

SOA vs MVSOA : Une architecture orientée services multivues

Adil Kenzi, Bouchra El Asri, Mahmoud Nassar, Abdelaziz Kriouile.

Laboratoire Systèmes d'Information, Multimédia et Mobile.
Ecole Nationale Supérieure d'Informatique et D'Analyse de Systèmes
Rabat
Maroc.

Adil.kenzi@gmail.com, {elasri, nassar, kriouile}@ensias.ma

RÉSUMÉ. Ce papier présente le concept de service multivue comme un nouveau concept de modélisation et d'abstraction capable de décrire des services adaptables aux besoins et aux spécificités des utilisateurs. Ce concept donne lieu à une architecture, MVSOA, orientée service multivue où les acteurs principaux d'une architecture orientée service agissent selon les différents profils des utilisateurs. Nous présentons dans cet article les éléments fondamentaux de cette architecture et les extensions des standards associées. Enfin, nous situons notre approche dans le contexte de l'approche MDA.

Abstract. This paper presents the concept of multiview service as a new element of abstraction and modelling able to describe adaptable services that deal with the profile of users. On the basis of this concept, we define the Multiview Service Oriented Architecture (MVSOA) which each element must take into account the user profile. In this paper, we present the fundamental elements of this architecture and their artefacts. Lastly, we situate our approach in the context of the MDA approach.

MOTS-CLÉS : Service Multivue, MVSOA, L'Informatique Orientée Service, MDA

KEYWORDS: Multiview Service, MVSOA, Service Oriented Computing, MDA.

1. Introduction :

L'Informatique orientée service (SOC – Service Oriented Computing) a émergé comme un nouveau paradigme de développement et de maintenance des systèmes distribués flexibles, réutilisables, évolutifs et interopérables. Elle se base sur l'architecture orientée service (SOA) dont l'objectif est d'organiser un ensemble d'applications isolées, en un ensemble de services interconnectés, chacun étant accessible à travers des interfaces et des protocoles de communication standards [7]. Dès lors, l'architecture orientée service se positionne comme une solution pour des environnements ouverts à grande échelle. Dans ce cadre, plusieurs travaux ont abordé la description, l'identification, la conception et le déploiement de services [3][8][9]. En revanche, la complexité inhérente aux besoins croissants des utilisateurs et à leurs spécificités reste un champ d'investigation pour les chercheurs. Par conséquent, des approches permettant la séparation des préoccupations fonctionnelles ont été proposées dans le but de capitaliser les besoins fonctionnels dans des entités modulaires. Dans cette perspective, plusieurs approches ont été menées pour intégrer les vues/points de vue[6]. Ces approches n'emploient pas forcément les termes de point de vue ou de vue mais introduisent des termes sémantiquement proches : sujet, rôle, aspect, etc.

Dans la perspective de combiner les bénéfices du développement par point de vue et celui du développement orienté service, nous proposons le concept du service multivue comme élément fondamental pour le développement des systèmes adaptables aux profils des utilisateurs. La spécificité d'un service multivues est la possibilité de répondre pertinemment aux besoins des utilisateurs selon leurs profils (rôle).

Cet article est organisé comme suit : La section 2 présente des travaux relatifs aux notre. La section 3 présente notre proposition d'une architecture orientée service multivues. La section 4 présente la mise en œuvre de l'architecture MVSOA par les services web dans le cadre de l'approche MDA.

2. Travaux relatifs:

Dans cette section, nous présentons des travaux relatifs suivant deux aspects. Premièrement, nous présentons les travaux concernant la modélisation des applications SOA. Deuxièmement, nous nous focalisons, plus particulièrement, sur les travaux combinant entre les vues et les services pour l'adaptabilité de ces derniers.

Pour la modélisation des applications SOA, (Stojanovic et al.) [8] proposent une approche pour la modélisation et la conception des solutions SOA. Cette approche se base sur le concept « service component » et sur le langage UML. Deux types de « service component » de différente granularité ont été définis : le BSC (Business Service Component) et le ASC (Application Service Component) pour la description de systèmes SOA. Dans le même sens, Zhang et al. [9] introduisent un framework de modélisation pour l'architecture SOA en se basant sur le concept de « service component ». Quant à Johnston et al.[3] et al., ils ont défini un profil UML pour

SOA se composant d'un ensemble de stéréotypes : service, message, spécification, etc. Dans une perspective d'adaptabilité de service, plusieurs travaux ont intégré le **concept de vue, d'aspect, ou de contexte à celui de service**. Ainsi, pour l'adaptation des services composites, Maamar et al. [5] définissent une vue comme étant une image dynamique de la spécification des services web composites selon un contexte donné. Le concept de vue est utilisé pour faire le suivi de l'exécution des services web composites et pour prendre les mesures correctives dans le cas où l'exécution de services ne se conformant pas aux préférences des utilisateurs. Par contre, Fink et al. [1] Définissent une approche à base de vue et un langage VPL (View Policy Language) pour la gestion des droits d'accès aux opérations d'un service. Enfin, pour le concept d'aspect, (Kim et al. 2005)[4] proposent une méthode de conception orientée service avec aspects appelée SODA (Service-Oriented Design with Aspects). Cette méthode définit un modèle de services représenté par le langage UML et ses mécanismes d'extensions ainsi que les réseaux de Petri pour la description du comportement du service.

3. SOA vs MVSOA (MultiView SOA) :

Dans cette section, nous proposons une architecture orientée service multivues. Pour ce faire, nous commençons, par la définition du concept de service multivue, avant de présenter les composants principaux d'une architecture MVSOA.

3.1 Le concept de service Multivues :

Dans un objectif d'une adaptabilité de service, le service multivue a été introduit comme un élément de modélisation capable de décrire les besoins des utilisateurs et leurs droits d'accès le plus tôt dans le cycle de développement des systèmes. D'un point de vue conceptuel, un service multivue est une extension du service. Il fournit des interfaces dont la description et l'accessibilité changent dynamiquement selon la vue active. Pour ce faire, un service multivues permet de fournir, en plus d'interfaces simples, **des interfaces multivues** qui décrivent le comportement du service et son interaction avec ses utilisateurs potentiels (Acteurs) tout en prenant en compte les besoins et les spécificités des acteurs ainsi que leurs droits d'accès.

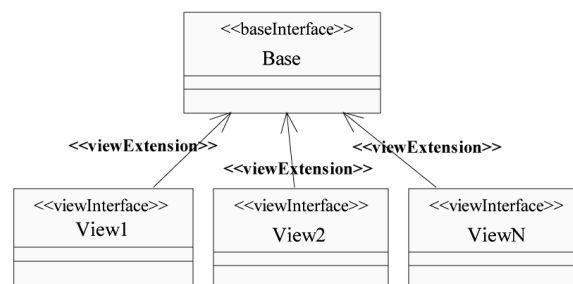


Figure 1 : la structure statique d'une interface multivue

La figure 1 ci dessus illustre la structure statique d'une interface multivue. Une interface multivues est composée d'une interface partagée par tous les points de vues appelée interface de base (stéréotypée *baseInterface*) et un ensemble d'interfaces visuelles (stéréotypée *viewInterface*). Une interface visuelle correspond aux spécificités d'un point de vue et est liée à l'interface de base par une relation de dépendance stéréotypée *viewExtension*. La relation *viewExtension* est une relation de dépendance. Une interface visuelle dépend de l'interface de base dans le sens où les opérations et les messages de l'interface de base sont partagés par toutes les vues. Au moment de l'exécution, l'interface multivue s'adapte à la vue courante en fournissant une interface se composant des opérations et messages de l'interface de base et les opérations et messages d'une interface visuelle « *viewInterface* » d'un service correspondant à la vue courante.

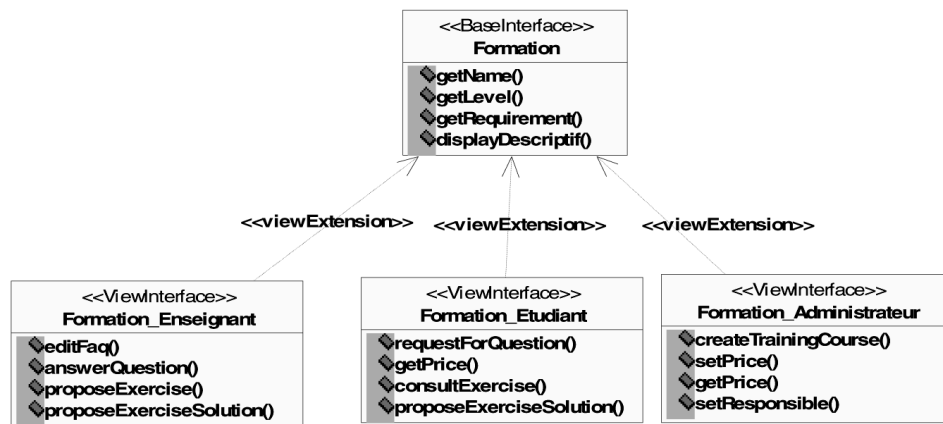


Figure 2 :L'interface multivue du service multivue Formation

Dans le cadre d'une application d'enseignement à distance, nous donnons l'exemple d'un service multivue Formation. Un tel service peut être invoqué selon trois types d'acteurs : Enseignant, Etudiant ou un Administrateur (cf. Figure 2). Ce service est spécifié par la spécification « Formation ». une telle spécification contient la description de l'interface de base qui reflète les fonctionnalités communes entre les trois types d'acteur interagissant avec le service « Formation » encapsulée dans la « BaseInterface » et les fonctionnalités concernant chaque type d'acteurs encapsulées dans les « ViewInterface ». Ainsi, pour tous ces trois types d'acteurs, ils peuvent exécuter les opérations à l'interface de base «baseInterface » Formation. Par contre, les interfaces associées à chaque type d'acteur : « viewInteface » Formation_administrateur, « viewInteface » Formation_Etudiant, « viewInteface » Formation_Enseignant associée respectivement à l'administrateur, à l'étudiant et à l'enseignant ne sont accessibles que si le type d'acteur associé est actif. Par exemple, un acteur de type administrateur ne peut invoquer que les opérations appartenant à « « viewInteface » Formation_administrateur alors qu'un acteur de type Etudiant ne peut invoquer que les opérations appartenant à « viewInteface » Formation_Etudiant

3.3 SOA vs MVSOA (MultiView SOA) :

L'architecture orientée service (SOA) définit trois acteurs et trois opérations : **le fournisseur de service** qui possède l'implémentation de service et la description du service **publie** (l'opération<<publish>>) cette dernière dans **l'annuaire des services**. L'Annuaire de service permet, par ailleurs à **un client de service** de trouver les services correspondant à ses besoins (l'opération<<find>>) et de l'invoquer en lui transmettant les données métiers requises (l'opération <<Bind>>). Une telle architecture ne prend pas en compte le profil de l'utilisateur interagissant avec le service. Pour répondre à cette limite, nous avons introduit le concept de service multivue pour la prise en compte des profils des utilisateurs. Pour intégrer le service multivue et le rendre opérationnel, nous avons étendu l'architecture orientée service appelée MVSOA (Multiview SOA) (cf. figure 3) par l'ajout d'un module de correspondance. Un tel module joue un rôle d'intermédiaire entre le client de service et l'annuaire et entre le fournisseur de service et l'annuaire de service. En relation avec le fournisseur de service (*l'opération publish*), le module de correspondance récupère la description du service multivue qui décrit en plus des interfaces simples, les interfaces multivues, stocke les informations concernant les profils d'acteurs et leurs interfaces correspondant, ensuite il normalise la description du service et la publie dans l'annuaire de service. En relation avec le client de service, le module de correspondance récupère la requête du client (*l'opération Find (Service, Profil actor)*) qui contient en plus des besoins du client de service en terme de services, les informations concernant le profil du client de service. Ensuite, il cherche dans l'annuaire de service les services correspondant aux besoins du client de service et il les adapte au profil du client de service avant de l'envoyer au client de service.

La figure 3, ci-dessous, présente l'architecture MVSOA :

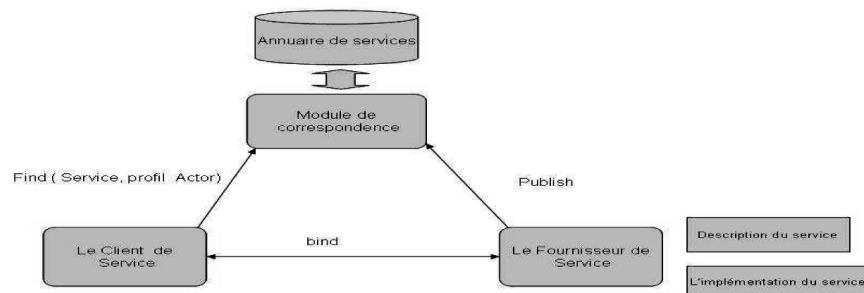


Figure 3 : l'architecture MVSOA

4. Mise en œuvre de MVSOA par les services web dans le cadre de l'approche MDA :

L'architecture orientée service est réalisée fondamentalement par la technologie des services web qui fournit l'infrastructure nécessaire pour le développement et le déploiement de services. Cette infrastructure se base sur un ensemble de standards SOAP, WSDL, UDDI permettant respectivement l'invocation, la description et la découverte des services. Aussi, plusieurs frameworks technologiques (.Net, J2EE,...) ont fourni les API nécessaires pour implémenter les services web. Dans le but de s'abstraire de ces technologies et standards d'une part, et de la prise en compte de l'utilisateur d'autre part, nous avons introduit le concept de service multivue comme une entité de modélisation et d'abstraction dans les précédentes sections.

Dans cette section, l'objectif principal est de traiter l'implémentation des services multivues en tenant compte les standards de la technologie des services web. Nous commençons tout d'abord par présenter une extension du Protocole WSDL appelée MVWSDL pour pouvoir décrire les services multivues. Ensuite, nous présentons le processus de mise en œuvre des services multivues selon l'approche MDA.(Model Driven Architecture).

4.1 MVWSDL : une extension de WSDL pour la description des Services multivues :

Dans le but de la description et la spécification des services multivues, nous nous sommes basés sur une légère extension du protocole WSDL. Ce protocole qui définit un schéma XML standard pour la description des services. Ce schéma XML se base sur six éléments principaux (types, messages, portType, Binding, Port, Service) permettant de définir les interfaces de services, leurs opérations, les paramètres d'entrée/sortie de chaque opération, le type de ces paramètres ainsi que les points d'entrée (URL) de ces opérations. Cependant, ce protocole ne permet pas la prise en compte du profil de l'acteur interagissant avec le service. Ainsi, nous avons défini une extension du protocole WSDL, appelée MVWSDL (MultiView WSDL), pour la description d'un service multivue. L'objectif de cette extension est d'une part, de décrire dans un seul document XML l'ensemble des interfaces fournies du services qu'elles soient simples ou multivues. D'autre part, elle permet d'adapter cette description à l'exécution suivant le profil de l'utilisateur interagissant avec le service en fournissant à chaque utilisateur la description WSDL standard correspondant à son profil.

Dans l'objectif de génération de code qui s'inscrit dans les principes et les standards de l'approche MDA, nous avons établi le métamodèle de MVWSDL comme une extension du métamodèle WSDL pour la prise en compte du type d'acteur interagissant avec le service via l'ajout de la métaclasse « actor » au métamodèle WSDL. Une telle métaclasse permet de définir le type d'acteur interagissant avec le service, et elle s'associe aux différentes métaclasses du métamodèle WSDL qui doivent s'adapter aux profils des utilisateurs.

4.2 Processus de mise en œuvre des services multivues selon MDA:

Notre approche se base sur le concept de service multivue comme une entité de modélisation permettant de capturer les besoins et les spécificité des utilisateurs sans tenir compte les technologies et les standards. Ainsi, nous établissons en premier lieu un modèle métier à base de services multivues reflétant la structure et les fonctionnalités des systèmes. Ce modèle métier définit le modèle indépendant de la plateforme (PIM) selon le vocabulaire MDA. Une fois notre PIM à base de services multivues élaboré, nous définissons les règles de transformations nécessaires pour transformer le modèle source en un modèle Cible Web Service. Pour ce faire, deux types de transformations ont été définis. La première consiste à transformer le modèle source à base de service multivues en un modèle Cible conformément au métamodèle MVWSDL pour ensuite générer un fichier XML permettant la description des interfaces du service. La deuxième transformation consiste en la transformation de notre modèle source vers le modèle J2EE dans un but de génération du code java qui constitue la réalisation du service multivues. La figure ci dessous illustre globalement notre processus de transformation :

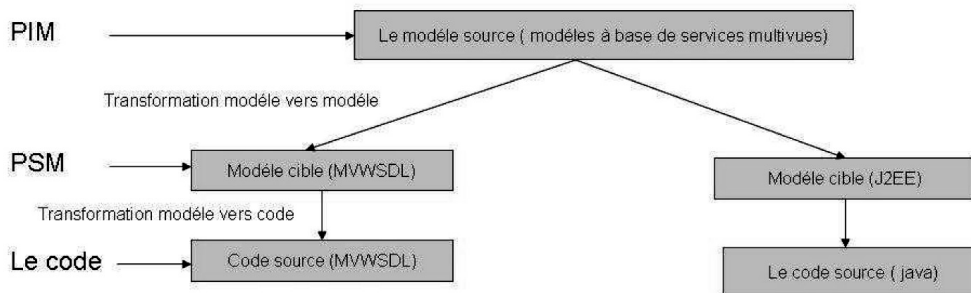


Figure 6 : Le processus de transformation de notre approche

5. Conclusion :

Dans un objectif de personnalisation, d'adaptabilité et de séparation des préoccupations fonctionnelles, nous avons proposé dans le cadre de cet article, une approche à base de services multivues pour la modélisation des systèmes. Premièrement, nous avons défini le concept de service multivues qui permet de décrire pertinemment les besoins fonctionnels des différents types d'utilisateurs et leurs droits d'accès. Nous avons, ensuite, étendu l'architecture orientée service SOA pour la prise en compte du service multivue en illustrant son fonctionnement. Enfin, nous avons situé notre approche dans le cadre de l'approche MDA.

Nos travaux de recherche, en cours, consistent en la mise en œuvre de tels services. Ainsi, nous sommes entrain de développer et d'implémenter les règles de transformation de modèles pour la l'automatisation de la génération de code.

Une direction future pour notre travail est **la composition des services multivues** dans le cadre de l'approche MDA. En effet, le développement et l'adoption de la technologie associée aux services permettent d'implanter de nouvelles applications avec valeur ajoutée en faisant la composition des services existants. L'objectif à moyen terme est de permettre le développement des systèmes d'informations flexibles et adaptables à base de services multivues.

6. REFERENCES:

- [1] Fink T, Manuel Koch, Cristian Oancea: Specification and Enforcement of Access Control in Heterogeneous Distributed Applications. ICWS-Europe 2003: 88-100.
- [2] Fuchs M, Adapting Web Services in a Heterogeneous Environment, Proceedings of the IEEE International Conference on Web Services (ICWS'04), 2004: 656-664
- [3] Johnston S , K, Alan W. Brown: A Model-Driven Development Approach to Creating Service-Oriented Solutions. ICSOC 2006: 624-636
- [4] Kim T., Chang C .K., "Service-Oriented Design with Aspects (SODA), Proceedings of the 2005 IEEE International Conference on Services Computing (SCC'05), 2005.
- [5]Maamar Z., Benslimane D., Chirine Ghedira: A View-based Approach for Tracking Composite Web Services. ECOWS 2005: 170-181
- [6] Nassar M., Coulette B., Guiochet J., Ebersold S., El Asri B., Crégut X., Kriouile A, « Vers un profil UML pour la conception de composants multivue ». Revue RSTI-L'Objet, vol.11 –n°4/2005.
- [7]Papazoglou M. P., « Service Oriented Computing : Concepts, characteristics and Directions », actes de la 4° conférence WISE'03, Rome, 10-12décembre 2003, Italie, IEEE Computer Society, p. 3-12.
- [8]Stojanovic Z., A. Dahanayake, H. Sol, " Modeling and design of Service Oriented Architecture", IEEE International Conference on Systems, Man and Cybernytics, 2004.
- [9] Zhang T., S.Ying, S. Cao, and X.Jia, "A Modeling Framework for Service-Oriented Architecture", Proceedings of the Sixth International Conference on Quality Software (QSIC'06), 2006.