

ARCHITECTURE ORIENTEE SERVICES INTENTIONNELS (ISOA) : CONCEPTS, CARACTERISTIQUES ET DIRECTION

Yves Roger. NEHAN

Centre de Recherche en Informatique, CRI
Université Paris 1 – Panthéon Sorbonne
90 rue de Tolbiac
75013 Paris
habas-yves-roger.nehan@univ-paris1.fr

Carine. SOUVEYET

Centre de Recherche en Informatique, CRI
Université Paris 1 – Panthéon Sorbonne
90 rue de Tolbiac
75013 Paris
Carine.soveyet@univ-paris1.fr

RESUME : Le travail présenté dans cet article est de montrer le paradigme service afin de modéliser, exécuter, composer et rechercher des services intentionnels. Cette approche vise d'une part, à combler le fossé entre la vision utilisateur et la vision technique du concept de service et, d'autre part, d'offrir une architecture orientée services intentionnels (ISOA) permettant de piloter l'architecture technique du système. Cette approche orientée services à deux niveaux d'abstractions : niveau intentionnel et niveau opérationnel, permet de composer dynamiquement des services techniques en fonction des intentions des utilisateurs. Le niveau intentionnel guide le processus décisionnel de l'utilisateur alors que le niveau opérationnel prend en charge la réalisation des intentions exécutables (atomiques) par l'exécution du service logiciel le plus approprié. Aussi cette approche propose un modèle intentionnel de services (MIS) permettant la description et la composition de services au niveau intentionnel et un modèle opérationnel de services (MOS) spécifiant le module logiciel à exécuter pour atteindre l'intention d'un service MIS indépendamment de la plateforme cible.

MOTS-CLES : *Ingénierie dirigée par les modèles, transformation de modèles, coopération de services, Architecture Orientée services, services intentionnels, interopérabilité des services*

1. INTRODUCTION

L'adhésion croissante des entreprises aux technologies émergentes orientées e-services vient de la promesse de celles-ci à favoriser la conception, la construction et le déploiement des systèmes d'information coopératifs et flexibles. En effet, l'architecture orientée services appelée SOA (W3C, 2004) et les techniques émergentes (WSCl, 2002) (WSCL, 2001) (Andrews, 2003) (Alonso et al, 2004) (Arsanjani, 2004) utilisées pour sa mise en œuvre permettant le développement de systèmes très faiblement couplés où les sous-systèmes indépendants, autonomes, déployés sur des environnements fortement hétérogènes, peuvent coopérer et évoluer sans modifier leurs architectures. Les services sont considérés comme des modules logiciels autonomes, auto-descriptifs qui sont exposés, publiés, découverts, composés et négociés à la demande. En outre, cette approche orientée services est d'autant plus prometteuse que c'est la première fois que le paradigme « services » a un sens au niveau technique et au niveau stratégique.

Toutefois pour que la paradigme orienté services soit transposé au niveau business de l'entreprise, il est nécessaire d'établir un pont entre les services de haut niveau au sens où le monde du business les comprend et les services techniques et logiciels de bas niveau tels que les développeurs d'applications informatiques les comprennent aujourd'hui (Arsanjani, 2004),

(Zimmermann, 2004). Le constat est qu'il existe un fossé entre ce concept de services à deux niveaux d'abstraction : niveau stratégique et niveau technique (figure 1) qu'il faudrait à combler.

Ce constat requiert que l'on s'interroge sur la notion de service au sens du monde du business, sur l'identification et la spécification de tels services et sur la correspondance avec les services logiciels qui les réalisent. Il nous semble que c'est à ce prix que le paradigme SOA peut porter ses meilleurs fruits.

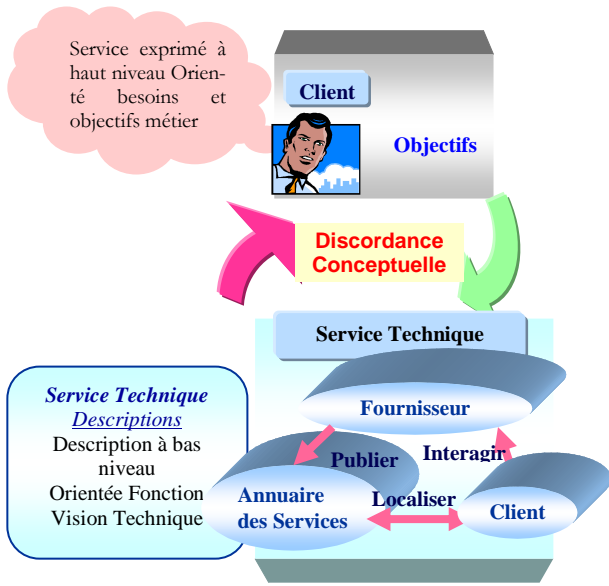


Figure 1. Problème de matching entre les besoins utilisateurs et les services techniques.

Dans la figure 1, la description du service au niveau opérationnel (niveau techniques) se focalise sur le « quoi » et le « comment » des opérations et méthodes ; par exemple, sur les données qu'elles doivent fournir et les opérations qui doivent être effectuées. Rétablir une mise en correspondance entre ces services et les besoins des clients n'est pas une tâche simple. D'un côté, les clients qui souhaitent utiliser un service possèdent un certain nombre d'objectifs à satisfaire, d'un autre côté, les services techniques sont implémentés à travers différents outils hétérogènes.

Le souhait est donc de faire une mise en correspondance au niveau le plus haut entre leurs objectifs des clients et les objectifs que les services pourraient aider à réaliser. L'expression des objectifs est intentionnelle, contrairement aux services qui sont exprimés de manière opérationnelle.

L'illustration de notre travail part de la vue hiérarchique du cycle de développement des services techniques de (Papazoglou, 2006a). (Papazoglou, 2006a) conçoit que les services et les processus du business dans une application orientée services sont implémentés comme des composants en termes de fonctions opérationnelles et financières et des données disponibles de ressources tel que les ERP, les bases de données, les CRM (customer service module) et d'autres systèmes.

Dans ses travaux (figure 2), il part du principe qu'il est commode de faire une distinction entre les processus du business et les services du business qui sont compris dans la vue logique du cycle de développement de service et la vue physique du cycle de développement des services qui est constituée quand à elle de services d'infrastructures (services techniques) et les implémentations des composants des composants qui élaborent les services logiques aux ressources existantes (Papazoglou, 2006b). La vue logique permet de comprendre les processus et les services du business tel qu'un ordre de gestion de processus qui fourni les services du business

(lesquels sont des services indivisibles) pour la création, la modification, la cessation, la suppression, des ordres d'interrogation. Les services d'infrastructure (ou services techniques) sont habituellement les services de gestion et de contrôle dans les infrastructures de gestion et de contrôle de services tels que ceux qui fournissent l'utilité technique comme la sécurité, les contrôles d'accès, les transactions, l'authenticité et ceux qui gèrent ces ressources.

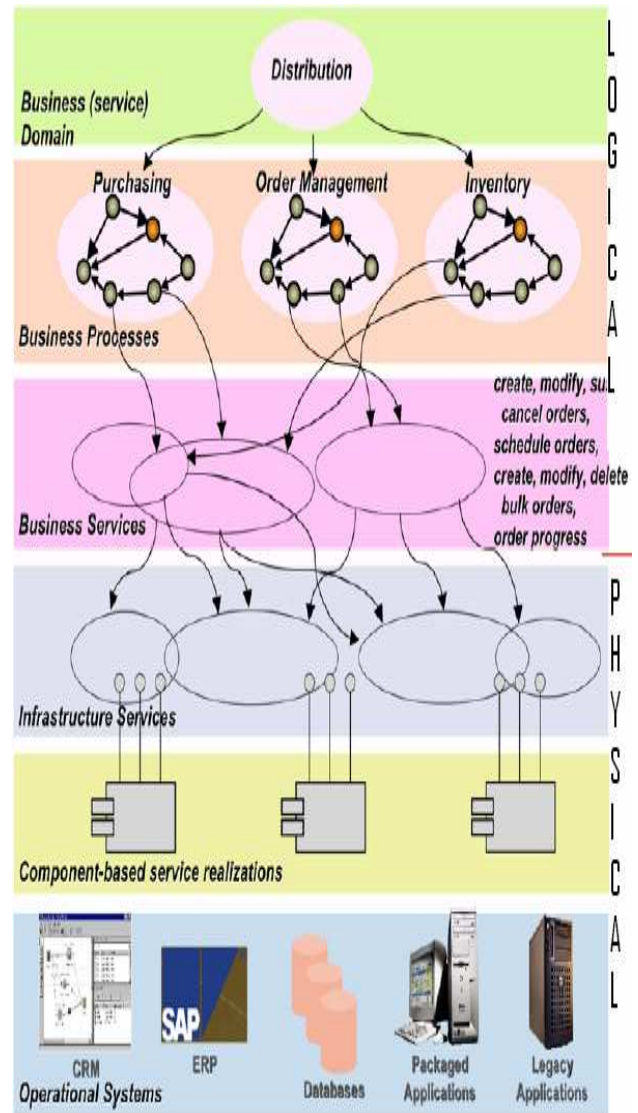


Figure 2. Vue hiérarchique du cycle de développement des services techniques (Papazoglou, 2006)

Comme l'a montré la figure 1, la difficulté des applications SOA entre la vue logique et la vue physique (figure 2) réside dans le fait la spécification métier au niveau logique est non flexible alors que celle au niveau physique l'est. Ceci vient du fait du lien de transformation qui se reside entre la spécification métier (niveau logique) et la spécification technique (niveau physique). Cette transformation (figure 3) empêche les objectifs métiers d'être modifiables, flexibles. Cependant, les services techni-

ques seront quand à elle flexibles car elles sont basées sur l'architecture SOA (figure 1).

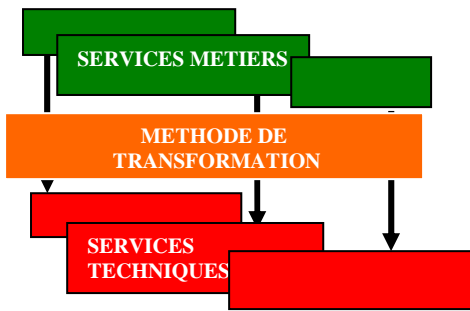


Figure 3. Transformation de la spécification métier à la spécification technique dans une SOA

Dans la figure 3, notre constat est que les services techniques sont obtenus par la transformation des services métiers. Cette méthode de transformation constitue aujourd'hui un problème majeur dans les applications SOA car elle ne favorise pas la flexibilité au niveau métier mais seulement au niveau technique basé sur la plateforme SOA qui est une architecture technique flexible et interopérable. C'est donc à cette technique et à ce problème que nous nous attaquons pour résoudre le problème de discordance conceptuel entre les objectifs utilisateurs exprimés dans les services à haut niveau orienté besoin et objectifs métiers Et les services techniques basés SOA qui sont la transformation des objectifs métiers.

2. ARCHITECTURE ISOA : SOLUTION AUX PROBLEMES DE DISCORDANCE CONCEPTUELLE

2.1. Concepts de l'architecture à deux niveaux

L'idée adoptée dans cet article est que de tels services de haut niveau, pour être en phase avec le monde du business, doivent être exprimés dans un *mode intentionnel*, c'est-à-dire par référence à des buts et stratégies que l'organisation choisit pour les atteindre. Nous les qualifions de *services intentionnels*. Nous pensons qu'une telle vue des services peut minimiser la discordance conceptuelle entre la définition des services logiciels et l'énoncé des exigences des utilisateurs en réduisant le décalage pénalisant entre des expressions fonctionnelles et celles des besoins organisationnels tels que le monde du business les formule naturellement.

Notre croyance est que la découverte et la spécification des services ne peut pas être déconnectée de leur usage, c'est-à-dire des processus du business qu'ils servent. Notre position est donc de découvrir et de décrire les services intentionnels nécessaires à une organisation en développant une perspective de modélisation des processus. Par homogénéité entre la notion de service intentionnel et le processus d'élucidation des services du business, nous proposons de modéliser les processus du

métier de l'entreprise dans une perspective elle-même intentionnelle.

Notre position nous conduit à proposer un déplacement de toute la perspective SOA centrée *'fonction'* vers une position équivalente mais centrée *'intention'*. Dans cette perspective, les services sont décrits dans les termes intentionnels du business, c'est-à-dire en termes d'intentions et de stratégies pour les atteindre et leur publication, leur recherche et leur composition se fait sur la base de ces descriptions intentionnelles. De cette façon, on peut parler d'un portage de SOA au niveau intentionnel que nous qualifions de *ISOA* (intentional Service Oriented Architecture).

La proposition dans cet article repose donc sur le concept de services intentionnels qui abstrait les détails du service logiciel et de ses fonctionnalités pour se concentrer sur son essence, à savoir le but qu'il permet d'atteindre.

En conséquence, la mise en correspondance entre un service et un besoin organisationnel, s'établira au niveau intentionnel au moyen de rapprochement entre modèles de buts. Dans cette approche, nous ajoutons une autre couche d'architecture à deux niveaux d'abstractions : le niveau intentionnel et le niveau opérationnel. Au niveau intentionnel, un élément intentionnel sera opérationnalisé par un élément opérationnel. L'élément intention est un modèle exécutable alors que l'élément opérationnel est un élément logiciel. Le lien qui permet d'opérationnaliser l'élément intentionnel en élément opérationnel est un lien de délégation : on a donc une méthode de délégation. D'un côté, il aura un appel (call) de l'élément opérationnel par l'élément intentionnel et de l'autre, on aura une réponse (return) de l'élément opérationnel. C'est donc le principe de délégation de type Appel/Réponse ou Call/return qui est utilisé en lieu et place de la transformation. La couche intentionnelle est une vision centrée usage des services techniques : ce sont les intentions des utilisateurs. Comme conséquence, le niveau opérationnel sera piloté par le niveau intentionnel. Un élément intentionnel est constitué d'un élément atomique non décomposable et d'élément composite décomposable en élément atomique (Figure 4.). C'est donc ce dernier élément atomique qui sera opérationnalisé par le niveau opérationnel.

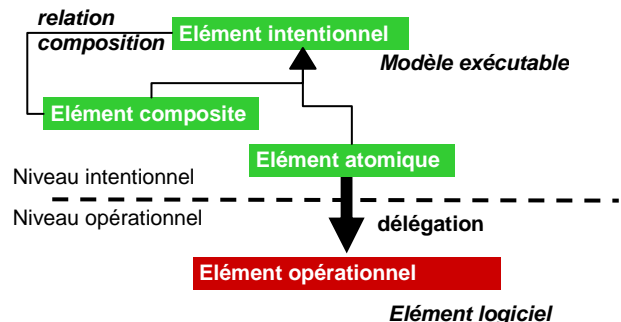


Figure 4. Architecture à deux niveaux (ISOA) : niveau intentionnel et niveau opérationnel

La figure 4 représente la solution au problème de transformation rencontré dans les architectures SOA. Notre solution consiste à remplacer le lien de transformation précédent par le lien de délégation de type Appel/Réponse (Call/Return) entre un élément atomique au niveau intentionnel centré intention ou but utilisateur et un élément opérationnel. Avec notre approche, nous arrivons très facilement à une flexibilité non seulement au niveau opérationnel car couplé faiblement et de plus interchangeable car basé sur l'architecture SOA mais aussi une flexibilité au niveau intentionnel, ce qui n'est pas le cas avec l'architecture vue précédemment. Le niveau intentionnel comporte désormais une variabilité intentionnelle. Il y a donc désormais une variation dans la réalisation d'une intention. Les variantes correspondent aux différentes manières d'atteindre l'intention (ou le but). Comme un élément intentionnel peut être composé d'éléments intentionnels ayant chacun leur propre but et donc des variantes associées, il en résulte que l'élément intentionnel se définit comme un réseau de variante attaché à des points de variation. De ce fait, il découle une composition dynamique du niveau opérationnel dirigée par les intentions.

2.2. Le modèle de la carte

Les besoins utilisateurs sont identifiés à la première étape. Ces besoins sont décrits avec un modèle orienté buts appelé carte (Rolland, 2000). La carte exprime les besoins des utilisateurs par des buts et des stratégies pour les atteindre. Le formalisme de la carte représente les besoins par un ensemble de stratégies pour atteindre un même but et par des compositions de buts et de stratégies pour satisfaire un but final. Dans ce qui suit, nous détaillons les deux étapes, à savoir la construction du modèle de capture des besoins et la génération du modèle de représentation des services.

Nous présentons le formalisme de la carte que nous avons choisi pour construire le modèle de capture des besoins des utilisateurs.

Le modèle de la carte est un système de représentation des processus dans un mode intentionnel. Il fait partie de la classe des modèles de buts. Le système de représentation de la carte utilise le concept de but et se différencie des autres modèles par l'introduction du concept de stratégie pour atteindre un but. Dans ce système de représentation, un but est ce qu'on cherche à atteindre. Une stratégie est une manière de réaliser un but. Le modèle de la carte repose sur un ordonnancement déclaratif et flexible de buts et de stratégies. La Figure 5 présente le méta-modèle de la carte, ses concepts clés et leurs relations en utilisant la notation UML.

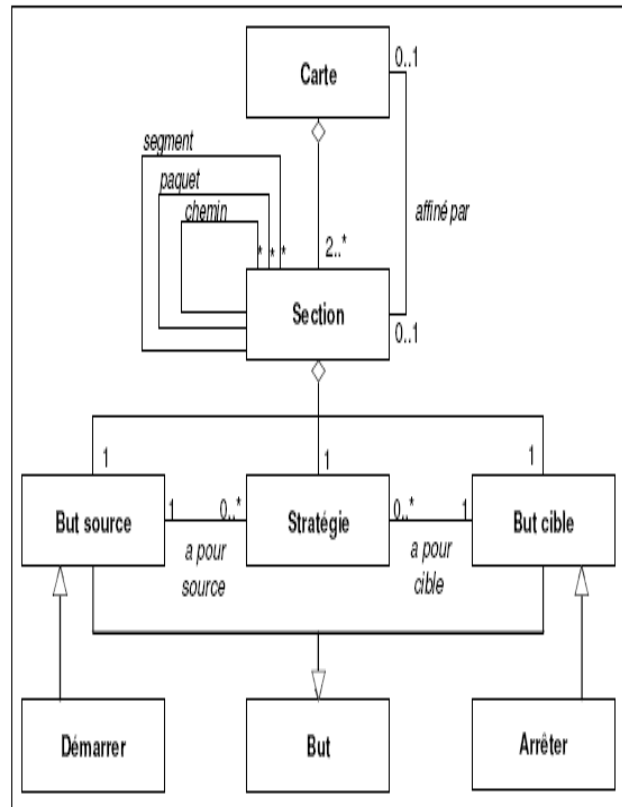


Figure 5. Méta modèle de la carte

Le méta-modèle de la carte (Figure 5) montre qu'une carte est composée de deux ou plusieurs sections. Une section est une agrégation de deux types de buts, un but source et un but cible, et d'une stratégie. Chaque section correspond à une stratégie qui peut être utilisée pour réaliser un but cible, une fois que le but source a été atteint.

La carte est représentée par un graphe orienté et étiqueté. Les buts sont les noeuds et les stratégies en sont les arcs. La nature orientée de la carte traduit le flux du but source au but cible via la stratégie. Une section est ainsi représentée par deux noeuds reliés par une flèche.

Dans l'exemple de la Figure 6, nous proposons un exemple de carte qui a pour but Satisfaire efficacement les besoins en produits.

L'exemple de la Figure 6 comporte quatre buts (Démarrer, Acquérir des produits, Contrôler le stock et Arrêter), douze stratégies (Par seuil de réapprovisionnement, Par prévisions stratégiques, Manuellement, Entrer en stock des produits livrés, Entrer les produits directement, Périodique, Echantillonnage, En continu, Par valuation, Par inspection de la qualité, Par transfert des produits, Par contrôle du paiement des factures) et douze sections.

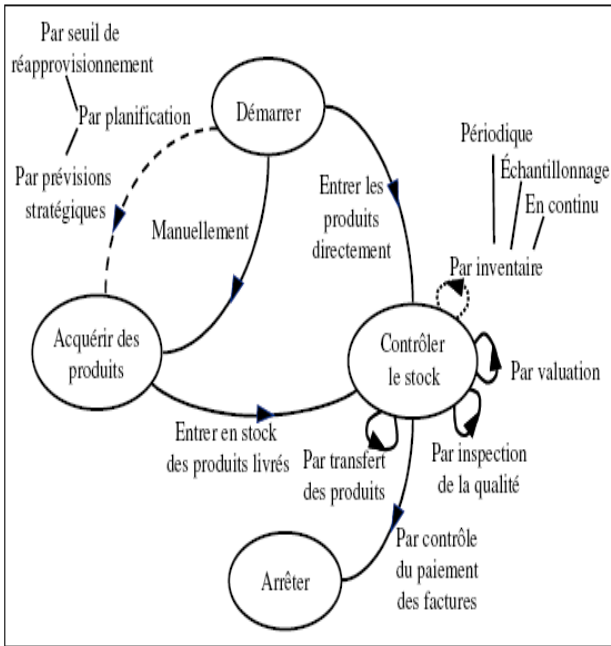


Figure 6. Exemple d'une carte Satisfaire efficacement les besoins en produits

Un but représente ce qu'on cherche à atteindre. Selon (Jackson, 1995), un but est une déclaration 'optative' qui exprime ce que l'on veut, un état ou un résultat que l'on cherche à atteindre. Par exemple, comme le montre la Figure 6, Acquérir des produits et Contrôler le stock sont deux buts dans le domaine de la gestion de stock. Chaque carte possède deux buts particuliers : Démarrer et Arrêter pour respectivement commencer et terminer l'exécution de la carte.

Une stratégie est une approche, une manière ou un moyen pour réaliser un but cible à partir du but source. Dans une carte, les stratégies correspondent aux différentes façons de réaliser les buts. Une stratégie s'associe au but auquel elle s'applique. Elle a pour objectif principal d'extérioriser la façon d'atteindre ce but puisqu'elle permet de distinguer le but et la façon de le réaliser. En outre, fournir plusieurs stratégies pour atteindre le même but permet de suggérer des façons alternatives de réaliser ce but. Ceci permet plus d'adaptabilité et de souplesse dans l'exécution des processus métiers.

Dans l'exemple de la Figure 6, les demandes d'achats de produits peuvent être créées Manuellement. Les demandes peuvent être générées automatiquement et sont transformées en ordres d'achat à l'aide de la stratégie Par planification. Les deux stratégies Manuellement et Par planification sont des manières différentes de réaliser le but Acquérir des produits.

Une section est un triplet <Bs,Bc,S> composé d'un but source Bs, d'un but cible Bc et d'une stratégie S. Une section exprime la réalisation du but cible en utilisant la stratégie une fois que le but source a été réalisé. Par exemple, dans la Figure 6, l'agrégation du but source Acquérir des produits, du but cible Contrôler le stock et de la stratégie Entrer en stock des produits livrés définit

la section <Acquérir des produits, Contrôler le stock, Entrer en stock des produits livrés>. Ici la stratégie Entrer en stock des produits livrés caractérise le flux du but source Acquérir des produits au but cible Contrôler le stock et la manière de réaliser la cible.

Ceci signifie que les livraisons sont contrôlées par rapport aux ordres d'achat, et envoyées sur le lieu de stockage ou de consommation.

2.3. Architecture ISOA : architecture de système dirigé par un modèle intensionnel exécutable

La figure 7 présente l'architecture du système d'information dirigée par un niveau non technique centré utilisateur. Au sein de cette architecture, nous avons notre architecture à deux niveaux avec cette fois-ci les éléments intentionnel et opérationnel convertis en service intentionnel et service opérationnel. Le service intentionnel est décomposé en deux services distincts. Le service composite, constitué de plusieurs services atomiques et ne pouvant pas être opérationnalisés ; le service atomique, composé d'un seul service et opérationnalisable par le niveau opérationnel. Dans notre approche, les besoins des utilisateurs sont capturés à travers le modèle de la carte puis convertis en service intentionnel atomique opérationnalisable à travers l'architecture orienté services intentionnels (ISOA). L'architecture ISOA est une sorte de boîte noire constitué d'un ensemble de modèle intentionnel exécutable et d'un modèle opérationnel.

L'établissement d'une architecture orientée services intentionnel (ISOA) engendre quatre avantages principaux qui sont :

- Introduction de la notion de service intentionnel et de service opérationnel. Ceci permet une meilleure homogénéité dans l'expression intentionnelle des besoins des clients et celles des services et conduit à repenser la notion de services technologiques et sa « raison d'être » pour s'adapter à l'échelle stratégique et intentionnelle des clients. L'objectif est de mettre en avant le but que ce service permet de réaliser et non la fonctionnalité qu'il permet d'atteindre : solution aux problèmes de discordance conceptuelle entre les besoins des clients et les services techniques. Dans le cadre de l'architecture ISOA, les services intentionnels peuvent être (i) publiés par les fournisseurs dans l'annuaire des services et (ii) localisés par le client dans le cas où ils répondraient à leurs attentes par coïncidence d'intentions.

- Introduction de la notion de variabilité. Le besoin de variabilité dans le monde des services émerge à la suite d'un changement du comportement des utilisateurs, ne voulant plus s'adapter aux capacités des logiciels et préférant que les logiciels se configurent en fonction de leurs besoins. Les variantes correspondent aux différentes manières d'atteindre le but. Etant donné qu'un service intentionnel peut être composé d'autres services intentionnels, ayant chacun son propre but et par conséquent des variantes associées, il en résulte que le service intentionnel est défini comme un réseau de variantes attachées à des points de variation. Comme conséquence,

nous remarquons une variabilité opérationnelle par rapport au niveau intention, puis une composition dynamique au niveau opérationnel dirigé par les intentions. La composition de services est un des atouts majeurs des approches à base de services. Les services sont conceptuellement limités à des fonctionnalités modélisées par une collection d'opérations qui s'apparentent à des méthodes du monde objet. Pour répondre aux exigences de certains types d'application, il est donc nécessaire de combiner un ensemble de services en services plus complexes appelés services composites.

- Introduction de la notion de réutilisation, la réutilisation des services est mise en oeuvre par des mécanismes de réflexivité : un service composite étant un service à part entière, réutilisable dans d'autres compositions de services.
- Evolution et maintenance facilitée de l'architecture à deux niveaux dirigée par un modèle usage et non technique.

2.4 Le Modèle Intentionnel de Services (MIS) et le Modèle Opérationnel de Services (MOS) de l'architecture ISOA

Après définition et construction de l'architecture orientée service intentionnel (ISOA), il est important dorénavant de définir un modèle de représentation des services que nous appellerons MIS (Modèle Intentionnel de Services). Il s'agit d'un modèle exécutable qui permet de représenter, à un niveau intentionnel, les services du système à développer et d'aider un client à exprimer son intention et se concentrer sur les buts qu'un service permet de réaliser. Le modèle MIS définit chaque service intentionnel en tant que brique de construction d'applications en lui associant les connaissances situationnelle et intentionnelle sous la forme d'une interface.

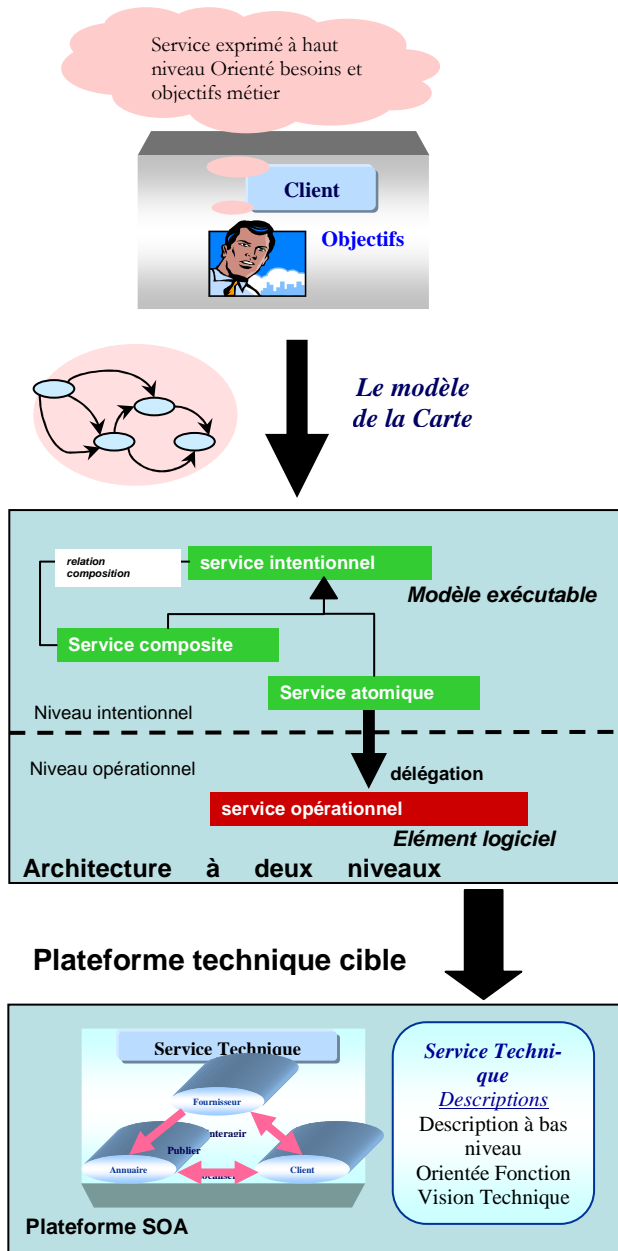


Figure 7. Architecture du système d'information dirigée par un niveau non technique centré sur l'utilisateur

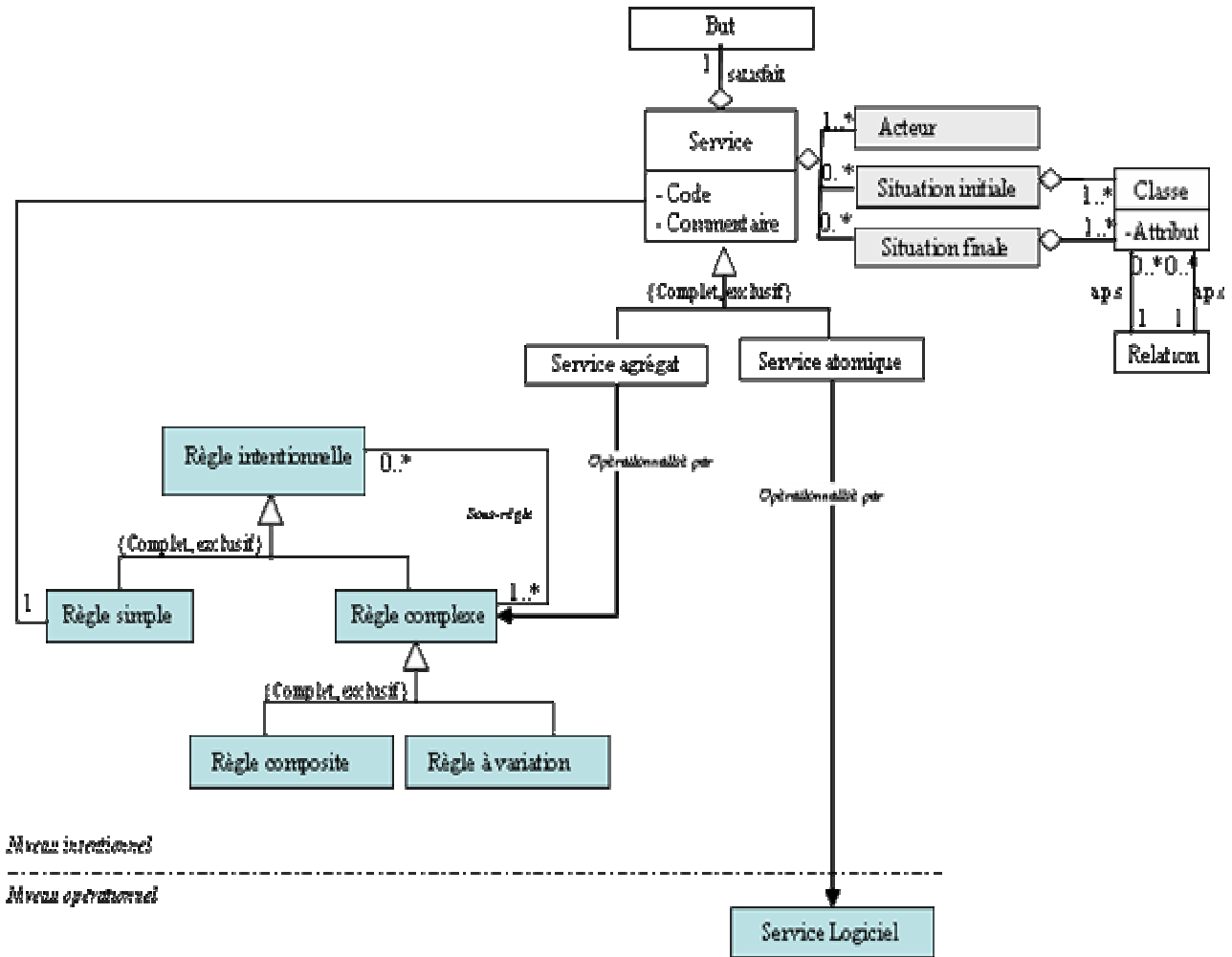


Figure 8. Le modèle exécutable MIS (modèle intentionnel de services)

La Figure 8 met en évidence le concept central du modèle du service intentionnel. La perspective intentionnelle se traduit par le fait qu'un service exhibe une intentionnalité formulée par le but qu'il permet à ses clients d'atteindre.

Un but reflète une intention, un objectif que l'on cherche à atteindre sans pour autant dire comment l'atteindre. Un but est associé à un résultat souhaité matérialisé par un ensemble d'états d'objets. Par exemple le but Payer une réservation par carte de crédit correspond à un résultat escompté : le paiement effectif de la réservation, et plus précisément se décline par l'état de l'objet réservation : l'état réservation. statut= 'payée'.

Dans l'optique intentionnelle de MIS, un service se comporte de façon à satisfaire le but qui lui est associé. Sachant qu'un but est caractérisé par un ensemble d'états d'objets qui traduisent le résultat obtenu par la satisfaction du but, il est usuel de définir le processus de réalisation d'un but par deux ensemble d'états : celui de la situation initiale (au départ du processus de réalisation d'un but) et celui de la situation finale (lorsque le processus de recherche d'atteinte du but est achevé). Nous proposons donc dans MIS, de qualifier le comportement d'un service par (i) deux conditions d'états désignées pré-condition et post-condition et (ii) une règle de transi-

tion. Les conditions sont respectivement des conditions des états des objets de la situation initiale et la situation finale. La Règle de transition permet de préciser les contraintes régissant la réalisation d'opérations entraînant le changement d'états d'objets. Ainsi, un service peut être vu comme la transition d'une pré-condition vers une post-condition résultant de la réalisation du but du service par l'exécution de règles de transition associées à un service. La pré-condition précise les états des objets constituant les conditions devant être satisfaites pour que le but du service puisse être satisfait. La post-condition précise les états des objets obtenus suite à la réalisation du but du service.

Il y a différents types de compositions de services. Au premier niveau de spécialisation les services sont soit atomiques soit agrégats. Un service atomique n'est pas décomposable en d'autres services intentionnels alors qu'un service agrégat l'est. Cette classification prend sa source dans la nature des buts associés aux services.

Un service atomique est associé à un but que l'on qualifie 'd'opérationnalisable', c'est-à-dire pour lequel on est capable de définir une séquence d'actions qui permet de réaliser le but.

Dans notre cas, ce sont les actions qui composent le service atomique qui permettent d'assurer

l'opérationnalisation du but du service atomique. Cela permet de dire qu'un service atomique est exécutable. Par exemple le service permettant de Payer une réservation par carte de crédit est un service atomique car le but qui lui associé est opérationnalisable dans le sens où l'on sait définir un processus de paiement d'une réservation par carte de crédit.

Nous proposons au niveau opérationnel une démarche pour guider l'identification de services qui correspondent aux services intentionnels de type MIS. Ceci est réalisé à l'aide la représentation explicite des services opérationnels qui découlent des services intentionnels dans les termes d'un modèle spécifique MOS (Modèle Opérationnel des Services).

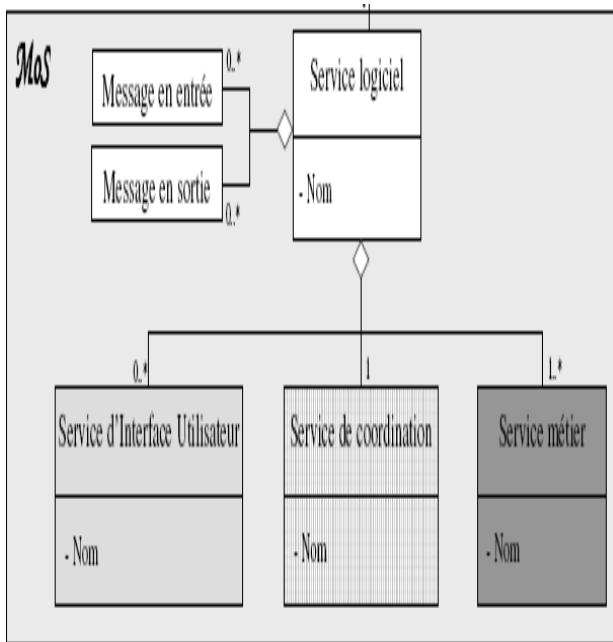


Figure 9. Le modèle opérationnel de service (MOS)

Le service logiciel est le moyen d'opérationnaliser le service intentionnel atomique.

L'exécution du service logiciel permet d'interagir avec le service d'interface utilisateur, qui délègue au service de coordination la réalisation d'une transaction métier complexe et distribuée, qui à son tour, délègue aux services métier pour la réalisation de transaction métier. A la fin de cette chaîne de délégation, la réalisation du but se traduit par le fait que le service d'interface délivre le résultat du service logiciel.

Le service logiciel est caractérisé par les éléments suivants :

- un nom permettant de l'identifier de manière unique,
- Message en entrée : est une structure de données passée en entrée pour adapter le comportement du service logiciel au contexte d'utilisation,
- Message en sortie : est une structure de données représentant le résultat produit et permettant de constater la délivrance du service à l'utilisateur.

- Le service d'interface utilisateur représente la spécification nécessaire au développement de l'application d'avant plan. Ceci peut être fait à l'aide de techniques de développement d'interface web.

Un service d'interaction utilisateur est défini par un nom, et un ensemble ordonné d'interactions que le service d'interface utilisateur doit mettre en oeuvre pour gérer les interactions avec l'application d'arrière plan.

- Le service métier décrit les transactions métier. La spécification ne décrit pas la manière dont les transactions métier doivent être implémentées mais elle s'attache à décrire comment elles sont utilisées indépendamment de leur implémentation. Le service métier est, par conséquent, décrit par son interface.

- Le service de coordination permet, d'une part, de coordonner l'ensemble des services métier à travers un ensemble d'activités ordonnées et, d'autre part, de rendre le service d'interface utilisateur indépendant des services métier mis en oeuvre. Ceci permet de développer des applications d'avant plan indépendamment des applications métier.

Un service de coordination permet l'exécution d'un processus déclenché par l'arrivée d'une action de communication. Il évalue les données en entrée et appelle le service métier concerné. Ce dernier renvoie son résultat au service de coordination.

Les structures du service d'interface utilisateur, du service métier et du service de coordination sont présentées à travers des modèles définis en utilisant la notation UML. Dans ce papier, ces trois modèles : modèle du service d'interface utilisateur, modèle du service métier et modèle du service de coordination ne sont pas représenter.

3. CONCLUSION

Cet article a exploré un certain nombre de problèmes posés par un changement d'échelle visant à permettre une exploitation de SOA au niveau intentionnel (ISOA), c'est à dire celui où les souhaits, besoins et exigences sont formulés par des buts à atteindre. Nous pensons que ce changement d'échelle est nécessaire pour permettre un raisonnement en termes de services et de composition de services tel que les acteurs du monde du business peuvent/souhaitent le conduire.

Les propositions élaborées pour répondre aux problèmes de ISOA et que nous avons présenté, comportent les éléments suivants :

- Une architecture à deux niveaux d'abstractions : niveau intentionnel et niveau opérationnel
- Un processus pour identifier les services intentionnels à partir des besoins des utilisateurs,
- Une démarche méthodologique pour l'opérationnalisation des services intentionnels en services du modèle MOS.
- Un modèle intentionnel de représentation des services : le modèle MIS,
- Un modèle opérationnel de services : le modèle MOS, Nous avons défini un processus pour construire le modèle à partir des besoins des utilisateurs. Ce processus

propose d'élucider les besoins des utilisateurs à l'aide d'une carte des besoins. Nous considérons que le modèle de la carte est adapté à la découverte des services. En effet, il met en évidence les différentes stratégies pour satisfaire le même but et aussi les différentes combinaisons de stratégies et de buts pour atteindre un but. Il aide donc aussi à la découverte de la variabilité des services. Chaque combinaison possible de buts et de stratégies identifie une variante de services possible pour atteindre le but du service global.

Les perspectives futures seraient d'établir un template architectural pour un ISOA prenant en compte plusieurs couches dont l'agrégation d'une couche présentation devenue de nos jours très important dans les applications SOA. L'accroissement des standards, tel que le WSRP (Web services Remote Portlets) et autres technologies, nous amène à constater l'importance de la prise en compte du comportement de l'utilisateur dans une SOA. Une seconde perspective est d'utiliser l'architecture ISOA pour le développement d'un environnement logiciel permettant assister le concepteur lors de la modélisation des services et la génération de la solution logicielle. Enfin, l'enrichissement du modèle MIS par des exigences non fonctionnelles n'est pas en reste.

REFERENCES

- Alonso, G., Casati, F., Kuno, H., and Machiraju, V., 2004. *Web Services: Concepts, Architectures and Applications*. Springer-Verlag, 1st edition, 2004.
- Andrews, T., Curbera, F., Dholakia, H., Golland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., and Weerawarana. S., 2003. *Business Process Execution Language for Web Services (BPEL4WS) version 1.1*, May 2003.
- Azsanjani, A., 2004. *Service-oriented modelling and architecture*.
<http://www.128.ibm.com/developerworks/webservices/library/ws-soa-design1/>, November 2004.
- Jackson, M., 1995 *Software Requirements and Specifications A Lexicon of Practice, Principles and Pejudices*. ACM Press, Addison-Wesley, 1995.
- Papazoglou, M-P., and van den Heuvel, W-J., 2006a. *Business Process Development Lifecycle Methodology*. To appear in *communications of ACM*, 2006.
- Papazoglou, M-P., and van den Heuvel, W-J., 2006b. *Service-Oriented Design and Development Methodology*. *Int. J. of Web Engineering and Technology (IJWET)*, 2006.
- Rolland, C., and Prakash, N., 2000. *Bridging the gap between Organizational needs and ERP functionality*. *Requirements Engineering Journal*, 2000.
- WSCL, 2001. Kuno, H., Lemon, M., Karp, A., Beringer, D., *Conversations + Interfaces = Business Logic*, in *Proc of the 2nd VLDB International Workshop on Technologies for e-Services (VLDB-TES 2001)*, Rome, Italy, 2001.
- WSCI 2002. BEA, SAP, Sun, *Web Service Choreography Interface (WSCI) 1.0*:
<http://www.sun.com/software/xml/developers/wsci/wsci-spec-10.pdf>, 2002.
- Zimmermann, O., Krogdahl, P., and Gee, C., 2004. *Elements of Service-Oriented Analysis and Design*.
<http://www.128.ibm.com/developerworks/webservices/library/ws-soad1/>, June 2004.