
Approche orientée services pour la gestion de modèles

Jorge Luis PEREZ-MEDINA - Dominique RIEU - Sophie DUPUY-CHESSA

***LIG – Université de Grenoble*

B.P. 53 38041 Grenoble Cedex 9, France

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

RÉSUMÉ. Le développement de systèmes à base de 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 et les usages autour des modèles ne sont pas consensuels. Pour remédier à l'hétérogénéité et aux limitations fonctionnelles des outils de gestion de modèles, nous proposons une approche de gestion de modèles à base de services pour la création d'environnements de modélisation adaptables aux besoins des concepteurs. Les besoins que nous nous prenons en compte portent sur deux niveaux d'abstraction : le niveau opérationnel et le niveau organisationnel.

ABSTRACT. The development of systems based on models considers any software artefact as a model. The models' management groups a set of features allowing 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, we propose an approach models' management based on services for the creation of environments of modelling adapted to the needs of the designers. The considered needs are related to two different abstract levels: the operational level and the organisational level.

MOTS-CLÉS : Modèle, service, gestion de modèles, IDM, outils de modélisation.

KEYWORDS: Model, service, models' management, MDE, modelling tools.

1. Introduction

Le développement de systèmes à base des modèles est une branche de l'ingénierie 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.

Celle-ci regroupe l'ensemble des fonctionnalités permettant de représenter, créer, stocker et manipuler les modèles. 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. Actuellement l'IDM est utilisé pour décrire l'ensemble des concepts et technologies autour de la gestion de modèles. La plupart des outils IDM sont amplement 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.

Cependant, les besoins des concepteurs en termes de gestion de processus et de produits sont divers. Il n'existe pas aujourd'hui d'outil consensuel et complet de gestion de modèles. Les outils sont limités en fonctionnalités et souvent réduits à quelques méta-modèles et modèles. Un autre aspect à considérer est le fait que les outils actuels ne permettent pas la gestion coordonnée et coopérative de modèles requis par plusieurs spécialistes, qui travaillent ensemble dans un contexte de conception collaborative. En conclusion, les outils ne sont pas complets car les besoins et les usages autour des modèles ne sont pas consensuels. Aussi de nombreux travaux ont adressé le problème de l'intégration d'outils (M. Wicks, 2006). Si les outils proposés sont intéressants pour résoudre le problème technique d'intégration, ils ne répondent pas aux questions du choix des outils de modélisation en fonction des acteurs et des processus de conception. La question que nous cherchons donc à résoudre est comment construire un environnement de gestion de modèles qui répondent aux besoins organisationnels et techniques des entreprises.

Pour remédier à l'hétérogénéité et aux limitations fonctionnelles des outils de gestion de modèles, nous pensons qu'il est nécessaire de disposer d'un environnement de modélisation qui permet de trouver, de coupler et d'utiliser des briques fonctionnelles répondant aux besoins spécifiques de chaque concepteur. Un concepteur de modèles doit pouvoir construire l'environnement de modélisation qui lui convient, quel que soit le rôle qu'il joue dans un processus de conception. Pour trouver une solution à cette problématique, nous nous proposons d'adapter l'approche à base de services (Bieber et al, 2002).

Notre approche concerne la gestion de modèles comme services pour la création d'environnements de modélisation adaptés aux besoins des différents acteurs, il s'agit de :

- Intégrer des outils de gestion de modèles (AGL et IDM) comme des services dans une plateforme ouverte permettant d'ajouter, modifier et supprimer les services.
- Documenter les services de manière à faciliter leur recherche et leur usage.
- Définir un langage de modélisation de processus de développement permettant de définir les rôles et les besoins de chaque acteur du développement en terme de gestion de modèles.

Nous proposons deux types de service : un service «opérationnel» qui correspond à un ensemble d'outils de modélisation utilisable par un concepteur et un service «organisationnel» qui permet de réutiliser de façon coordonnée des services opérationnels et de créer et d'administrer des fragments de modèles de processus.

Le plan de cet article est le suivant. La section 2 présente la notion de service qui est une vision générale de notre approche basée sur deux niveaux de modélisation. La section 3 détaille le modèle de la couche opérationnelle. La section 4 présente le modèle de la couche organisationnelle. Enfin la conclusion et quelques perspectives sont présentées en section 5.

2. Approche générale

2.1. La notion de service

Un service est défini comme une fonctionnalité réutilisable décrite de façon contractuelle dans un annuaire de services (Cervantes, 2004). Dans les applications à base de services, un fournisseur de services doit offrir les objets de service qui implémentent l'interface du service et permettent à un client d'utiliser sa fonctionnalité. Dans notre contexte, un fournisseur de services peut être l'entité qui offre n'importe quel éditeur pour réaliser des traitements sur les modèles, c'est-à-dire qu'il offre des outils de gestion de modèles (figure 1).

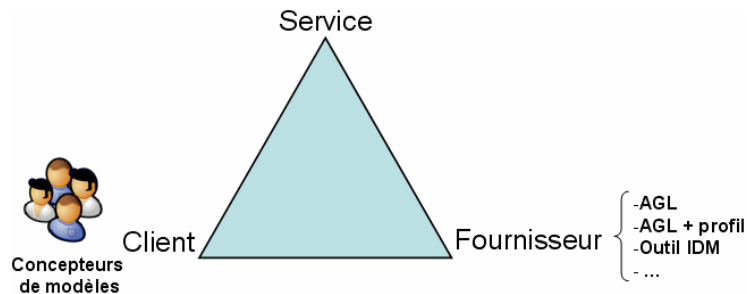


Figure 1. *L'approche à base de services.*

Les clients sont les utilisateurs qui ont pour exigences d'utiliser les services de gestion de modèles. Ils représentent n'importe quel concepteur de modèles. Nous prendrons comme exemples : un « utilisateur lambda » peut avoir besoin d'un environnement de modélisation qui lui offre l'édition de modèles UML et transforme ces modèles dans un autre langage de modélisation ou de programmation. Un « chef du projet » peut avoir besoin de traiter la totalité ou un fragment d'un processus de modélisation, de définir, d'administrer des rôles et des activités des concepteurs intervenant dans le projet. Dans un autre contexte, un « enseignant » qui donne un cours de Génie Logiciel pourrait créer l'environnement de travail qu'utiliseront ses étudiants.

2.2. Deux niveaux de service

Notre approche à base de services ne se limite pas à considérer comme « service » tout outil de gestion de modèles. Elle s'appuie sur deux niveaux de modélisation (voir figure 2) dont les fournisseurs, les clients et les services sont différents. Le premier niveau correspond à la couche opérationnelle. Cette couche permettra de définir l'infrastructure de modélisation qu'un concepteur de modèles veut utiliser. Il s'agit d'offrir les services aux concepteurs de modèles, 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).

La couche organisationnelle permettra de modéliser le processus de conception d'un système. Il s'agit d'offrir un support à base de services à un groupe de projet, donc, dans cette couche, les rôles et activités seront exprimés sous la forme de processus de développement simplifiés. Cette couche permettra de réutiliser de façon coordonnée les services opérationnels. Les clients seront, dans ce cas, les chefs de projet cherchant à définir et à administrer des rôles et des activités sous la forme de processus de développement. Dans notre cas, 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.

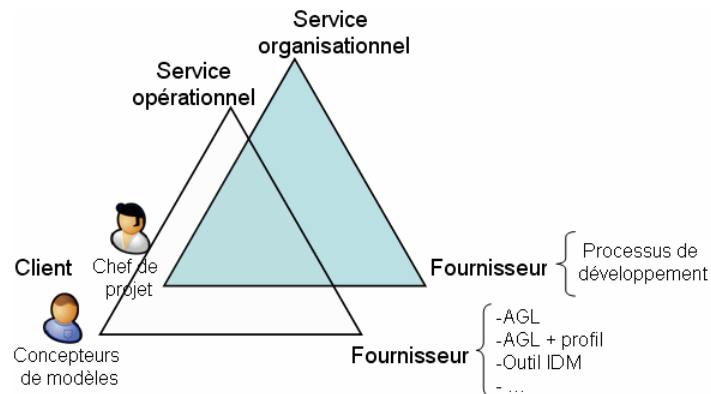


Figure 2. Deux niveaux de service.

En considérant la notion de deux niveaux de modélisation, nous devons garantir que n'importe quel client, quel que soit son rôle, peut chercher et assembler des services.

2.3. Des fonctionnalités offertes par les outils de gestion de modèles

Tant au niveau commercial que dans le monde de la recherche, de nombreux outils de gestion de modèles sont disponibles ou en cours de développement. On peut distinguer trois catégories d'outils de gestion de modèles : Atelier de Génie Logiciel (AGL), outils de transformation de modèles, outils de méta-modélisation.

Les Ateliers de Génie Logiciel (AGL) sont un ensemble d'outils logiciels structurés au sein d'une même interface. Ils sont dotés de fonctionnalités d'édition de modèles adaptés aux différentes phases de la production d'un logiciel. Généralement, ils sont des outils limités à la création de modèles conformes à des modèles propriétaires ou bien ils s'appuient sur le langage de modélisation UML. Actuellement, ils n'offrent que la possibilité de génération de code, de simulation et de test.

Les outils de transformation permettent la transformation d'un modèle vers un ou plusieurs modèles cibles ou la modélisation de la cohérence inter modèles. Ces outils sont nombreux et divers, la plupart d'entre eux sont des outils prévus pour être plus ou moins intégrables dans les environnements de développement standard.

Les outils de méta-modélisation sont un ensemble d'outils qui permettent de réaliser la modélisation et de modifier le méta-modèle sur lequel il est basé, ou tout au moins, il doit permettre d'étendre ce méta-modèle.

Ces outils englobent tous les types de fonctionnalités qui permettent la représentation et/ou la manipulation de modèles. Par rapport à notre approche, nous avons synthétisé les fonctionnalités des outils au moyen de cas d'utilisation. Dans le but de situer les besoins de modélisation de chaque type de client, nous avons séparé les fonctionnalités offertes pour les outils dans deux diagrammes de cas d'utilisation. Par réaliser cette séparation, nous avons pris en compte les deux niveaux de modélisation présentés dans la section précédente.

Au niveau opérationnel, les fonctionnalités offertes par les outils sont représentées par la figure 3. Ces fonctionnalités sont : la gestion de modèles, faire la transformation de modèles, la vérification de la cohérence, la simulation et les tests.

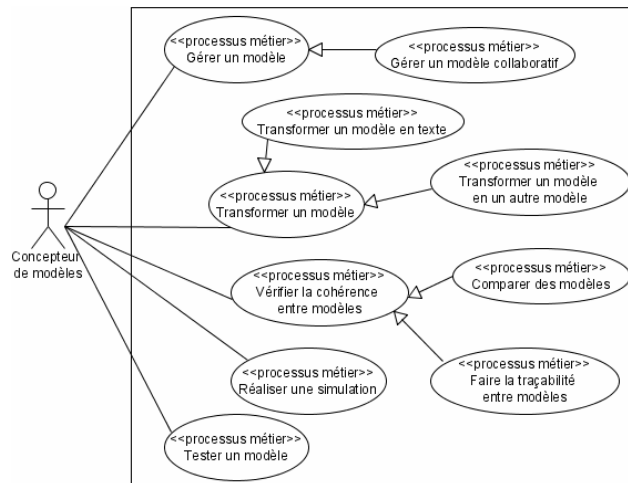


Figure 3. Diagramme de cas d'utilisation des fonctionnalités associées à la gestion de modèles au niveau opérationnel.

Le cas d'utilisation « gérer un modèle », englobe toutes les fonctionnalités d'édition de modèles, quelque soit sa nature et son niveau d'abstraction. En ce qui concerne ce cas d'utilisation, nous l'avons spécialisé pour indiquer la collaboration qui peut exister parmi les concepteurs de modèles, qui doivent effectuer des tâches collaboratives, comme par exemple, l'édition partagée d'un modèle.

Les autres cas d'utilisation de la figure 3 se rapportent à la manipulation des modèles, par exemple : le cas d'utilisation « Transformer un modèle » consiste à faire la transformation d'un modèle dans du code ou dans d'autre langage qui peut être réutilisé par un autre concepteur dans le processus de développement.

Au niveau organisationnel, les fonctionnalités offertes par les outils peuvent être synthétisées en termes de la gestion de démarches de conception de systèmes (voir la figure 4). Gérer une démarche de conception de systèmes consiste à guider les activités de modélisation utilisant des ressources humaines. Il s'agit de définir les rôles que jouent les concepteurs de modèles, les activités à réaliser et leurs enchaînements. Également, dans le processus de développement, il est nécessaire de définir le langage de modélisation à utiliser par le groupe de projet.

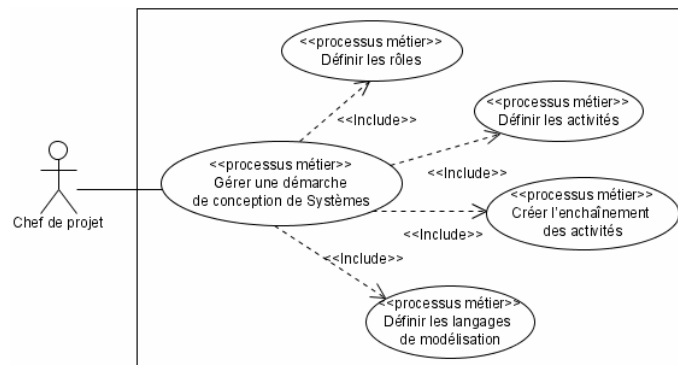


Figure 4. Diagramme de cas d'utilisation des fonctionnalités associées à la gestion de processus au niveau organisationnel.

Dans cette section nous avons exprimé comment se situent les fonctionnalités offertes par les outils de modélisation en relation à notre démarche de gestion de modèles à base de services. Nous avons considéré deux niveaux d'abstraction. Les sections suivantes présentent les détails des modèles aux niveau opérationnel et organisationnel.

3. La couche Opérationnelle

3.1. Modélisation d'un service opérationnel

Cette section présente le modèle de la couche opérationnelle. 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. La figure 5 montre le modèle opérationnel. Un service opérationnel correspond à une application exécutable composée de 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: un AGL, un outil de méta-modélisation ou un outils de transformation de modèles. Un service est associé à divers types de

fonctionnalités. Un type de service est représenté par une composition de services élémentaires. Un service élémentaire définit la façon dont un concepteur de modèles pourra effectuer les opérations sur les modèles (texte, graphique ou les deux), les langages supportés pour stocker l'information et les opérations possibles du service (création, édition, suppression)

Pour prendre en compte la réutilisation de ressources de modélisation, les services opérationnels peuvent porter un ou plusieurs modèles. En suivant l'architecture de modélisation à quatre niveaux (OMG97, 1997), un modèle de niveau=1 définit des représentations concrètes du monde réel (modèles) ainsi que des descriptions de ces représentations. Un modèle de « niveau= 1 » est conforme à un modèle de « niveau > 1 ». Une autre caractéristique que nous avons considérée est le fait qu'un modèle peut être représenté par plusieurs modèles, par exemple : UML se décompose en un ensemble de 13 diagrammes spécifiant les différents aspects d'un logiciel.

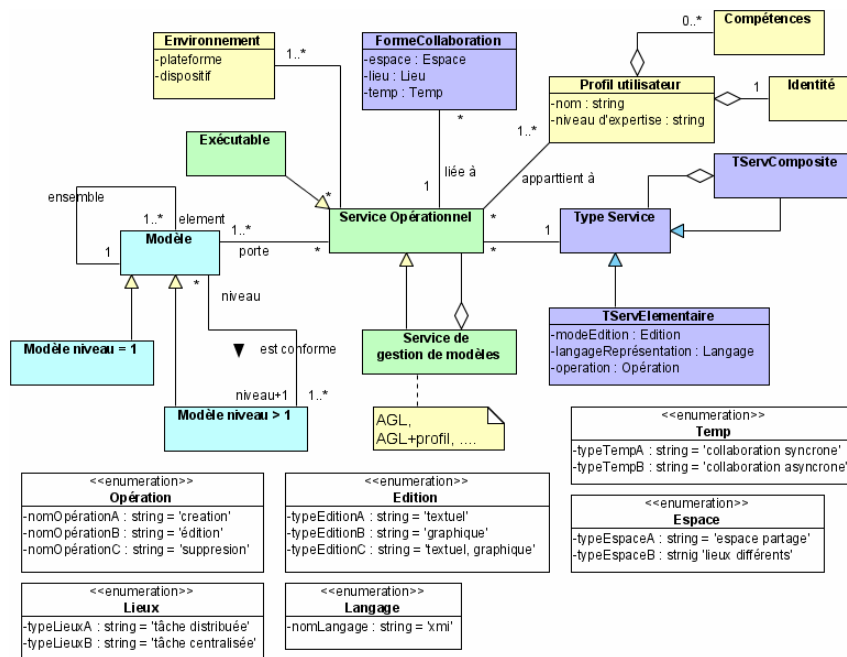


Figure 5. Modèle de service opérationnel.

La modélisation de service opérationnel n'est pas nouvelle. Nous avons voulu ici augmenter les approches classiques pour répondre aux besoins que nous avons identifiés en étudiant une méthode de conception qui intègre la conception de l'interaction homme-machine (IHM). Cette étude à montrer que divers acteurs (spécialiste génie logiciel, spécialiste IHM, ergonomes) travaillent de manière

coordonnée sur des modèles différents. Ces conclusions nous amènent à proposer un modèle de services opérationnels original par 1) les caractéristiques de l'utilisateur ; 2) les aspects collaboratifs du travail de conception.

Pour supporter les besoins de créer des environnements de travaux collaboratifs, un service opérationnel est lié à plusieurs formes de collaboration. Une forme de collaboration est définie en fonction de trois axes : le lieu, l'espace, et le temps (Salber, 1995). Le lieu consiste en la manière comme les tâches sont effectuées, nous nous intéressons à savoir si les tâches sont centralisées ou distribuées. L'espace est le lieu où les tâches sont exécutées, c'est-à-dire dans les espaces partagés ou lieux différents. Finalement, le temps est le moment de la collaboration. Nous nous rapportons alors aux types de collaboration synchrone ou asynchrone.

Un service opérationnel peut être exécuté dans une ou plusieurs plateformes, et dispositifs et peut appartenir à un ou plusieurs profils d'utilisateurs. La notion de profil contient des informations sur l'identité de l'utilisateur et sur ses compétences (par exemple, le niveau d'expertise relatif à ses activités) qui peuvent être utiles au moment de trouver le service plus adéquat aux utilisateurs.

3.2. Illustration

En nous basant sur le modèle opérationnel, nous avons créé un scénario d'exemple (figure 6). Notre scénario considère un concepteur individuel spécialiste en interaction homme-machine qui désire faire un des modèles de sa spécialité : un arbre de tâches. Le contexte de l'utilisateur est qu'il est novice et qu'il travaille sur un PC sous Windows. D'autres concepteurs travailleront sur l'arbre des tâches mais de manière asynchrone et dans des lieux différents. Les besoins fonctionnels se résument en la création d'arbres de tâches. L'arbre de tâches qu'il souhaite réaliser s'appuie sur la notation CTT (Paternò, 2001). Il a donc besoin de définir le méta-modèle basé sur la notation CTT et de créer ses diagrammes respectifs (arbres de tâches). Il s'agit de fournir un environnement de modélisation à ce concepteur.

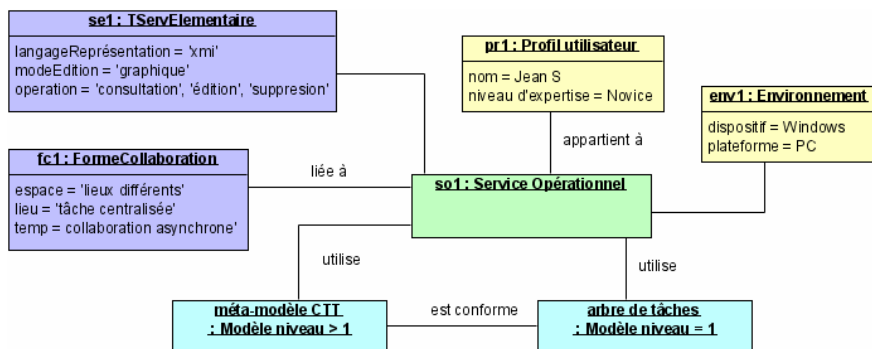


Figure 6. Un diagramme d'objets du modèle de la couche opérationnelle.

Pour résoudre les besoins du concepteur, le service opérationnel « so1 » lui permet d'éditer le méta-modèle CTT. De même, ce service lui permettra de créer les arbres de tâches conformes au méta-modèle de CTT.

4. La couche Organisationnelle

4.1. Modélisation d'un service organisationnel

Pour aider au choix des services opérationnels, nous avons considéré les services à un autre niveau. Contrairement à (Rolland, 2007) qui considère les services intentionnels pour prendre en compte leurs buts, nous prenons un autre point de vue et considérons leurs aspects organisationnels. Notre modèle de service organisationnel est inspiré des travaux de Hug (Hug, 2007). Ces travaux se rapportent en la méta-modélisation des processus. Nous y introduisons la notion de service. Notre modèle de service organisationnel doit donner les moyens pour créer et administrer des fragments de processus de modélisation. La figure 7 présente un modèle de processus minimal dont l'objectif est d'identifier les rôles et les activités en termes de gestion de modèles de chaque participant responsable d'une ou de plusieurs activités dans un processus de conception de systèmes.

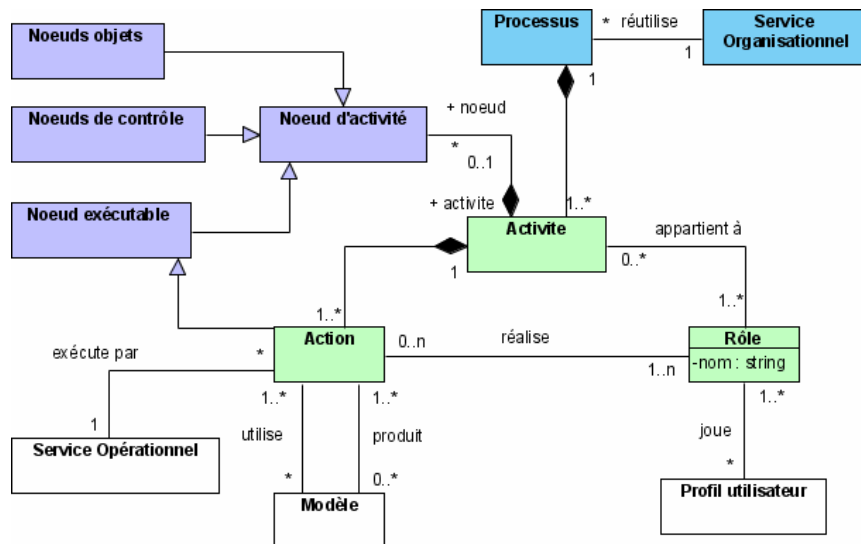


Figure 7. Modèle de processus de la couche organisationnelle.

Un service organisationnel consiste en la réutilisation de plusieurs processus. En ce qui concerne nos travaux, un processus est composé d'activités, lesquelles doivent être définies en termes de manipulation de modèles. Par exemple, dans la définition d'un processus de transformation de modèle, une activité pourrait être l'édition du méta-modèle source.

Une activité définit un comportement décrit par un séquençement organisé d'unités dont les éléments simples sont les actions. La notion de nœud d'activité permet de représenter les étapes le long du flot d'une activité. Un nœud exécutable est un nœud d'activité qu'on peut exécuter. Un nœud de contrôle est un nœud d'activité utilisé pour coordonner les flots entre les nœuds d'une activité. Un nœud d'objet permet de définir un flot d'objets dans un diagramme d'activités. Dans notre contexte, ce nœud représente l'existence d'un modèle généré par une action dans une activité et qui peut être utilisé par d'autres actions.

Les activités appartiennent aux rôles que les concepteurs de modèles peuvent accomplir. Un utilisateur qui joue un rôle peut réaliser plusieurs actions qui appartiennent à une activité. En prenant en compte qu'une activité peut être réalisée de forme collaborative, nous remarquons qu'une action peut être effectuée par différents rôles.

Un autre aspect considéré par le modèle opérationnel est le fait que les actions sont les mécanismes qui permettent de lier des services opérationnels avec des modèles. La réalisation d'une action peut utiliser plusieurs modèles, mais aussi, une action peut produire plusieurs modèles. Par exemple, dans un processus de transformation d'un modèle, l'action exécuter la transformation requiert d'utiliser un méta-modèle source, un méta-modèle cible, un modèle source et produire un modèle cible.

4.2. Illustration

Cette section présente un cas d'expérimentation. Notre objectif est donner une meilleure compréhension du modèle organisationnel en montrant comment les services organisationnels s'intègrent lors dans une démarche de conception. Notre contexte d'expérimentation est basé sur la méthode de conception de systèmes de réalité mixte proposée par Juras (Juras, 2006). Cette méthode prend en compte les collaborations entre spécialistes de différents domaines (génie logiciel, interaction homme-machine, ergonomie) qui travaillent dans la conception de systèmes interactifs liant mondes réel et virtuel. Le cœur de cette méthode se base sur la démarche Symphony (Hassine, 2005). Nous résumons dans la figure 8 des extraits de la méthode qui servent de fil conducteur pour la présentation de notre exemple.

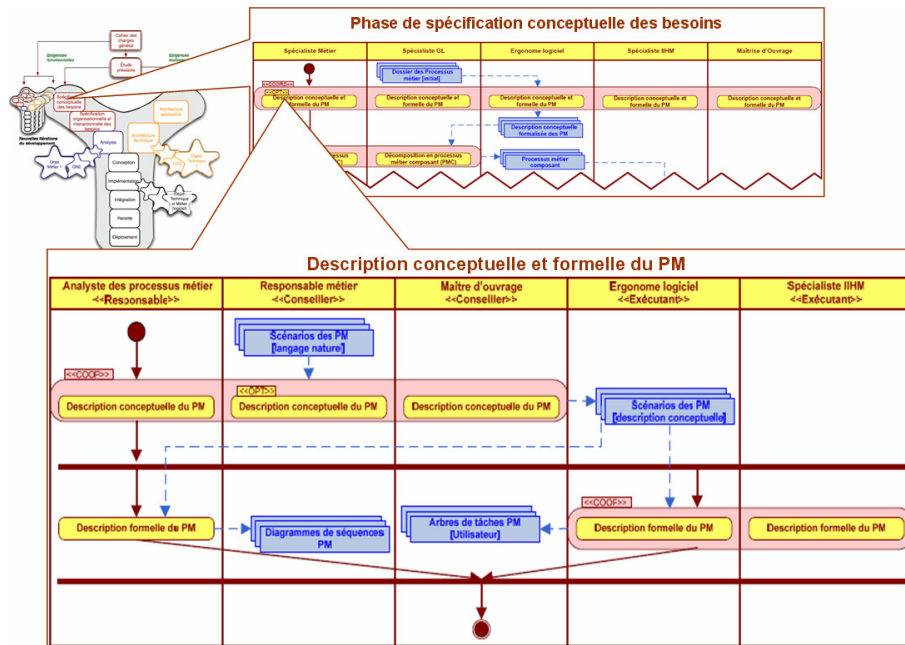


Figure 8. *Processus de description conceptuelle et formelle du Processus Métier correspondant à la phase de spécification conceptuelle des besoins de la méthode Symphony étendue pour le développement de systèmes mixtes.*

Afin de simplifier l'exemple, nous nous concentrons sur le processus de description conceptuelle et formelle du Processus Métier (PM). Les activités de « description conceptuelle du PM » et « description formelle du PM » doivent être partagées par différents rôles. Par exemple, l'analyste des processus métier, le responsable métier, le maître d'ouvrage doivent participer à la réalisation de l'activité de description conceptuelle du PM alors que, l'analyste des processus métier, l'ergonome logiciel et le spécialiste IHM sont responsables de la réalisation de l'activité de description formelle du PM.

Au niveau opérationnel, l'approche que nous proposons doit garantir un environnement de modélisation pour chaque concepteur, donc, en nous basant sur le modèle opérationnel de la figure 5, nous devons créer les services opérationnels adéquats à chaque concepteur. Par manque d'espace dans ce document, nous considérons seulement le rôle de l'analyste de processus métier. Ce concepteur de modèles doit collaborer dans la réalisation de deux activités : la description conceptuelle du PM et la description formelle du PM.

Pour réaliser son travail, un concepteur a besoin d'un environnement de modélisation basé sur la consultation des scénarios de processus métier en langage

naturel, et l'édition partagée des scénarios structurés des PM, ainsi que l'édition de diagrammes de séquences. Nous remarquons que parmi les caractéristiques que doit avoir son environnement de modélisation, se trouve l'utilisation d'un espace partagé d'édition de diagrammes de cas d'utilisation et de diagrammes de séquences, où les tâches sont réalisées sous forme distribuée et non synchronisées. Toutes ces informations nous permettent de définir les services opérationnels voulus par les concepteurs dans le cadre de la méthode Symphony augmentée pour l'IHM.

Au niveau organisationnel, un chef de projet doit gérer le processus de description conceptuelle et formelle du Processus Métier (PM). En ce qui concerne l'enchaînement des activités de modélisation, la figure 9 illustre comment les actions doivent être conduites en termes de gestion de modèles. Notre but consiste à montrer comment au niveau organisationnel, un chef de projet doit modéliser les actions de gestion de modèles d'une façon coordonnée et coopérative.

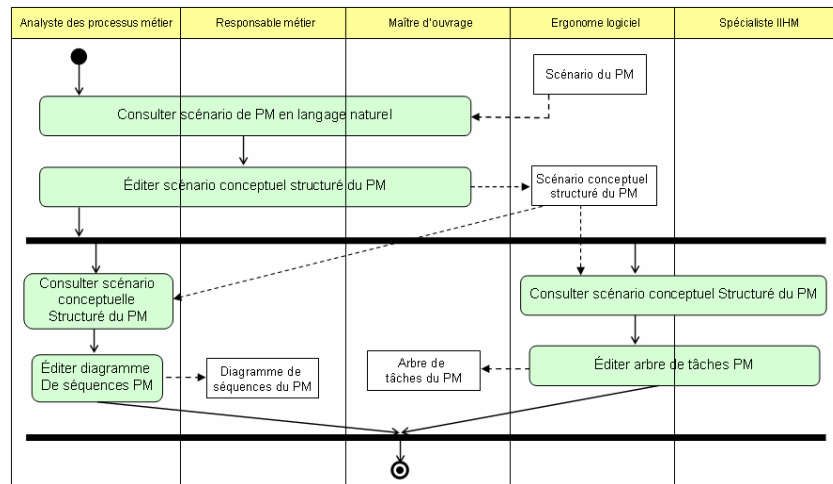


Figure 9. Processus de description conceptuelle et formelle du Processus Métier en termes des actions sur les modèles.

En nous basant sur le modèle organisationnel de la figure 7, nous avons créé une instance du modèle organisationnel (figure 10), qui correspond au rôle de l'analyste de processus métier, le concepteur que nous avons considéré au niveau opérationnel.

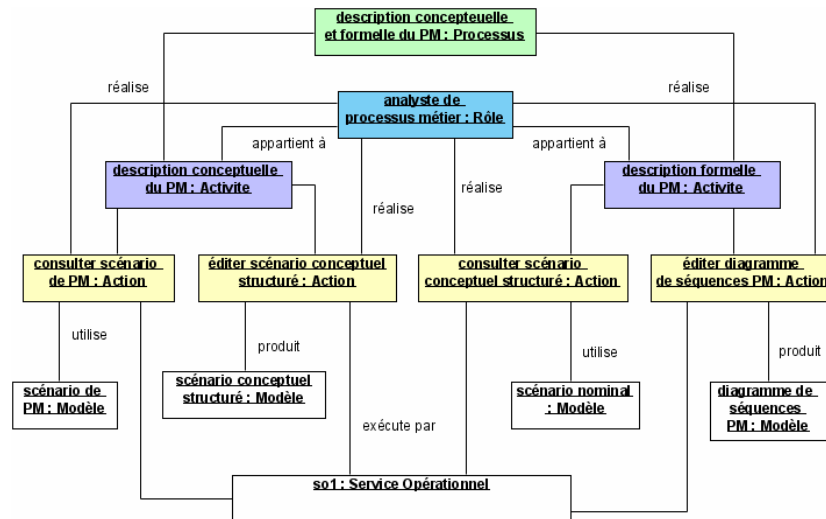


Figure 10. Un diagramme d'objets du modèle de la couche organisationnelle pour représenter les actions que doit réaliser l'analyste de processus métier.

Le diagramme d'objets de la figure 10, montre que l'activité de description conceptuelle du PM qui appartient à l'analyste de processus métier, est basée sur deux actions qu'il doit réaliser. La première action correspond à la consultation des scénarios de PM : il s'agit d'utiliser un outil de modélisation pour la consultation de modèles de scénarios de PM. La deuxième action, doit permettre l'édition partagée des scénarios structurés des PM. Ces deux actions sont supportées par le service opérationnel « so1 » défini au niveau opérationnel.

La deuxième activité représentée par le diagramme de la figure 7, est la description formelle du PM pour laquelle il s'agit de consulter les scénarios structurés des PM et de créer des diagrammes de séquences. De la même manière, ces actions sont supportées par le service opérationnel défini au niveau opérationnel.

Au travers de cet exemple, nous avons montré la faisabilité de notre approche pour considérer les besoins organisationnels et techniques des concepteurs. Nous avons montré l'intérêt des services organisationnels et leurs liens avec les services opérationnels.

5. Conclusion et perspectives

Nous avons présenté, à travers cet article les principes qui seront utilisés pour l'implémentation de notre approche orientée services pour la gestion de modèles. L'approche que nous proposons ne se limite pas à considérer comme « service » tout

outil de gestion de modèles. Elle s'appuie sur deux niveaux de modélisation dont les fournisseurs, les clients et les services sont différents.

Le premier niveau correspond à la couche opérationnelle. Cette couche permettra de définir l'infrastructure de modélisation qu'un concepteur de modèles veut utiliser. Il s'agit de donner les services aux concepteurs de modèles, pour faciliter la création de son environnement de modélisation. L'originalité de notre travail se trouve dans la description d'un service opérationnel, ce service considère les aspects tels que : type de service, ressources de modélisation, forme de collaboration, plateforme, dispositifs et profil de l'utilisateur.

Au deuxième niveau, le modèle organisationnel permet de réutiliser de façon coordonnée les services opérationnels, mais aussi de créer et d'administrer des fragments de modèles de processus, en utilisant la notion de rôles, d'activités et d'actions. Ce modèle organisationnel permettra aux chefs de projets de modéliser le processus de conception d'un système. Il s'agit de donner un support à base de services à un groupe de projet, donc, à ce niveau, les rôles et activités seront exprimés sous la forme de processus de développement simplifiés.

Par la suite, il est nécessaire de spécifier une couche intentionnelle. Cette couche correspond à la modélisation des buts qui peuvent être proposés pour représenter l'intention d'un sujet individuel, un groupe d'individus, une unité de travail ou toute organisation qui participe au processus de développement de systèmes. Ce niveau intentionnel permettra de justifier l'existence de processus, des activités, des acteurs et ses rôles, ainsi que les systèmes d'information qui les appuient.

Ensuite, il faudra concevoir un langage de modélisation de processus qui permet d'unir les notions des objectifs, des acteurs et ses rôles, processus et d'activités en termes de gestion de modèles avec les services opérationnels. Ce langage de modélisation devra prendre en compte la gestion coordonnée et coopérative de modèles requis par plusieurs spécialistes.

En parallèle, nous devons considérer l'utilisation d'une plateforme pour la mise en œuvre des services. Les outils d'intégration existants tels que ModelBus (Blanc, 2005) seront des références importantes dans ce domaine. Actuellement, nous avons perçu que la plateforme de développement ouverte Eclipse apporte une solution aux problèmes d'interopérabilité entre outils hétérogènes via l'ajout de plug-in. Enfin, il s'agira d'intégrer dans cet environnement de développement l'approche que nous proposons.

Remerciements

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

6. Bibliographie

- Action IDM. – Ingénierie dirigée par les modèles. <http://www.actionidm.org/>, consultation février 2007.
- Bieber, G., Carpenter, J. Introduction to Service-Oriented Programming (Rev 2.1), Avril 2001. (<http://www.openwings.org/download.html>).
- Blanc X., Gervais M.-P., Sriplakich P., Modeling Services and Web Services: Application of ModelBus, Proceedings of the 2005 International Conference on Software Engineering Research and Practice (SERP'05), Las Vegas, USA, June 2005.
- Cervantes H., « Vers un modèle a composants orienté services pour supporter la disponibilité dynamique », Thèse de Doctorat soutenue à l'Université Joseph Fourier, Grenoble, Mars 2004, 187 pages.
- Hassine I., Spécification et formalisation des démarches de développement à base de composants métier : la démarche Symphony, Thèse de Doctorat soutenue à l'Institut National Polytechnique de Grenoble, Grenoble septembre 2005.
- Hug C., Front A., Rieu D., « Ingénierie des processus : une approche à base de patrons ». Congrès Inforsid'07. Perros-Guirec, France, May 2007. pp. 471-486.
- Juras D., Rieu D., Dupuy-Chessa S., Front A., « Conception collaborative pour les Systèmes Mixtes », Congrès Inforsid'06. Hammameth, Tunisie, Juin 2006. pp. 33-48.
- OMG97, OMG/MOF Meta Object Facility (MOF) Specification. OMG Document AD/97-08-14, September 1997.
- OMG00, Trading Object Service Specification. Version 1.0, OMG Document 2000-06-27. <http://www.omg.org/cgi-bin/doc?formal/2000-06-27>, consultation février 2007.
- OMG03, Object Management Group, « MDA Guide Version 1.0.1 », Juin 2003. <http://www.omg.org/docs/omg/03-06-01.pdf>, consultation février 2007.
- Paternò F., Mori G., Galimberti R., CTTE : An environment for Analysis and Development of Task Models of Cooperative Applications. Proceedings ACM CHI'01, ACM Press, Seattle, 2001.
- Rolland C., Kaabi R.-S., An Intentional Perspective to Service Modeling and Discovery, Proceedings of 31st Annual International Computer Software and Applications Conference, COMPSAC 2007 - Vol. 2. Volume 2, Issue, 24-27 July 2007, pp 455 – 460.
- Salber D., « De l'interaction individuelle aux systèmes multi-utilisateurs. L'exemple de la Communication Homme-Homme-Médiatisée », Thèse de Doctorat soutenue à l'Université Joseph Fourier, Grenoble, Septembre 1995, 303 pages.
- Wicks M., Tool Integration within Software Engineering Environments: An Annotated Bibliography, Technical Report. Ref. HW-MACS-TR-0041, 4th August 2006.