HAL
open science

Approche orientée services pour la construction des environnements de modélisation

Jorge-Luis Pérez-Medina, Sophie Dupuy-Chessa, Dominique Rieu

To cite this version:

Jorge-Luis Pérez-Medina, Sophie Dupuy-Chessa, Dominique Rieu. Approche orientée services pour la construction des environnements de modélisation. Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information, 2010, 15 (4), pp.113-137. hal-00953328

HAL Id: hal-00953328

https://inria.hal.science/hal-00953328v1

Submitted on 28 Feb 2014

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

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

Approche orientée services pour la construction des environnements de modélisation

Jorge Luis Pérez Medina, Sophie Dupuy-Chessa, Dominique Rieu

Laboratoire d'Informatique de Grenoble, Université de Grenoble
385 rue de la Bibliothèque
B.P. 53 3841 Grenoble Cedex 9, France

{Jorge-Luis.Perez-Medina, Sophie.Dupuy, Dominique.Rieu}@imag.fr

RÉSUMÉ. *L'ingénierie des modèles considère tout artefact logiciel comme un modèle. La gestion de modèles regroupe tout un ensemble de fonctionnalités permettant de représenter, créer, stocker et manipuler les modèles. Actuellement les besoins des concepteurs en termes de gestion de processus et produits sont divers et les outils de modélisation ne sont pas complets car les besoins autour des modèles ne sont pas consensuels. Pour remédier à l'hétérogénéité et aux limitations fonctionnelles des outils de gestion de modèles, l'objectif de nos recherches, est de faciliter le travail des concepteurs de modèles et chef de projets en les aidant dans le choix de processus, des modèles et d'environnements de modélisation adaptés à leurs besoins spécifiques. Cet article détaille l'utilisation d'une approche orientée services pour la gestion de modèles, selon les besoins des concepteurs. Nos propositions portent sur trois niveaux d'abstraction : opérationnel, organisationnel et intentionnel. Le niveau opérationnel, permet de choisir l'ensemble d'outils appropriés, le niveau organisationnel facilite la sélection d'un processus et le niveau intentionnel permet d'expliciter les besoins des concepteurs en termes de gestion de modèles.*

ABSTRACT. *Models engineering considers any software artefact as a model. Models management groups a set of features which permit to represent, create, store and manipulate the models. Nowadays, the needs of the designers in terms of the process management and products are diverse and the modelling tools are not complete because the needs and usages around the models are not consensual. To remedy the heterogeneity and the functional limitations of the models' management tools, the aim of our researches is to facilitate the work of model designers and project managers by helping them in choosing processes, models and modeling environments adapted to their specific needs. This paper details the use of a service-oriented approach for model management, according to the needs of the models designers. Our propositions are related to three different abstract levels: the operational level, the organisational level and the intentional level. The operational level allows to choose the appropriate tool, the organisational level facilitates the selection of a process and the intentional level allows to define modelling goals.*

MOTS-CLÉS : *Modèle, service, gestion de modèles, outil de modélisation.*

KEYWORDS: *Model, service, model management, modeling tools.*

1. Introduction

L'ingénierie des modèles est une branche du génie logiciel où tout artefact logiciel est considéré comme un modèle. Dans ce contexte, un modèle est une spécification souvent partielle des fonctionnalités, de la structure et/ou du comportement d'un système ou d'une application (OMG03, 2003). Dans le processus de développement du logiciel, les concepteurs de modèles jouent différents rôles et s'appuient sur des outils classiques tels que les Ateliers de Génie Logiciel (AGL), et les outils d'Ingénierie Dirigée par les Modèles (IDM), pour résoudre leurs besoins en termes de gestion de modèles.

La gestion de modèles regroupe l'ensemble des fonctionnalités permettant de représenter, créer, stocker et manipuler les modèles, avec pour but d'expliquer et de représenter les systèmes sous différents aspects (statique, dynamique, fonctionnel, interactionnel) et à différents niveaux d'abstraction (expression de besoins, analyse, conception,…). Dans le domaine du Génie Logiciel, l'Object Management Group a proposé l'approche dirigée par les modèles, MDA, dont les concepts sont orientés-modèles plutôt que orientés-objets. L'approche MDA (OMG03, 2003) offre le pouvoir d'abstraction, de raffinement et des vues multiples sur les modèles. Surtout, elle donne la possibilité de concevoir des modèles indépendants des plateformes et de l'environnement d'implantation. Le MDA a été étendu à l'IDM (action IDM, 2007) qui ne se limite pas aux modèles UML et met les modèles, et non pas les programmes, au centre de la démarche en Génie logiciel. Donc, l'IDM est utilisé pour décrire l'ensemble des concepts et technologies autour de la gestion de modèles. La plupart des outils IDM sont utilisés pour automatiser les transformations de modèles ce qui comprend l'évolution, l'évaluation, la génération de code, la validation, le test, la traçabilité.

Au-delà de l'IDM, les AGL sont un ensemble cohérent d'outils informatiques formant des environnements d'aide à la conception, au développement et à la mise au point des logiciels. Ils sont dotés de fonctionnalités plus ou moins avancées adaptées aux différentes phases de la production d'un logiciel. Actuellement les besoins des concepteurs en termes de gestion de processus et produits sont divers et les outils de modélisation ne sont pas complets car les besoins et les usages autour des modèles ne sont pas consensuels.

L'objectif global de notre travail est de faciliter la construction d'environnements de modélisation adaptés aux besoins spécifiques des concepteurs de modèles. Pour ce faire nous proposons une plateforme permettant de trouver, de coupler et d'utiliser des briques fonctionnelles répondant aux besoins spécifiques de chaque concepteur. Notre approche est basée sur la réutilisation de processus existants, de modèles et d'outils pour trouver une solution aux besoins de concepteurs de modèles et des chefs de projets. Pour atteindre cet objectif de réutilisation, l'approche basée sur les services (Bieber *et al.*, 2001) nous a paru

intéressante. Elle offre de plus la possibilité de sélectionner, voire de découvrir, les services répondant à certaines caractéristiques, puis de les composer pour offrir un service plus complet. Dans notre cas, l'objectif est de faciliter et d'améliorer le choix d'environnements de modélisation. Il faut donc, dans un premier temps, caractériser des services de modélisation pour les rendre plus accessibles.

Pour ce faire, nous avons choisi une approche proche de celle de SATIS (Crescenzo, 2009) proposée pour la recherche de services web pour la communauté des neuroscientifiques. Dans ce travail, la recherche d'un service est basée sur la représentation des processus métier des neurosciences et par celle des intentions des utilisateurs. De façon similaire, le choix d'un environnement de modélisation sera issu des buts des concepteurs et de leur processus de modélisation. Aussi nous avons identifié trois niveaux d'abstraction : opérationnel, organisationnel et intentionnel. Les trois niveaux offrent des services de gestion de modèles de natures différentes. Les services opérationnels réalisent des opérations automatisées, par exemple l'édition d'un modèle. Les services organisationnels proposent des fragments de méthode de conception, dédiés à l'ingénierie des systèmes à base de modèles. Le choix d'un processus de conception détermine les modèles qui vont être utilisés, et donc, leurs environnements de modélisation. Les services intentionnels correspondent à la modélisation des buts visés par toute personne ou toute organisation manipulant des modèles (Rolland et al., 2008). Le niveau intentionnel permet de justifier l'existence des buts organisationnels qui répondent aux besoins des concepteurs de modèles.

Ce papier est organisé comme suit. La section 2 présente des travaux concernant la réutilisation et la gestion de modèles dans le contexte de l'ingénierie de méthodes situationnelles. La section 3 décrit un scénario expérimental. Cet exemple est basé sur une méthode de conception de systèmes d'information et nous l'appliquerons dans le domaine de la conception des systèmes interactifs[1]. La section 4 présente notre approche orientée services pour la gestion de modèles. Les sections 5, 6 et 7 détaillent respectivement les modèles du service intentionnel, organisationnel et opérationnel. La section 8 présente une plateforme de support pour notre approche orientée service. Enfin, la section 9 contient nos conclusions et perspectives.

2. Gestion de modèles et réutilisation

Notre approche se base sur la convergence de quatre axes : l'ingénierie des besoins, l'ingénierie de méthodes situationnelles, la réutilisation et les outils de gestion de modèles. L'ingénierie des besoins est utilisée comme un moyen pour identifier et représenter les objectifs de l'entreprise dans lequel le système sera utilisé. Donc, le contexte le plus abstrait de fonctionnement du système est centré

1. Un système interactif est un système qui dépend des actions d'un utilisateur pour réaliser une tâche, c'est-à-dire tout système dans lequel interviennent en action une personne et une machine.

autour des concepts de scénarios et de buts. Cette approche nous aidera à définir et organiser les besoins des concepteurs de modèles.

La réutilisation permet de construire un système nouveau à partir d'éléments (objets, composants, etc.) existants, éprouvés et réutilisables. Nous utiliserons cette approche pour réutiliser des produits (modèles) et processus (méthodes) afin de faciliter la construction des environnements adaptés aux besoins des concepteurs.

L'étude de l'axe « outils de gestion de modèles » aide à aligner les besoins intentionnels et organisationnels avec les outils permettant d'assister les méthodes.

Par la suite nous présentons l'ingénierie de méthodes situationnelles et plus particulièrement les différents aspects concernant la réutilisation et la gestion de modèles, ce qui nous permettra de positionner notre approche orientée services pour la construction des environnements de modélisation.

2.1. L'ingénierie des méthodes situationnelles

L'ingénierie de méthodes situationnelles promeut la notion de composants de méthodes réutilisables, de processus de sélection et d'assemblage de ces composants selon la situation particulière de chaque projet (Brinkkemper *et al.*, 1998), (Ralyté *et al.*, 2001b). Cette discipline vise à construire des nouvelles méthodes d'ingénierie des systèmes d'information en réutilisant et assemblant différents fragments de méthodes qui ont déjà fait leurs preuves.

Plusieurs types de fragments ont émergé dans la littérature. Les plus connus - sont les "***method fragment***" (Harmsen *et al.*, 1994), les "***method chunks***" (Ralyté *et al.*,2001a), (Mirbel *et al.*, 2006), les "***components***" (Wistrand *et al.*, 2004), et les "***method services***" (Guzélian, 2007) (Deneckère *et al.*, 2008) (Rolland, 2008). Historiquement, le terme fragment a été le premier à apparaître, bien avant les termes composant, "*methode chunk*", etc.

Le rôle principal d'un fragment de méthode est de fournir des directives au concepteur de systèmes pour comprendre certaines activités de développement (par exemple : la modélisation métier, la spécification des besoins, la conception, etc.) et pour fournir les définitions de concepts à utiliser dans cette activité (Ralyté *et al.*, 2006). Les fragments de méthode sont alors, des blocs de constructions réutilisables qui permettent de définir des méthodes de manière modulaire. Les méthodes ainsi obtenues sont elles-mêmes modulaires et peuvent être modifiées et étendues.

Le domaine de l'ingénierie des méthodes situationnelles s'intéresse à la réutilisation de fragments de méthodes pour la construction de solutions de conception adaptées aux spécificités du projet. Ces fragments de méthodes doivent être stockés dans une base de méthodes. Les sections suivantes traitent le stockage et la construction d'une méthode.

2.1.1. *Stockage de méthodes*

La plupart des approches d'assemblage utilisent la notion de « base de méthodes » pour stocker les méthodes. On trouve une base de méthodes dans les approches présentées dans la section précédente (section 2.1). Parmi ces approches, Rolland et Praksh proposent une structure pour représenter et stocker des méthodes (sous la forme de composants) de différents niveaux d'abstraction et de différents niveaux de granularité (Rolland *et al.*, 1996). L'abstraction est constituée de deux niveaux : conceptuel et technique. Selon (Brinkkemper, 1998), le niveau conceptuel comporte des fragments de méthodes de conception tandis que le niveau technique propose des fragments qui représentent des spécifications implémentables de parties opérationnelles de méthodes, c'est-à-dire des outils.

2.1.2. *La construction d'une méthode*

L'ingénierie de méthodes est basée sur la réutilisation des méthodes existantes. L'existence d'une base de connaissances méthodologiques fait émerger une large panoplie de possibilités pour construire d'autres méthodes. Un de ses objectifs est d'assembler des composants de méthodes et de construire des outils de support pour la réingénierie de méthodes existantes (Harmsen *et al.*, 1994).

Les approches de construction de méthodes par assemblage de fragments sont basées sur le regroupement de fragments de méthodes disjoints et complémentaires (Brinkkemper, 1998). L'existence d'une multitude de méthodes montre que chaque méthode a des avantages et des inconvénients relatifs au domaine du problème (domaine d'application, taille des projets, etc). De plus, l'expérience sur l'usage des méthodes montre que les concepteurs de systèmes adaptent et modifient les méthodes en fonction de la situation et de leurs préférences personnelles. Les concepteurs peuvent avoir besoin de créer une nouvelle méthode à partir de rien, de modifier (c'est-à-dire : améliorer, compléter ou adapter) une méthode existante ou de réutiliser les parties de diverses méthodes et de les assembler pour créer une nouvelle méthode.

Les méthodes sont décomposées en fragments de méthodes et postérieurement ces fragments sont stockés dans la base de méthodes. Les méthodes une fois décomposées et stockées, peuvent être utilisées dans les projets. Donc, les concepteurs de systèmes doivent trouver dans la base de méthodes des fragments qui correspondent le mieux aux spécificités du projet. Le principe d'extraction/sélection définit la première étape pour identifier un fragment approprié. L'étape suivante consiste à construire une nouvelle méthode à partir des fragments choisis.

2.2. Des méthodes vers les services

Au cours des dix dernières années, les approches d'ingénierie de méthodes ont largement adopté une vision modulaire d'une méthode afin d'appliquer des approches orientée composants.

Plus récemment, des approches à base de services ont été proposées. L'approche **SO2M** de Guzélien (Guzélien, 2007) apparaît comme la première tentative pour appliquer une approche orientée services dans le contexte de l'ingénierie de méthodes. **SO2M** considère les "***method chunks***" comme des services « méthode ». **SO2M** est définie comme un ensemble de services « méthode » auxquels est associé un processus de composition pour assurer la recherche, la sélection et la composition de services. Les services méthodes sont des unités réutilisables qui proposent un ou plusieurs fragments de démarches pour résoudre des problèmes de développement de systèmes d'information. Ils permettent ainsi de répondre aux besoins de capitalisation et de réutilisation des connaissances en ingénierie des SI.

L'approche **MaaS** propose par Rolland (Rolland, 2008) consiste à concevoir une méthode comme un service. L'approche vise à adapter les technologies des services Web aux besoins de l'ingénierie de méthodes. Dans cette perspective, une méthode devient des services méthode qui sont mises en œuvre sous la forme des services web. Il s'agit d'offrir une solution pour réorganiser les méthodes dans un portefeuille de services méthodes qui sont distribués via le réseau Internet.

2.3. Positionnement de notre approche

Cette brève étude bibliographique a permis de constater que l'ingénierie de méthodes offre des concepts et des techniques pour construire des processus spécifiques aux projets par l'assemblage/composition de fragments de méthodes. Parmi la littérature étudiée, nous distinguons plus particulièrement l'approche **SO2M**, plus représentative de nos objectifs, qui utilise la notion de services pour construire des méthodes adaptées aux besoins des concepteurs/développeurs. Par contre, cette approche ne traite pas de l'implémentation de services « méthode » considérée par l'approche MaaS. Ces deux approches n'offrent aucun mécanisme de support pour construire des environnements de modélisation pour les concepteurs/développeurs qui utilisent les services méthodes. Pour ce faire, nous nous basons sur la notion des services méthodes comme source d'inspiration de notre approche.

La section suivante introduit un cas d'étude qui servira à illustrer notre approche.

3. Cas d'étude

L'essence de notre étude vise le choix de processus et d'outils de modélisation appropriés. Pour illustrer notre approche, nous nous appuierons sur une méthode de conception de systèmes interactifs (Godet-bar et al., 2007) utilisée dans des projets de recherche de notre laboratoire. Cette méthode est une extension de la démarche de conception Symphony (Hassine *et al.*, 2002), qui intègre les pratiques de conception de l'Interaction Homme-Machine (IHM) et un processus de développement du génie logiciel. Symphony est une démarche inspirée des pratiques

de l'Unified Process : la démarche est itérative, incrémentale, orientée composants métier et pilotée par les cas d'utilisation. Elle a été initialement proposée par la société UMANIS. Cependant, Symphony a été étendue récemment pour intégrer la conception de systèmes Réalité Augmentée (Godet-bar et al., 2007).

Cette nouvelle méthode prend en compte les aspects fonctionnels (méthodes de GL pour le développement du noyau fonctionnel) et l'étude de l'interaction de l'utilisateur. La conception des fonctionnalités est effectuée par les « Spécialistes GL » et les « Spécialistes Métier » qui utilisent des modèles UML. Concernant les aspects méthodologiques en IHM, Symphony utilise une approche basée sur la modélisation des tâches et des diagrammes spécifiques à la description des dispositifs d'interaction. Les concepteurs sont les « Spécialistes IHM » et les « Ergonomes logiciels ».

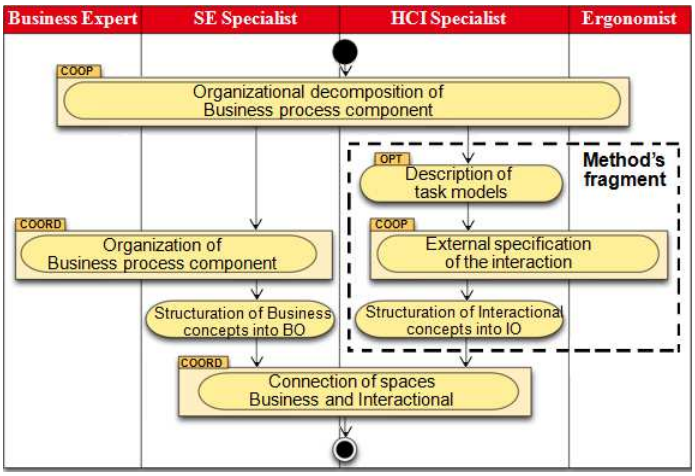

Figure 1. Phase de spécification organisationnelle et interactionnelle des besoins

Dans cet article, nous nous concentrons sur la phase de "***Spécification Organisationnelle et Interactionnelle des Besoins***" (représentée par la figure 1). Cette phase a pour objectif d'analyser un processus métier à un niveau organisationnel (le « qui fait quoi et quand » du futur système), et d'envisager une interaction satisfaisante pour réaliser les activités des différents acteurs. Elle débute par une activité de coopération entre tous les spécialistes qui a pour but de décomposer le processus métier étudié en sous-processus (« Décomposition organisationnelle des processus métier composants »). Cette décomposition accomplie, les spécialistes GL et métier organisent les sous-processus puis identifient et structurent le métier sous forme d'objets métier nommés BO pour Business Objects. En parallèle, des activités spécifiques aux aspects IHM sont réalisées par le spécialiste IHM et l'ergonome : "***Description des modèles de***

tâches" pour spécifier les tâches de l'utilisateur et du système, "***Spécification externe de l'interaction***" pour définir l'interface homme-machine et finalement "***Structuration des concepts interactionnels en Objets Interactionnels*** » pour structurer les concepts pour l'interaction. La phase se termine par une activité de coordination entre spécialiste GL et IHM qui consiste à faire le lien entre les objets métier et les objets interactionnels. Nous utiliserons cet exemple dans le reste de cet article pour illustrer notre approche.

4. Approche générale

4.1. Introduction

Dans notre approche, la modélisation de services permet de structurer l'ensemble des connaissances nécessaires pour décrire les besoins des concepteurs de modèles, les méthodes adaptées à ces besoins et les outils supports de ces méthodes. Donc, afin de faciliter et d'automatiser les étapes de développement de logiciels utilisant des modèles (par exemple : l'édition du modèle, la transformation du modèle, la vérification de la cohérence, …), plusieurs services de modélisation sont nécessaires. Les services de modélisation sont les opérations automatisées qui peuvent être appliquées sur les modèles (Blanc X., 2005). Les utilisateurs des services de modélisation sont donc des concepteurs de modèles qui souhaitent appliquer des opérations automatisées sur leurs modèles afin de réaliser des activités de développement de logiciels. Cette section propose les concepts de notre approche orientée services pour la gestion de modèles.

4.2. La notion de service

L'ingénierie dirigée par les services (Papazoglou M-P., 2003) est un paradigme qui utilise des services comme éléments fondamentaux dans le développement d'applications. Cette approche accélère le développement d'applications en permettant la composition de services distribués (Papazoglou M-P., 2005). Un service est un ensemble de modules logiciels autonomes qui sont décrits, publiés, découverts, composés et négociés à la demande d'un client.

Les services exécutent des fonctions qui peuvent être aussi bien de simples requêtes dans un formulaire, que des processus métiers complexes (Papazoglou M-P, 2005). Pour soutenir l'intégration des applications basées sur différents processus métier, la modélisation de services est supportée par l'architecture orientée service (SOA) (Papazoglou M-P, 2005). L'objectif de SOA est de fournir des services aux utilisateurs finaux ou applications. Cette approche définit une interaction entre agents logiciels comme un échange de messages entre clients et fournisseurs. Les agents peuvent être simultanément des clients et des fournisseurs. Les fournisseurs sont des agents logiciels qui offrent les services, des organismes qui procurent la mise en œuvre du service, des descriptions techniques et un support métier. Ils ont

pour responsabilité de publier une description du service(s) qu'ils offrent. Les clients sont les agents logiciels qui demandent l'exécution d'un service.

4.3. Trois niveaux de services

Notre approche s'appuie sur trois niveaux de modélisation (voir figure 3) dont les fournisseurs, les clients et les services sont différents.

Le premier niveau correspond à **la couche opérationnelle**. Cette couche permet de définir l'infrastructure de modélisation d'un concepteur de modèles. Il s'agit d'offrir des services aux concepteurs de modèles (« models designers »), pour faciliter la création de son environnement de modélisation. Le client est donc un concepteur de modèles qui désire gérer des modèles de manière individuelle ou collaborative (avec d'autres concepteurs). Il a besoin de définir un environnement de modélisation offrant des outils adaptés à ses préférences en termes de gestion de modèles. Par exemple, un « utilisateur lambda » a besoin d'un environnement de modélisation pour l'édition de modèles UML et la transformation de ces modèles dans un autre langage de programmation ou de modélisation.

La couche organisationnelle est inspirée des travaux de Ralyté (Ralyté, 1999) qui a adopté les idées de l'Ingénierie des Méthodes Situationnelles (Kumar et al., 1992). Dans notre travail, nous utilisons la notion de service pour soutenir la construction de processus de conception de systèmes en assemblant des fragments de méthodes. Il s'agit d'offrir un support à base de services à des groupes de projet. Dans cette couche, les rôles et les activités sont exprimés sous la forme de processus de développement simplifiés. L'objectif est de capitaliser des fragments de méthodes dans le but de fournir à chaque concepteur, donc chaque rôle du groupe projet, l'environnement de modélisation qui lui convient. Un fragment de méthode est modélisé par un ensemble d'activités réalisées par des rôles. Les activités sont décrites en termes d'actions sur des modèles.

La **couche organisationnelle** permet de réutiliser de façon coordonnée les services opérationnels. Les clients sont, dans ce cas, les chefs de projet (« projets manager ») cherchant à définir et à administrer des rôles et des activités sous la forme de processus de développement. Les chefs de projet vont choisir des services organisationnels (parties de processus de conception) qui nécessitent la mise en place de services opérationnels de gestion de modèles. Ainsi ils définissent la création des environnements de gestion de modèles pour les concepteurs impliqués dans leurs processus de développement.

La couche intentionnelle permet la modélisation des buts. Il s'agit de conceptualiser des besoins stratégiques de modélisation requis par un sujet individuel, un groupe d'individus, une unité de travail ou toute organisation qui participe au processus de développement de systèmes. Cette couche permet de réutiliser les services organisationnels. Le fournisseur correspond à l'ingénieur de l'environnement (« environment engineer »). Il s'agit d'un nouveau métier qui

s'occupe de l'administration et la gestion de la plateforme. Les clients sont ceux des couches organisationnelle et opérationnelle, c'est-à-dire les concepteurs de modèles (« model designer ») et les chefs de projets (« project manager »). Pour ces clients, les services sont les buts proposés par l'ingénieur de l'environnement qui peuvent être vus comme les buts à atteindre pour la gestion de modèles, par exemple : « spécifier un Système Interactif ».

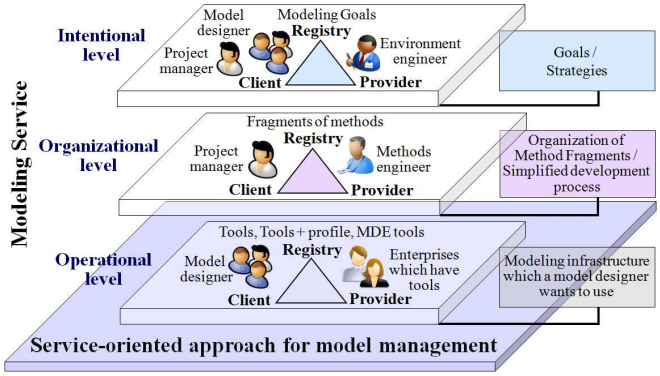

Figure 3. Trois niveaux de service

Nous faisons l'hypothèse d'une mixité entre ceux qui conçoivent les environnements et ceux qui les utilisent. Ce haut niveau de maturité entre client et fournisseur peut néanmoins être réduit par l'alignement des vocabulaires c'est-à-dire par l'utilisation d'un vocabulaire contraint et commun spécifié sous forme d'ontologies.

4.4. Utilisation des ontologies

Pour faciliter le choix de services adaptés, il est nécessaire que les services aient une sémantique riche. Pour répondre à ce besoin, les ontologies sont aujourd'hui largement utilisées. Elles définissent une terminologie pour partager un vocabulaire commun au sein d'un domaine. Nous pouvons noter en particulier, OWL-S (OWL-S, 2004) qui permet une description des services enrichie notamment par un modèle de processus exprimant comment le service peut être utilisé. Si cette solution est intéressante, nous la voyons comme une solution technique qui pourrait contraindre notre solution conceptuelle (nous tenons à avoir 3 niveaux conceptuellement distincts). Aussi nous retiendrons de ces travaux l'intérêt d'utiliser des ontologies pour décrire les services.

Dans le contexte de la gestion de modèles, les ontologies (buts, produits, processus et rôles) proposées par Guzélian (Guzélian, 2007) définissent une terminologie réutilisable et partageable par les fournisseurs des services « méthode »

(ingénieurs de méthodes) et par les clients qui les utilisent (concepteurs/développeurs de SI). Les ontologies permettent aussi d'enrichir la sémantique de la description des services. Ce niveau sémantique joue un rôle essentiel dans l'automatisation de la recherche/sélection, dans l'adaptation et dans la composition de services. Mais dans ces travaux, modèles de services et ontologies sont dissociés, l'ontologie servant à « typer » des éléments du modèle.

Pour notre part, nous avons choisi d'intégrer les ontologies dans nos modèles de services de manière à offrir une vision intégrée de chaque niveau d'abstraction. Les ontologies généralement intégrées sous la forme d'une adaptation du patron Item-Description auquel nous avons rajouté deux classes énumérées. Nous appellerons ce patron : Patron « **Concept – Term** » (Fig. 4).

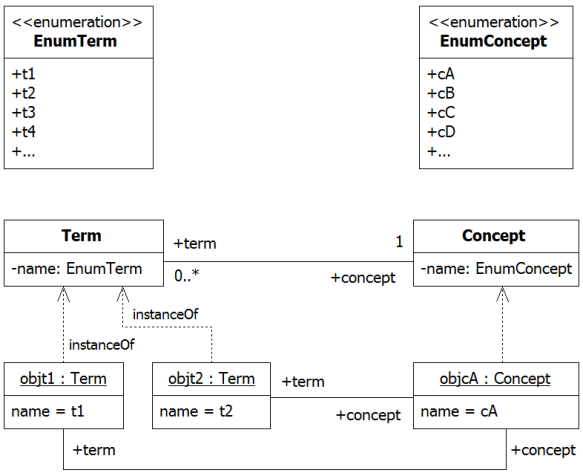

Figure 4. Patron « Concept-Term »

Nous avons utilisé ce patron Concept-Term pour les ontologies de verbes d'intention, des produits de modélisation, d'acteurs, de processus et de facteurs de qualité logicielle. La modélisation des verbes est particulièrement intéressante. En se basant sur la théorie des actes de langage, un service intentionnel peut être considéré comme un acte de langage c'est-à-dire que c'est un moyen mis en œuvre par un locuteur pour agir sur son environnement. Ainsi, le verbe d'une intention peut être considéré comme performatif (Austin, 1962) puisqu'il accomplit une action. Néanmoins, comme il nous est difficile de catégoriser nos verbes performatifs suivant les travaux d'Austin et Searle, nous avons choisi de réaliser une ontologie de verbes suivant les travaux spécifiques en gestion de modèles.

Notre modélisation des verbes utilise les ontologies de buts (Guzélian, 2007) qui décrivent les intentions spécifiques pour la gestion de modèles. D'un côté, nous souhaitons identifier des catégories des verbes (instances de la classe ConceptVerb)

selon ses propriétés communes et de l'autre, des verbes qui seront utilisés pour la modélisation des intentions (instances de la classe TermVerb). Les verbes correspondant au même type d'activité de développement appartiennent à la même catégorie de verbes. Donc, les verbes et leur catégorisation sont utilisés, dans le modèle de la couche intentionnelle, pour exprimer les intentions des concepteurs de modèles.

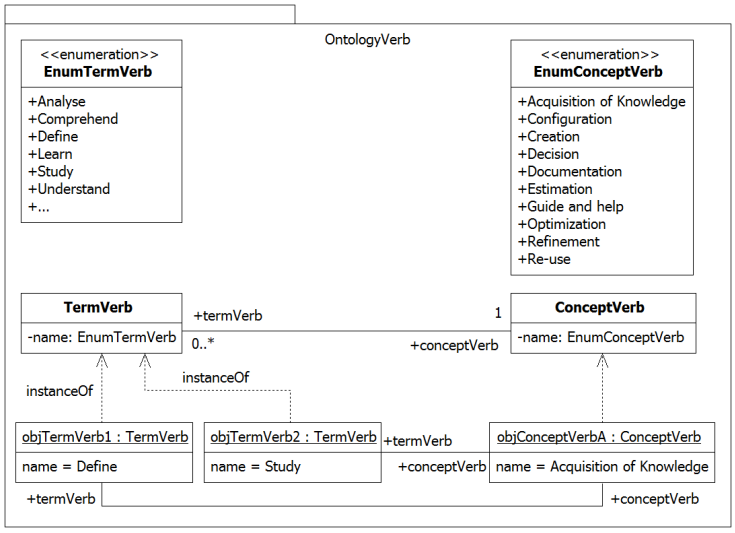

Figure 5. Imitation du patron Concept-Term pour l'ontologie des verbes

Chaque concept de verbe d'ontologie est représenté par (Fig. 5):

- Une valeur du type énuméré « **EnumConceptVerb** », par exemple : Acquisition of Knowledge.
- Une instance de la classe « **ConceptVerb** » dont l'attribut « **name** » est valué par une valeur du type énuméré « **EnumConceptVerb**

Chaque terme d'ontologie, (donc le verbe) est représenté par **:**

- Une valeur du type énuméré « **EnumTermVerb** », par exemple : Define.
- Une instance de la classe « **TermVerb** » dont l'attribut « **name** » est valué par une valeur du type énuméré « **EnumTermVerb** »..

Chaque instance de la classe ConceptVerb conserve une référence vers ses instances de la classe TermVerb via le rôle termVerb. Inversement chaque instance de la classe TermVerb pointe vers ConceptVerb.

Nous verrons par la suite comment cette ontologie et celles des produits de modélisation, des acteurs, des processus et des facteurs de qualité logicielle sont utilisées dans nos modèles de services.

5. La couche intentionnelle

5.1. Modélisation d'un service intentionnel

Un service intentionnel est un service orienté métier décrit d'un point de vue intentionnel (par exemple : spécifier un système interactif, identifier les cas d'utilisation, élaborer un scénario, personnaliser le processus de spécification de besoins…). Il correspond à la modélisation des buts de gestion de modèles. Il peut être composé d'autres services intentionnels plus élémentaires (cf. figure 6) et est réalisé par des services organisationnels correspondant à des fragments de méthodes.

Un service intentionnel est défini par **un verbe**. Un service intentionnel est également associé aux **produits** utiles à la réalisation du service (association « comprise ») et aux produits générés par le service (association « reach »). De plus, un service intentionnel est lié à une façon de faire, une manière, pour réaliser l'intention.

L'élément central d'un service intentionnel est le **verbe,** qui est représenté par la classe « TermVerb » de l'ontologie de verbes. Les verbes décrivant les intentions spécifiques pour la gestion de modèles, par exemples : **améliorer** un processus métier, **personnaliser** le processus de spécification de besoins, **élaborer** un scenario, **automatiser** une procédure administrative, sont donc des instances de « TermVerb ». Ils sont rattachés à leur catégorie (instance de « ConceptVerb »).

Le produit est l'élément qui complète le verbe pour décrire l'intention, cet élément est représenté par la classe « TermProduct » de l'ontologie de produits « OntologyProduct » qui regroupe les différents objets qui peuvent être élaborés, modifiés ou utilisés au cours de la gestion de modèles comme scénario ou processus métier. Ce sont les objets sur lesquels portent les actions. Ils sont instance de « TermProduct » et sont rattachés à leur catégorie (instance de « ConceptProduct ») : par exemple, modèle, méta-modèle, documentation. Dans la figure 6 seule la classe TermProduct apparait.

La manière est une façon complémentaire d'exprimer l'intention du verbe, c'est-à-dire l'élément qui permet de décrire une approche possible pour réaliser le service intentionnel. Par exemple, dans le but « spécifier les besoins avec l'utilisation des scénarios », la phrase « avec l'utilisation des scénarios » correspond à la manière de procéder pour résoudre le but atteindre.

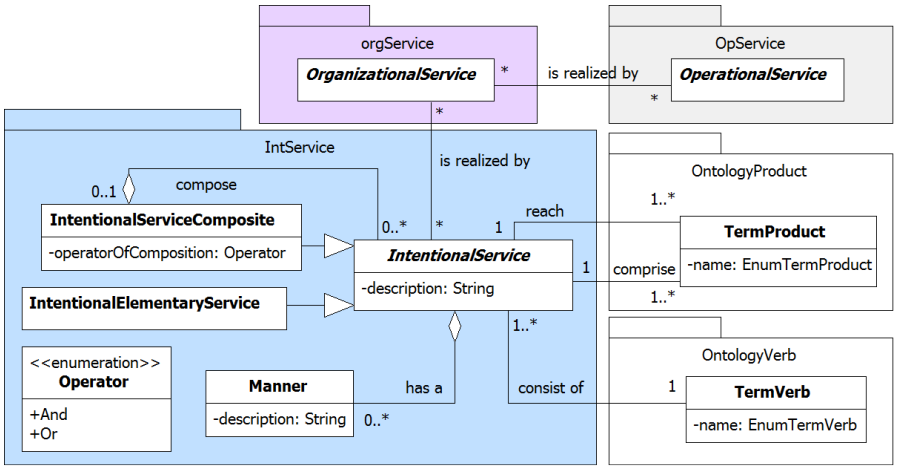

Figure 6. Modèle de services intentionnels

5.2. La construction lexicale des intentions

Nous utilisons une approche linguistique pour formuler une intention. Cette approche s'appuie sur la déclaration structurelle d'une intention proposée par Rolland (Rolland, 2007). Ces travaux sur les intentions ont déjà été utilisés pour les services web (Lee, 2005). Dans ce travail, un but est constitué de contenus (action, contraintes et paramètres tels que l'objet, le résultat, la source, la destination), de propriétés (compétences, vues, pré et post-condition) et d'un plan. De façon similaire, nous avons retenu dans notre travail les éléments constitutifs du contenu. La déclaration structurelle reprend les éléments du modèle de service intentionnel (Fig.6).

Intention<
<TermVerb(name)>
{<ObjectComprise(name)>} -- { } element repetitive
{<ObjectReach(name)>}
[*Manner*] -- [] element facultative
>

Le tableau 1 propose deux exemples.

Exemple 1 :	
Intention	Spécifier les fonctionnalités d'un système interactif
Forme générale	Intention< <TermVerb(Spécifier)> <ObjectComprise(les fonctionnalités)> <ObjectReach(Cartographie des processus métier d'un système interactif)>

	>
Exemple 2 :	
Intention	Spécifier les besoins d'un système avec l'utilisation des scénarios
Forme générale	Intention< <TermVerb(Spécifier)> <ObjectComprise(les besoins d'un système)> <ObjectReach(Diagramme de scénarios)> [*Manner*(avec l'utilisation des scénarios)] >

Table 1. Exemples de la structure d'une Intention

5.3. Exemple

Notre exemple est basé sur la méthode de conception présentée dans la section 3. Au niveau intentionnel, l'approche que nous proposons doit garantir un environnement de modélisation des buts stratégiques requis par les spécialistes en GL et les spécialistes de l'IHM, qui participent au processus de développement de systèmes interactifs. Les services intentionnels sont représentés sous la forme de graphes et/ou dont les nœuds sont structurés suivant la syntaxe définie dans la section 5.2.

La représentation de la figure 7 montre une composition des buts à atteindre pour le but principal de spécification d'un système interactif. Dans la figure ci-dessus, les rectangles correspondent aux services et les liens décrivent la composition des services intentionnels. Par exemple : le service "*Spécifier un Système Interactif*" est décomposé par les services : "*Spécifier les aspects fonctionnels*", "*Spécifier les aspects Interactionnels*" et "*Spécifier l'architecture logicielle*".

De la même façon, le service "Spécifier les aspects Interactionnels" peut être décomposé par les services : "Concevoir les interfaces utilisateurs", "Spécifier le processus d'interaction" et "Concrétiser l'interface utilisateurs". Tous les services peuvent être liés à des services organisationnels pour proposer une solution (en termes de processus) aux buts correspondants.

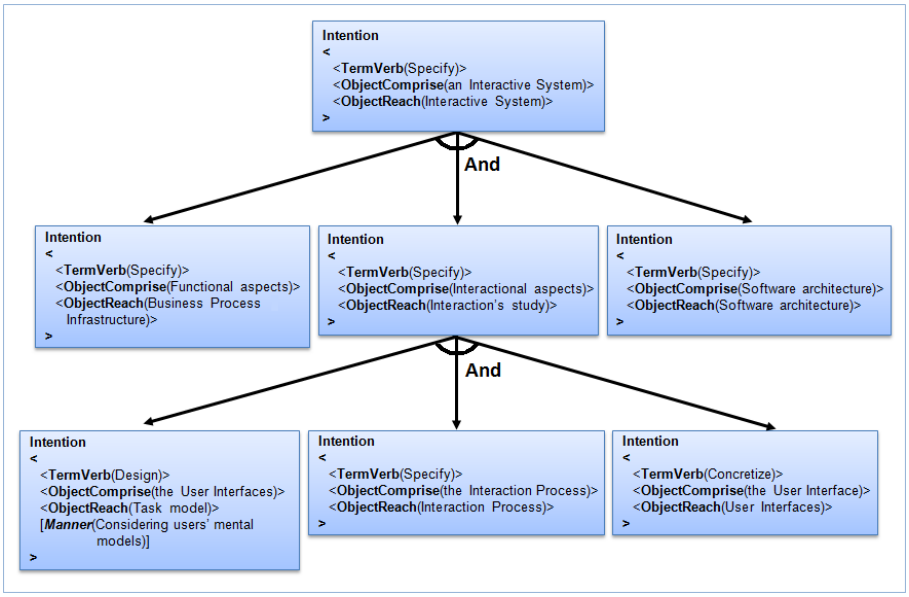

Figure 7. Modèle de service intentionnel

6. La couche organisationnelle

6.1. Modélisation d'un service organisationnel

Un service organisationnel consiste en une composition de fragments de méthodes qui peuvent être réutilisés par des concepteurs de modèles, pour répondre aux buts des services intentionnels. Donc, un service organisationnel complexe est composé par d'autres services organisationnels. La composition d'un service organisationnel est représentée dans le paquetage « orgService ». Comme dans les méthodes situationnelles, il est difficile de composer des fragments de processus. Aussi la pertinence d'une composition de service est laissée à l'appréciation de l'ingénieur de méthodes. Néanmoins la composition est facilitée par l'utilisation de modèles de processus identiques et par un vocabulaire commun défini dans nos ontologies.

Dans notre modèle (Fig. 8), un fragment de méthode est représenté par un service élémentaire organisationnel défini en termes de manipulations de modèles. **La manipulation des produits** consiste en la description de chaque service élémentaire sous la forme d'actions sur les produits utilisés ou produits : par exemple, un processus de spécification interactionnelle des besoins, produit un modèle de tâche de l'utilisateur qui peut être utilisé par d'autres fragments de méthode. Nous utilisons ici l'ontologie des produits (« OntologyProduct »).

Un service organisationnel est effectué par un ou plusieurs rôles. Les acteurs correspondent aux rôles responsables de la réalisation d'un service organisationnel. Pour ce faire, nous utilisons l'ontologie des acteurs « OntologyActor » qui définit un ensemble de rôles (concepteur, développeur, architecte, etc.) pour les acteurs qui ont en charge les problèmes d'ingénierie (Guzélian, 2007).

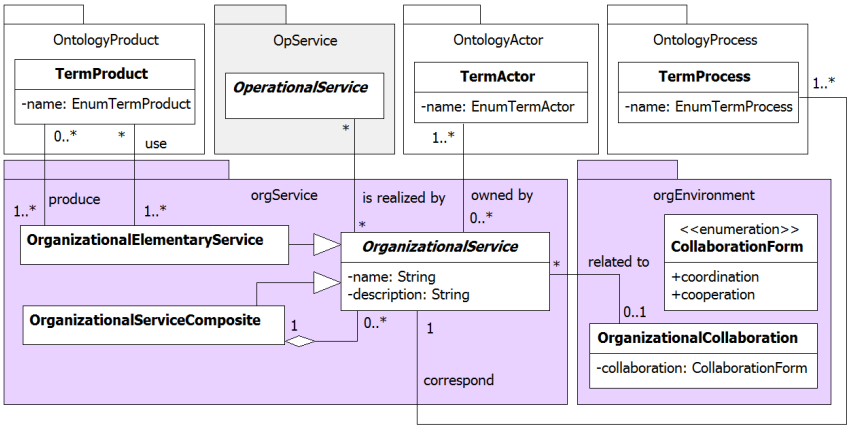

Figure 8. Modèle de services Organisationnels

Un concepteur de modèles peut définir et réutiliser plusieurs services organisationnels. A ce niveau, le terme de collaboration (Blanco *et al.*, 2007) est utilisé pour représenter les tâches de coordination et de coopération entre les concepteurs. La collaboration prend en compte la cohérence et la synchronisation des activités entre les différentes spécialités intervenant dans un processus de conception de systèmes interactifs. Les activités de **coordination** reposent sur une décomposition du travail en activités de buts similaires, alors que les activités de **coopération** sont basées sur un objectif de modélisation commun. Chaque spécialiste apporte ses modèles et la coopération permet de produire des modèles consensuels ou communs.

Un autre aspect considéré par notre modèle organisationnel est le fait que les services opérationnels sont supportés par les services organisationnels. Cela signifie que les services organisationnels utilisent les services opérationnels pour supporter la gestion des activités de modélisation. Par exemple, un service organisationnel qui comprend la description du modèle de tâches peut être lié aux services opérationnels qui offrent toutes les fonctionnalités pour faciliter l'édition des arbres de tâches.

6.2. Exemple

Comme nous l'avons souligné dans la section précédente, nous considérons qu'un service intentionnel peut être réalisé par plusieurs services organisationnels. Donc, il convient de s'appuyer sur les processus et les méthodes existants. Le service intentionnel "*Spécifier les aspects interactionnels*" d'un système interactif peut être réalisé par une partie de la phase "Spécification organisationnelle et interactionnelle des besoins" de la méthode de conception Symhony décrite dans la section 3. Ce lien entre service intentionnel et service opérationnel est réalisé statiquement par un ingénieur de méthodes car il nous semble difficilement informatisable. Ainsi la pertinence de fragments de processus pour un but est garantie par les connaissances d'un spécialiste.

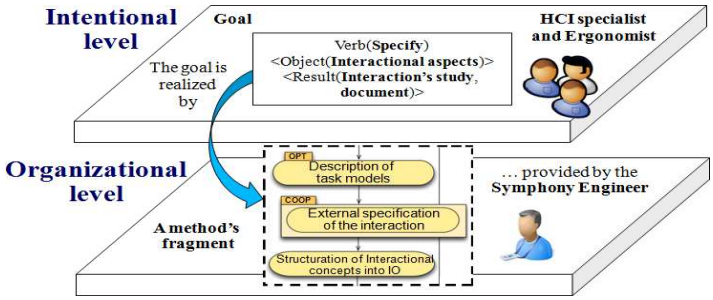

Figure 9. Modèle de service organisationnel

Pour être intégré dans notre environnement de gestion de modèles, le fragment de méthode (encadré de la figure 1) doit être défini en termes de services organisationnels. Ce fragment constitue un service organisationnel composé de trois sous-services (un par activité). Le service organisationnel composite est lié au niveau intentionnel au but "*Spécifier les aspects Interactionnels*" (figure 9).

Les services élémentaires sont décrits sous la forme d'actions sur les modèles utilisés ou produits. On peut noter aussi d'autres informations telles que le type de collaboration. Ainsi l'activité de « Spécification externe de l'interaction » est caractérisée par un service qui a la particularité de nécessiter une coopération entre le spécialiste IHM et l'ergonome. Cette propriété du service est importante et doit permettre de ne le choisir que dans les conditions adéquates.

Un autre exemple est l'activité de description des modèles de tâches qui est exprimée au niveau organisationnel comme un service d'édition de modèles de tâches. Nous verrons dans la section 7.2 qu'un service organisationnel peut être supporté pour plusieurs outils définis en termes de services opérationnels.

Il est évidement possible de lier les buts intentionnels à d'autres processus existants. Un des autres processus possibles qui répond au but "*Spécifier les aspects*

Interactionnels" est le processus d'élaboration de la méthode RUP proposé par (Jacobson *et al.*, 1999). Dans ce processus, les concepteurs de modèles analysent et décrivent les exigences non fonctionnelles de l'utilisateur et conçoivent l'interface de l'utilisateur. Dans notre approche, ces activités correspondent à deux services organisationnels. L'un pour décrire les interfaces et l'autre pour concevoir les interfaces.

7. La couche opérationnelle

7.1. Modélisation d'un service opérationnel

Cette section présente le modèle de la couche opérationnelle (Fig. 10). L'objectif de ce modèle est de décrire la structure statique de l'application qui soutiendra la recherche et le couplage des services qui sont requis par la gestion de modèles.

Un **service opérationnel** (voir le paquetage « **opService** » dans la figure 10) correspond à une application exécutable composée d'autres services de gestion de modèles. Un service de gestion de modèles est un outil de gestion de modèles offert par un fournisseur. Ces services peuvent être fournis par des outils de gestion de modèles (par exemple : un AGL, un outil de modélisation ou un outil de transformation de modèles, etc.).

Les services opérationnels peuvent porter sur un ou plusieurs modèles (par exemple : le modèle des arbres de tâches, les modèle de classes, le modèle de cas d'utilisation, etc.). De la même façon que dans les couches intentionnelle et organisationnelle, nous avons utilisé ici l'ontologie des produits « **OntologyProduct** ».

Un service opérationnel fournit plusieurs fonctionnalités (paquetage « **opFunctionalityFactor** » dans la figure 10). La fonctionnalité d'un service opérationnel définit la façon dont un concepteur de modèles pourra effectuer les opérations sur les modèles (par exemple, texte, graphique ou les deux), les langages supportés pour stocker l'information (par exemple, XMI, etc.) et les éventuelles opérations de service (par exemple, la création, l'édition, suppression, etc.).

L'étude des différents outils de gestion de modèles a mis en évidence l'ensemble des fonctionnalités offertes par les outils. Ces fonctionnalités sont : la gestion de modèles, la transformation de modèles, la vérification de la cohérence, la simulation et les tests.

Pour supporter les besoins de création d'environnements collaboratifs, un service opérationnel est lié à plusieurs formes de collaboration. Une **forme de collaboration** est caractérisée suivant le modèle de Denver de Salvador (Salvador, 1996) et sur la classification Espace-Temps proposée par Ellis (Ellis, 1991). Le modèle de Denver établit que les utilisateurs peuvent être proches ou éloignés et interagir de façon **synchrone** (temps réel) ou **asynchrone**. La classification Espace-

Temps proposée par Ellis repose sur deux caractéristiques, à savoir **où et quand une action est exécutée** par un des utilisateurs par rapport aux autres utilisateurs. Donc, dans le contexte de notre travail, une collaboration est définie en fonction de deux axes : l'espace (« space »), et le temps (« time »).

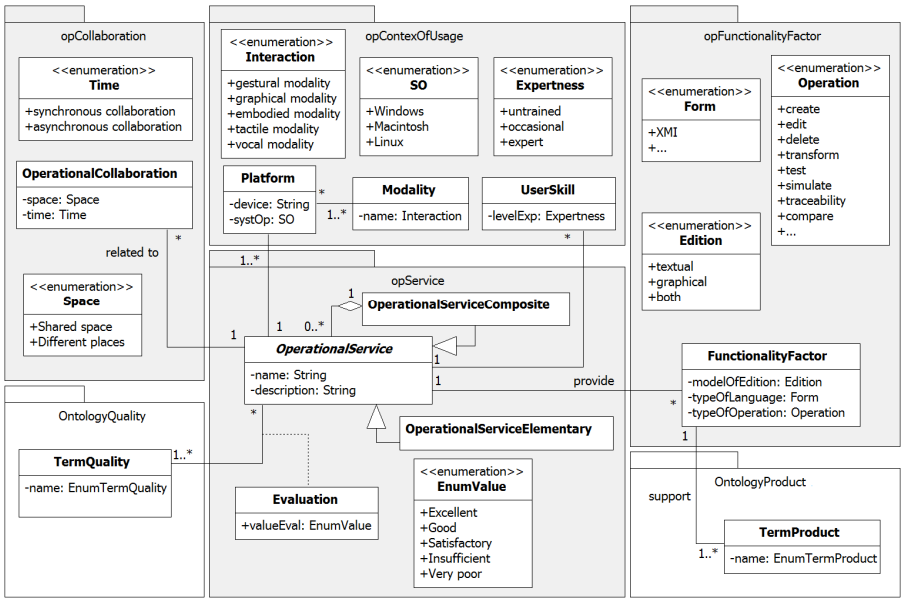

Figure 10. Modèle de service Opérationnel

En ce qui concerne le **profil de l'utilisateur** et le **contexte d'utilisation** d'un service opérationnel, un service opérationnel peut être exécuté dans différents contextes d'utilisation par des spécialistes qui ont divers niveaux d'expertise. Le contexte d'utilisation est modélisé par le profil de l'utilisateur, la plateforme et la modalité d'interaction. Un service opérationnel est caractérisé par **le niveau d'expertise** nécessaire pour l'utilisateur. Nous considérons, dans nos travaux, les niveaux d'expertise suivants : expert, occasionnel et novice. Le **contexte d'utilisation** est un espace d'information structuré qui inclut les plateformes logicielle et matérielle exigées par le service. Il est relié à une variété de modalités d'interaction. La modalité d'interaction est inspirée du terme « **multimodalité** » dans lequel l'activité de l'utilisateur implique simultanément multiples modalités d'interaction. Par exemple, nous considérons ici la possibilité d'avoir des services de gestion de modèles en mode graphique, gestuelle, tactile ou vocale.

Pour garantir la qualité dans l'utilisation d'un outil de gestion de modèles, nous avons inclus dans notre approche la notion de qualité logicielle. La qualité logicielle est une appréciation globale d'un logiciel basée sur de nombreux indicateurs (par

exemple : l'interopérabilité, la maintenance, …). Les définitions de qualité logicielle sont généralement axées sur l'excellence. Nous avons intégré dans notre modèle opérationnel l'ontologie des facteurs de qualité « OntologyQuality » modélisée par imitation du patron Concept-Term. Dans la figure 10, seule la classe TermQuality est représentée. Cette ontologie a été établie à partir du standard IEEE et de l'ISO sur la qualité logicielle.

Chaque critère de qualité est apprécié à l'aide d'une évaluation. Cette évaluation peut être exprimée comme une valeur d'excellence. **La valeur d'excellence** est évaluée à l'aide d'une classe énumération appelée « **EnumValue** ».

7.2. Exemple

Comme nous l'avons souligné précédemment un service organisationnel peut être réalisé par plusieurs services opérationnels. Donc, il convient de s'appuyer sur les outils de modélisation existants. Par rapport à l'action : "*description de modèles de tâches*", le spécialiste IHM, responsable de l'exécution de l'action, a des compétences spécifiques et des besoins. Cette action qui est représentée sous forme de service organisationnel dans notre approche, est liée à différents services opérationnels, correspondant à des outils pour aider la réalisation de la modélisation des tâches. Ces outils sont des éditeurs de modèles de tâches tels que **CTTE**, **TERESA**, **Tamot**, **K-MADe**.

Pour choisir parmi eux, les caractéristiques des services opérationnels sont utilisées. Ici le spécialiste IHM a généralement un Mac, est habitué à travailler avec des modèles graphiques et collabore avec d'autres concepteurs de modèles (ergonomes et spécialistes GL). Il a besoin d'un environnement qui supporte la collaboration asynchrone et permet la réalisation de la tâche de manière distribuée. Il est donc important que 1) l'outil tourne sur plusieurs plateformes dont Mac ; 2) présente les modèles sous forme graphique ; 3) permette la collaboration asynchrone et distribuée. En considérant ces besoins, l'outil CTTE (Paterno, 1997) sera sélectionné.

8. Plateforme de support

Actuellement nous implémentons un prototype expérimental, que nous présentons dans cette section. Divers solutions techniques étaient envisageables. Nous aurions pu, en particulier, nous baser sur les technologies web dont OWL-S. Mais nous avons voulu conserver trois niveaux de services distincts et ne pas nous limiter aux environnements web. Aussi notre solution est basée sur une plateforme de gestion de services « générique ». Nos services pourront ensuite évoqués des services dans différentes technologies dont OWL-S (Fig. 11). Le lien entre les services de modélisation et les services exécutables s'effectueront en particulier via les paramètres d'entrée et de sortie des services exécutables.

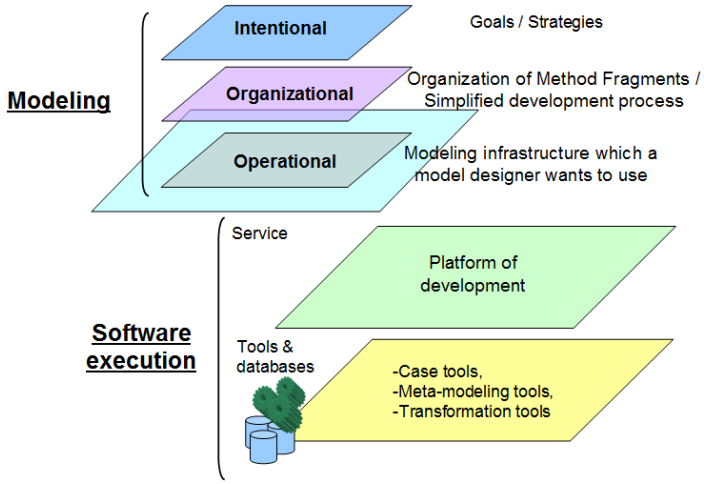

Figure 11. Plateforme de gestion de services « générique »

Le prototype doit garantir l'enregistrement, la consultation, la recherche et la conception de nos trois niveaux de services. A ce jour, le prototype est constitué de deux blocs indépendants mais complémentaires.

Le **premier bloc** (figure 12b) considère la mise en œuvre d'un registre de services avec l'atelier de gestion de composition de services « ChiSpace » (Yu *et al.*, 2008). L'objectif de cet environnement est de simplifier le travail des développeurs lors de la réalisation d'applications par composition de services. ChiSpace a été implémenté sur *"Eclipse Modeling Framework"* (Eclipse-EMF) et la technologie JET2 d'Eclipse. L'environnement est composé d'un ensemble d'éditeurs basés sur une perspective personnalisée d'Eclipse (Yu *et al.*, 2008). Le registre de services correspond à la base de données où sont stockées les descriptions des trois niveaux de services. Ces descriptions sont basées sur les trois modèles de services présentés dans les sections précédentes. Le **deuxième bloc** permet d'ajouter, d'afficher, de sélectionner et de valider les services qui sont stockés dans le registre de services. Il est basé sur *"Eclipse Rich Platform"* (Eclipse-RCP). Ainsi, Eclipse-RCP est utilisé pour développer les interfaces qui permettant d'utiliser les fonctionnalités de la plateforme pour les clients et les fournisseurs. La figure 12 (partie a) montre l'interface de recherche des services intentionnels. Les autres interfaces ne sont pas présentées dans ce document pour cause de place limitée.

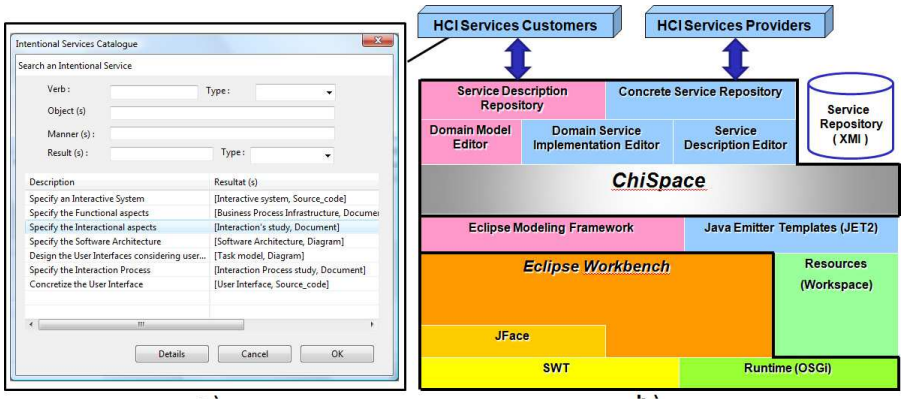

Figure 12 Plateforme Orientée Service

9. Conclusion et perspectives

Nous avons présenté dans cet article une approche de gestion de modèles orientée services facilitant le choix des processus et des environnements de modélisation en fonction des besoins des concepteurs de modèles. Nos propositions portent sur trois niveaux de services : le niveau intentionnel, le niveau organisationnel et le niveau opérationnel. Les trois niveaux offrent des services de gestion de modèles de natures différentes. Le niveau opérationnel permet de définir des environnements de modélisation adaptés aux concepteurs de modèles. Le niveau organisationnel garantit la réutilisation des services opérationnels d'une manière coordonnée, mais également, il garantit la création et la gestion des fragments de méthodes. Le niveau intentionnel correspond aux buts stratégiques des concepteurs de modèles, ces buts sont mis en œuvre par des démarches de conception décrites au niveau organisationnel.

Nous présentons également un aperçu de la plateforme en cours d'élaboration qui permettra la gestion de nos trois niveaux de service. Actuellement notre prototype permet d'ajouter et de rechercher des services intentionnels.

Les travaux à venir incluront la finalisation de la plateforme. Cette plateforme permettra de tester notre approche afin de valider nos propositions, d'analyser leur impact et leur utilisabilité. Pour ce faire, nous réaliserons des expériences auprès de concepteurs. Même si ce n'est qu'une première étape dans l'évaluation de notre solution, cela nous permettra également d'explorer d'autres fonctionnalités et de les intégrer dans notre solution.

Remerciements : Les auteurs remercient la Fondation « Gran Mariscal de Ayacucho, l'Université UCLA-Venezuela » pour leur soutien financier.

10. Bibliographie

Action IDM. - Ingénierie dirigée par les modèles. http://www.actionidm.org/, consultation Juin 2009.

Austin J.: How to Do Things with Words, Oxford, Oxford University Press, 1962.

Bieber G., Carpenter J., Introduction to Service-Oriented Programming (Rev 2.1), Avril 2001.

Blanco E., Grebici K., Rieu D.: A unified framework to manage information maturity in design process. In: International Journal of Product Development. Volume 4, Number 3-4, pp. 255--279, (2007);

Brinkkemper S., Saeki M., Harmsen F., Assembly Techniques for Method Engineering. Proc. of the 10th Conf. on Advanced Information Systems Engineering, Pisa Italy, 1998.

Blanc X., Gervais M.-P., Sriplakich P., « Modeling Services and Web Services: Application of ModelBus », *Proc. of SERP'05*, Las Vegas, Nevada, USA, June 2005, p. 557-563.

Crescenzo P., Mirbel I.: Improving Collaborations in Neuroscientist Community, Web2Touch workshop in conjunction with International Conference on Web Intelligence, Milan, Italy, Septembre, 2009.

Deneckère R., Iacovelli A. , Kornyshova E. , Souveyet C., « From Method Fragments to Method Services », In Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'08), Montpellier, France, 2008.

Ellis C., Gibbs S., Rein G. : Groupware : Some Issues and Experiences. Journal, Communications of the ACM (CACM), volume 34, Nro. 1, p. 38-58, ACM Press, 1991.

Godet-Bar G., Juras D., Dupuy-Chessa S., Rieu D., « Vers une méthode de conception de systèmes mixtes : Principes et mise en oeuvre », *RSTI-ISI*. Vol 12, 2007, p. 39-66.

Guzélian G., « Conception de systèmes d'information : une approche orienté services », Thèse de Doctorat soutenue à l'université Paul Cezanne, Marseille, juillet 2007.

Harmsen A.F., Brinkkemper S., Oei H., Situational Method Engineering for Information System Projects. Proc. of the IFIP WG8.1 Working Conference CRIS'94, pp. 169-194, North-Holland, Amsterdam, 1994.

Harmsen, A.F., Situational Method Engineering. In Moret Ernst and Young Management Consultants, Utrecht, Netherlands, 1997.

Hassine I., Rieu D., Bounaas F., Seghrouchni O., Symphony: a conceptual model based on business components, *in SMC'02*, IEEE International Conference on Systems, Man, and Cybernetics. Volume 2. (2002).

Jacobson, I., Booch, G., Rumbaugh, J. The Unified Software Development Process, Addison-Wesley, 1999.

Kumar K., Welke R.J., Method Engineering, A Proposal for Situation-specific Methodology Construction. In Systems Analysis and Design: A Research Agenda, Cotterman and Senn (eds), Wiley, pp.257-268, 1992.

Lee Chiung-Hon, Liu A.: Toward Intention Aware Semantic Web Service Systems. IEEE SCC'05, pp. 69-76, 2005.

Mirbel I., Ralyté J., Situational Method Engineering: Combining Assembly-Based and Roadmap-Driven Approaches. Requirements Engineering, 11(1), pp. 58–78, 2006.

OMG03, Object Management Group, « MDA Guide Version 1.0.1 », Juin 2003. http://www.omg.org/docs/omg/03-06-01.pdf, consultation Juin 2009.

OWL-S, W3C, « OWL-S : Sementic Markuo for Web Services », Novembre 2004. http://www.w3.org/Submission/OWL-S/, consultation Octobre 2009.

Papazoglou et al., Guest editor introduction: Service oriented computing ACM SIGSOFT 2003.

Papazoglou M-P., Traverso P., Dustdar S., Leymann F., « Service-Oriented Computing : A Research Roadmap », Dahstuhl Seminar 05462, pp. 1-29, 2005.

Paternò, « ConcurTask Tree : a diagrammatic notation for specifying task models », *Proc. of Interact'97*, Sydney, Australia, July 1997, p. 362-369.

Ralyté J., Backlund P., Kühn H., Jeusfeld M. (2006). Method Chunks for Interoperability. Proc. of the 25th Int. Conf. on Conceptual Modelling (ER'06). Tucson, Arizona, USA. LNCS 4215, Springer-verlag, pp. 339-353, November 6-9, 2006.

Ralyté J., Reusing Scenario Based Approaches in Requirement Engineering Methods: CREWS Method Base. Proc. of the 10th Int. Workshop on Database and Expert Systems Applications (DEXA'99), 1st Int. REP'99 Workshop, Florence, Italy, 1999.

Ralyté J., Rolland C., An Approach for Method Reengineering. Proc. of the 20th Int. Conf. on Conceptual Modeling (ER'01), Yokohama , Japan , November 2001. H. Kunii, S. Jajodia, A. Solvberg (Eds.), LNCS 2224, Springer-Verlag, pp.471-484 2001a.

Ralyté J., Rolland C., An Assembly Process Model for Method Engineering. Proc. of the 13th Conf. on Advanced Information Systems Engineering (CAISE'01), Interlaken, Switzerland, June 2001. K. Dittrich, A. Geppert, M. Norrie (Eds.). LNCS 2068, Springer-Verlag, pp. 267-283, 2001b.

Rolland C., Method Engineering: Towards Methods as Services", In International Conference on Software Process (ICSE-ICSP), Springer-Verlag, Leipzig, Germany, Mai 2008.

Rolland C., Capturing System Intentionality with Maps, Conceptual Modeling in *Information Systems Engineering*, Springer-Verlag, Berlin, Germany, 2007, p. 141-158.

Rolland C., Prakash N., *A proposal for context-specific method engineering*, IFIP WG 8.1 Conf. on Method Engineering, pp 191-208, Atlanta, Gerorgie, USA, 1996.

Rolland C., Souveyet C., Kraeim N., « An intentional view of service-oriented computing », RSTI-ISI Vol 13, 2007, p. 107-137, 2008.

Salvador T., Scholtz J., Larson J.: The Denver Model for Groupware Design. Journal, ACM Special Interest Group Computer-Human Interface bulletin (SIGCHI), vol. 28, Nro. 1, ACM Press, 1996.

Wistrand, K., Karlsson, F.: Method components: Rationale revealed. In: CAISE'04, Springer-Verlag. Riga, Latvia, 2004.

Yu J., Lalanda P., Chollet S. « Development Tool for Service-Oriented Applications in Smart Homes », Proc. of SCC'08, Washintong, DC, USA, 2008, p. 239-246.