail du *comportement* de certains aspects de la structure des *processus métier* formalisant une *communauté* de réingénierie d'un système de formation dans le cadre ODP-RM.

Nous avons identifié le profil UML EDOC comme un instrument capable de spécifier le travail des concepteurs dans leurs *activités* de création, d'utilisation, de modification et de maintenance de ces composants diffusés par des communautés.

L'objectif n'est pas de valider le langage de spécification proposé par le comité OMG mais de présenter deux exemples d'utilisation. Nous montrons ainsi le potentiel de cet outil à spécifier certains aspects du *comportement* des communautés de conception. Ainsi formalisés, ces modèles permettent d'engager des réflexions sur les pratiques des concepteurs dans un système de formation.

Un tel outil envisage la spécification des *activités* des concepteurs d'une *communauté* de réingénierie d'un système de formation comme une ingénierie de composants dans les systèmes d'information (Barbier, Cauvet et al. 2004). Un composant peut être réutilisé pour concevoir d'autres composants ou pour supporter la négociation entre des composants. Les deux exemples présentés illustrent ces potentiels : le premier présente le *comportement* d'une structure de composants et le deuxième présente un protocole de négociation entre deux composants. Nous pouvons voir dans ce nouveau formalisme une nouvelle réponse aux préoccupations des différents projets sur les technologies éducatives normées ou les modèles de spécification diffusés par le consortium IMS pour la gestion des plates-formes de formation à distance supportent l'interopérabilité dans un système de formation. Le formalisme défini par le profil UML EDOC vient compléter ces initiatives et nous aide à expliciter les comportements et les échanges entre les *interfaces* des composants de notre *point de vue métier* sur un système de formation mettant en œuvre ces technologies (cf. Figure 30, page 75).

Les deux exemples présentés dans ce sous-chapitre montre qu'un tel formalisme est une première contribution à la problématique cherchant à fédérer les propositions sur les technologies éducatives normées et les outils logiciels diffusés par les communautés du logiciel libre diffusant les composants d'un système de formation.

4.4 Synthèse du chapitre

Dès les premières initiatives de projet sur les technologies éducatives normées, les problématiques posées sont de disposer d'outils pour anticiper et supporter l'émergence de communauté de pratiques autour des alternatives sur les processus de développement. Très tôt, le développement dirigé par les architectures à composants a été l'une des alternative à explorer (Roschelle, Kaput et al. 1999). Actuellement, la communauté EIAH (Adam, Bessagnet et al. 2005) s'interroge sur la pertinence d'appliquer les solutions adoptées par la communauté génie logiciel pour répondre aux problématiques d'intégration des technologies dans un système de formation.

Dans ce chapitre, nous faisons jouer le *rôle* de fournisseur de composants à la fois aux communautés du logiciel libre et au consortium IMS. L'analyse de ces composants dans le cadre ODP-RM nous amène à modéliser les choix des concepteurs et à spécifier de nouvelles interactions au niveau du *point de vue métier* ou au niveau de l'articulation entre les *points de vue* de l'architecture du cadre ODP-RM.

Nous rappelons que ce cadre se présente comme un guide pour le développement de standards de conception d'un système distribué. Il répond à la problématique d'échange d'information entre les groupes dans une organisation et entre des organisations coopérantes. Plusieurs catégories de standards peuvent être définies [(ISO/IEC-10746-1 1998), sous-partie 6.3.2] :

- Cadres architecturaux additionnels, qui viennent compléter le cadre ODP-RM sur les domaines spécifiques liés au nommage, à la sécurité et à la cohérence entre les modèles.
- Standards de notation : ils définissent les notations pour exprimer les spécifications des différents aspects de l'intégration ou la distribution d'un système, et les règles relatives à ces différentes spécifications.
- Standards de composant : ils définissent le composant prenant en compte une ou un ensemble de fonctions définies par le cadre ODP-RM. Ces composants correspondent directement à un élément logiciel du système développé.
- Standards de composition de composants : ils définissent la composition et la coordination dans l'utilisation d'un ensemble de composants pour atteindre un certain objectif d'un système considéré dans sa globalité en ne spécifiant qu'un aspect particulier et en en cachant d'autres.

L'utilisation conjointe du cadre ODP-RM et du profil UML EDOC nous amènent à analyser les productions des communautés du logiciel libre et les modèles de spécification produits par les comités diffusant des modèles de spécification sur les technologies éducatives. Le cadre ODP-RM en définissant un ensemble de termes permet de supporter la communication entre les différents acteurs d'un système : les utilisateurs, l'équipe chargée de gérer un tel système et les différents fournisseurs de composants du système. Le profil UML EDOC vise à représenter par un système de notation graphique la collaboration entre les composants du système considéré. Les représentations obtenues se destinent à être directement interprétées par les environnements de développement logiciel utilisés par le concepteur informatique dans son travail de modélisation, de structuration et de spécification. En effet, toutes les spécifications promues par le groupe OMG sont vouées à être interprétées par les applications conçues par l'industrie informatique. Un formalisme spécifique lui est associé, XMI (XML Metadata Interchange), et lui garantit l'interopérabilité syntaxique des données échangées.

Les concepteurs ont donc la possibilité de partager, de réutiliser et de modifier les spécifications. Cependant, le travail de spécification illustré par nos deux exemples nous montre que la sémantique liée aux négociations entre les concepteurs (exemple 1) ou entre le concepteur informatique et les modèles de spécification (exemple 2) est perdue. En effet, le profil UML EDOC, utilisé comme système de notation, ramène ces échanges respectivement à une séquence d'activités ou à un protocole d'échanges entre composants.

L'objectif du travail présenté dans ce chapitre a été d'illustrer l'utilisation de ces deux outils et de démontrer leurs capacités à formaliser l'activité des concepteurs mettant en œuvre des outils, des formalismes et des collaborations avec d'autres concepteurs.

CHAPITRE 5 : CONCLUSION ET PERSPECTIVES

5.1 Synthèse des travaux

Le travail présenté dans cette thèse s'inscrit dans le projet de recherche « Réingénieirie des EIAH dirigés par les modèles » (REDiM) dont un des objectifs est de proposer et de valider un cadre d'ingénierie et de réingénierie de la formation à distance en intégrant les récents efforts de normalisation des technologies éducatives. Ce cadre s'adresse aux concepteurs, afin de les accompagner dans leur travail, de favoriser l'acquisition de nouvelles connaissances et compétences, notamment d'ordre conceptuel, organisationnel et méthodologique. En amenant les concepteurs à spécifier de manière explicite leurs intentions de conception d'un système de formation dans des modèles, et en confrontant ces modèles de spécification avec les usages observés durant les sessions d'apprentissage, nous accompagnons la nécessaire démarche réflexive de l'équipe de conception sur ses méthodes, et favorisons le développement et la capitalisation des savoir-faire et des démarches pédagogiques au sein d'une communauté de concepteurs.

Les concepteurs visant à spécifier un système de formation cherchent des modèles pour appréhender leurs pratiques pédagogiques. Actuellement, les technologies éducatives normées apportent deux réponses ; la première est centrée sur la gestion des contenus et la seconde sur le processus d'apprentissage. Dans le premier cas, ces technologies doivent intégrer les contraintes d'accessibilité, de modularité et d'interopérabilité (Friesen 2001). En cela, nous avons analysé les travaux du groupe LTSC du comité IEEE. Dans le second cas, elles visent à optimiser les échanges sur les styles ou sur les approches d'apprentissage entre les concepteurs pédagogiques. Elles se situent au cœur des projets d'une démarche qualité dans un système de formation (Pawlowski 2002). Nous nous sommes intéressés plus particulièrement aux modèles informationnels de spécification du consortium IMS Global Learning.

Nous avons proposé le cadre de référence du processus distribué ouvert de l'ISO et de l'IEC comme le méta-standard à adresser aux concepteurs de la communauté EIAH. La vision globale d'un système est explicitée par des mécanismes de transformations des différents points de vue. Ce cadre applique les théories systémiques (Wegmann et Naumenko 2001) qui ont pour rôle d'intégrer différentes disciplines telles que la théorie de l'information, la cybernétique et la théorie des systèmes.

Un tel cadre a pour vertu d'adresser un ensemble de termes à la communauté de concepteurs soucieux de partager les pratiques. Ce vocabulaire permet d'expliciter les invariants et les principes généraux, structuraux et

fonctionnels d'un système de formation. Le travail présenté dans cette thèse vise à présenter différentes instances d'un tel méta-standard et d'illustrer ainsi les différents apports de ce modèle à la communauté des concepteurs des EIAH. Nous les avons présentés suivant trois axes.

- Le premier analyse les négociations identifiées dans les communautés de normalisation intégrant les réactions des communautés scientifiques et les pratiques des concepteurs les mettant en œuvre. L'arrivée des premiers consensus autour des technologies éducatives normées proposées par les deux consortiums IEEE et IMS engagent de nouvelles négociations. Elles portent sur les limites des propositions actuelles, les mécanismes de transformation des modèles de spécification proposés, leurs perspectives d'évolution et l'activité de négociation engagée par ces propositions. De tels constats nous ont amené à proposer un point de vue supplémentaire, le point de vue métier du cadre ODP-RM. Nous avons proposé et adressé une instance de ce point de vue aux concepteurs d'un système de formation.
- Le deuxième analyse l'impact de l'évolution du développement d'un système informatique sur la conception d'un EIAH. Nous avons mis en relation les modèles répondant aux récentes problématiques de rétroconception et de réingénierie d'un système informatique et les modèles de spécification proposés par les communautés de normalisation sur les technologies éducatives. Le cadre ODP-RM nous a aidé à identifier les cohérences entre ces différents modèles. Sa terminologie nous a permis dans un premier temps d'affiner notre modèle métier d'un système de formation et de décrire deux exemples illustrant les actes de rétroconception et de réingénierie du concepteur sur un tel système.
- Le troisième présente, suivant la perspective de l'évolution des processus de développement logiciel, l'utilisation de deux instruments : les concepts du cadre ODP-RM et le système de notation du profil UML EDOC du comité OMG. Ces deux outils ont guidé ainsi notre analyse de l'impact des modèles de spécification sur les technologies éducatives et des composants diffusés par les communautés du logiciel libre. Nous avons illustré par deux exemples le travail de modélisation et de spécification des concepteurs dans notre vision d'un processus de conception d'un système de formation dirigé par les modèles. L'utilisation du profil UML EDOC est présentée en exemple pour démontrer son potentiel à spécifier les deux exemples précédents.

Comme nous venons de le voir, notre travail s'est structuré autour de ces trois analyses guidées par le cadre ODP-RM. Elles initient trois nouvelles réflexions sur la conception d'un EIAH portant sur l'utilisation des technologies éducatives normées, sur la réingénierie dans les EIAH et sur le processus de développement dirigé par les modèles. De plus, les différentes illustrations présentées dans le mémoire montrent le potentiel du cadre choisi à formaliser ces différents constats et à créer un ensemble d'instances décrivant de nouveaux éléments méthodologiques à adresser à la communauté de concepteurs d'un système de formation.

5.2 Apports de la thèse

Tout concepteur définissant une session de formation est amené à s'interroger sur le contenu, le savoir-faire à transmettre, les relations entre la tâche et l'apprenant, ainsi que entre l'apprenant et le logiciel. Un concepteur cherche à exploiter les potentiels du système utilisé et à comprendre les intentions et les actions des apprenants (Linard 2003). Il est amené à décrire, à choisir différents composants réutilisables pour construire, modifier, adapter, un scénario pédagogique *a priori* et l'analyser *à posteriori* (Barré, Choquet et al. 2003).

L'émergence des formalismes de spécification définis par les technologies éducatives constitue une opportunité pour supporter la description des observables, leur traçage lors des sessions et leur analyse en retour de formation. Ces nouveaux outils adressés à la communauté EIAH ne sont pas suffisants.

Le premier apport de cette thèse est de présenter une analyse critique des projets à l'origine des propositions de standards sur les technologies éducatives :

- L'architecture LTSA du comité IEEE a pour finalité de guider la démarche du concepteur. Aujourd'hui promue au rang de standard, cette architecture n'engage plus aucune activité au sein du comité IEEE. Cependant, ce travail est à considérer comme l'initiateur des projets sur les technologies éducatives actuelles menés par les différents consortiums. Le modèle à processus au cœur de cette architecture présente une vision cyclique d'un dispositif d'apprentissage centrée sur les interactions entre l'apprenant et le système. Les différentes *instances* du modèle formalisent une première taxonomie des comportements d'un dispositif de formation. Formulées sous la forme de contraintes, elles sensibilisent le concepteur soucieux de décrire et d'observer les sessions d'apprentissage. Comme il est précisé dans la documentation [(IEEE/LTSA 2003), page 92], cette architecture est une *instance* du cadre général définissant un ensemble de concepts liés à la description d'une architecture d'un système informatique du comité IEEE (IEEE/ADS-IS 2000). Ces concepts, la dimension cyclique du modèle et la nécessité de stocker les ressources pédagogiques et les informations des apprenants sont repris dans notre proposition d'une communauté de réingénierie d'un système de formation.

- Les langages EML décrivent de manière formelle le déroulement d'une session de formation. En prenant comme référent le consortium IMS, la rigueur au niveau de l'organisation des différents modèles produits induit des règles pour la communauté des concepteurs de ressources quant à leur structuration et à leur syntaxe. En choisissant la technologie XML, ce consortium montre la volonté d'inscrire ces projets dans une perspective d'automatisation du processus d'apprentissage en donnant les moyens aux systèmes d'interpréter et d'agir en conséquence sur le déroulement de la session d'apprentissage (Koper 2004). Notre

approche diffère en positionnant les concepteurs au cœur d'une communauté de réingénierie d'un EIAH. Notre apport a été d'identifier et de leur adresser le cadre ODP-RM comme l'outil les aidant à expliciter leurs travails de modélisation et de spécification d'un système de formation mettant en œuvre ces langages.

Le second apport a été de présenter sous un nouvel angle de vue le travail de réingénierie et d'ingénierie d'un système de formation. Il correspond à l'*instance* du cadre ODP-RM proposé par l'ISO sur un système de formation. Cette *instance* s'adresse aux concepteurs d'un EIAH et aux développeurs des outils produits par des communautés de pratique. Appliquer un tel cadre au domaine de la formation, nous a amené à considérer dans un premier temps un système de formation comme une organisation structurée et instrumentée, et dans un deuxième temps à disposer d'un ensemble de concepts pour spécifier les différents artefacts obtenus à la suite des mises en pratique des composants diffusés par des communautés du logiciel libre.

L'apport principal a été de choisir et d'utiliser un cadre considérant les deux rôles, le premier de nature « anthropocentrée » et le second de nature « technocentrée » (Rabardel 1995), comme complémentaires sur l'utilisation des propositions de standards sur les technologies éducatives. Le premier permet de comprendre le travail des concepteurs de systèmes de formation souhaitant prendre part aux innovations pédagogiques, qu'ils soient enseignants, institutions de formation, communauté scientifique ou communauté du logiciel libre. Le second est mettre en exergue des réalités économiques de la formation à distance, où les technologies sont utilisées comme de simples médiateurs de contenu.

5.3 Perspectives des travaux

Avant d'exposer les perspectives de nos travaux, nous rappelons les trois analyses précédemment présentées sur lesquelles reposent la recherche menée dans le cadre de cette thèse.

L'analyse de la synergie entre une communauté du logiciel libre et une approche dirigée par les modèles nous a permis de formaliser notre contexte de travail à l'aide d'un *point de vue* spécifique sur un système de formation. Ce résultat a été possible en choisissant le cadre ODP-RM, défini par l'ISO, qui met à disposition un nouveau *point de vue* ; le *point de vue métier*, et un ensemble de concepts définissant le travail de modélisation et de spécification des concepteurs d'un système de formation. Ce cadre nous a servi de guide pour mener une réflexion sur les processus de standardisation sur les technologies éducatives de deux consortiums IEEE et IMS. Cette approche est très similaire au travail mené par le groupe OMG dont la principale mission est de gérer le processus de standardisation sur les langages de spécification sur les technologies orientées objet. Afin de redéfinir ces nouvelles missions, ce comité a fait le choix d'adopter un cadre structurant intitulé MDA^{TM}. Un tel cadre a permis au comité OMG de définir ces nouveaux axes de travail. Le premier axe est d'instrumenter l'ingénierie des modèles où la problématique est d'identifier les mécanismes

liés au travail de modélisation des paradigmes de différents domaines (Bézivin 2004). Le rôle du consortium OMG est de fédérer les groupes de travail sur la définition des DSL (Domain Specific Language). Un DSL se définit comme « un langage de programmation ou langage de spécification exécutable qui offre, à travers une notation, une abstraction appropriée d'un domaine de problèmes particuliers » (Deursen, Klint et al. 2000). Dans le domaine de la formation à distance, la communauté du logiciel libre OpenUSS (Grob, Bensberg et al. 2005) propose une des premières contributions à ce nouveau chantier. En mettant en œuvre ce premier résultat, la perspective est de s'interroger sur le rôle de notre proposition du *point de vue métier* d'un système de formation et sur l'apport du cadre ODP-RM dans la définition des nouveaux éléments d'un DSL dans le domaine de la formation à distance.

L'analyse de la synergie entre une communauté du logiciel libre et un processus de développement logiciel nous a permis d'illustrer une réingénierie d'un système de formation. Nous avons adhéré à la communauté du logiciel libre « FreeStyle Learning ». L'un des principaux acteurs de ce projet communautaire, Jan von Brocke, mène actuellement une réflexion scientifique visant à définir un cadre lui permettant de définir les différentes tâches de « dissémination » sur les points de vue applicatif, organisationnel, technologique et méthodologique (Brocke 2005). Une telle perspective nous montre que le concept de « FreeStyle » engage différents enjeux ne se focalisant pas uniquement sur des aspects technologiques. En adhérant à cette communauté, les membres bénéficient de ressources de qualité. Cela se matérialise au niveau du code source élaboré par les développeurs mettant en œuvre les techniques de « refactoring » (Fowler, Beck et al. 1999) et des patrons de conception (Gamma, Helm et al. 1999). De plus, ses membres mettent en pratique un style d'apprentissage spécifique à cet outil. En choisissant le cadre ODP-RM pour guider les activités de rétroconception et de réingénierie, une perspective est de poursuivre l'analyse de ce méta-standard afin de déterminer ses limites à formaliser et à communiquer les observations sur les différents points de vue d'un système de formation.

L'analyse de la synergie entre une approche dirigée par les modèles et un processus de développement logiciel montre qu'il est nécessaire d'engager avec la communauté du logiciel libre OpenUSS une coopération plus étroite. Deux constats viennent motiver cette perspective. Le premier vient du fait que ce travail est reconnu par le consortium IMS dans la spécification d'un cadre abstrait d'un système de formation (Smythe 2003). Le second est que l'outil a une certaine légitimité auprès des institutions de formation les mettant en pratique. Une moyenne de trois cents téléchargements est effectuée chaque mois sur la plate-forme de développement collaboratif SourceForge[57]. Deux enjeux peuvent se dégager de cette coopération. Le premier permet de mener une analyse critique sur les langages de description sur l'architecture afin de rendre plus explicite l'évolution des outils diffusés par cette communauté. Le deuxième vise à accompagner l'initiative du développeur Ingo Düppe, membre de la communauté OpenUSS, cherchant

[57] Voir, http://sourceforge.net/project/stats/detail.php?group_id=6259&ugn=openuss&mode=alltime&type=prdownload

à intégrer la fonctionnalité de présentation des statistiques sur le comportement des composants de la plate-forme OpenUSS.

Le travail présenté dans ce mémoire va nous permettre de proposer le cadre ODP-RM comme base d'échanges entre différentes communautés. Nous pensons notamment aux organisations et consortiums déjà bien développés, tels que l'OMG, le W3C et l'IMS. La proposition que nous faisons vise à montrer à ces groupes que l'univers du logiciel libre est un exemple pertinent et élaboré de communauté de pratique au sens de Wenger, et qu'adopter ces modes de fonctionnement permettraient de placer les usagers, et notamment les concepteurs, au centre du processus d'ingénierie et réingénierie des EIAH.

BIBLIOGRAPHIE

Adam, J.-M., M.-N. Bessagnet, A. Bouzeghoub, P.-A. Caron, T. Carron, C. Choquet, B. David, A. Derycke, S. George, S. Iksal, J.-M. Labat, P. Laforcade, X. le Pallec, C. Lecocq, V. Luengo, T. Nodenot, L. Oubahssi, Y. Peter, I. Rebaï, M. Rosselle et T. Vantroys (2005). *Contributions de l'Action Spécifique Conception d'une Plate-forme pour la recherche en EIAH à l'ingénierie des EIAH*, In: Revue Sciences et Technologies de l´Information et de la Communication pour l´Éducation et la Formation (STICEF), vol. 12, n°HS.

ADL/SCORM (2004a). *Sharable Content Object Reference Model Overview, 2nd Edition*, edt. Advanced Distributed Learning, 57 p.

ADL/SCORM (2004b). *Sharable Content Object Reference Model Runtime Environment, version 1.3.1*, edt. Advanced Distributed Learning, 209 p.

Alexander, C., S. Ishikawa et M. Silverstein (1977). *A Pattern Language : Towns, Buildings, Construction*, edt. Oxford University Press, ISBN 195019199.

Amorim, R. R., M. Lama, E. Sánchez, A. Riera et X. A. Vila (2006). *A Learning Design Ontology based on the IMS Specification*, In: Educational Technology & Society, vol. 9, n°1, p. 38-57, ISSN 14364522.

Avgeriou, P., A. Papasalouros, S. Retalis et M. Skordalakis (2003). *Towards a Pattern Language for Learning Management Systems*, In: Educational Technology & Society, vol. 6, n°2, p. 11-24, ISSN 14364522.

Bär, H., M. Bauer, O. Ciupke, S. Demeyer, S. Ducasse, M. Lanza, R. Marinescu, R. Nebbe, O. Nierstrasz, M. Przybilski, T. Richner, M. Rieger, C. Riva, A.-M. Sassen, B. Schulz, P. Steyaert, S. Tichelaar et J. Weisbrod (1999). *The FAMOOS Object-Oriented Reengineering Handbook*, 232 p.

Bar, M. et K. Fogel (2003). *Open Source Development with CVS, 3rd edition*, edt. Paraglyph Press, ISBN 1932111816.

Barbier, F., C. Cauvet, M. Oussalah, D. Rieu, S. Bennasri et C. Souveyet (2004). *Concepts clés et techniques de réutilisation dans l'ingénierie des systèmes d'information*. Ingénierie des composants dans les systèmes d'information, edt. Hermes, vol. 10, p. 11-35.

Barbot, M.-J. et G. Camatarri (1999). *Autonomie et apprentissage, l'innovation dans la formation*, edt. Puf, ISBN 2130501222.

Barré, V. (2004). *EMLs : case study in distance learning education*, In: International Conference on Computer Aided Learning in Engineering Education (CALIE 2004), Grenoble, p. 155-160.

Barré, V. et C. Choquet (2005). *Language Independent Rules for Suggesting and Formalizing Observed Uses in a Pedagogical Reengineering Context*, In: IEEE International Conference on Advanced Learning Technologies (ICALT 2005), Kaohsiung (Taïwan), p. 550-554.

Barré, V., C. Choquet, A. Corbière, P. Cottier, X. Dubourg et P. Gounon (2003). *MOCA, une approche expérimentale de l'ingénierie des EIAH*, In: Conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH 2003), Strasbourg, p. 55-66.

Bass, L., C. Buhman, S. Comella-Dorda, F. Long, J. Robert, R. Seacord et K. Wallnau (2001). *Volume I: Market Assessment of Component-Based Software Engineering*, CMU/SEI, T. N. CMU/SEI-2001-TN-007, Carnegie Mellon University, edt. Software Engineering Institute, 41 p.

Bass, L., P. Clements et R. Kazman (2003). *Software Architecture in Practice, second edition*, edt. Addison Wesley, ISBN 321154959.

Bélisle, C. et M. Linard (1996). *Quelles nouvelles compétences des acteurs de la formation dans le contexte des TIC ?*, In: Education permanente, vol. 127, n°2.

Berners-Lee, T., J. Hendler et O. Lassila (2001). *The Semantic Web*, In: Scientific American, vol. 284, n°5, p. 35-43.

Bertrand, I. (2003). *Les dispositifs de FOAD dans les établissements d'enseignement supérieur : transfert ou intégration ?*, In: Distances et savoirs, vol. 1, p. 61-78.

Bézivin, J. (2004). *Sur les principes de base de l'ingénierie des modèles*. Des octets aux modèles. Vingt ans après: où en sont les objets ?, edt. Hermes, vol. 10, p. 145-156.

Blanc, X., M.-P. Gervais et R. L. Delliou (1999). *Using the UML Language to Express the ODP Enterprise Concepts*, In: International IEEE EDOC Conference (EDOC 1999), Mannheim (Allemagne), p. 50-59.

Blandin, B. (2003). *French Comments on ISO/IEC JTC1 SC36 WG5 N0011 Learning Technology Systems Architecture (LTSA)*, N0014, edt. ISO/IEC JTC1 SC36 WG5, 5 p.

Blandin, B. (2004). *Are e-learning standards neutral ?*, In: International Conference on Computer Aided Learning in Engineering Education (CALIE 2004), Grenoble, p. 51-62.

Blandin, B. (2005). *What kind of Framework do we need to describe ITLET environments?*, N0057, edt. ISO/IEC JTC1 SC36 WG5, 4 p.

Booch, G., J. Rumbaugh et I. Jacobson (2000). *Le guide de l'utilisateur UML*, edt. Eyrolles, Addison Wesley, ISBN 2212091036.

Brocke, J. v. (2001). *Freestyle Learning - Concept, Platforms, and Applications for Individual Learning Scenarios*, In: 46th International Scientific Colloquium, Ilmenau (Allemagne), p. 149-151.

Brocke, J. v. (2005). *Multi-channel-based dissemination of e-learning systems: a reference process model for the design of applications, technologies, methods, and organisations*, In: 4th International Symposium on Information and Communication Technologies (WISICT 2005), Cape Town (Afrique du Sud), p. 8 - 13.

Bruillard, E. (1999). *Informatique et education : Quels liens entre connaissances et technologie ?* Comment penser la communication des connaissances ? Du CD-Rom à Internet, edt. Harmattan, p. 195-207.

Bruillard, E. et M. Vivet (1994). *Concevoir des EIAO pour des situations scolaires : approche méthodologique*, In: Recherches en Didactique des Mathématiques, vol. 14, n°1/2, p. 273-302.

Buschmann, F., R. Meunier, H. Rohnert, P. Sommerlad et M. Stal (1996). *Pattern-Oriented Software Architecture, A System of Patterns*, edt. John Wiley & Sons, ISBN 4-7195-8697.

Charlier, B., A. Daele et N. Deschryver (2002). *Vers une approche intégrée des technologies de l'information et de la communication dans les pratiques d'enseignement*, In: Revue des sciences de l'éducation, vol. XXVIII, no 2, p. 345-365.

Chikofsky, E. J. et J. H. Cross (1990). *Reverse Engineering and Design Recovery: A Taxonomy*, In: Software IEEE, p. 13-17.

Choquet, C., A. Merceron, F. Pozzi et F. Verdejo (2005). *Design patterns for recording and analysing usage of learning systems*, Jointly Excuted Integrating Research Projects (JEIRP), Kaleidoscope.

Cockburn, A. (2001). *Agile Software Development*, edt. Addison Wesley, ISBN 201699699.

Corbière, A. (2005). *Les technologies éducatives standards : un nouvel éclairage sur l'approche qualité d'un système de formation à distance*, In: Conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH 2005), Montpellier, p. 273-284.

Corbière, A. et C. Choquet (2004a). *Designer integration in training cycles: IEEE LTSA model adaptation*, In: International Conference on Computer Aided Learning in Engineering Education (CALIE 2004), Grenoble, p. 51-62.

Corbière, A. et C. Choquet (2004b). *Re-engineering method for multimedia system in education*, In: IEEE IS Multimedia Software Engineering (MSE 2004), Miami, Floride (USA), p. 80-87.

Corbière, A. et C. Choquet (2005). *ODP-RM : Un cadre de réingénierie des systèmes de formation*, In: Revue Sciences et Technologies de l´Information et de la Communication pour l´Éducation et la Formation (STICEF), vol. 12, n°HS.

De Rosnay, J. (1975). *Le macroscope, vers une vision globale*, edt. Seuil, ISBN 2020028182.

Debauche, B. et P. Mégard (2004). *Business Process Management*, edt. Hermes.

Deckmyn, O. et P.-J. Grizel (2003). *Zope, 2^{ième} édition*, edt. Eyrolles, ISBN 2212112270.

Demeyer, S., S. Ducasse et O. Nierstrasz (2003). *Object-Oriented Reengineering Patterns*, edt. Morgan Kaufmann, ISBN 1558606394.

Derycke, A. (2002). *Sept questions sur le E-learning: vers une problématique nouvelle pour la recherche ?* Les technologies en éducation : perspectives de recherche et questions vives. Paris:, edt. INRP : IUFM Basse-Normandie, p. 27-39.

Deursen, A. v., P. Klint et J. Visser (2000). *Domain-specific languages: an annotated bibliography*, ACM SIGPLAN Notices, p. 26-36.

Dewanto, B. L. (2002). *ObjectWeb and OpenUSS: a win-win situation ?*, In: Conference objectweb 2002, INRIA Rocquencourt, Le Chesnay (France).

D'Souza, D. F. et A. C. Wills (1999). *Objects, components, frameworks with UML: the Catalysis Approach*, edt. Addison Wesley, ISBN 0201310120.

Duqesnoy, L. (2002). *Aide à la conception, évaluation et démarche qualité pour le déploiement de formations multimédias en milieu industriel*, Thèse de doctorat, spécialité informatique, Institut National des Sciences Appliquées de Lyon, 228 p.

Durand, D. (1979). *La systèmique*, edt. que sais-je ?, Puf, ISBN 2130523455.

El-Kechai, H. et C. Choquet (2006). *Analyse d'une activité de conception collective par les objets intermédiaires,*, In: Colloque organisé par l'INRP et l'ERTé e-Praxis dans le cadre de la 8ème Biennale de l'Education, Lyon, p. 25-30.

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

Engels, G. et S. Sauer (2002). *Object-oriented modeling of multimedia applications*, In: S.K. Chang (Editor), Handbook of SEKE, p. 21-53.

Faerber, R. (2003). *Groupements, processus pédagogiques et quelques contraintes liés à un environnement virtuel d'apprentissage*, In: Conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH 2003), Strasbourg, p. 199-210.

Farance, F. (2003). *A Roadmap of ICT Standardization and Interoperability For Learning, Education, and Training*, In: Symposium, atelier sur l'élargissement du domaine de normalisation, Versailles.

Farance, F. et J. Tonkel (1998). *LTSA Specification Learning Technology Systems Architecture (LTSA)*, edt. Edutool Division.

Ferraris, C., A. Lejeune, L. Vignollet et J.-P. David (2005). *Modélisation de scénarios pédagogiques collaboratifs*, In: Conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH 2005), Montpellier, p. 285-296.

Fowler, M., K. Beck, J. Brant, W. Opdyke et D. Roberts (1999). *Refactoring: improving the design of existing code*, edt. Addison Wesley, ISBN 201485672.

Friesen, N. (2005). *Interoperability and Learning Objects: An Overview of E-Learning Standardization*, In: Interdisciplinary Journal of Knowledge and Learning Objects, vol. 1, p. 23-31.

Gagné, R. (1992). *Principles of instructional design, fourth edition*, edt. Wadsworth Publishing;, ISBN 030347572.

Gamma, E., R. Helm, R. Johnson et J. Vlissides (1999). *Design Patterns : Catalogue de modèles de conception réutilisables*, edt. Editions Vuibert, ISBN 2711786447.

Garlan, D. et M. Shaw (1994). *An introduction to Software Architecture*, In: Software Engineering and Knowledge Engineering, Singapore, p. 1-39.

Grob, H. L., F. Bensberg et B. L. Dewanto (2004). *Developing, Deploying, Using and Evaluating an Open Source Learning Management System*, In: Journal of Computing and Information Technology, vol. 12, n°2, p. 127-134.

Grob, H. L., F. Bensberg et B. L. Dewanto (2005). *Model Driven Architecture (MDA): Integration and Model Reuse for Open Source eLearning Platforms*, In: Eleed Journal (E-learning and Education), vol. 1, ISSN 18607470.

Hermans, H., R. Koper, A. Loeffen, J. Manderveld et E. Rusman (2000). *Reference Manual for Edubox-EML/XML binding 1.0*, 243 p.

Hernández-Leo, D., J. I. Asensio-Pérez et Y. Dimitriadis (2004). *IMS Learning Design Support for the Formalization of Collaborative Learning Patterns*, In: IEEE International Conference on Advanced Learning Technologies (ICALT 2004), Joensuu (Finlande), p. 350-354.

Hernández-Leo, D., E. D. Villasclaras-Fernández, J. I. Asensio-Pérez, Y. Dimitriadis, I. M. Jorrín-Abellán, I. Ruiz-Requies et B. Rubia-Avi (2006). *COLLAGE: A collaborative Learning Design editor based on patterns*, In: Educational Technology & Society, vol. 9, n°1, p. 58-76.

Highsmith, J. (2000). *Retiring lifecycle dinosaurs*, Software Testing & Quality Engineering Magazine, vol. 2, p. 22-28.

Highsmith, J. (2002). *Agile Software Development Ecosystems*, edt. Addison Wesley, ISBN 201760436.

Houssaye, J. (2000). *Théorie et pratiques de l'éducation scolaire 1 : Le triangle pédagogique, 3ième édition*, edt. Peter Lang, ISBN 3906754952.

Hummel, H., J. Manderveld, C. Tattersall et R. Koper (2004). *Educational modelling language and learning design: new opportunities for instructional reusability and personalised learning*, In: International Journal of Learning Technology, vol. 1, n°1, p. 111-126.

IEEE/ADS-IS (2000). *1471-2000 IEEE Recommended Practice for Architectural Description for Software-Intensive Systems*, edt. IEEE Computer Society, 30 p.

IEEE/LOM (2002). *1484.12.1-2002 IEEE Standard for Learning Object Metadata*, edt. IEEE Computer Society, 44 p.

IEEE/LTSA (2003). *1484.1-2003 IEEE Standard for Learning Technology-Learning Technology Systems Architecture (LTSA)*, edt. IEEE Computer Society, 103 p.

Iksal, S., V. Barré, C. Choquet et A. Corbière (2004). *Comparing Prescibed and Observed for the Re-Engineering of E-Learning Systems*, In: EEE IS Multimedia Software Engineering (MSE 2004), Miami, Florida (USA), p. 106-109.

Iksal, S. et C. Choquet (2005). *Usage Analysis Driven by Models in a Pedagogical Context*, In: Workshop International Conference on Artificial Intelligence in Education (AIED 2005) : Usage Analysis in learning systems, Amsterdam (Pays-Bas), p. 49-56.

ISO/IEC-10746-1 (1998). *Open Distributed Processing Reference Model, Part 1: Overview*, edt. ISO/IEC JTC1 SC7, 84 p.

ISO/IEC-10746-2 (1996). *Open Distributed Processing Reference Model, Part 2: Foundations*, edt. ISO/IEC JTC1 SC7, 28 p.

ISO/IEC-10746-3 (1996). *Open Distributed Processing Reference Model, Part 3: Architecture*, edt. ISO/IEC JTC1 SC7, 68 p.

ISO/IEC-15414 (2002). *Open Distributed Processing Reference Model, Entreprise language*, edt. ISO/IEC JTC1 SC7, 25 p.

ISO/IEC-19796 (2004). *Information technology - Quality management, assurance, and metrics - Part 1 : General approach, final committee draft*, N0045, edt. ISO/IEC JTC1 SC36 WG5, 139 p.

Jackl, A., A. Panar et B. Towle (2004). *IMS Shareable State Persistence Information Model, version 1.0 Public draft*, edt. IMS Global Learning Consortium, 31 p.

Jacobson, I., G. Booch et J. Rumbaugh (2000). *Le processus unifié de développement logiciel*, edt. Eyrolles, ISBN 2212091427.

Johnson, R. E. (1992). *Documenting Frameworks using Patterns*, In: Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA 1992), Vancouver (Canada), p. 63-70.

Kilof, H. (2004). *Semantic interoperability: using RM-ODP to bridge communication gaps between stakeholders*, In: Workshop International EDOC Conference (EDOC 2004) : ODP for Enterprise Computing, Monterey, California (USA), p. 4-14.

Koper, R. (2001). *Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML*, edt. Educational Technology Expertise Centre Open University of the Netherlands, 40 p.

Koper, R. (2004). *Use of the Semantic Web to Solve Some Basic Problems in Education*, In: Journal of Interactive Media in Education, vol. 6, Special Issue on the Educational Semantic Web.

Koper, R., B. Olivier et T. Anderson (2003a). *IMS Learning Design Best Practive and Implementation Guide, version 1.0 Final Specification*, edt. IMS Global Learning Consortium, 138 p.

Koper, R., B. Olivier et T. Anderson (2003b). *IMS Learning Design Information Model, version 1.0 Final Specification*, edt. IMS Global Learning Consortium, 87 p.

Krasner, G. E. (1988). *A cookbook for using the model-view controller user interface paradigm in Smalltalk-80.*, In: Journal of Object-Oriented Programming, vol. 1, n°6, p. 26-49.

Kruchten, P. (1995). *The 4+1 view model of architectur*, In: IEEE Software, vol. 12, n°6, p. 42-50.

Le Boterf, G. (2001). *Construire les compétences individuelles et collectives, 2ième édition*, edt. Editions d'organisations, Paris, ISBN 2708126458.

Le Moigne, J.-L. (1990). *La modélisation des systèmes complexes*, edt. Dunod, ISBN 210004382.

Lefébure, R. et G. Venturi (2001). *Data mining: Gestion de la relation client et Personnalisation de sites web*, edt. Eyrolles, ISBN 2212092032.

Leinonen, T., O. Virtanen, K. Hakkarainen et G. Kligyte (2002). *Collaborative Discovering of Key Ideas in Knowledge Building*, In: Computer Supported Collaborative Learning Conference (CSCL 2002), Boulder, Colorado (USA), p. 529-530.

Linard, M. (1996). *Des machines et des hommes*, edt. L'Harmattan, ISBN 2738442919.

Linard, M. (2001). *Concevoir des environnements pour apprendre: l'activité humaine, cadre organisateur de l'interactivité technique*. Sciences et techniques éducatives, edt. Hermes, vol. 8, p. 211-238.

Linard, M. (2002). *Conception de dispositifs et changement de paradigme en formation*. Les TIC au service des nouveaux dispositifs de formation, edt. Education permante, vol. 152, p. 143-155.

Linard, M. (2003). *Autoformation, éthique et technologies : enjeux et paradoxes de l'autonomie*, edt. Hermes, p. 241-263.

Lindner, R. (2003). *German Comments on IEEE P1484.1/D11, 2002-11-28 Draft Standard for Learning Technology -- Learning Technology Systems Architecture (LTSA)*, N0015, edt. ISO/IEC JTC1 SC36 WG5, 4 p.

Loy, M., R. Eckstein, D. Wood, J. Elliott et B. Cole (2002). *Java Swing*, edt. O'Reilly, ISBN 0596004087.

Merrill, D. (1997). *Instructional Transaction Theory: An Instructional Design Model Based on Knowledges Objects*. Instructional Design: International Perspectives, edt. Lawrence Erlbaum Associates, vol. 1, p. 381-394.

Monson-Haefel, R. (2002). *Enterprise JavaBeans, 3e édition*, edt. O'Reilly, ISBN 2841772187.

Moreau, C. et M. Majada (2002). *Nouveaux dispositifs de formation : de la pratique à l'ingénierie et de l'ingénierie à la pratique*. Les TIC au services des nouveaux dispositifs de formation, p. 133-142.

Morin, E. (1977). *La méthode : 1. La Nature de la Nature*, edt. Seuil, ISBN 2020057719.

Nagase, Y., D. Hashimoto et M. Sato (2004). *Applying Model-Driven Development to Business Systems using RM-ODP and EDOC*, In: Workshop International EDOC Conference (EDOC 2004) : ODP for Enterprise Computing, Monterey, California (USA), p. 36-42.

Nakakoji, K., Y. Yamamoto, Y. Nishinaka, K. Kishida et Y. Ye (2002). *Evolution Patterns of Open-Source Software Systems and Communities*, In: International Workshop on Principles of Software Evolution (IWPSE 2002), Orlando, Florida (USA), ACM Press, p. 76-85.

Nodenot, T. (2005). *Contribution à l'ingénierie dirigée par les modèles en EIAH : le cas des situations-problèmes coopératives*, Habilitation à Dirigée des Recherches en Informatique, Université de Pau et des Pays de l'Adour, 182 p.

Norton, M. et A. Panar (2003). *IMS Simple Sequencing Information and Behavior Model, version 1.0*, edt. IMS Global Learning Consortium, 122 p.

OMG/ECA (2004). *Enterprise Collaboration Architecture (ECA) Specification, version 1.0*, edt. Object Management Group, 202 p.

OMG/MDA (2001a). *Model Driven Architecture : A Technical Perspective*, edt. Object Management Group, 27 p.

OMG/MDA (2001b). *Model-Driven Architecture and Integration : Opportunities and Challenges*, edt. Object Management Group, 32 p.

OMG/MDA (2003). *MDA Guide, version 1.0.1*, edt. Object Management Group, 62 p.

OMG/QVT (2003). *Revised submission for MOF 2.0 Query / Views / Transformations RFP, version 1.1*, edt. Object Management Group, 109 p.

OMG/UML (2005). *Unified Modeling Language: Infrastructure, version 2.0*, edt. Object Management Group, 218 p.

Oubahssi, L., M. Grandbastien et G. Claës (2004). *Ré-ingénierie d'une plate-forme fondée sur la modélisation d'un processus global de FOAD*, In: Colloque International sur les Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie (TICE 2004), Compiègne, p. 32-38.

Oussalah, M. C., N. Sadou et D. Tamzalit (2005). *SAEV, a Model to ensure a Static and Dynamic Software-Architecture Evolution*, In: WSEAS transactions on systems, vol. 4, n°5, p. 671-681.

Papert, S. (1987). *Computer Criticism vs. Technocentric Thinking*, In: Educational Researcher, vol. 16, n°1, p. 20-33.

Pawlowski, J. M. (2002a). *Project Team Quality Assurance And Guidlines*, N0003, edt. ISO/IEC JTC1 SC36 WG5, 63 p.

Pawlowski, J. M. (2002b). *Report on Quality Assurance Standards / Proposal for Future Work*, Project Team Quality Assurance and Guidelines, edt. CEN / ISSS Workshop on Learning Technologies, 29 p.

Pawlowski, J. M. (2005). *Information technology - Quality management, assurance, and metrics : draft proposal for future work in WG5 on quality*, N0055, edt. ISO/IEC JTC1 SC36 WG5, 8 p.

Pernin, J.-P. et A. Lejeune (2004). *Dispositifs d'Apprentissage Instrumentés par les Technologies : vers une ingénierie centrée sur les scénarios*, In: Colloque International sur les Technologie de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie (TICE 2004), Compiègne, p. 407-414.

Rabardel, P. (1995). *Les hommes et les technologies. Approche cognitive des instruments contemporains*, edt. Armand Colin, ISBN 220021569X.

Randriamalaka, N. et S. Iksal (2006). *Patterns Approach in the Re-Engineering Process of Learning Scenario*, In: IEEE International Conference on Advanced Learning Technologies (ICALT 2006), Kerkrade (Pays-Bas), (à paraître).

Rastetter, Y. (2002). *Le logiciel libre dans les entreprises*, edt. Hermes, ISBN 2746204428.

Rawlings, A., P. v. Rosmalen, R. Koper, M. Rodríguez-Artacho et P. Lefrere (2002). *Survey of Educational Modelling Languages (EMLs)*, edt. CEN/ISSS WS/LT, 78 p.

Raymond, E. S. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, edt. O'Reilly, ISBN 596001088.

Roques, P. et F. Vallée (2004). *UML 2 en action : De l'analyse des besoins à la conception J2EE*, edt. Eyrolles, ISBN 2212114621.

Roschelle, J., C. DiGiano, M. Koutlis, A. Repenning, J. Phillips, N. Jackiw et D. Suthers (1999). *Developing Educational Software Components*, In: IEEE Computer, vol. 32, n°9, p. 50-58.

Roschelle, J., J. Kaput, W. Stroup et T. M.Kahn (1999). *Scaleable Integration of Educational Software: Exploring the Promise of Component Architectures*, In: Journal of Interactive Media in Education, vol. 6.

Sauer, S. et G. Engels (2001). *UML-based Behavior Specification of Interactive Multimedia Applications*, In: IEEE Symposia on Human-Centric Computing Languages and Environments, Stresa, Italie, p. 248-255.

Smythe, C. (2003). *IMS Abstract Framework: White Paper, version 1.0*, edt. IMS Global Learning Consortium, 70 p.

Smythe, C., G. Collier, C. Etesse et W. Veres (2002). *IMS Enterprise Information Model, Version 1.1 Final Specification*, edt. IMS Global Learning Consortium, 43 p.

Smythe, C. et A. Cooper (2003). *IMS Content Packaging Information Model, version 1.1.3*, edt. IMS Global Learning Consortium, 17 p.

Szyperski, C. (2003). *Component Technology - What, Where, and How?*, In: 25th International Conference on Software Engineering, Portland, Oregon (USA), p. 684-674.

Tattersall, C., J. Manderveld, H. Hummel, P. Sloep, R. Koper et F. d. Vries (2003). *IMS Learning Design Frequently Asked Questions*, 10 p.

Tchounikine, P., M. Baker, N. Balacheff, M. Baron, A. Derycke, D. Guin, J.-F. Nicaud et P. Rabardel (2004). *Platon-1: quelques dimensions pour l'analyse des travaux de recherche en conception d'EIAH*, In: STIC-CNRS.

Varela, F. J. (1996). *Invitation aux sciences cognitives*, edt. Seuil, ISBN 2020287439.

Verbert, K. et E. Duval (2004). *Towards a Global Component Architecture for Learning Objects: A Comparative Analysis of Learning Object Content Models*, In: World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2004), Lugano (Suisse), p. 202-208.

Wegmann, A. et A. Naumenko (2001). *Conceptual Modeling of Complex Systems Using an RM-ODP Based Ontology*, In: International EDOC Conference (EDOC 2001), Seattle, Washington (USA), p. 200-211.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems: Computational Cognitive Approaches to the Communication of Knowledge*, edt. Morgan Kaufmann, ISBN 0934613265.

Wenger, E. (1998). *Communities of Practice: Learning, Meaning, and Identity*. New York, edt. Cambridge University Press, ISBN 0521663636.

Wenger, E., R. McDermott et W. M. Snyder (2002). *Cultivating communities of pratice*, edt. Harvard business school press, ISBN 1578513308.

Yourdon, E. (1989). *Modern Structured Analysis*, edt. Yourdon Press, Prentice-Hall, Englewood Cliffs, ISBN 0135986249.

LISTE DES ABREVIATIONS

Ce glossaire définit les abréviations utilisées.

2TUP	Two Tracks Unified Process
ADL	Advanced Distributed Learning
CCA	Component Collaboration Architecture
CVS	Concurrent Versions System
DSL	Domain Specific Language
ECA	Entreprise Collaboration Architecture
EDOC	Entreprise Distributed Object Computing
EIAH	Environnements Informatique pour l'Apprentissage Humain
EJB	Entreprise JavaBeans
EJOSA	Enterprise Java Open Source Architecture
EML	Educational Modeling Language
FLE	Futur Learning Environment
FSL	FreeStyle Learning
HTML	Hypertext Markup Language
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMS	Instructional Management Systems
ISO	International Organization for Standardization
J2EE	Java 2 Entreprise Edition
JOnAS	Java Open Application Server
LD	Learning Design
LOM	Learning Object Metadata
LTSA	Learning Technology Systems Architecture
LTSC	Learning Technology Standards Committee
MDA™	Model-Driven Architecture

ODP-RM	Object Distributed Processsing - Reference Model
OMG	Object Management Group
OPENUSS	Open University Support System
QVT	Query Views Tranformations
SCORM	Sharable Content Object Reference Model
SMIL	Synchronized Multimedia Integration Language
UML	Unified Modeling Langage
UP	Unified Process
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

GLOSSAIRE ET LEXIQUE

Nous donnons les traductions des termes du cadre ODP-RM utilisés dans ce mémoire et définis dans les quatre documents (ISO/IEC-10746-2 1996; ISO/IEC-10746-3 1996; ISO/IEC-10746-1 1998; ISO/IEC-15414 2002).

Acteur (Actor [(ISO/IEC-15414 2002), sous-partie 6.3.1]) : *objet métier* qui participe à l'*action*, pages 41, 109, 120, 121 et 134, sous-partie 3.3.3 (page 84).

Action (Action [(ISO/IEC-10746-2 1996), partie 8.3]) : quelque chose qui se passe. Toute action ayant un intérêt pour les besoins de la modélisation, pages 51, 75, 84, 86, 105 et 119.

Activité (Activity [(ISO/IEC-10746-2 1996), partie 8.5]) : graphe d'*actions* acyclique ayant une seule racine, pages 84, 132 et 134.

Artefact (Artefact [(ISO/IEC-15414 2002), sous-partie 6.3.2]) : *objet métier* qui est référencé dans l'*action*, pages 41, 85, 94, 113 et 137.

Canal (Channel [(ISO/IEC-10746-3 1996), sous-partie 8.1.8]) : composition d'*objets ingénierie* spécifiques assurant la liaison entre un ensemble d'*interfaces* d'*objets ingénierie* de base pour supporter leurs *interactions*, page 41.

Capsule (Capsule [(ISO/IEC-10746-3 1996), sous-partie 8.1.4]) : ensemble d'*objets ingénierie* regroupés dans une unité logicielle de traitement et de stockage, page 41.

Communauté (Community [(ISO/IEC-15414 2002), sous-partie 7.3.1]) : composition d'*objets métier*. Ces derniers sont soumis aux règles définies par le *contrat* de la *communauté* qui (1) indique l'*objectif* sur lequel l'existence de la *communauté* est fondée ; (2) régit la structure, le *comportement* et les *stratégies* de la communauté ; (3) contraint le *comportement* des membres de la *communauté* et (4) indique les règles d'attribution d'*objets métier* à des *rôles*, pages 41, 85, 85, 93 et 134, paragraphe 2.2.4.3, page 71, sous-partie 2.3.2, page 74, paragraphe 4.3.4.2, page 132.

Compatibilité comportementale (Behavioural compatibility [(ISO/IEC-10746-2 1996), partie 9.4]) : propriété de deux *objets* quand l'un d'eux peut remplacer l'autre dans un *environnement* donné sans qu'aucune différence soit constatée. La compatibilité comportementale est réflexive, mais pas forcément symétrique ou transitive, page 121, paragraphe 4.3.3.4, page 127.

Comportement (Behavior [(ISO/IEC-10746-2 1996), partie 8.6]) : ensemble d'*actions* ainsi que les contraintes portant sur les circonstances dans lesquels ces *actions* peuvent se produire. Les contraintes exprimables dépendent du *langage* de spécification utilisé, pages 51, 66, 74, 78, 81, 85, 111, 114 et 122, sous-partie 2.2.4.3, page 71, sous-partie 4.3.1, page 105, paragraphe 4.3.2.3, page 116, sous-partie 4.3.4, page 130.

Composition (Composition [(ISO/IEC-10746-2 1996), partie 9.1]) : regroupement de deux ou plusieurs *objets* ou *comportements* qui peut être référencé comme un nouvel *objet* à un niveau d'abstraction donné ou un nouveau *comportement*, page 81 et 132.

Contrat (Contract [(ISO/IEC-10746-2 1996), sous-partie 11.2.1]) : accord qui régit la coopération entre un certain nombre d'objets. Il spécifie les obligations, les permissions et les interdictions d'une partie du *comportement* de cet ensemble d'*objets*, page 67.

Décomposition (Decomposition [(ISO/IEC-10746-2 1996), partie 9.3]) : spécification des *objets* ou des *comportements* constituant un *objet* ou un *comportement* donné, pages 41, 81, 91, 96 et 121, sous-partie 4.3.3.2, page 121.

Editeur de liens (Binder [(ISO/IEC-10746-3 1996), sous-partie 8.1.10]) : *objet ingénierie* pour un *canal* donné qui maintient la liaison distribuée entre deux *objets ingénierie* de base, page 41.

Environnement (Environment [(ISO/IEC-10746-2 1996), partie 8.2]) : pour un *objet*, partie du modèle qui ne fait par partie de cet *objet*. Dans de nombreux *langages* de spécification, on peut estimer que l'*environnement* comprend au moins un

objet suceptible de participer sans contrainte à toutes les *intéractions* possibles, représentant le travail d'observation, pages 67, 74, 84 et 105.

Etape (Step [(ISO/IEC-15414 2002), sous-partie 6.3.6]) : abstraction d'une *action* utilisée dans un *processus*, pages 41, 77, 85, 88, 106 et 119, partie 4.3.2, page 113.

Etat (State [(ISO/IEC-10746-2 1996), partie 8.7]) : condition d'un *objet* à un instant donné qui détermine l'ensemble des séquences d'*actions* auxquelles l'*objet* prend part, pages 51, 66, 81, 96, 105, 108, 113, 114, 119, 135 et 139, sous-partie 3.4.2.6, page 92, sous-partie 4.3.2.2, page 115, sous-partie 4.3.2.3, page 118.

Flux (Flow [(ISO/IEC-10746-3 1996), sous-partie 7.1.5]) : abstraction d'un ensemble ordonné d'une ou plusieurs *intéractions* supportant l'échange d'information d'un *objet* producteur vers un *objet* consommateur, pages 41, 47, 91 et 138, sous-partie 2.3.2, page 74.

Fonction (Function [(ISO/IEC-10746-1 1998), partie 8.8 & (ISO/IEC-10746-3 1996), chapitre 11]) : dans le cadre ODP-RM, des fonctions sont définies comme fondamentales ou largement applicables dans la réalisation d'un système. Le modèle de référence du cadre ODP-RM distingue quatre grandes catégories de fonctions : fonctions de gestion, fonctions de coordination, fonctions de dépôt et fonctions de sécurité, page 65.

Gabarit (Gabarit [(ISO/IEC-10746-2 1996), partie 9.11 & (ISO/IEC-10746-1 1998), sous-partie 7.2.4) : spécification des caractéristiques communes d'un ensemble d'*objets* à un niveau de détail suffisant pour que ces caractéristiques soient *instanciées*, pages 121 et 130, sous-partie 4.3.3.3, page 125.

Grappe (Cluster [(ISO/IEC-10746-3 1996), sous-partie 8.1.2]) : ensemble d'*objets ingénierie de base* regroupés dans une unité bénéficiant des mécanismes de désactivation, de reprise après arrêt, de réactivation, de récupération et de migration, pages 41, 95 et 99.

Implantation (Implementation [(ISO/IEC-10746-3 1996), sous-partie 9.1.2]) : démarche d'instanciation dont la validité peut être testée, page 41, 72, 120 et 126.

Informations supplémentaires sur l'exécution des mises à l'essai (IXIT : Implementation eXtra Information Testing [(ISO/IEC-10746-3 1996), sous-partie 9.1.3]) : informations complémentaires pour observer le *comportement* lors des tests de conformité, page 41.

Instance (Instance [(ISO/IEC-10746-2 1996), partie 9.18]) : entité satisfaisant à un *type* particulier.

Interaction (Interaction [(ISO/IEC-10746-2 1996), partie 8.3]) : ensemble d'*actions* associées à un *objet* et se produisant avec la participation de l'*environnement* de l'*objet*, pages 41, 51, 76, 81, 84 et 105.

Interface (Interface [(ISO/IEC-10746-2 1996), partie 8.4]) : abstraction du *comportement* d'un *objet* qui regroupe dans un sous-ensemble les *interactions* de cet *objet* et un ensemble de contraintes portant sur les circonstances dans lesquelles se produisent ses *interactions*, pages 41, 61, 67, 70, 93, 114 et 141, sous-partie 4.3.1, page 105, paragraphe 4.3.2.2, page 115.

Langage (Language [(ISO/IEC-10746-3 1996), paragraphe 4.2.1.1]) : définition de concepts, de structures et de règles permettant de développer, de représenter et d'analyser la spécification d'un système à partir d'un *point de vue* du cadre ODP-RM, pages 51, 71, 74, 84, 85, 115 et 120.

Nœud (Node [(ISO/IEC-10746-3 1996), sous-partie 8.1.7]) : composition d'*objets ingénierie* formant une unité localisée dans un espace donné et comprend un ensemble de fonctions de traitement, de stockage et de communication, page 41.

Noyau (Nucleus [(ISO/IEC-10746-3 1996), sous-partie 8.1.6]) : *objet ingénierie* qui coordonne les fonctions de traitement, de stockage et de communication des *objets ingénierie* repgroupés dans un *nœud*, page 41.

Objectif (Objective [(ISO/IEC-15414 2002), sous-partie 6.2.1]) : définit l'avantage pratique ou l'effet attendu, exprimé comme des préférences sur les *états* futurs d'un *objet métier*, pages 41, 75, 85 et 132.

Objet (Object [(ISO/IEC-10746-2 1996), partie 8.1]) : modèle d'une entité. Un objet est caractérisé par son *comportement* et, de manière duale, par son *état*. Un *objet* est distinct de tous autres *objets*. Un *objet* est encapsulé, c.-à-d. n'importe quel changement de son *état* peut seulement se produire en raison d'une *action* interne ou en raison d'une *interaction* avec son *environnement*.

Opération (Operation [(ISO/IEC-10746-3 1996), sous-partie 7.1.2]) : *interaction* entre un *objet* client et un *objet* serveur qui correspond à une interrogation ou à une invocation, pages 41 et 91.

Point de vue (Viewpoint [(ISO/IEC-10746-3 1996), sous-partie 4.1.2 & (ISO/IEC-10746-1 1998), sous-partie 8.1.1]) : subdivision de la spécification d'un système complet. Le cadre ODP-RM définit les *points de vue* nécessaires et suffisants pour répondre aux besoins des standards. Les *points de vue* ne sont pas complètement indépendants les uns des autres. On trouvre dans chacun d'eux des éléments essentiels qui sont apparentés à des éléments appartenant à d'autres *points de vue*. Ils restent cependant suffisament indépendants pour qu'en puisse être simplifié le raisonnement qui conduit à la spécification complète.

Point de vue computationnel (Computational viewpoint [(ISO/IEC-10746-3 1996), paragraphe 4.1.1.3]) : *point de vue* sur la *décomposition* fonctionnelle d'un système et les *interactions* entre les *interfaces* des différents *objets*, à l'aide des concepts *signal*, *opération* et *flux*, pages 41, 51, 58, 83, 96, 109, 116, 118, 122 et 124, paragraphe 3.4.2.5, page 91.

Point de vue informationnel (Information viewpoint [(ISO/IEC-10746-3 1996), paragraphe 4.1.1.2]) : *point de vue* sur les informations traitées par les différentes *ressources* systèmes, à l'aide des concepts *schéma dynamique*, *schéma statique* et *schéma invariant*, pages 41, 57, 66, 83, 82, 98, 113, 112, 115, 118, 126, 127, 128 et 128, paragraphe 3.4.2.3, page 90.

Point de vue ingénierie (Engineering viewpoint [(ISO/IEC-10746-3 1996), paragraphe 4.1.1.4]) : *point de vue* décrivant les moyens mis en œuvre pour que les *objets* d'un système réparti interagissent, à l'aide des concepts *grappe*, *capsule*, *noyau*, *nœud*, *canal*, *souche* et *éditeur de liens*, pages 41, 58, 83, 95 , 96, 98, 99, 124, 125 et 138, paragraphe 3.4.2.4, page 90.

Point de vue métier (Entreprise viewpoint [(ISO/IEC-10746-3 1996), paragraphe 4.1.1.1]) : *point de vue* se focalisant sur les *objectifs*, le domaine d'application et les *stratégies* de ce système, à l'aide des concepts *rôle*, *communauté*, *processus*, *étape*, *objectif*, *artefact*, *acteur* et *ressource*.

Point de vue technologique (Technology viewpoint [(ISO/IEC-10746-3 1996), paragraphe 4.1.1.5]) : *point de vue* définissant les technologies logicielles et matérielles utilisées, à l'aide des concepts *standard implantable*, *implantation* et *informations supplémentaires sur l'exécution des mises à l'essai*, pages 41, 53, 65, 66 et 93, paragraphe 3.4.2.6, page 92.

Processus (Process [(ISO/IEC-15414 2002), sous-partie 6.3.5]) : série d'*étapes* prescrites et conduisant à un *objectif*.

Qualité de service (Quality of Service [(ISO/IEC-10746-2 1996), sous-partie 11.2.2]) : exigence qualité sur un *comportement* collectif d'un ou de plusieurs *objets*, page 67.

Responsabilité (Accountability [(ISO/IEC-15414 2002), sous-partie 6.5]) : description de l'*action* d'un *objet métier* précisant ses intentions utilisant les concepts : de partie, d'engagement, de déclaration, de délégation, d'évaluation, de prescription, d'agent, de mandant et de partie contractualisée, pages 72 et 112.

Ressource (Resource [(ISO/IEC-15414 2002), sous-partie 6.3.3]) : *objet métier* qui est essentiel pour un *comportement* donné exigeant l'attribution ou pouvant devenir indisponible. Dans le premier cas, l'attribution d'une *ressource* peut contraindre d'autres *comportements* pour lesquels cette *ressource* est essentielle. Dans le second cas, une *ressource* consommable peut devenir indisponible après un certain nombre d'utilisations ou après une certaine durée.

Rôle (role [(ISO/IEC-10746-2 1996), partie 9.14 & (ISO/IEC-15414 2002), sous-partie 6.3.4]) : identificateur d'un *comportement* qui peut se présenter sous la forme de paramètres dans un *gabarit* d'*objet* composite et qui est associé à l'un des *objets* composant l'*objet* composite. Dans le *point de vue métier*, le *rôle* d'une *communauté* indique le *comportement* qui intervient avec la participation d'*objets métier* qui ne sont pas membres de cette *communauté*, pages 41, 51, 67, 76, 84, 85, 86, 90, 92, 110, 133 et 141, sous-partie 3.4.3, page 94.

Schéma dynamique (Dynamic schema [(ISO/IEC-10746-3 1996), sous-partie 6.1.3]) : spécification des changements d'*états* qui sont autorisés pour un ou plusieurs *objets informationnel*. Ce schéma doit respecter les contraintes de tous les *schémas invariants*, pages 41, 90, 91, 93 et 127.

Schéma invariant (Invariant schema [(ISO/IEC-10746-3 1996), sous-partie 6.1.1]) : spécification des prédicats sur un ou plusieurs *objets informationnel* qui doivent être toujours vrais pour tout *comportement* valide du système. Ces prédicats définissent les contraintes des *états* possibles et des changements d'*états* possibles des *objets* auxquels ils s'appliquent, pages 41, 90 et 127.

Schéma statique (Static schema [(ISO/IEC-10746-3 1996), sous-partie 6.1.2]) : spécification des relations entre les *objets informationnel* valides à un instant donné. Ce schéma doit respecter les contraintes de tous les *schémas invariants*, pages 41, 90, 127, 128, 129 et 129.

Signal (Signal [(ISO/IEC-10746-3 1996), sous-partie 7.1.1] : *action* atomique partagée résultant d'un échange entre un *objet* initiant la communication et un *objet* y répondant, pages 41, 91, 92, 93, 115, 116, 116, 117, 116 et 118.

Souche (Stub [(ISO/IEC-10746-3 1996), sous-partie 8.1.9]) : *objet ingénierie* appartenant à un *canal* qui interprète les *interactions* transportées dans ce *canal*, et qui effectue, suivant cette interprétation, les transformations ou les surpervisions nécessaires, page 41.

Standard implantable (Implementable standard [(ISO/IEC-10746-3 1996), sous-partie 9.1.1]) : *gabarit* d'un *objet technologique*, page 41.

Stratégie (Policy [(ISO/IEC-15414 2002), sous-partie 6.4.1]) : ensemble de règles relatives à un *objectif* particulier. Une règle peut être exprimée sous la forme d'une obligation, d'une autorisation, d'une permission ou d'une interdiction, pages 41, 61, 72, 73, 74, 113 et 131.

Trace (Trace [(ISO/IEC-10746-2 1996), partie 9.6]) : enregistrement de l'*interaction* d'un *objet*, de son *état* initial à un quelconque autre *état*, pages 58, 58, 66, 76, 97, 106, 108, 118, 128, 130, 130 et 137.

Transparence (ODP distribution transparencies [(ISO/IEC-10746-1 1998), partie 8.9 & , (ISO/IEC-10746-3 1996) partie 4.4]) : principe permettant de cacher à l'élaboration d'une application répartie les détails liés aux mécanismes complexes qui servent à résoudre les problèmes dus à la répartition (par exemple, migration, persistance, transaction , etc.), page 65.

Résumé :

Ce travail de thèse propose un cadre méthodologique pour l'ingénierie et la réingénierie de la formation à distance qui s'appuie sur les récents efforts de normalisation des technologies éducatives. Ce cadre s'adresse aux concepteurs, afin de les accompagner dans leur travail, et de favoriser l'acquisition de nouvelles connaissances et compétences, notamment d'ordre conceptuel, organisationnel et méthodologique. En amenant les concepteurs à spécifier de manière explicite leurs intentions de conception d'un système de formation dans des modèles, et en confrontant ces modèles de spécification avec les usages observés durant les sessions d'apprentissage, nous accompagnons la nécessaire démarche réflexive de l'équipe de conception sur ses méthodes, et favorisons le développement et la capitalisation des savoir-faire et des démarches pédagogiques au sein d'une communauté de concepteurs.

Nous avons proposé le cadre de référence du processus distribué ouvert de l'ISO (International Organization for Standardization) et de l'IEC (International Electrotechnical Commission) comme le méta-standard à adresser aux concepteurs de la communauté EIAH. La vision globale d'un système est explicitée par des mécanismes de transformations de différents points de vue.

Un tel cadre a pour intérêt d'adresser un ensemble de termes à la communauté de concepteurs soucieux de partager les pratiques. Ce vocabulaire permet d'expliciter les invariants et les principes généraux, structuraux et fonctionnels d'un système de formation. Le travail présenté dans cette thèse vise à définir différentes instances d'un tel méta-standard et à illustrer ainsi les différents apports de ce modèle à la communauté des concepteurs des EIAH. La présentation du travail se structure autour de trois analyses instrumentées par le cadre ODP-RM (Object Distributed Processsing - Reference Model), trois réflexions sur la conception d'un EIAH portant sur l'utilisation des technologies éducatives normées, sur la réingénierie dans les EIAH et sur le processus de développement dirigé par les modèles. Les illustrations présentées dans le mémoire montrent le potentiel du cadre choisi à formaliser ces différents constats et à créer un ensemble d'instances décrivant de nouveaux éléments méthodologiques à adresser à la communauté de concepteurs d'un système de formation.

Abstract :

This thesis proposes a methodological framework for the engineering and the reengineering of an e-learning system which uses Technology Enhanced Learning (TEL). This framework is addressed to designers, to assist them in their work, and to support the acquisition of new knowledge and expertise, in particular of a conceptual, organisational and methodological nature. By leading the designers to specify, and explicit their e-learning system design intentions in models, and by confronting these specification models with the uses observed during the training sessions, we support the necessary reflexive process by the designer community.This also supports the development and the capitalization of know-how and the pedagogical process within this community.

We proposed the open distributed process model reference of the ISO (International Organization for Standardization) and the IEC (International Electrotechnical Commission) as the meta standard to be recommended to the designers of computer assisted language learning (CALL) environments. The global vision of a system is clarified by transformation mechanisms of from different points of view.

The interest of this framework is to propose concepts to the designer community eager to share the practices. This vocabulary makes it possible to clarify the invariants and the general, structural and functional principles of an e-learning system. The work presented in this thesis aims to define various instances of such a meta standard and thus to illustrate the various contributions from this model for the computer assisted language learning (CALL) environment designer community. The work is strutured around three analyses instrumented by a ODP-RM (Object Distributed Processsing - Model Reference) on the design of a CALL environment. These three considerations are : (1) on the use of normalized educational technologies, (2) on the reengineering in the EIAH and (3) on the process of development directed by the models. The illustrations presented in the report show the potential of the framework chosen to formalize these various reports and to create a group of authorities describing new methodological elements to assist the designer community.

Mots clés : Object Distributed Processsing - Reference Model (ODP-RM), réingénieirie, ingénierie, Environnements Informatique pour l'Apprentissage Humain (EIAH)