



**HAL**  
open science

## Towards an Automatic Adaptation of Heterogeneous Multimedia Mobile Applications

Adel Alti, Philippe Roose, R. Saffidine, Sébastien Laborie

► **To cite this version:**

Adel Alti, Philippe Roose, R. Saffidine, Sébastien Laborie. Towards an Automatic Adaptation of Heterogeneous Multimedia Mobile Applications. NOTERE / CFIP 2012, Oct 2012, Anglet, France. pp.79-87. hal-00751515

**HAL Id: hal-00751515**

**<https://hal.science/hal-00751515>**

Submitted on 14 Nov 2012

**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.

# Towards an Automatic Adaptation of Heterogeneous Multimedia Mobile Applications

A. Alti<sup>1</sup>, P. Roose<sup>2</sup>, R. Saffidine<sup>1</sup>, S. Laborie<sup>2</sup>

<sup>1</sup> Computer Science Department, Engineering Faculty, Ferhat ABBAS University of Setif, 19000 Setif, Algeria

<sup>2</sup>LIUPPA / IUT Bayonne, 2 Allée du Parc Montaury, 64600 Anglet - France

**Abstract**—The key aspect of any mobile multimedia application is the multimedia document. Currently, multimedia documents can be executed on multiple platforms (laptop, Smartphone, tablet, etc.), that resulting in a birth of new information system called pervasive. The different execution contexts of a multimedia presentation introduce different constraints on the presentation itself. This includes device constraints and user preferences, resulting to the overall system heterogeneity increase. In this paper, we propose a specific ontology for adapting multimedia applications. Thus, we propose rules allowing automatic generation of dynamic and quality assembly of heterogeneous components. The proposed ontology has the great advantage to offer to users a flexible infrastructure where they can easily govern the response time and the quality assembly of their own applications at runtime.

**Keywords**—Ontology; Context; QoS; multimedia application, dynamic composition, mobility;

## I. INTRODUCTION

Ces dernières années, la technologie des terminaux mobiles a connu un développement très rapide. Le téléphone mobile ne sert plus uniquement à de simples communications vocales ou textuelles, les assistants personnels ne sont plus de simples gadgets de planning et d'organisation, les ordinateurs de poche ne sont plus isolés de l'Internet. La fusion de ces terminaux au sein de Smartphone donne naissance aux systèmes d'informations dits «pervasifs», permettant de rendre l'information disponible partout et à tout moment. Dans le cadre de l'informatique ubiquitaire, l'environnement d'exécution d'une application est constitué de machines hétérogènes (ordinateur portable, PDA, Smartphone, etc.) en ressources matérielles (*taille écran, modes d'interactions, mémoire, batterie, interfaces réseaux, etc.*) appartenant à des utilisateurs avec des besoins différents et manipulant des médias de différents types (*vidéo, son, image, texte*). Ces caractéristiques imposent de structurer l'application en une organisation d'entités logicielles relativement indépendantes qui coopèrent et interagissent afin de faciliter son adaptation au contexte d'utilisation. L'adaptation de documents multimédia est considérée comme une solution efficace, adéquate et intéressante à ces problématiques. Elle consiste en la transformation et la personnalisation du document multimédia afin qu'il soit compatible avec les préférences de l'utilisateur et les contraintes du contexte d'exploitation.

On distingue généralement trois types d'adaptations des documents multimédias: le transcodage (*modification du format d'encodage, JPEG vers PNG par exemple*), le transmodage (*modification de la modalité, Texte vers Son par*

*exemple*) et la transformation (*modification de résolution, résumé d'une vidéo, traduction d'un contenu en différentes langues, etc.*). La difficulté, lorsque de telles applications traitent et échangent des documents, est l'automatisation des adaptations nécessaires, ce qui se traduit par la recherche et l'assemblage des services assurant ces fonctions d'adaptation (*transcodage, transmodage, transformation*).

Dans ce travail, nous proposons une approche d'adaptation sémantique automatique fondée sur l'utilisation d'ontologies. Il consiste à mettre en évidence les besoins d'adaptation d'une part, et d'autre part faciliter la recherche et l'identification des services afin d'appliquer les adaptations nécessaires. Au sein de notre travail, différentes catégories d'informations doivent être gérées : 1) - les informations contextuelles liées aux interactions entre l'utilisateur et l'appareil mobile, les préférences utilisateurs, la localisation et les préférences de qualité ; 2) - les informations liées aux constituants des documents (type, format) ainsi qu'au document multimédia dans sa globalité sachant que ces informations doivent être indépendantes des caractéristiques des multiples plateformes hétérogènes.

L'idée clé de l'ontologie proposée est de fournir un niveau sémantique qui permet la génération automatique d'un assemblage de qualité des composants hétérogènes. Il prévoit également la composition dynamique de la sélection de services basée sur une métadonnée de paramètres de contexte (*utilisateur, environnement, dispositif, et le contexte fournisseur de service*) et surtout de mettre en place des relations de dépendances, de substitutions (*un service peut « jouer » le rôle d'un autre...*) et d'équivalences.

Notre travail consiste à 1) – définir une ontologie décrivant les informations sémantiques des services d'adaptation multimédia (*nom, localisation, catégorie, rôle*) et les caractéristiques coût/qualité des services d'adaptation (*temps moyen d'exécution du service d'adaptation, temps moyen de transfert et le pourcentage de qualité de sortie par rapport à l'entrée*), 2) - un moteur d'inférence chargé d'optimiser et de planifier l'adaptation à mettre en œuvre sur un serveur multimédia ; 3) - utiliser des services web pour exécuter physiquement l'adaptation. Nos objectifs visent donc à mieux capter la sensibilisation des services aux ressources et à mieux contrôler la qualité des services au niveau sémantique, plus précisément de fournir la meilleure composition de services d'adaptation via des propriétés sémantiques.

Le reste de cet article est organisé comme suit : la section 2 présente nos motivations et d'autres travaux connexes. La section 3 détaille notre approche. La réalisation est décrite

dans la section 4. La section 5 propose quelques scénarios possibles. Enfin, nous concluons dans la section 6.

## II. MOTIVATIONS ET TRAVAUX CONNEXES

### A. La nécessité et l'importance des ontologies

Pour étudier l'importance de l'ontologie du point de vue sémantique, il y a quatre raisons :

- *Homogénéité*: différentes informations contextuelles et différentes qualités de services (QoS) utilisant la même ontologie, confère à l'ontologie un rôle d'homogénéité au moins pour les informations de qualité de service et les informations de contexte qu'elle décrit.
- *Interopérabilité*: les ontologies fournissent un bon moyen pour traiter l'interopérabilité sémantique entre les représentations hétérogènes de service web comme les paramètres de QoS et les paramètres de contexte par une représentation sémantique commune.
- *Expressivité* : la plupart des langages d'ontologie tels que RDF(S), DAML-S et OWL-S sont efficaces avec une expressivité élevée pour mieux exprimer la sémantique des divers dispositifs portables (PDA, Smartphones, iPod) et divers paramètres de QoS.
- *Facilité de détection des problèmes d'hétérogénéités*: les ontologies rendent la détection automatique des problèmes d'hétérogénéités générés par la manipulation des différents types de données multimédia. La composition dynamique des services hétérogènes devient plus facile, plus rapide et automatique. Le principal mécanisme pour la détection des problèmes d'hétérogénéités des services est le *service registry* et la sémantique peut être utilisée pour la génération automatique d'assemblages des services de qualité.

### B. Travaux connexes

Plusieurs efforts ont été dépensés pour le support de l'adaptation et l'assemblage des services. Nous citons en premier lieu les standards OWL-S [12], WSMO [8], DAML-S [11], ConteXtML [9], CWSC4EC [6], CxG [10]. Ces langages fournissent des moyens de description de haut niveau des services, des techniques pour vérifier la composition de deux services et l'assemblage sémantique des services. Notre approche tient compte des limites des approches existantes, ce qui suggère des mécanismes pour fournir une composition dynamique des services d'un point de vue technique (type et format des données multimédia) mais également d'un point de vue sémantique (lieu de service, rôle de service, catégorie). En effet, la liste exhaustive des assemblages possibles de composants ne peut pas être a priori connue, il faut donc trouver des moyens sémantiques basés sur des paramètres de métadonnées contextuels (*l'utilisateur, l'environnement, le dispositif et les informations contextuelles liée aux services*) pour rechercher des « types » de services pouvant constituer une application donnée. Boukadi et al. [6] exploitent une nouvelle catégorie de contexte dans un environnement de collaboration des entreprises. Ils se focalisent sur une approche

à base d'ontologie pour décrire et sélectionner des services web. Néanmoins, les ontologies proposées se focalisent uniquement sur des informations contextuelles nécessaires à la collaboration des services, et de ce fait, elles ne profitent pas réellement des concepts liés aux qualités des services et leurs sémantiques qui permet une meilleure composition des différents services vis-à-vis des contraintes contextuelles.

Récemment, Keskes et al. [2] proposent un modèle de sélection automatique du meilleur fournisseur de service. Ils se focalisent sur la composition du contexte avec l'ontologie QoS pour donner un ensemble des paramètres de QoS. L'ontologie considérée comme perspective permet une bonne interopérabilité entre les services hétérogènes et offre une meilleure amélioration des QoS. Cependant, elle ne gère pas la composition automatique des services en utilisant les QoS multidimensionnelles et ne traite pas le problème des ressources limitées des appareils mobiles. Lamolle et al. [7] introduit une plateforme à base d'ontologies pour le déploiement et la configuration des systèmes multimédia reliés automatiquement. Basé sur des normes bien connues, ce système intègre le multimédia, la distribution et le concept de plateforme. Ce travail propose une excellente gestion de la qualité de service et le déploiement, mais ne gère pas l'adaptation sémantique dans des contextes complexes.

Dans la plupart des plateformes d'adaptation et d'auto-adaptation, par exemple dans [3, 4, 5, 14], nous trouvons qu'ils manquent de génération automatique d'assemblages de composants hétérogènes à un niveau sémantique.

Notre contribution dans cet article se démarque des travaux précédents. Dans notre approche, l'ontologie permet la détection automatique des points d'hétérogénéité entre composants/services, de plus l'utilisation des moteurs d'inférence permet la résolution de ces hétérogénéités. Derdour & al. [1] proposent un profil UML permettant de mettre en évidence les besoins d'adaptation. Elle permet la description d'architectures logicielles à l'aide d'un profil UML dédié exprimant un système logiciel comme une collection de composants qui manipulent différents types et formats de données, et qui interagissent entre eux par l'intermédiaire de connecteurs incluant les connecteurs d'adaptation. Il reste la recherche et l'identification des services afin de résoudre les adaptations nécessaires. Pour ce faire, il est nécessaire de :

- développer une ontologie de données contextuelles ;
- développer une ontologie de périphériques mobiles (Smartphone, Tablet, etc.) capables de traiter des données multimédia ;
- à partir des 3 opérateurs de base que sont les transcodage/transmodage/transformation, montrer que toute adaptation de données multimédia se ramène à une succession de telles opérations (avec les bons paramètres à définir) ;
- développer un service web qui, en se basant sur les ontologies de données multimédia et celle des périphériques mobiles en déduit les adaptations à mettre en place et les sélectionne parmi une collection disponible.

### III. ONTOLOGIE DÉVELOPPÉE

#### A. Le modèle de service

Les services d'adaptations modifient les données produites/requises et donc provoquent des problèmes d'hétérogénéités de ces dernières (problème de format et de type des données, problème de résolution). Notre objectif visent à mieux sélectionner les composants d'adaptation afin de fournir une solution d'assemblage la meilleure possible pour répondre aux préférences des utilisateurs. Figure 2 présente la structure de notre ontologie. Elle permet aux fournisseurs de services la publication des métadonnées au sujet de leurs services avec leurs contraintes de contexte dans le registre UDDI. Les fournisseurs vont instancier les classes de l'ontologie et publier les individus résultant sous forme de fichiers OWL. Figure 1 illustre notre ontologie qui a les concepts suivants

- *Service* : a un *nom*, une *version* et un *identificateur* unique, longueur, numéro de chemin;
- *Service\_Role* = {Transmoding, Transcoding, redimensionner, compression, etc.}.
- *Lieu du service*: la ville où le service est localisé.
- *Temps de service*: l'heure où le service est exécuté.
- *Contenu multimédia* = {texte, image, son, vidéo}.
- *Step*: *Start*, *Current*, *Passé*, *Finish*.
- *Service\_Relation*: représente les relations sémantiques de *dépendances*, la relation *d'équivalence* et la relation de *substitution*. Les relations sémantiques permettent de guider l'assemblage automatique dans des applications hétérogènes. Lorsque la configuration de service n'est pas possible, l'ajustement peut être effectué en remplaçant un ou plusieurs services de l'assemblage par des services offrant des fonctionnalités et localisation identiques mais proposant des qualités de service différentes.
- *Relation de dépendance* : nous distinguons les dépendances de *successions* et les dépendances de

*prédécesseurs* (par exemple, le service compression est toujours suivi par un service de décompression).

- *Relation d'équivalence* : est une relation sémantique entre deux services jouant le même rôle, ayant la même catégorie et localisé dans la même ville.
- *Relation de substitution* : cette relation apparaît par exemple dans le cas de l'existence de plusieurs versions d'un service.
- *Qualité de service (QoS)* : la liste des paramètres de QoS. Ce concept ne concerne pas seulement les services mais aussi les ressources du périphérique où l'exécution a lieu.
- *Service d'adaptation* : Cette classe consiste à transformer un objet multimédia en un autre objet multimédia satisfaisant un profil donné. On distingue :
  - *Adaptation technique* : cette adaptation se divise en trois parties comme suit :
    - *Transmodage* : permet de changer le type de média. Par exemple : *JPEG vers PNG*.
    - *Transcodage* : permet de changer le format du média sans changer son type. Par exemple : *Texte vers Son*.
  - *Adaptation sémantique*: permet de modifier le contenu du média sans changer son type et son format. Par exemple, résumé d'une vidéo ou traduction d'un texte.
  - *Service de passerelle*: permet de changer le protocole de communication. Par exemple: *WIFI vers GSM*.
- *Paramètres contextuelles* : Chaque catégorie de contexte a des paramètres de contexte. Par exemple, le contexte lié au dispositif matériel est une collection de paramètres : la taille de la mémoire, la puissance de CPU, la bande passante, le cycle de vie de la batterie, etc. Certains paramètres de contexte peuvent utiliser des termes sémantiquement similaires, par exemple la *puissance du processeur* et la *vitesse du processeur*.

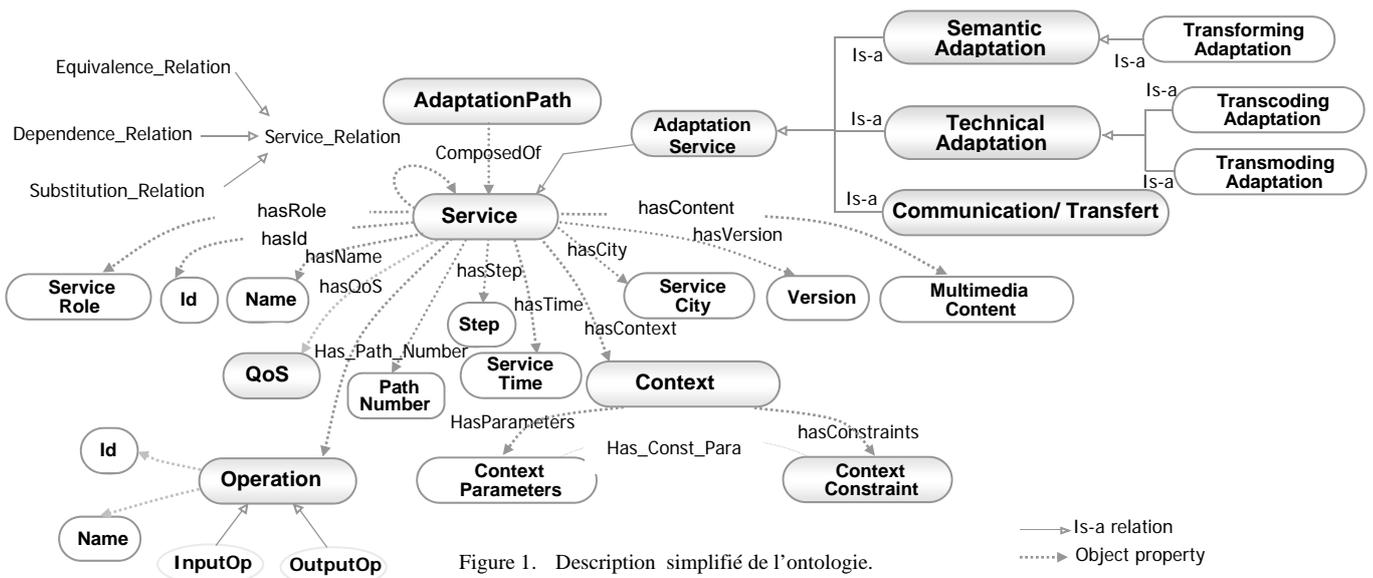


Figure 1. Description simplifiée de l'ontologie.

- *Contrainte contextuelles*: se compose des expressions sémantiques simples ou complexes. Par exemple, une contrainte de contexte peut être comme suit : *bande\_passante = Haut et Taille\_mémoire = Bas et location\_utilisateur = Proche\_Maison*
- *Expression contextuelle* : est une expression qui se compose des : paramètres de contexte, de l'opérateur logique et de la valeur logique.

### B. Le modèle de contexte

Le contexte est représenté par toutes informations qui peuvent être collectées auprès du *document multimédia* (type, format, résolution et taille), du *hardware* (type de dispositif et taille de l'écran) et de l'*utilisateur* (préférences). Notre ontologie utilise ces informations pour choisir la qualité de service dans un environnement dynamique, en confrontant un *contexte requis* (utilisateur, service) avec le *contexte fourni* (nœud mobile). La figure 2 montre les sous-classes de base de la classe contexte de l'ontologie proposée.

- *Contexte utilisateur*: un profil décrit un ensemble de contraintes d'adaptation. Pour qu'un document multimédia puisse être exécuté sur une plate-forme disposant d'un profil, il est primordial que ce document satisfasse l'ensemble des contraintes imposées par ce profil. Le contexte utilisateur décrit ses capacités logicielles et matérielles au sein d'un profil : *la résolution, le format, la taille*, etc. Les propriétés importantes comprennent également les préférences de l'utilisateur : *âge, langue et localisation*.
- *Contexte lié aux appareils mobiles* : Les composants matériels sont des appareils mobiles comme les tablettes ou Smartphone, et chacun d'entre eux a une taille d'écran différente à l'autre.

- *Contexte de document* : Le document est l'information élémentaire recherchée par le client. Cette information élémentaire peut avoir différentes formes : *texte, image, vidéo et son*.

- *Document* : Texte, Image, Son, Vidéo.
- *Texte* : alignement, font, couleur, format.
- *Image* : hauteur, largeur, résolution, taille, nombre de couleurs, format.
- *Vidéo* : titre, couleur, résolution, taille, format.
- *Son* : Fréquence, taille, résolution, format.

- *Contexte d'un service* : décrit les propriétés fonctionnelles et non fonctionnelles d'un service. La qualité de service inclut les informations sémantiques comme les pré-conditions, les effets et les post-conditions. Le rôle de service (transcodage, compression, décompression, rédimensionner) et le niveau de confiance.

### C. Le modèle de qualité du service

Les paramètres de qualité du service dans notre ontologie sont comme le *Coût* et le *Bénéfice* (Figure 3).

- *Coût*: on distingue deux critères : Temps moyen d'exécution du service d'adaptation et temps moyen de transfert d'un document adapté.
- *Bénéfice* : pourcentage de qualité de sortie par rapport à l'entrée.

Deux types de qualité de service : *statique* et *dynamique*. La gestion de la QoS statique est assurée par un processus de choix entre plusieurs services fournissant différentes qualités, tandis que la gestion de la QoS dynamique est assurée par le connecteur d'adaptation qui manipule les paramètres du service d'adaptation (type, format, contenu\_multimédia, rôle, localisation, temps) afin de satisfaire au contexte d'exécution.

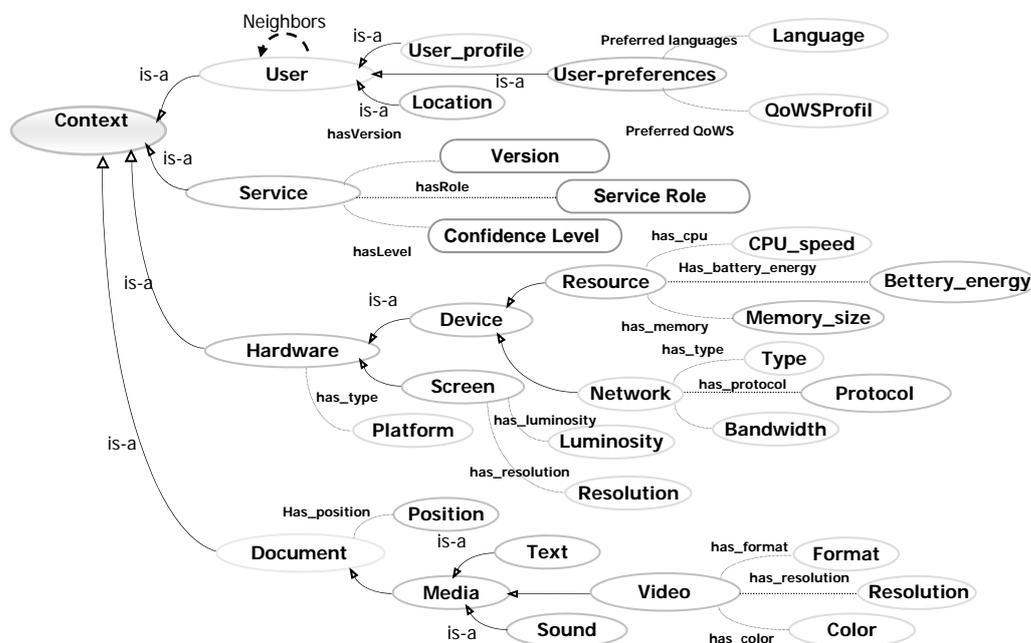


Figure 2. Catégories des contextes.

Les contrats de niveau de service peuvent contenir de multiples mesures de performances (du service) correspondant aux objectifs du niveau de service. Prenons, par exemple, l'adaptation des images, les paramètres que l'on mesure dans ce cas sont généralement :

- TA : (Temps d'Adaptation) : temps moyen d'exécution du service d'adaptation.
- TT : (Temps de Transfert) : temps moyen de transfert d'une image adaptée.

QA : (Qualité d'adaptation) : pourcentage de qualité de la sortie par rapport à l'entrée.

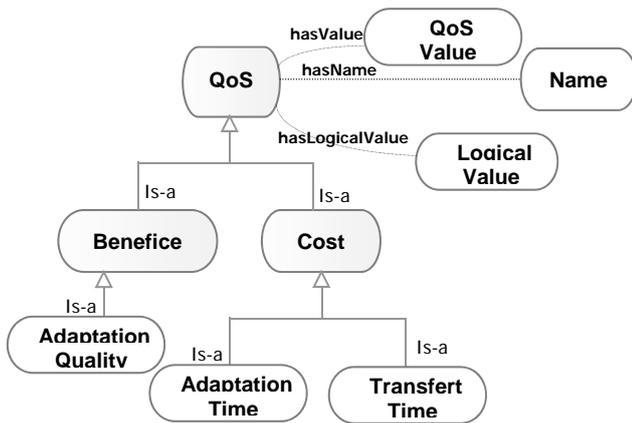


Figure 3. Description des paramètres de qualité

#### D. Profil différence

Ce concept contient les conflits entre les préférences du client et la ressource. Après avoir extrait les caractéristiques du profil et de la ressource, on a deux descriptions la première du profil et l'autre de la ressource. On fait la comparaison entre ces deux descriptions, et on aura la différence entre les utilisateurs et la ressource.

#### E. Le modèle de processus d'adaptation sémantique

Un processus d'adaptation sémantique est composé des services, et il est défini en deux sous-classes (Figure 4) : *ensemble des services d'adaptation atomiques* et *chemin d'adaptation*.

- *Services d'adaptation* : permet d'assembler les services qui répondent aux préférences de l'utilisateur et aux contextes d'utilisation.
- *Chemins d'adaptation* : permet de regrouper les services d'adaptation qui se trouvent dans « Ensemble des services d'adaptation », tel que la sortie d'un service et l'entrée de l'autre. De plus, il faut calculer les temps d'adaptation et les temps de transfert de chaque chemin et l'ajouter dans *Path\_Cost*, ainsi il est possible de mesurer la qualité d'adaptation et d'indiquer le résultat dans *Path-Benefice*. Après avoir trouvé les différents chemins d'adaptation, on sélectionne le chemin optimum offrant une meilleure qualité.

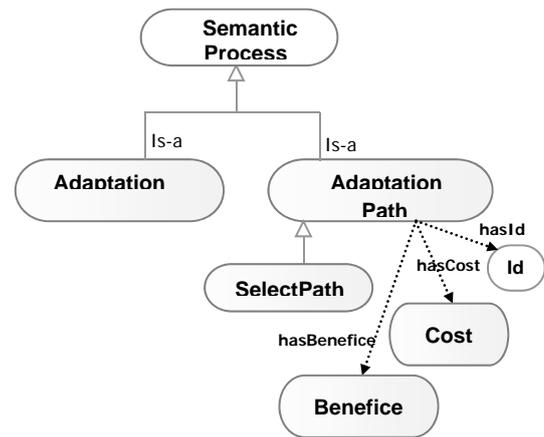


Figure 4. Processus d'adaptation sémantique.

## IV. ARCHITECTURE DU SYSTEME D'ASSEMBLAGE DYNAMIQUE DES SERVICES DE QUALITE

### A. Architecture générale du système

Le système proposé est un système générique, dynamique et interopérable en ce qui concerne la gestion de la qualité des services web sensibles aux contextes. Le système facilite la découverte sémantique et l'interopérabilité des services web, gère les qualités des services touristiques. L'approche proposée est présentée dans la figure 5.

Il offre deux fonctionnalités : la première est la découverte des services de qualité. Cela prend comme entrée la requête de l'utilisateur, et renvoie la liste des services à partir de l'annuaire des services. La deuxième est de permettre aux fournisseurs de services de publier des métadonnées à propos de leurs services de façon dynamique et de définir leurs propres spécialisations des classes en se basant sur notre ontologie. Les composants de l'architecture sont :

- *Client* : c'est le dispositif que l'utilisateur utilise. Ce sont des ordinateurs fixes, des ordinateurs portables, tablettes, Smartphone, etc.
- *Serveur multimédia* : il est constitué d'un planificateur, Plan d'adaptation, Moteur d'inférence, Ressources multimédia et Service Web Sémantique;
- *Ressources multimédia* : un composant doit permettre d'obtenir une *description du document multimédia* à adapter (source, nature...);
- *Moteur d'inférence* : il prend en entrée la description du document multimédia et la description du contexte et produit un *plan d'adaptation*, c'est-à-dire la description d'une série d'étapes élémentaires, éventuellement soumises à des conditions, exécutées en parallèle ou en séquence, qui permettent d'obtenir le document adapté;
- *Exécutif* : est un service web qui exécute le plan et supervise l'exécution des services retenus; à la fin de l'exécution, il fournit le résultat de l'adaptation ou un lien vers ce résultat ;
- *Service Web Sémantique* : il est composé d'un service d'annuaire UDDI et d'une Ontologie.

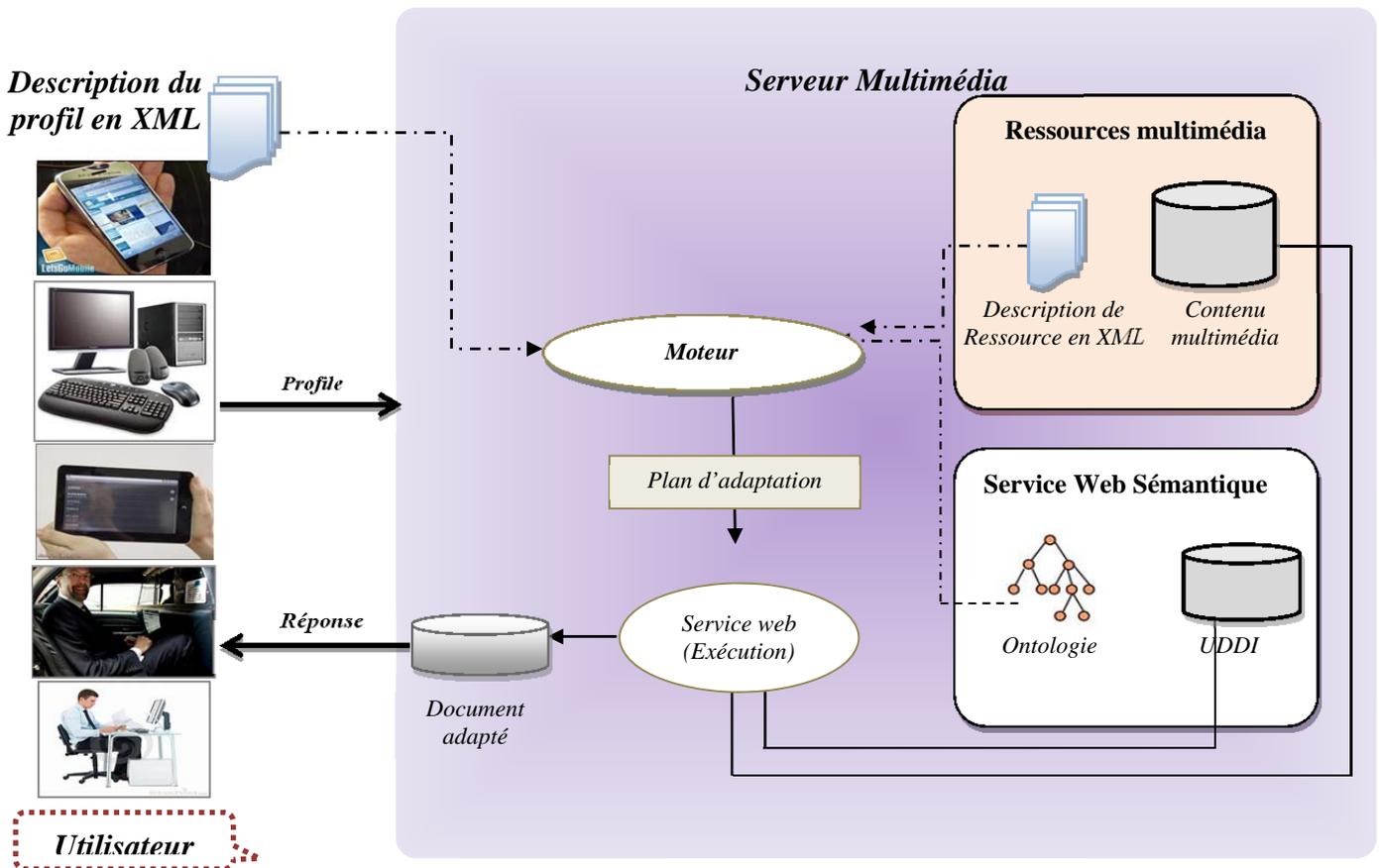


Figure 5. Architecture général de la prise automatique de décision d'adaptation.

### B. Algorithme de construction des services d'adaptation

**Entrée :** Description du profil, Description de ressource

**Sortie :** Ressource adapté

#### Les étapes d'adaptation

**Etape 1 - Initialisation :** Le moteur d'inférence initie l'adaptation automatique des documents multimédia. Il se base sur la description des caractéristiques du profil utilisateur et de la ressource, s'il n'y a pas de conflit *Fin* sinon aller à l'étape 2.

**Etape 2 - Recherche des chaînes adaptation :** Le moteur d'inférence utilise des règles d'adaptation sémantique, il utilise l'ontologie pour la description sémantique des services d'adaptation afin de trouver des correspondances avec les descriptions précédentes « description de profil et de la ressource ». Le moteur d'inférence fait sortir des chemins d'adaptation.

**Etape 3 – Sélection de meilleur chemin :** Parmi les chemins trouvés, il faut sélectionner la chaîne optimale de services offrant la meilleure qualité. Pour trouver les chaînes de services d'adaptation, la sortie d'un service correspond à l'entrée du service suivant sans oublier de calculer le coût et le bénéfice de chaque chaîne. Une fois toutes les possibilités d'assemblage calculées, on sélectionne le chemin optimum en terme de coût avec la meilleure qualité de service. Le sélectionneur de services assure la sélection du meilleur chemin d'adaptation à partir du graphe dans le cas où on trouve

plusieurs services qui font la même chose ou plusieurs chemins fournissant l'adaptation demandée.

**Etape 5 – Extraire des services d'adaptation :** La chaîne est envoyée à l'*Exécutif* dont il utilise le service UDDI pour trouver les services disponibles.

**Etape 6 – Exécution de l'adaptation:** Le document adapté est envoyé au client.

### C. L'ontologie en OWL/XML

L'implémentation de notre ontologie s'est effectuée à travers l'éditeur d'ontologies Protégé 3.4 [13]. Le procédé que nous avons choisi pour définir la hiérarchie des classes est le procédé de haut en bas, on a commencé par définir les concepts les plus généraux, et par la suite spécialisés. Nous avons introduit les propriétés pour chaque classe de l'ontologie, ainsi que les relations entre ses classes. Les figures 6 présentent notre ontologie.

SWRL [13] est un langage de requêtes sur des descriptions RDF. Ce langage est souvent couplé à un moteur de requête qui gère les mécanismes d'inférence. Nous avons également utilisé SWRL pour la description des règles d'inférences et pour manipuler des concepts sémantiques définis dans des ontologies. Le tableau 1 présente quelques règles implémentées.

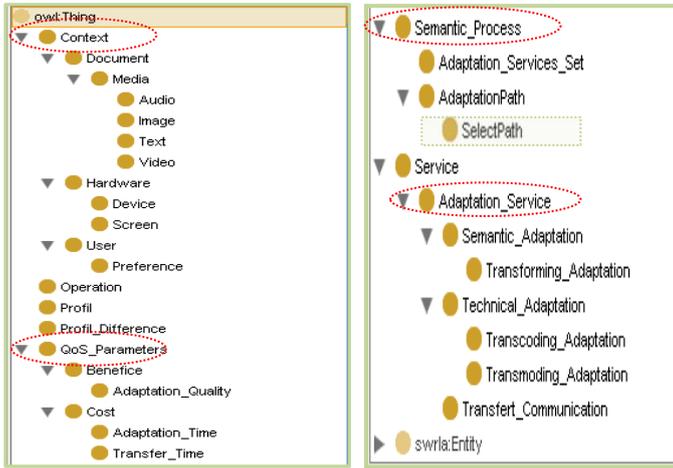


Figure 6. Les classes et les sous classes de l'ontologie.

## V. SCÉNARIOS POSSIBLES ET VALIDATION

### A. Scénarios possibles

À travers les interfaces présentées ci-dessous, nous visons à donner une vue générale de notre prototype conçu dans le but de sélectionner le meilleur chemin d'adaptation à partir des paramètres de QoS préférés par l'utilisateur, le contexte dans lequel l'utilisateur se trouve (en voiture, en parking, en dehors), d'une liste des services d'adaptation disponibles, les exigences du service pour l'exécution, les capacités du dispositif mobile pour déployer et exécuter le service. Notre prototype permet de préciser le type du document, son format, le type du contexte et ses propriétés (exemple : l'âge et sa localisation). Pour mieux comprendre notre approche sémantique prenons l'exemple suivant :

**Scénario 1 :** On suppose qu'un auteur A réalise sa présentation orale dans une conférence (Figure 7). Il met sa présentation dans quelques PCs. Un participant B est intéressé par des services de qualité qui permettent de recevoir la

présentation multimédia sous forme d'image au sein d'un document de type PDF. L'adaptation d'une présentation orale dans une conférence en un fichier PDF peut être réalisée en utilisant les règles d'inférences sémantiques qui génèrent automatiquement les chaînes des services d'adaptation. Ainsi, nous pouvons trouver les chemins suivants :

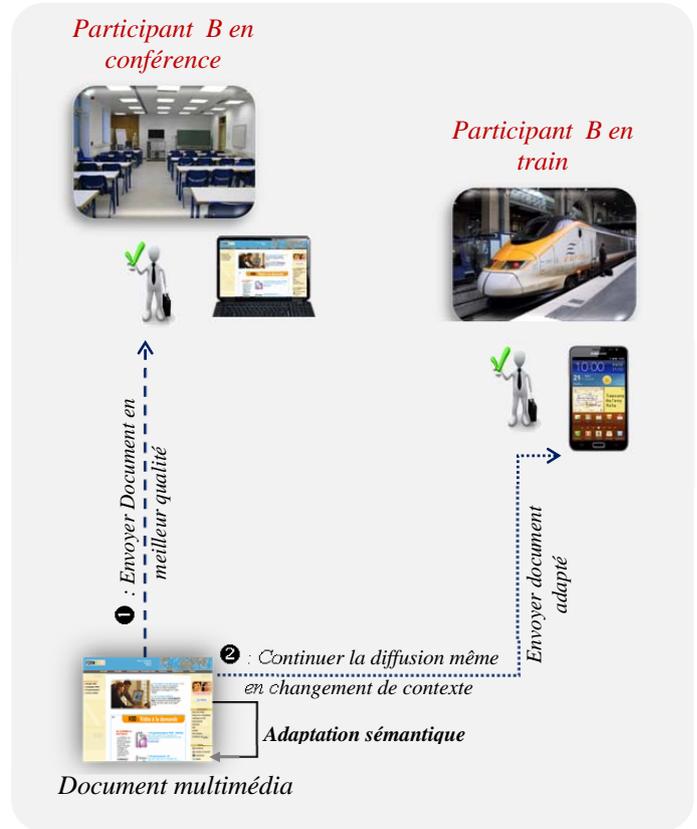


Figure 7. Scénarios possibles et validation.

TABLE I. QUELQUES REGLES D'INFERENCE EN SWRL.

Description de la règle	Langage SWRL
<b>Transcoding_Transcoding_Rule</b> : permet de trouver le format de sortie d'un service d'adaptation et le format d'entrée du prochain service d'adaptation	$Profil\_difference(?dp) \wedge Service\_role(?dp, ?act) \wedge swrlb:equal(?act, "Transcoding") \wedge Adaptation\_Services\_Set(?c) \wedge Compose\_of\_service(?c, ?s) \wedge Has\_set\_of\_operations(?s, ?op5) \wedge Has\_parametre\_out(?op5, ?m1) \wedge Has\_type(?m1, ?type1) \wedge Has\_format(?m1, ?format) \wedge Transcoding\_Adaptation(?m1) \wedge Has\_set\_of\_operations(?m1, ?op3) \wedge Has\_parametre\_out(?op3, ?out2) \wedge Has\_type(?out2, ?type\_out) \wedge Has\_format(?out2, ?forme) \wedge Has\_parametre\_in(?op3, ?in) \wedge Has\_type(?in, ?type\_in) \wedge Has\_format(?in, ?formatin) \wedge swrlb:equal(?type1, ?type\_in) \wedge swrlb:equal(?format, ?formatin) \wedge Transcoding\_Adaptation(?m2) \wedge Action\_type(?m2, ?act1) \wedge swrlb:equal(?act, ?act1) \wedge Has\_set\_of\_operations(?m2, ?op4) \wedge Has\_parametre\_out(?op4, ?out3) \wedge Has\_type(?out3, ?type\_out1) \wedge Has\_format(?out3, ?forme1) \wedge Has\_parametre\_in(?op4, ?in1) \wedge Has\_type(?in1, ?type\_in1) \wedge Has\_format(?in1, ?formatin1) \wedge swrlb:equal(?type\_out, ?type\_in1) \wedge swrlb:equal(?forme, ?formatin1) \rightarrow Compose\_of\_service(?c, ?m1) \wedge Compose\_of\_service(?c, ?m2) \wedge Has\_step(?m1, "Passe") \wedge Has\_step(?m2, "Current") \wedge Has\_successor(?m1, ?m2) \wedge Has\_predecessor(?m2, ?m1)$
<b>Max_Benefice_Rule</b> : permet de calculer et de sélectionner un chemin d'adptation avec maximum de bénéfice.	$SelectPath(?p2) \wedge Adaptation\_Path(?p) \wedge Path\_Benefice(?p, ?b1) \wedge Path\_Cost(?p, ?c1) \wedge sqwrl:makeBag(?b, ?b1) \wedge sqwrl:max(?max, ?b) \rightarrow Path\_Benefice(?p2, ?max)$
<b>Min_Benefice_Rule</b> : permet de calculer et de sélectionner un chemin d'adptation avec minimum de coût.	$SelectPath(?p1) \wedge Adaptation\_Path(?p) \wedge Path\_Cost(?p, ?c) \wedge Path\_Benefice(?p, ?b) \wedge sqwrl:makeBag(?mg, ?c) \wedge sqwrl:min(?min, ?mg) \rightarrow Path\_Cost(?p1, ?min)$

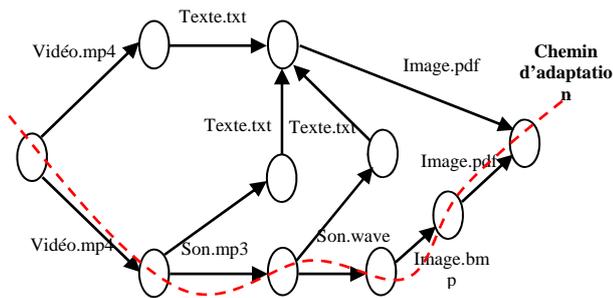


Figure 8. Chemins d'adaptations possibles.

On sélectionne le meilleur chemin. Pour ce faire, on utilise une règle qui permet de calculer le coût minimum et le bénéfice maximum pour sélectionner ce meilleur chemin d'adaptation. À la fin de ce scénario, on utilise un service web pour exécuter la chaîne d'adaptation. Figure 9 illustre le résultat de la recherche du meilleur chemin d'adaptation.

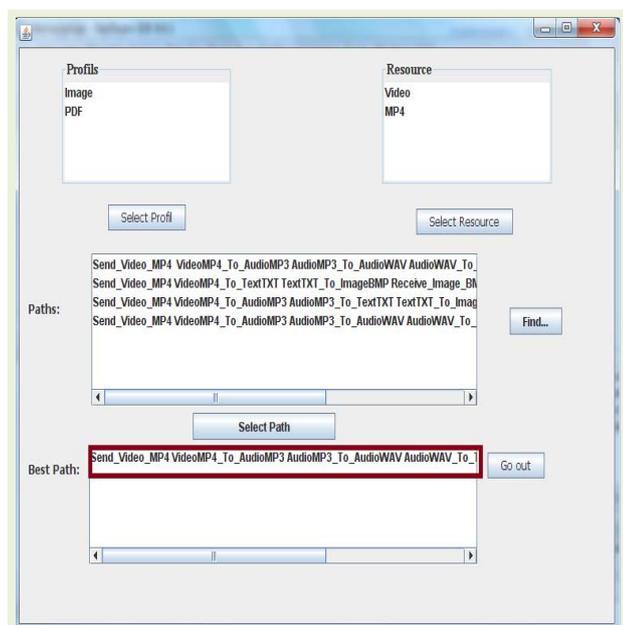


Figure 9. Résultat de recherche pour le premier scénario.

**Scénario 2 :** Alors que l'adaptation d'un document est en cours, le participant B a reçu un appel téléphonique afin de rejoindre un ami à l'aéroport. Il est obligé de quitter la conférence, mais veut quand même suivre la présentation. Pour cela, il dispose d'un Smartphone avec une connexion sans fil 3G. En restituant la connexion, l'application de diffusion détecte ses nouveaux paramètres, elle trouve que le Smartphone n'a pas de lecteur « real media » et ne peut pas recevoir une diffusion dans le même format que sur le PC à cause des caractéristiques matérielles (*taille d'écran, contrainte d'énergie*) et de la qualité de la connexion (*bande passante, type de connexion*).

Pour maintenir la connexion, l'application a besoin d'adapter la vidéo de la présentation afin de répondre au nouveau besoin. On doit donc chercher des services

d'adaptation qui transcode le format de la vidéo de *.MP4* en *.MOV* qui lui est supporté par le Smartphone, et un autre service qui diminue sa résolution afin de réduire encore plus la bande passante pour être compatible avec la 3G. Après la recherche, la sélection et l'intégration des services d'adaptation l'utilisateur peut regarder la présentation pendant son voyage à l'aéroport.

### B. Evaluation des performances

Nous avons réalisé quelques tests de performances pour valider notre approche d'adaptation dynamique et sémantique des documents multimédia. Nous avons effectué ces expériences sur deux configurations hétérogènes mobiles. La configuration 1 est un ordinateur portable fonctionnant sous Windows 7 (x64) avec 5 Go de RAM et un processeur i5 (2.30 GHz). La configuration 2 est une tablette Samsung Galaxy Tab sous Android 3.2 avec 1 Go de RAM et un double coeur Tegra 2 (1 GHz). Dans la figure 10, on compare le temps de recherche des chaînes de service sur des profils composés par des problèmes de changement de format tout d'abord. Ensuite on ajoute d'autres services qui résolvent d'autres problèmes (changement de résolution, localisation de l'utilisateur) aux chaînes trouvées.

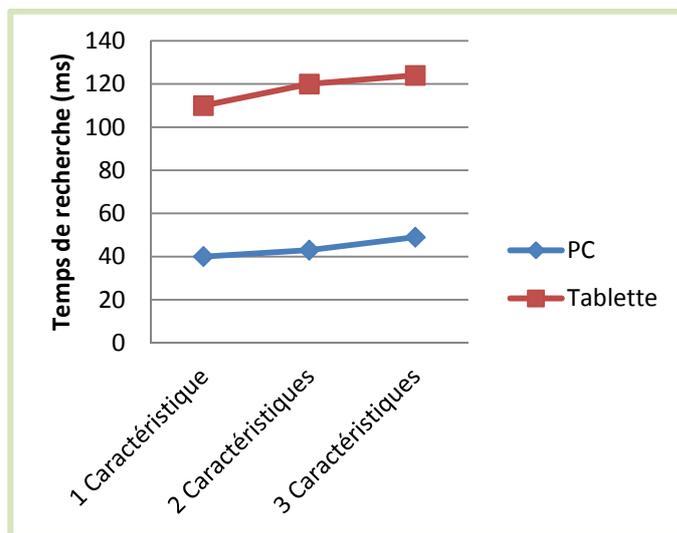


Figure 10. Le temps de recherche des chaînes qui résolvent tous les problèmes

Notre objectif est d'identifier l'impact de la taille du profil sur le temps d'exécution des requêtes. Ainsi, nous notons que l'exécution de la requête est cinq fois plus rapide sous Windows que sous Android. Ces expériences nous permettent de valider la possibilité d'exécuter des requêtes sémantique sur des configurations mobiles avec des performances suffisantes surtout avec des situations de mobilité de l'utilisateur. Notre stratégie d'adaptation fournit une chaîne d'adaptation rapide par l'établissement des relations sémantiques de dépendance, d'équivalence et de substitution et l'élimination des services qui ne change pas le format de document multimédia.

## VI. CONCLUSION ET PERSPECTIVES

S'appuyant sur les progrès récents dans le monde des technologies mobiles, les environnements pervasifs suscitent un intérêt croissant. Cependant, les limitations liées aux ressources des terminaux mobiles, l'hétérogénéité des dispositifs et des données, la multiplicité des demandes et des préférences utilisateurs génèrent des problèmes indésirables. L'adaptation de documents multimédia transmise aux utilisateurs offre un cadre de solution à ces problématiques.

Dans cet article, nous avons proposé une approche d'adaptation sémantique automatique des documents multimédia en se basant sur une ontologie décrivant les caractéristiques (sémantique, coût, qualité) des services d'adaptation multimédia. Cette ontologie est utilisée pour sélectionner et composer les services d'adaptation les plus appropriés au contexte d'utilisation et sur les services web qui exécutent physiquement l'adaptation. Notre travail reste bien sûr ouvert, notamment sur l'amélioration de l'ontologie proposée pour la représentation de la qualité des services sensibles au contexte mais aussi pour compléter l'approche proposée en se basant sur des services (SOA) adaptée à une exécution de type Cloud Computing.

### REFERENCES

- [1] Makhoulf Derdour, Nacira Ghoulmi Zine, Philippe Roose, Marc Dalmau, Adel Altı - *UML-Profile for Multimedia Software Architectures* - International Journal of Multimedia Intelligence and Security (IJMIS) Inderscience Publishers, Vol. 1, No. 3, pp. 209-231, 2010.
- [2] Keskes, N., Lehireche, A., and Rahmoun, A. *Web Services Selection Based on Context Ontology and Quality of Services*, International Arab Journal of e-Technology, 1 (3): 98-105, 2010.
- [3] Romain Rouvoy, Paolo Barone, Yun Ding, Frank Eliassen, Svein Hallsteinsen, Jorge Lorenzo, Alessandro Mamelli, and Ulrich Scholz. MUSIC: Middleware Support for Self-Adaptation in Ubiquitous and Service-Oriented Environments. Springer-LNCS 5525, pp. 164–182, 2009.
- [4] G. Grondin, N. Bouraqadi, and L. Vercoüter. MaDcAr: an Abstract Model for Dynamic and Automatic (Re-) Assembling of Component-Based Applications. In Proceedings of the 9th International SIGSOFT Symposium on Component-Based Software Engineering (CBSE 2006), LNCS 4063, pages 360-367, June 2006, Västerås, Sweden. Springer.
- [5] Luc Fabresse, Christophe Dony, and Marianne Huchard. Foundations of a Simple and Unified Component-Oriented Language. Journal of Computer Languages, Systems & Structures, editor Elsevier, Volume 34/2-3 (July-October 2008), p. 130-149.
- [6] Boukadi, K., C. Ghedira, S. Chaari, L. Vincent, E. Bataineh, CWSC4EC: How to Employ Context, Web Service, and Community in Enterprise Collaboration, *Proceedings of 8<sup>th</sup> International Conference on New Technologies in Distributed Systems (NOTERE'08)*, Lyon, France, 1: 22 – 33, 2008.
- [7] Lamolle, M., Gomez, J., Exposito, E., MODA : une architecture multimédia dirigée par les ontologies pour des systèmes multimédia en réseau, *Proceedings of 4<sup>ème</sup> Conférence francophone sur les Architectures Logicielles (CAL'2010)*, Pau, France, pp. 137–151, 2010.
- [8] WSMO. Web Service Modeling Ontology, 2005, [http://www.wsmo.org/TR/d2/v1.2/20050413/D2v1-2\\_20050414.pdf](http://www.wsmo.org/TR/d2/v1.2/20050413/D2v1-2_20050414.pdf).
- [9] Rayn, N. ConteXtML: Exchanging Contextual Information between a Mobile Client and the FieldNote Server, 1999, <http://www.cs.kent.ac.uk/projects/mobicomp/fnc/ConteXtML.html>.
- [10] Brezillon, P. Context-based modelling of operators Practices bu Contextual Graphs, *Proceedings of 14th Mini Euro Conference in Humman Contered Processes*, 2003.
- [11] Horrocks, I. CDAML+OIL: a Description Logic for the Semantic Web, *IEEE Data Engineering Bulletin*, 25: 4–9, 2002.
- [12] Dean. M., OWL-S: Semantic Markup for Web Services, Version 1.0, 2004.
- [13] The *Protégé* Ontology Editor and Knowledge Acquisition System, <http://protege.stanford.edu/>
- [14] Laborie, S., Euzenat, J., Layaïda, N.,: Semantic adaptation of multimedia documents. *Multimedia Tools Appl.* 55(3): 379-398 -2011.