

ODP-RM : Un cadre de réingénierie des systèmes de formation

Alain Corbière, Christophe Choquet [LIUM, Laval]

■ **RÉSUMÉ** : Les spécifications candidates au statut de standards, proposées récemment dans le domaine des technologies éducatives, interrogent les communautés de conception d'EIAH sur leurs mises en pratique. Les travaux récents en ingénierie des modèles montrent qu'il est nécessaire pour la communauté EIAH de se définir un cadre de travail qui formalise l'utilisation de ces spécifications tout en assurant la qualité des productions. Cet article est une discussion sur l'opportunité pour la communauté EIAH d'adopter le modèle ODP-RM (Open Distributed Processing - Reference Model) proposé par l'ISO (International Organization for Standardization) pour expliciter l'organisation de conception et l'ingénierie d'un EIAH. Un tel modèle met à disposition un ensemble de concepts liés aux tâches de spécification, d'observation et de modélisation qu'il nous semble pertinent d'utiliser pour formaliser la négociation et la communication dans une telle communauté. Nous démontrons dans un premier temps la pertinence d'instancier un tel cadre à la fois pour intégrer les différents projets de normes sur les technologies éducatives et pour prendre en compte les évolutions de la réingénierie logicielle; puis dans un deuxième temps nous proposons deux premières instances spécifiques à la rétroconception et à la réingénierie d'un système de formation.

■ **MOTS CLÉS** : Réingénierie, ingénierie, EIAH, processus métier.

■ **ABSTRACT** : Specifications recently proposed as standards in the domain of Technology Enhanced Learning (TEL), question the designers of TEL systems on how to put them into practice. Recent studies in Model Driven Engineering have highlighted the need for a framework which could formalise the use of these specifications as well as enhance the quality of the developments for the TEL community. This paper deals with the opportunity for the TEL community to adopt the ISO/ODP-RM model (International Organization for Standardization/Open Distributed Processing - Reference Model), to express and formalise the design organisation and the engineering process of a TEL system. Such a model provides a set of concepts allowing the description of specification, modelisation and observation tasks one needs to perform to define the negotiation and the communication between actors in such a community. In a first part, we stress the need for the instantiation of this framework so as to integrate the specifications to TEL (such as SCORM, LOM or IMS/LD) as well as taking into account the evolution of software reengineering. In a second part, we propose two instances of this framework; the first is dedicated to the retro design and the second to the reengineering of a TEL system.

■ **KEYWORDS** : Reengineering, engineering, computer environment of human learning, business process

1. Introduction

2. Proposition d'un cadre unifié pour la réingénierie des EIAH

3. Exemples

4. Perspectives et conclusion

BIBLIOGRAPHIE

1. Introduction

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 technologies éducatives normées 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 supporter la spécification du comportement du dispositif d'apprentissage essentiellement a priori. Or, la démarche de conception d'un EIAH (Bruillard et Vivet, 1994), (Tchounikine et al., 2004) ne peut pas être réduite simplement à la spécification du dispositif mais doit 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.

Pour Bélisle et Linard, cités dans (Barbot et Camatarri, 1999), 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 suivants des parcours qui ne soient pas simplement informatifs ou comportementalistes, mais qui leur assurent une redéfinition existentielle de leur rôle".

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'explicitier son organisation et l'ingénierie mise en œuvre. Elles leur permettent tout à la fois de définir leur activité dans ce nouveau cadre, 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 de logiciels libres s'est traduite par la mise au point d'outils supports effectifs de communication structurée et située autour des 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, centré sur les problématiques d'interopérabilité et de réutilisabilité. (Chikofsky, 1990), par exemple, a finalisé une première terminologie pour définir les tâches de rétroconception et de réingénierie d'un logiciel. Tout au long de cet article, nous utilisons ces termes, ou la tâche de rétroconception 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" et la réingénierie 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". 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 et al., 1999) définissant à la fois des éléments méthodologiques et un vocabulaire commun.

Nous proposons dans cet article d'utiliser ces résultats proposés par la communauté du génie logiciel pour définir un cadre capable de supporter une communauté de concepteurs en situation de réingénierie d'un système de formation. La proposition que nous faisons ici est d'instancier le cadre ODP-RM (Open Distributed Processing – Reference Model) sur le processus de réingénierie d'un EIAH. Le modèle de référence choisi se présente comme un cadre de coordination pour la standardisation des processus distribués ouverts en définissant avec beaucoup de détails un ensemble de concepts issus des pratiques de développement. Les objectifs et les motivations d'un tel cadre sont détaillés dans (ISO/IEC-10746-1, 1998).

Dans une première partie, nous définissons notre approche de l'instance du cadre ODP-RM sur le processus de réingénierie d'un EIAH. Nous illustrons ensuite le modèle proposé 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 notre proposition.

2. Proposition d'un cadre unifié pour la réingénierie des EIAH

2.1. Présentation du cadre ODP-RM

2.1.1 Modèle général

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 classifient les objets du système par des liens de composition ;
- 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 implémenter dans le système sont définies.

L'objectif avoué par le cadre ODP-RM est de favoriser l'émergence d'un consensus (un standard) dans une communauté de concepteurs donnée.

ODP-RM est structuré en trois documents distincts :

- le premier document "Overview" définit l'univers du discours d'un concepteur en précisant le spectre d'application du modèle (ISO/IEC-10746-1, 1998).
- le second document "Foundation" définit les concepts génériques liés aux actes de spécification, d'observation 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* (Les mots en italiques correspondent aux termes définis par le cadre ODP-RM) sur la conception architecturale d'un système:

- le *point de vue métier* focalise sur les *stratégies*, les *ressources*, les contraintes du système, représentées à l'aide des concepts *communauté*, *acteur*, *rôle*, *artefact* et *activité* ;
- 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.
- le *point de vue technologique* définit les technologies logicielles et matérielles utilisées.

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*. Ceux-ci sont une des réponses de ce modèle de référence pour appréhender la complexité d'un système lors de sa spécification.

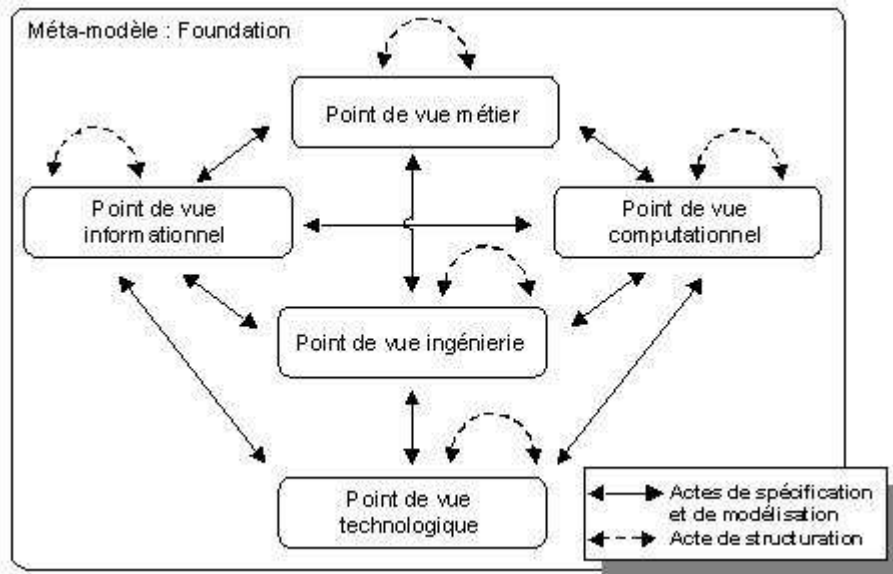


Figure 1 : Les cinq points de vue inscrits dans le méta-modèle ODP-RM

Pour illustrer ODP-RM, nous montrons comment il peut s'instancier sur deux exemples.

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

Le projet européen ESPRIT 21975 (Bär 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.

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 adressent les acteurs de la réingénierie en décrivant et guidant leurs différentes tâches, constituent l'instanciation du *point de vue métier* d'ODP-RM sur la réingénierie du logiciel. Les patrons orientés structures instancient les *points de vue ingénierie*, *computationnel* et *technologique* en optimisant la rétroconception des composants logicielles analysés et 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 de patrons de conception est d'être indépendant de tout domaine d'application.

ODP-RM se présente alors comme un cadre générique capable de mettre en relation ces résultats sur la réingénierie du logiciel avec un domaine spécifique, constituant une instantiation du *point de vue informationnel*.

2.1.3. Instantiation d'ODP-RM sur l'ingénierie des EIAH

Nous définissons par l'expression ingénierie éducative normée les travaux des communautés de conception exploitant et/ou définissant des propositions de normes et standards des technologies éducatives. Les trois propositions majeures que nous retiendrons ici sont :

- Le standard de description LOM (Learning Object Metadata) du comité IEEE¹ qui permet de décrire une ressource pédagogique afin qu'elle puisse être référencée dans une banque distribuée 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 IMS² qui permettent de décrire de manière cohérente le *point de vue informationnel* sur le 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 qui 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 diffusée par le consortium IMS décrit le déroulement d'une session de formation. Une structuration au niveau du contenu exploité en session 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 et al., 2002), 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 schémas des 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 comme OpenUSS³ et des spécifications produites par des industriels (SUN, 2003). Le développeur doit ici s'employer à respecter les *interfaces* et le *comportement* des différents composants 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 la définition d'un modèle abstrait explicitant les composantes fonctionnelles et les différents flux d'un processus cyclique d'un dispositif d'apprentissage centré sur les interactions entre l'apprenant et le système. 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 montrer les manques actuels de ces travaux. L'objectif initial des technologies éducatives normées est de 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*.

2.2. Modèle métier de la réingénierie d'un 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 alors par les retours quantitatifs des usages observés mais également par la recherche d'indicateurs qualitatifs manquants (ISO/IEC-10746-1, 1998).

L'instance du modèle ODP-RM sur l'ingénierie normée des EIAH montre l'absence des *points de vue technologique* et *métier*. Or, le premier permet de communiquer les choix techniques de conception des composantes d'un système de formation et d'optimiser les phases de rétroconception. Le deuxième permet d'explicitier dans un cadre unique 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, 2002) en constitue cependant une tentative mais se limite à synthétiser les différents projets d'assurance qualité des systèmes de formation.

Les travaux présentés dans (Oubahssi 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 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 et les mises en pratique des 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*.
- *Communauté* : Regroupe un ensemble d'*objets métiers* ayant des *objectifs* communs.
- *Processus* : Définit un ensemble d'*étapes* agencées de manière prédéfinie et conduisant à un *objectif*.
- *Etape* : Définit l'abstraction d'une *action* dans un *processus*.
- *Comportement* : Exprime les *actions* de négociations entre les différents *objets métiers*.
- *Objectif* : Qualifie une *communauté* en précisant ses intentions.
- *Artefact* : Qualifie un *objet métier* référencé dans une *action*.
- *Acteur* : Qualifie un *objet métier* participant à une *action*.
- *Ressource* : Qualifie un *objet métier* nécessaire à son *comportement*.

La définition d'un modèle *métier* d'un système de formation, la création d'instances, l'identification de correspondances avec les *instances* de *points de vue* sur les technologies éducatives sont présentées dans la suite de cet article.

2.3. Point de vue métier d'ODP-RM instancié sur la réingénierie d'un EIAH

Notre proposition est d'utiliser les technologies éducatives comme un support à la réingénierie des systèmes de formation. Le modèle présenté reprend les conventions graphiques du profil UML pour ECA (Enterprise Collaboration Architecture) présenté par le consortium OMG (OMG-ECA, 2004). Il est destiné à supporter un processus de développement dirigé par les modèles utilisant le cadre ODP-RM (Nagase et al., 2004). Nous reprenons les règles de transcription graphique associées aux termes du cadre ODP-RM.

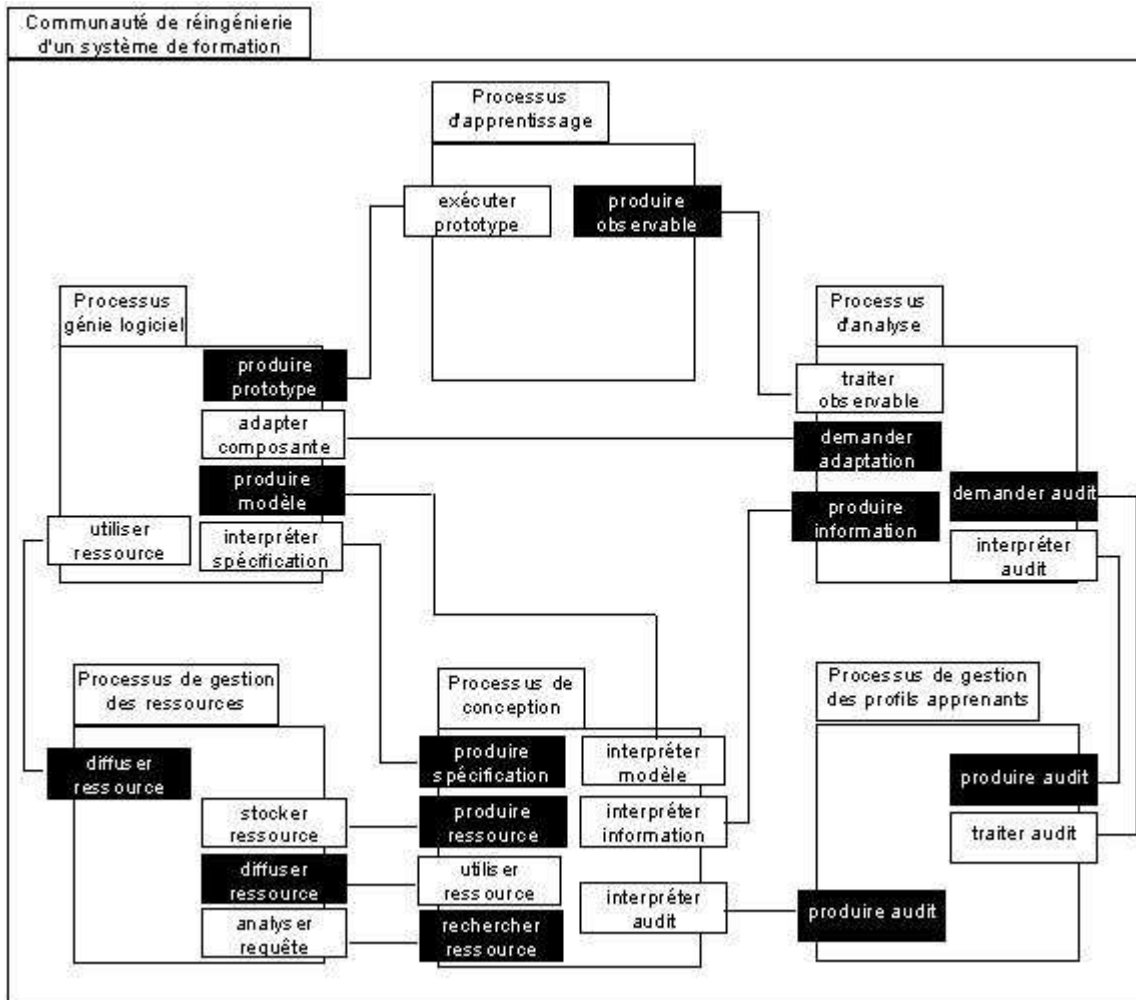


Figure 2 : Diagramme de paquetage UML représentant le processus de réingénierie d'un EIAH suivant le système de notation du profil UML ECA de l'OMG

En appliquant la terminologie et les règles du langage de spécification sur le *point de vue métier*, il est possible de définir avec un minimum d'ambiguïté les *communautés* et les *objectifs* du modèle de spécification d'un tel *point de vue*. Notre vision métier de la réingénierie d'un EIAH se décompose en six *processus*.

- 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 système.
- Le *processus* génie logiciel met en œuvre une démarche de réingénierie sur les différentes composantes de l'environnement de formation. 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 des sessions. La spécification du scénario repose sur la définition des activités effectuées, ou dans la terminologie ODP-RM des *comportements*, par chacun des *acteurs* de la session d'apprentissage.
- Le *processus* d'analyse met en œuvre les techniques d'analyse des usages, 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 métiers* dont un des type est l'objet pédagogique.
- Le *processus* de gestion des profils apprenants administre les *ressources métiers* des *processus* d'analyse et de conception du cycle de réingénierie.

Chaque processus prend part à l'*action* liée aux cycles de réingénierie d'un système de formation. En effectuant des regroupements, ils peuvent être considérés de manière plus générale comme des *communautés* d'objets *métiers*. Cette nouvelle considération permet de s'interroger sur les *objectifs*, les *rôles* et les *processus* de ces *communautés*. Ce concept est au cœur des différentes instances du *point de vue métier* d'un *processus* de réingénierie d'un EIAH.

3. Exemples

3.1. Présentation

Les deux exemples présentés sont extraits des mises à l'essai effectuées sur le site de l'IUT de Laval. Au court de ces quatre dernières années, une cinquantaine d'étudiants ont 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 mise en œuvre, 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'enseignement : 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* d'un système de formation considéré se définit comme étant composé de six processus : apprentissage, analyse, conception, génie logiciel, gestion de ressources et gestion des profils apprenants. Au sein de chacun d'eux sont mises en œuvre des composantes logicielles et opérationnelles

issues des communautés ouvertes utilisant les technologies informatiques orientées objets. Le tableau ci-dessous établit la liste des communautés auxquelles nous avons adhéré pour instancier les six processus mis en œuvre dans nos mises à l'essai.

| Processus | Composantes logicielles utilisées |
|---|---|
| Processus de conception | Mozilla ⁴ , OpenOffice ⁵ , OpenUSS et FreeStyle Learning ⁶ |
| Processus d'apprentissage | Mozilla, OpenUSS et FreeStyle Learning |
| Processus génie logiciel | Eclipse ⁷ , AndroMDA ⁸ et Poseidon CE ⁹ |
| Processus d'analyse | Weka ¹⁰ et WUM ¹¹ |
| Processus de gestion de ressources | CVS ¹² et GForge ¹³ |
| Processus de gestion des profils apprenants | OpenLDAP ¹⁴ |

Tableau 1 : Projets des composantes logicielles utilisées

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

3.2. Exemple de rétroconception sur un système de formation

3.2.1. Introduction

Partant de la définition de la rétroconception logicielle au sens de (Chikofsky, 1990) où la finalité de la 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", la rétroconception d'un EIAH correspond au souhait de spécifier avec plus de détail le système. Dans le cas présenté, la négociation entre le *processus* génie logiciel et le *processus* de conception est illustrée. Elle aboutit à l'*artefact métier* représenté sur la figure 7.

La description du cas d'usage est formalisée en cinq étapes.

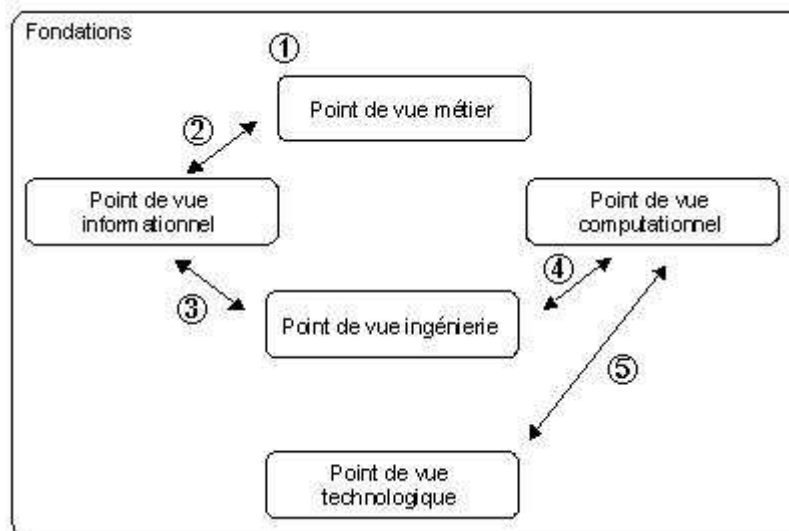


Figure 3 : Représentation des cinq étapes du cas d'usage présenté

3.2.2. Etape 1 : Instance du point de vue métier sur la rétroconception des EIAH

La rétroconception d'un EIAH est représentée au niveau métier par un dialogue entre les deux objets métiers le *processus* génie logiciel et le *processus* de conception. Les deux *processus* sont regroupés au sein d'une *communauté*, ce qui permet la négociation entre les *processus*.

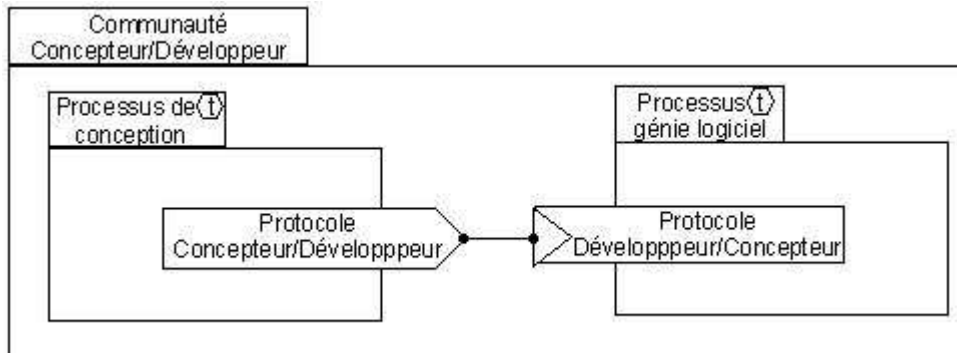


Figure 4 : Instance du modèle métier suivant le système de notation du profil UML ECA de l'OMG

3.2.3. Etape 2 : Instance du point de vue informationnel sur la rétroconception des 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 5 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 objets. L'objet "knowledge-object" est la page HTML "author_text1.html" 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. Comme le précise (Koper R., 2001), le concepteur en spécifiant un contenu pédagogique ("knowledge-object") explicite son souhait d'y associer un objet support ("tool-object").

```
<imsld:environment identifi er="ENV-textStudy-e1">
  <imsld:learning-object identifi er="LO-textStudy-e1" type="knowledge-object">
    <imsld:item identifi er="ITEM-textStudy-e1" identiofierref="RES-textStudy-e1" />
  </imsld:learning-object>
  <imsld:learning-object identifi er="LO-textStudyManagerFSL" type="tool-object">
    <imsld:item identifi er="ITEM-textStudyManager" identiofierref="RES-textStudyFSL" />
  </imsld:learning-object>
</imsld:environment>
...
<imscp:resource identifi er="RES-textStudy-e1" adlcp:scormtype="asset" type="webcontent"
href="/textStudy/author_text1.html">
  <imscp:file href="/textStudy/author_text1.html" />
</imscp:resource>
...
<imscp:resource identifi er="RES-textStudyFSL" adlcp:scormtype="sco" type="">
  <imscp:file href="/learningUnitViewManagers/textStudy/manager.jar" />
</imscp:resource>
```

Figure 5 : Extrait du scénario d'apprentissage

3.2.4. Etape 3 : Instance du point de vue ingénierie sur la rétroconception des 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 du concepteur. 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) :

- *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 correspond aux différents modèles définis par les propositions IMS-LD (IMS Learning Design), IMS-CP (IMS Content Packaging) et ADL-CP (ADLNet Content Package), auxquels appartiennent les objets *informationnels* `imscp:file`, `imscp:resource`, `imslid:item`, `imslid:environment`, `imslid:learning-object` et `adlcp:scormtype`. Le *schéma invariant* définit les liens entre ces différents objets. Il est représenté ici dans le formalisme graphique du diagramme de classe UML (OMG-UML, 2003).

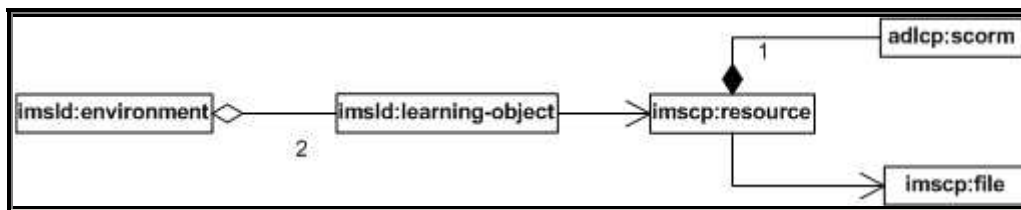


Figure 6 : Représentation du schéma invariant du cas d'usage

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 *informationnels* permettant l'établissement du *schéma dynamique*. Il s'agit ici de spécifier le *comportement* de ces différents objets.

3.2.5. Etape 4 : Instance du point de vue computationnel sur la rétroconception des EIAH

Afin de spécifier le comportement dynamique, des détails liés à la conception vont être extraits. Le patron "Capturer les détails du modèle" (Demeyer et al., 2002) 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 du consortium W3C) 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 (ISO/IEC-10746-3, 1996) : 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 pour concevoir l'interface utilisateur graphique (Loy 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.

3.2.6. Etape 5 : Instance du point de vue technologique sur la rétroconception des EIAH

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

En reprenant le paradigme évènementiel, il est alors nécessaire d'identifier les différents objets jouant les rôles d'initiateurs ou de répondeurs 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 et al., 1996), est implémenté dans la librairie java "swing/JFC". Chaque composante appliquant ce paradigme est un objet à observer.

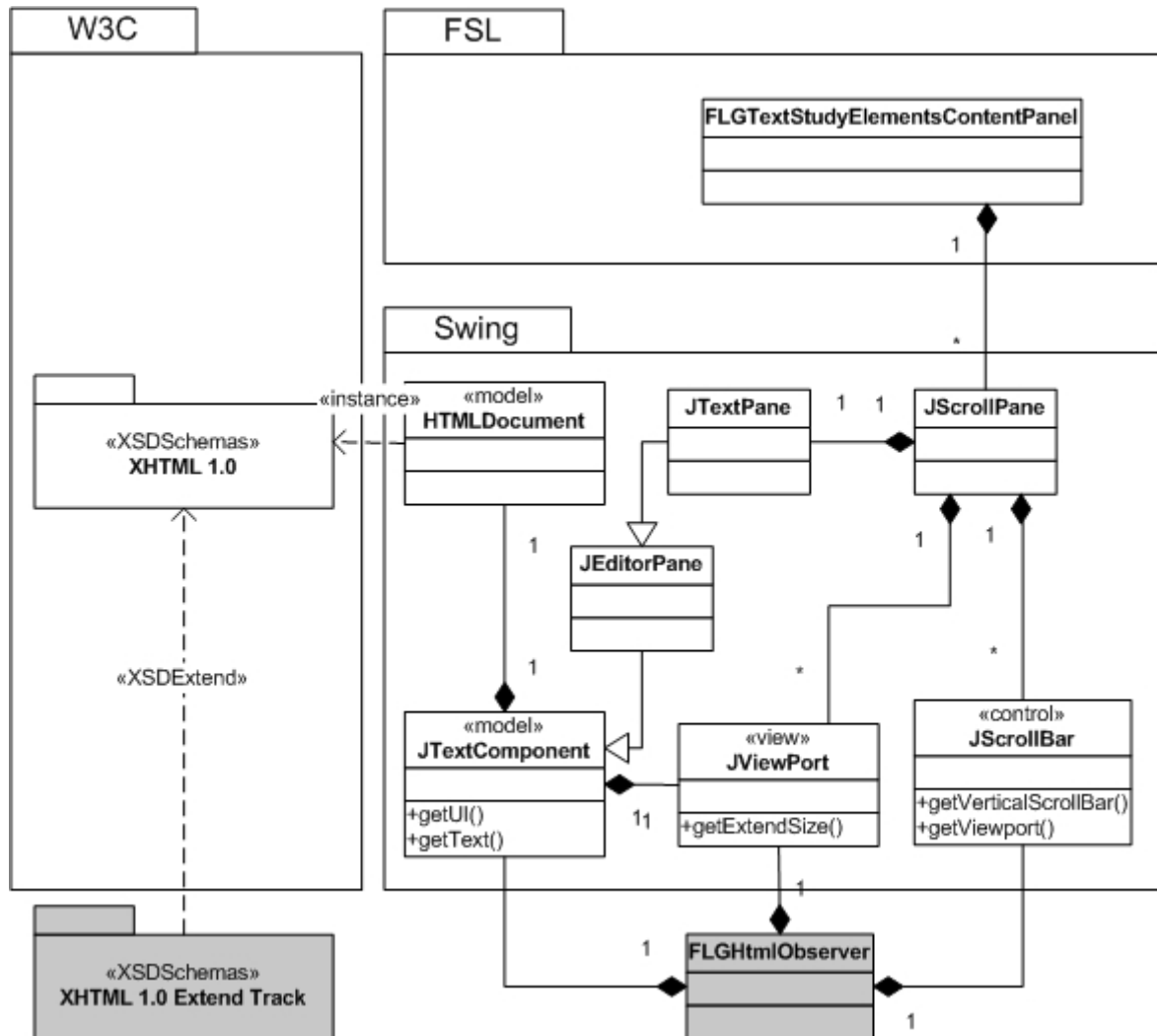


Figure 7 : Modèle émergent intégrant la sonde logicielle

3.2.7. Conclusion

La tâche de rétroconception des EIAH illustrée dans ce cas d'usage est 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. Des sondes sont alors intégrées, 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 le concepteur à négocier sur (i) le scénario pédagogique par la prise en compte de la sémantique des signaux générés par les sondes, (ii) l'objet pédagogique et la restructuration de son contenu et (iii) 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 usage.

D'un point de vue métier, ce cas d'usage a produit un artefact qui est représenté par le modèle de la figure 7. Cette production sert à apporter des précisions sur la description de l'artefact métier et à initier des réflexions sur ses interactions avec les autres objets d'une communauté.

3.3. Exemple de réingénierie sur un système de formation

3.3.1. Introduction

Selon la définition de la réingénierie du logiciel au sens de (Chikofsky, 1990) où la finalité est d'effectuer les tâches "d'examen et d'altération d'un système afin de le reconstituer sous une nouvelle forme suivies 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'explicitier ces modifications ainsi que les décisions de réingénierie prises.

La description du cas d'usage décompose en plusieurs étapes le cycle d'une communauté de réingénierie d'un système de formation. Dans (ISO/IEC-15414, 2002), le concept de *rôle* est défini comme "une abstraction du comportement d'une communauté". En conséquence, une communauté de réingénierie des systèmes de formation est définie par des *règles* prenant en compte les différents *rôles* joués par chaque *processus métier*. Ainsi, le concept de *rôle* est repris dans la description du cas d'usage présenté sur la réingénierie des EIAH.

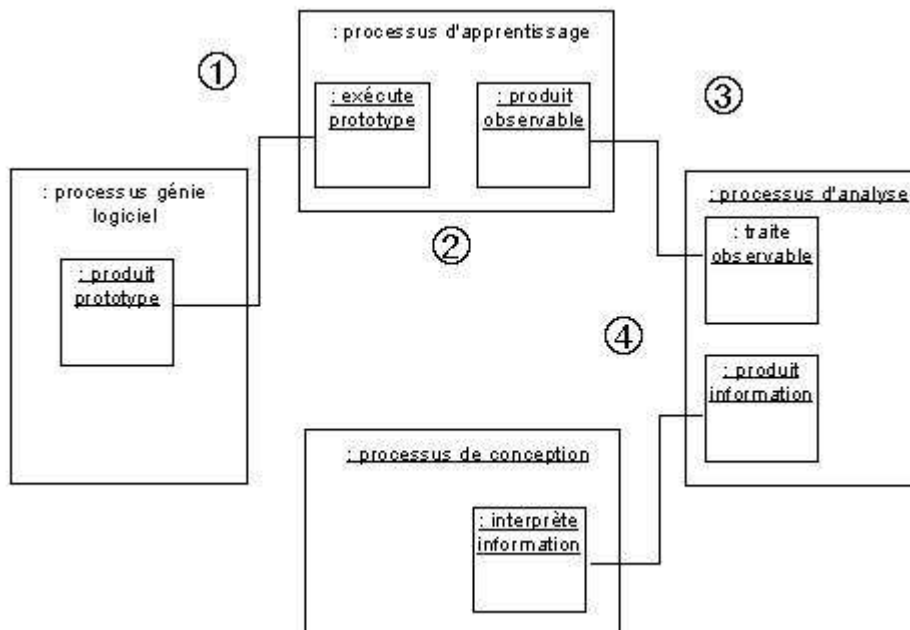


Figure 8 : Diagramme de collaboration UML du second cas d'usage suivant le système de notation du profil UML ECA

3.3.2. Etape 1 : Rôle du processus génie logiciel sur la réingénierie des 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". Des sondes sont intégrées afin de produire des observables liés aux décisions prises sur le plan ingénierie.

Dans l'exemple présenté, l'outil de déploiement utilisé est JavaWebStart¹⁵ de SUN. Suivant la terminologie du cadre ODP-RM, les objets *ingénieries* sont regroupés dans une entité intitulée *cluster* qui gère ces objets. Dans le cas d'usage présenté, son instance est le gestionnaire de diffusion d'unités d'apprentissage "FreeStyle Learning". Une décomposition des différentes composantes de ce système, déployées sur le poste client (côté apprenant), est présentée dans le script de déploiement (cf. [figure 9](#)).

```

<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>
  ...
  <application-desc main-class="freestyleLearning.homeCore.FSLApp"/>
</jnlp>

```

Figure 9 : Script de déploiement pour le dispositif "FreeStyle Learning"

3.3.3. Etape 2 : Rôle du processus d'apprentissage sur la réingénierie des EIAH

Les précédentes considérations du *point de vue ingénierie* sont corrélées avec les spécifications *computationnelles* définies au niveau client par SCORM d'ADLNet et côté 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 *ingénieries*. Chaque *état* de ces objets est tracé en respectant la terminologie de la spécification technique SCORM.

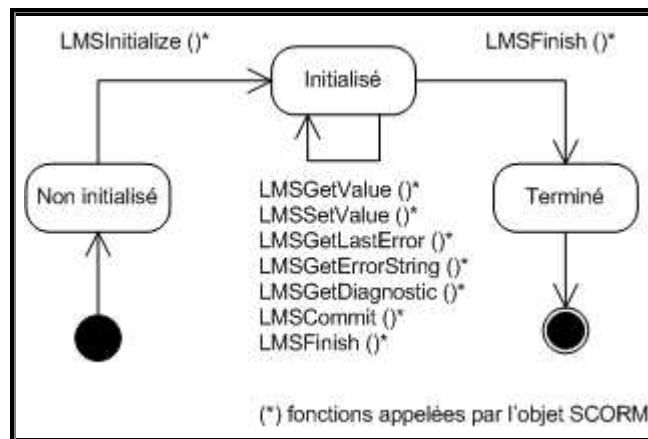


Figure 10 : Diagramme etat-transition UML du cycle de vie d'un objet SCORM

3.3.4. Etape 3 : Rôle du processus d'analyse sur la réingénierie des 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 capture, le paquetage auquel appartient l'objet *ingénierie* et l'intitulé de l'événement capté.

```

...
<observer id="src2e05">
  <date>
    <element name="Format" value="ISO 8601"/>
    <element name="Value" value="31/03/2004:10:45:42 +0407"/>
  </date>
  <event>
    <element name="Package" value="freestyleLearning.learningUnitViewManagers.checkUp"/>
    <element name="Event" value="LMSInitialize()"/>
  </event>
</observer>
...

```

Figure 11 : Exemple de trace générée par le composant "FreeStyle Learning"

Dans cet exemple, le *processus* d'analyse 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'arbre ou de règles (Lefebvre et Venturi, 2001). Il s'agit de permettre la communication autour de ces structures afin de faire émerger leurs sémantiques dans le domaine d'apprentissage. Ce *processus métier* s'intègre parfaitement dans le cadre ODP-RM qui partage le même objectif d'émergence de sens. Plus concrètement, 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. Le langage utilisé est celui du *point de vue ingénierie*. Il permet de spécifier les différents comportements des groupes d'apprenants.

3.3.5. Etape 4 : Rôle du processus de conception sur la réingénierie des EIAH

Une négociation, à l'initiative du processus d'analyse, est effectuée avec le *processus* de conception. 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 *cluster* (regroupement d'objets *ingénieries*) est à mettre en relation avec l'instance des objets *informationnels* décrivant chacune la structure des activités du scénario pédagogique. La *figure 12* montre l'évolution du *point de vue ingénierie*, en corrélation avec le *point de vue informationnel*.

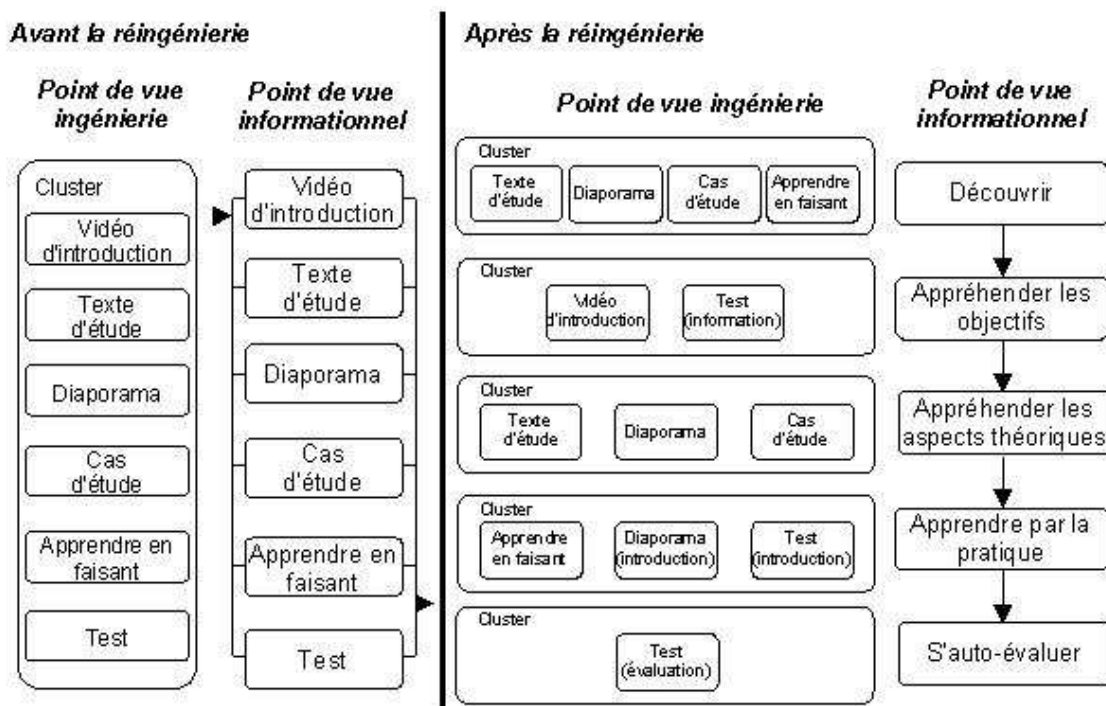


Figure 12 : Diagrammes état-transition UML d'objets ingénieries et informationnels

3.3.6. Conclusion

La tâche de réingénierie des EIAH illustrée dans ce cas d'usage fait intervenir les différents rôles des processus métiers 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'explicitier les correspondances entre les différents points de vue met en évidence les enjeux suivants (ISO/IEC-15414, 2002) :

- d'un point de vue ingénierie, cinq nouveaux questionnaires de cluster sont développés pour l'application "FreeStyle Learning". Ils correspondent aux activités pédagogiques "Découvrir", "Appréhender les objectifs", "Appréhender les aspects théoriques", "Apprendre par la pratique" et "S'auto-évaluer". Une fois codées, ces modifications sont communiquées à la communauté de logiciel libre concernée.
- le cas présenté peut être proposé comme un exemple d'instance du modèle "Learning design" du consortium IMS.

Dans le cadre ODP-RM, l'ensemble des mises en correspondance entre les points de vue produisent des règles de conformité spécifiant le cycle de réingénierie. Mises à disposition de la communauté de réingénierie des EIAH, elles interrogent les concepteurs sur l'opportunité de changer certains faits afin de mettre à l'essai une nouvelle instance du cycle. Derrière cette opportunité se définit la démarche qualité d'un système au sens d'ODP-RM (ISO/IEC-10746-1, 1998).

4. Perspectives et conclusion

Dans cet article, nous avons présenté un cadre unique permettant d'explicitier les mécanismes de transformation sur des modèles ayant pour origine une intention pédagogique et/ou technologique. Ces derniers utilisent des formalismes et des expertises différents suivant les communautés concernées. L'instrumentation des mécanismes de transformation entre – et au sein de – les différents niveaux d'abstraction et l'explicitation des liens de composition et de décomposition définissent l'un des enjeux de la conception des EIAH.

En inscrivant les technologies éducatives dans le cadre ODP-RM, nous souhaitons sensibiliser la communauté de conception sur son potentiel à fédérer les expériences de formation. 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. Le mouvement du logiciel libre illustre parfaitement les perspectives de nos travaux car il cherche à faire la promotion à la fois d'une philosophie de conception (Raymond, 2001), des outils (Bar et Fogel, 2003) et des supports de communication comme par exemple l'outil GForge. Le concepteur ou l'utilisateur membre d'une telle communauté est pris dans une synergie où les ressources produites sont considérées comme des prototypes, par nature évolutives. Nous pensons qu'il y a là une vraie dynamique de structuration d'une communauté. Le cadre que nous avons proposé dans cet article constitue, à notre sens, un préalable à la création d'une telle dynamique dans la recherche de qualité des processus d'ingénierie et de réingénierie des EIAH.

BIBLIOGRAPHIE

Bär H., Bauer M., Ciupke O., Demeyer S., Ducasse S., Lanza M., Marinescu R., Nebbe R., Nierstrasz O., Przybilski M., Richner T., Rieger M., Riva C., Sassen A.-M., Schulz B., Steyaert P., Tichelaar S. (1999). *The FAMOOS Object-Oriented Reengineering Handbook*, rapport du projet européen Esprit 21975, 1999.

Bar M., Fogel B (2003). *Development with CVS*. Paraglyph Press, 3^{ième} édition, Paraglyph Press, 2003.

Barbot M.-J., Camatarri G. (1999). *Autonomie et apprentissage, l'innovation dans la formation*, Paris, Edition PUF, 1999.

- Bruillard E., Vivet M. (1994). *Concevoir des EIAO pour des situations scolaires : approche méthodologique*. Grenoble: in N. Balacheff et M. Vivet, *Didactique et Intelligence Artificielle*, La Pensée Sauvage, pp. 273-302.
- Buschmann F., Meunier R., Rohnert H., Sommerlad P., Stal M. (1996). *Pattern-Oriented Software Architecture: A System of Patterns*, Vol. 1, John Wiley & Sons, 1996.
- Chikofsky E.J. (1990). *Reverse Engineering and Design Recovery: A Taxonomy*, Software IEEE, vol. 7, pp. 13-17.
- Demeyer S., Ducasse S., Nierstrasz O. (2002). *Object-oriented reengineering patterns*, Morgan Kaufmann/Elsevier and DPunkt, 2002.
- ISO/IEC (1998). *Open Distributed Processing Reference Model, Part 1: Overview*, ISO/IEC, 1998.
- ISO/IEC (1996). *Open Distributed Processing Reference Model, Part 2: Foundations*, ISO/IEC, 1996.
- ISO/IEC (1996). *Open Distributed Processing Reference Model, Part 3: Architecture*, ISO/IEC, 1996.
- ISO/IEC (2002). *Information Technology : Open Distributed Processing Reference Model, Entreprise Language*, ISO/IEC, 2002.
- Koper R. (2001). *Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML*, Educational Technology Expertise Center Open University of The Netherlands, version 2, 2001.
- Lefébure R., Venturi G. (2001). *Data Mining: Gestion de la relation client personnalisation de sites web*, Editions Eyrolles, 2001.
- Loy M., Eckstein R., Wood D., Elliott J., Cole B. (2002). *Java Swing*, O'Reilly, 2^{ième} édition, 2002.
- Nagase Y., Hashimoto D., Sato M. (2004). *Applying Model-Driven Development to Business Systems using RM-ODP en EDOC*, Workshop on ODP for Enterprise Computing, EDOC2004, 36-42.
- OMG (2004). *Enterprise Collaboration Architecture (ECA) Specification, version 1.0*, 2004.
- OMG (2003). *UML 2.0 Infrastructure Specification*, 2003.
- Oubahssi L., GrandBastien M., Claës G. (2004). *Ré-ingénierie d'une plate forme fondée sur la modélisation d'un processus global de FOAD*, actes de la conférence TICE'04, Compiègne, pp. 32-38.
- Pawlowski J.M. (2002). *Report on Quality Assurance Standards / Proposal for Future Work. Project Team Quality Assurance and Guidelines*, rapport du CEN / ISSS WS/LT, 2002.
- Rawlings A., Rosmalen P., Koper R., Rodríguez-Artacho M., Lefrere P. (2002). *Survey of Educational Modelling Languages (EMLs)*, rapport du CEN/ISSS WS/LT, 2002.
- Raymond E.S. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, Hardcover Edition O'Reilly, 2001.
- SUN Microsystems (2003). *E-Learning Framework: Technical White Paper*, 2003.
- Tchounikine P., Baker M., Balacheff N., Baron M., Derycke A., Guin D., Nicaud J.-F., Rabardel P. (2004). *Platon-1: quelques dimensions pour l'analyse des travaux de recherche en conception d'EIAH*, rapport, AS: Fondements théoriques et méthodologiques de la conception.

¹ Institute of Electrical and Electronics Engineers, Inc

² Instructional Management Systems

³ Open Source University Support System, projet communautaire développant une plateforme gérant les sessions de formation à distance (Learning Management System), <http://openuss.sourceforge.net/openuss/>

⁴ Mozilla, projet communautaire développant un ensemble d'applications logicielles pour accéder aux services de l'Internet (navigateur web, messagerie électronique, ...), <http://www.mozilla.org>

⁵ OpenOffice, projet communautaire développant un ensemble d'applications logicielles bureautiques, <http://www.openoffice.org/>

⁶ FreeStyle Learning, projet communautaire développant une composante de la plateforme OpenUSS chargée de gérer la diffusion d'une unité d'apprentissage (Learning Content System), <http://www.freestyle-learning.de/>

⁷ Eclipse, projet communautaire développant une plateforme de développement, <http://www.eclipse.org/>

⁸ AndromDA, projet communautaire développant un cadre technique pour générer automatiquement le code d'une application à partir d'un modèle UML, <http://www.andromda.org/>

⁹ Poseidon CE, projet communautaire développant un atelier génie logiciel permettant d'éditer des modèles UML (Le code de l'application n'est pas accessible), <http://gentleware.com/index.php>

¹⁰ Weka, projet communautaire développant une application pour supporter le travail de fouille de données, <http://www.cs.waikato.ac.nz/ml/weka/>

¹¹ WUM, projet communautaire développant une application pour supporter le travail de fouille de données générées par les serveurs Web, <http://munch.wiwi.hu-berlin.de/hypknowsys/wum/>

¹² CVS, projet communautaire développant l'outil CVS (Concurrent Versions System), <https://www.cvshome.org/>

¹³ GForge, projet communautaire diffusant une plateforme de développement collaboratif, <http://gforge.org/>

¹⁴ OpenLDAP, projet communautaire diffusant un gestionnaire d'annuaire distribué LDAP (Lightweight Directory Access Protocol), <http://www.openldap.org/>

¹⁵ JavaWebStart, communauté Java proposant un protocole de déploiement des composants d'une application logicielle mettant en œuvre les technologies Java, <http://www.jcp.org/en/jsr/detail?id=56/>

■ A propos des auteurs

Alain CORBIERE finalise actuellement une thèse de doctorat sur la proposition d'un cadre méthodologique pour l'ingénierie et la réingénierie des EIAH. Il est membre du Laboratoire d'Informatique de l'Université du Maine.

Courriel : alain.corbiere@univ-lemans.fr

Toile : www.univ-lemans.fr/~acorb

Christophe CHOQUET est Maître de Conférences en informatique à l'Université du Maine et membre du Laboratoire d'Informatique de l'Université du Maine. Ses recherches portent sur l'ingénierie et la réingénierie des EIAH dirigées par les modèles.

Adresse : IUT Laval Département Services et Réseaux de Communication, LIUM , 52, rue des docteurs Calmette et Guérin, BP 2045, 53020 Laval Cedex 09, France.

Courriel : christophe.choquet@univ-lemans.fr

Toile : www.univ-lemans.fr/~choquet

Référence de l'article :

Alain Corbière, Christophe Choquet, ODP-RM : Un cadre de réingénierie des systèmes de formation, *Revue STICEF*, Volume 12, 2005, ISSN : 1764-7223, mis en ligne le 20/10/2005, <http://sticef.org>
© Revue Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation, 2005