Table des matières

TABLE DES MATIERES 9
TABLE DES FIGURES 12
LISTE DES TABLEAUX 14
CHAPITRE 1 INTRODUCTION GENERALE 15
1.1 Contexte et problématique 16
1.2 Objectifs et challenges 18
1.3 Contributions 19
1.4 Organisation du mémoire 21

PREMIERE PARTIE CONCEPTS FONDAMENTAUX ET ETAT DE L'ART 23

CHAPITRE 2 ARCHITECTURE ORIENTEE SERVICE 24
2.1 Introduction 25
2.2 Définitions et principes de la SOA 26
2.3 Service web 28
2.4 Service web sémantique 32
2.5 Services web d'accès aux données 33
2.6 Conclusion 36
CHAPITRE 3 DESCRIPTION DE SERVICES WEB 37
3.1 Introduction 38
3.2 Description syntaxique 38
3.3 Description sémantique 42
A. Approches basées sur des langages sémantiques 42
OWL-S 43
WSMO 46
B. Approches de description à base d'annotations 48
WSDL-S 49
USDL 51
SAWSDL 52
3.4 Travaux de description des services web 55
DIANE Service Description, 2005 55
Travail proposé par (Martin, et al., 2007) 56
WSMO-Lite, 2007 56
Le langage YASA4WSDL, 2008 57
La plateforme PASiS, 2011 57
Travail proposé par (Brut, et al., 2011) 58
Travail proposé par (Christensen, et al., 2015) 58
3.5 Etude comparative des approches de description 58

Adaptabilité / Réutilisabilité 59
Extensibilité / Flexibilité 60
Explicité / Expressivité 60
3.6 Conclusion 61
CHAPITRE 4 DECOUVERTE DE SERVICES WEB 62
4.1 Introduction 63
4.2 Les approches non logiques 64
AASDU 64
DSD-matchmaker 67
iMatcher1 67
4.3 Les approches logiques 68
WSC 68
ALS 70
GR 71
PCEM 72
4.4 Les approche hybrides 73
OWLS-MX 73
FC-Match 74
WSMO-MX 75
SAWSDL-MX 77
4.5 Etude comparative des approches de découverte 81
Langage de description 83
Type d'appariement 83
Elément d'appariement 83
Degré d'appariement 84
Niveau d'automatisation 84
4.6 Conclusion 86
DEUXIEME PARTIE CONTRIBUTIONS 87
CHAPITRE 5 L'APPROCHE SESAM 88
5.1 Introduction 89
5.2 Rappel de la problématique 89
5.3 L'approche SeSaM 94
5.4 Présentation des principales notions de l'approche SeSaM 96
5.4.1 Les ontologies 96
L'ontologie métier 96
L'ontologie multimédia 97
5.4.2 Les services web multimédias 99
Motivations 99

Les services MaaS 101
5.4.3 La requête utilisateur 102
5.5 Conclusion 104
CHAPITRE 6 LE LANGAGE SA4MAAS (DESCRIPTION SEMANTIQUE DES SERVICES MAAS) 105
6.1 Introduction 106
6.2 Motivations de la contribution 107
6.3 Le langage de description sémantique SA4MaaS 108
6.4 Exemple illustratif de description sémantique SA4MaaS 112
6.5 Conclusion 119
CHAPITRE 7 LE MATCHMAKER MAAS-MX (APPARIEMENT SEMANTIQUE DES SERVICES MAAS) 120
7.1 Introduction 121
7.2 Motivations de la contribution 121
7.3 Le matchmaker MaaS-MX 124
7.3.1 Appariement métier 125
7.3.2 Appariement multimédia 128
7.4 Exemple illustratif d'appariement sémantique MaaS-MX 132
7.5 Conclusion 135
CHAPITRE 8 PROTOTYPE ET EVALUATION 136
8.1 Introduction 137
8.2 Prototype 137
8.2.1 Module de description 137
8.2.2 Module d'appariement 139
8.3 Evaluation 140
8.3.1 Les mesures de performance 141
Rappel 141
Précision 141
F-Mesure 141
8.3.2 Types d'évaluation 142
8.3.2 Scénarios d'évaluation 143
Scénario Médical 143
Scénario Tourisme 145
8.4 Conclusion 152
CHAPITRE 9 CONCLUSION ET PERSPECTIVES 154
9.1 Résumé des contributions 155
9.2 Perspectives 157

LISTE DES PUBLICATIONS **159**

BIBLIOGRAPHIE **160**

Table des figures

Figure 1. *Interactions au sein de l'architecture SOA (Mrissa, 2007).* *27*
Figure 2 . *Couches de fonctionnalités et protocoles/langages correspondants (Mrissa, 2007).* *29*
Figure 3 : *Architecture et standards des services web.* *31*
Figure 4 : *Web services sémantiques.* *32*
Figure 5 : *Architecture des services DaaS (Malki, 2015).* *35*
Figure 6 : *Structures générale d'un document WSDL 2.0.* *40*
Figure 7 : *Description du service web GlobalWeather en WSDL 2.0.* *41*
Figure 8 : *Les éléments de haut niveau de OWL-S.* *43*
Figure 9 : *Description du service web GlobalWeather en OWL-S.* *44*
Figure 10 : *Le profil du service web GlobalWeather.* *45*
Figure 11 : *Le processus du service web GlobalWeather.* *45*
Figure 12 : *Le Service Grounding du service web GlobalWeather.* *46*
Figure 13 : *Les éléments fondamentaux de WSMO.* *47*
Figure 14 : *Extrait du service Web GlobalWeather décrit en WSML dans WSMO.* *48*
Figure 15 : *Exemple WSDL-S.* *50*
Figure 16 : *Principaux éléments de la description USDL.* *51*
Figure 17 : *Approche SAWSDL (Kopecky, 2007).* *53*
Figure 18 : *Description SAWSDL du service GlobalWeather.* *54*
Figure 19 : *Classification des approches de description de services web.* *55*
Figure 20 : *Architecture de l'approche AASDU (Palathingal, et al., 2004).* *66*
Figure 21 : *Règles d'affectation des degrés d'appariement des sorties.* *69*
Figure 22 : *Règles de tri des scores d'appariement.* *69*
Figure 23 : *Un modèle conceptuel pour le processus de découverte dans l'approche ALS (Keller, et al., 2005).* *70*
Figure 24 : *Architecture de l'approche PCEM (Botelho, et al., 2008).* *73*
Figure 25 : *Les degrés d'appariement dans OWLS-MX (Klusch, et al., 2006).* *74*
Figure 26 : *Exemple d'appariement dans SAWSDL-MX. (Klusch, et al., 2008)* *79*
Figure 27 : *L'architecture de SAWSDL-MX. (Klusch, et al., 2008)* *80*
Figure 28 : *Classification des approches de découverte de services web.* *81*
Figure 29 : *Scénario de motivation.* *93*
Figure 30 : *Vue globale de l'approche SeSaM.* *94*
Figure 31 : *La description sémantique dans SA4MaaS.* *110*
Figure 32 : *XSD du langage SA4MaaS.* *111*
Figure 33 : *L'ontologie métier cancerOnto.* *113*
Figure 34 : *Extrait de l'ontologie des ressources multimédias.* *114*

Figure 35 : *Service MaaS exemple.* *116*
Figure 36 : *Exemple d'un service MaaS présenté dans SA4MaaS.* *118*
Figure 37 : *Requête SPARQL pour le service S_2.* *119*
Figure 38 : *Le matchmaker MaaS-MX.* *125*
Figure 39 : *Algorithme d'appariement métier.* *128*
Figure 40 : *Algorithme d'appariement multimédia.* *129*
Figure 41 : *Algorithme de la fonction MDMatching.* *131*
Figure 42 : *Architecture du système DAMaS.* *138*
Figure 43 : *La préparation de l'annotation sémantique dans DAMaS.* *139*
Figure 44 : *L'annotation métier et multimédia dans DAMaS.* *139*
Figure 45 : *L'appariement dans le système DAMaS.* *140*
Figure 46 : *Le graphe de rappel des deux variantes de DAMaS.* *144*
Figure 47 : *Le graphe de précision des deux variantes de DAMaS.* *145*
Figure 48 *: Le graphe de F-Mesure des deux variantes de DAMaS.* *145*
Figure 49 : *L'ontologie métier TourismOnto.* *146*
Figure 50 : *Le graphe de rappel des deux variantes de DAMaS (Tourisme).* *151*
Figure 51 : *Le graphe de précision des deux variantes de DAMaS (Tourisme).* *151*
Figure 52 : *Le graphe de F-Mesure des deux variantes de DAMaS (Tourisme).* *151*

Liste des tableaux

Tableau 1 : *Exemples de services web. 18*
Tableau 2 : *Symboles d'évaluation. 59*
Tableau 3 : *Récapitulatif de l'étude comparative des approches de description. 61*
Tableau 4 : *Les degrés d'appariement dans WSMO-MX (Kaufer, et al., 2006). 76*
Tableau 5 : *Récapitulatif de l'étude comparative des approches de découverte. 85*
Tableau 6 : *Les services MaaS du scénario exemple. 91*
Tableau 7 : *Les services MaaS annotés du scénario exemple. 115*
Tableau 8 : *La requête annotée. 116*
Tableau 9 : *Les degrés de similarité dans MaaS-MX. 127*
Tableau 10 : *Le calcul des similarités des services MaaS du scénario exemple. 133*
Tableau 11 : *Rappel, Précision et F-Mesure de la variante 1. ..... 144*
Tableau 12 : *Rappel, Précision et F-Mesure de la variante 2. ..... 144*
Tableau 13 : *La requête annotée du scenario de tourisme. 147*
Tableau 14 : *Les services MaaS du scenario de tourisme. 148*
Tableau 15 : *Rappel, Précision et F-Mesure de la variante 1 (Tourisme). 149*
Tableau 16 : *Rappel, Précision et F-Mesure de la variante 2 (Tourisme). 150*
Tableau 17 : *Rappel, Précision et F-Mesure de SAWSDL-M0 (Tourisme). 150*

CHAPITRE 1

INTRODUCTION GÉNÉRALE

1.1 Contexte et problématique

1.2 Objectifs et Challenges

1.3 Contributions

1.4 Organisation du mémoire

Rapport-Gratuit.com

1.1 Contexte et problématique

La diversité des sources de données distribuées et leur hétérogénéité est une des principales difficultés rencontrées par les utilisateurs du Web. Cette hétérogénéité peut provenir du format ou de la structure des sources (sources structurées : bases de données relationnelles, sources semi-structurées, ou non structurées). Le web sémantique permet de répondre à cette problématique en fournissant des mécanismes d'accès, à des sources de données distribuées et hétérogènes, normalisés et intelligibles pour les machines et les humains (Berners-Lee, et al., 2001). Cependant, il doit relever le challenge d'un nouveau passage à l'échelle en permettant un accès efficace, transparent et approprié à des données qui se situent dans un environnement MSSD (Multi-Sources, multi-Sites et multi-Domaines) pour des utilisateurs dont les préoccupations deviennent de plus en plus d'ordre sémantique et non plus simplement syntaxique car l'exigence de qualité de l'information devient une caractéristique récurrente pour les utilisateurs. Les besoins d'accéder de façon uniforme à des sources de données multiples et multimédias sont chaque jour, de plus en plus forts.

Nos travaux de recherche s'intéressent aux systèmes multimédias. Ces systèmes sont devenus incontournables sur le web. On trouve ainsi tout naturel aujourd'hui d'intégrer la vidéo, le son ou la musique dans des applications diverses: éducatif, scientifique, médical, technique, administratif, commercial, culturel et artistique.

Par exemple, dans le domaine médical, à un patient peuvent être associées des images représentant des IRM ou des radiologies stockées sur un site 1, des documents sous forme XML contenant les comptes rendus de ces images stockés sur un site 2, des séquences audio-visuelles retraçant le film d'une opération médicale subie par le patient stockées sur un site 3, mais aussi d'autres documents multimédias traitant des pathologies ou encore d'études épidémiologiques stockés sur un site 4. Le besoin d'accéder à une information sémantiquement unifiée devient tout à fait légitime.

L'accès aux données multimédias dans les systèmes distribués nécessite de gérer la volumétrie, la diversité des interfaces, le format de représentation, la localisation, etc. L'exigence de plus en plus forte des utilisateurs et des applications à vouloir intégrer la sémantique dans la recherche d'information pose de nouvelles questions à résoudre. Actuellement, pour prendre en compte cette nouvelle complexité, on a tendance à s'orienter vers des solutions d'intégration ou d'interopérabilité qui sont souvent appliquées aux données relationnelles mais

peu satisfaisantes ou du moins qui ne répondent que partiellement à ces besoins et aux exigences des données multimédias. Par définition, un système d'intégration de données est un système d'information qui intègre des données de sources différentes et fournit à l'utilisateur une vue uniforme et centralisée des données distribuées.

Une des dernières techniques utilisée dans le cadre de l'intégration de données est l'utilisation des services web, considérés comme l'implantation la plus répandue de l'architecture SOA (Service Oriented Architecture). Par définition, les services Web sont des composants logiciels autonomes et auto-descriptifs et constituent par ce fait un nouveau paradigme pour l'intégration d'applications. La combinaison des technologies des services Web et du Web sémantique a mené au concept des services Web sémantiques. Cette combinaison doit permettre de décrire la sémantique des services web et de leurs fonctionnalités. Cependant, la prise en considération de l'aspect sémantique des services Web constitue par conséquent une proposition d'automatisation des différentes tâches de leur cycle de vie (description, découverte, composition, etc.).

La définition et la découverte d'un nouveau type de service adapté à l'accès aux données multimédia dans le contexte de systèmes de données distribués et hétérogènes est le noyau de nos travaux de recherche. L'état de l'art fournit de nombreux outils formels pour gérer les services d'accès aux données (au sens des bases de données relationnelles), on parle de DaaS (Data as a Service) ou service fournisseur de données (Vaculín, et al., 2008) (Carey, et al., 2012), mais on manque manifestement d'outils pour gérer des services web multimédias. C'est à ce verrou que nous consacrons ce travail de thèse qui se situe à la croisée des communautés RI (Recherche d'Information) et Services.

Notre état de l'art a révélé qu'il y a assez peu de travaux qui se basent sur les services web multimédia, pour répondre à des requêtes utilisateurs dans le contexte des données hétérogènes et distribuées, comme nous l'envisageons. Notre contribution consiste à proposer une approche, basée sur les services web multimédias, capable de réaliser des médiations syntaxiques et sémantiques afin de répondre aux besoins de l'utilisateur.

Pour montrer la faisabilité de notre approche, prenons par exemple ce scénario du domaine médical. Supposons que Bob, un étudiant en médecine, voudrait étudier les différentes méthodes de diagnostiques de la maladie de cancer du poumon. Il veut avoir comme résultat de sa recherche toutes les vidéos fournies par le « centre de cancérologie » au format « mpeg ». Supposons qu'il a à sa disposition les services web présentés dans le **Tableau 1**. Le service S_1 retourne

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

des images sur les différents traitements du cancer du poumon. Les deux autres services S_2 et S_3 retournent les différentes méthodes de diagnostic de cancer du poumon, sachant que le scanner thoracique est un type d'examen imagerie qui permet de diagnostiquer un patient. Le service S_1 n'a pas la même sémantique en termes de fonctionnalités des deux autres services S_2 et S_3, et il est fourni par le « Centre d'imagerie médicale ». Les deux autres services S_2 et S_3 ont une sémantique similaire en termes de fonctionnalités mais retournent différents types de fichiers et ils sont fournis par des fournisseurs différents. Bob peut utiliser les services web du **Tableau 1** pour répondre à ses besoins. Dans un premier temps, il compare les fonctionnalités de chaque service avec les fonctionnalités de la requête. Il élimine alors le service S_1 et garde les services S_2 et S_3 parce que leur sémantique correspond à la sémantique de sa requête. Dans un deuxième temps, il conserve uniquement le service S_2 car ses propriétés multimédias correspondent aux propriétés multimédias de sa requête.

Services	Fonctionnalités	Fournisseur	Format
S_1	Retourner des images visualisant les différents traitements possibles du cancer de poumon	Centre d'imagerie médicale	jpeg
S_2	Retourner des vidéos montrant les différentes méthodes de diagnostique du cancer de poumon	Centre de cancérologie	mpeg
S_3	Retourner des fichiers audio discutant les résultats d'un scanner thoracique d'un patient atteint d'un cancer de poumon	Faculté de médecine	wav

Tableau 1 : Exemples de services web.

1.2 Objectifs et challenges

Dans cette section, nous résumons les principaux challenges que nous avons traités dans cette thèse. Ils sont illustrés en relation avec l'exemple précédent.

– Proposer une approche de recherche de contenus multimédia

L'objectif principal étant de proposer une approche, basée sur l'architecture SOA, pour la recherche sémantique des données multimédia. Les principales questions, qui se posent alors, sont : Quelle sont les étapes nécessaires à suivre pour répondre à des requêtes utilisateurs dans le contexte d'un système hétérogène et distribué contenant des données multimédia ? Que sont ces services d'accès aux données multimédia ? En quoi sont-ils différents ?

– Représentation des services multimédia

Afin d'identifier les services pertinents répondant à la requête de Bob, nous avons besoin d'explorer la base des services disponibles et de comprendre la sémantique de chaque service. La question qui se pose à ce niveau est : Comment définir et représenter les services d'accès aux données multimédia avec leurs différentes vues sémantiques possibles ? De nombreux services peuvent avoir la même sémantique en termes de fonctionnalités, mais complètement différents en termes de propriétés multimédias. Par exemple, les services S_2 et S_3 ont la même sémantique de leurs fonctionnalités, ils ont la même entrée (cancer de poumon) et la même sortie (méthode de diagnostique), mais ils sont différents par rapport à leurs propriétés multimédias (type, fournisseur et format). Le challenge ici est de savoir comment représenter et décrire les services multimédia pour renforcer l'expressivité et la précision de l'approche.

– Sélection des services multimédia pertinents

Supposons maintenant que Bob est capable de décrire et comprendre la sémantique des services multimédia disponibles. La prochaine étape serait d'identifier les services pertinents pour répondre à ses besoins, un tel mécanisme est appelé appariement (matching en anglais). Le challenge ici est de montrer comment faire apparier les services MaaS avec la requête de l'utilisateur prenant en compte leurs descriptions enrichies. Bob devrait se rendre compte que seul le service S_2 satisfait ses besoins fonctionnels et multimédia en même temps.

1.3 Contributions

Nos contributions s'organisent tout naturellement autour des challenges de recherche précédemment évoqués et viennent ainsi se cristalliser sur une approche, basée sur l'architecture SOA, permettant la description et la découverte des services multimédia. Nos principales contributions sont résumées comme suit:

1. Proposer une approche de recherche de contenus multimédias

Afin d'atteindre notre premier objectif, nous proposons une approche full service pour la recherche de contenus multimédias, celle-ci s'intitule "**Se**mantic **Se**arch of **M**ultimedia content" (**SeSaM**). Elle repose sur la définition d'un nouveau pattern de services web permettant l'accès à des sources de données multimédias distribuées afin de répondre à des requêtes utilisateurs. Nous appelons ce nouveau type de service « service web multimédia », baptisé **MaaS** (**M**ultimedia **a**s **a S**ervice). Un service de ce type est tout simplement un service web qui accède aux données multimédia, il est capable de retrouver des données quel que soit leur type (image, vidéo, audio). L'approche proposée se base sur un

processus en trois phases : description, filtrage et restitution. Elle fournit la possibilité d'accéder aux données, les agréger et les filtrer conformément aux besoins de l'utilisateur.

2. Description sémantique des services MaaS

La définition et la description de la sémantique des services MaaS est la contribution clé de notre approche. Dans la littérature, de nombreux langages et approches ont été développés dans l'objectif de décrire la sémantique des services web. Nous citons par exemple OWL-S (OWL for Services), WSMO (Web Services Modeling Ontology) et SAWSDL (Semantic Annotations for WSDL and XML Schema). Ces approches utilisent les ontologies pour donner une description sémantique aux services web. Nous nous sommes intéressés dans ce travail de thèse à l'extension de ces approches pour décrire les services multimédia (MaaS). Nous avons défini le langage **SA4MaaS** (Semantic Annotation for MaaS services), une extension du langage SAWSDL, pour la description sémantique des services MaaS. Notre extension utilise deux types d'ontologies : (1) une ontologie de domaine contenant des concepts et des relations qui définissent l'aspect métier des services (e.g : médical, éduction, tourisme, etc.) ; (2) une ontologie multimédia contenant des concepts et des relations qui définissent l'ensemble des propriétés et annotations multimédia des services (e.g : format, location, information de création, etc.). À l'aide de cette nouvelle manière de décrire un service MaaS par le langage SA4MaaS, nous avons amélioré la précision de l'approche proposée. Les services S_2 et S_3 seront différents, ils ont la même description métier mais différent dans leur description multimédia.

3. Découverte et appariement sémantique des services MaaS

Afin de résoudre le problème d'appariement entre la requête de l'utilisateur et les services MaaS, nous proposons un nouveau mécanisme d'appariement en s'appuyant sur la description enrichie des MaaS. Différentes approches de découverte et d'appariement des services web sémantiques ont été développés au cours des dernières années. Le but de ces approches est d'identifier un degré de similarité entre une requête et des services web. Nous avons défini une nouvelle approche pour la découverte des services MaaS, le matchmaker **MaaS-MX** (MaaS services Matchmaker). Le mécanisme d'appariement de MaaS-MX est réalisé en deux étapes : (1) un appariement métier et (2) un appariement multimédia. La deuxième étape est effectuée si et seulement si la première a réussi. Pour l'appariement métier, nous proposons un algorithme basé sur le calcul du degré de similarité entre, d'une part, les concepts sémantiques annotant les services MaaS ; et, d'autre part, ceux annotant la requête. Pour l'appariement multimédia, nous proposons un deuxième algorithme permettant de comparer la description

multimédia des services MaaS et celle de la requête de l'utilisateur. La description multimédia des services MaaS est définie comme une requête SPARQL en utilisant des concepts d'une ontologie multimédia.

4. Validation et expérimentation

Nous avons développé un prototype intitulé DAMaS (**D**escription **a**nd **M**atching of M**a**aS services **S**ystem) pour valider nos différentes propositions théoriques. Ce prototype a été testé et évalué dans deux domaines différents : médical et tourisme.

1.4 Organisation du mémoire

Les différents aspects décrits dans cette introduction sont détaillés dans le reste de ce mémoire. Celui-ci est structuré en deux parties principales : la première partie, intitulée « Concepts fondamentaux et état de l'art », dresse un état de l'art des travaux qui s'apparentent à notre problématique. La deuxième partie, intitulée « Contributions », est consacrée à nos contributions dans le cadre de cette thèse.

La première partie **« Concepts fondamentaux et état de l'art »** est composée de trois chapitres. Le contenu de chaque chapitre est résumé dans ce qui suit.

- Le chapitre 2 « Architecture orientée service » présente les notions de base des concepts que nous jugeons nécessaires à la compréhension du contenu de cette thèse. Nous présentons d'abord les concepts de base de l'architecture orientée services. Ensuite, nous présentons les notions fondamentales des services web et des services web sémantiques. Enfin, nous terminons ce chapitre par la définition des services web d'accès aux données.
- Le chapitre 3 « Description de services web » dresse un état de l'art sur les langages et les approches de description de services web. Dans ce chapitre, nous commençons par présenter le standard de services web (WSDL), une description considérée comme purement syntaxique. Ensuite, nous présentons les approches sémantiques OWL-S, WSMO, WSDL-S, USDL et SAWSDL. Une étude comparative des différentes approches de description de services web est présentée à la fin de ce chapitre.
- Le chapitre 4 « Découverte de services web » constitue un état de l'art de la découverte de services. Nous présentons dans ce chapitre le principe et les travaux les plus connus sur la découverte de services. Nous commençons par exposer les approches de découverte non logiques, puis les approches de découverte logiques et finalement les approches hybrides. Une étude comparative des différentes approches de découverte de services web est présentée à la fin de ce chapitre.

La deuxième partie « **Contributions** » est composée de quatre chapitres. Le contenu de chaque chapitre est résumé dans ce qui suit.

- Le chapitre 5 décrit d'une manière globale l'approche SeSaM et ses caractéristiques. Il présente également la définition formelle des ontologies, des services MaaS et la requête de l'utilisateur.
- Le chapitre 6 se focalise sur la phase de description des services MaaS. Ce chapitre décrit le langage SA4MaaS, notre extension SAWSDL pour la description sémantique des services MaaS.
- Le chapitre 7 est consacré à la phase de découverte et d'appariement des services MaaS. Nous présentons dans ce chapitre le matchmaker MaaS-MX, notre approche de découverte des services MaaS.
- Le chapitre 8 présente le prototype développé DAMaS et les expérimentations réalisées.

Le chapitre 9 « Conclusions et perspectives » termine ce mémoire par une synthèse des principales contributions réalisées dans le cadre de nos travaux de thèse, ainsi que les perspectives de recherche envisagées.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

PREMIERE PARTIE

CONCEPTS FONDAMENTAUX ET ÉTAT DE L'ART

Chapitre 2 *Architecture orientée services*

Chapitre 3 *Description de services web*

Chapitre 4 *Découverte de services web*

CHAPITRE 2

ARCHITECTURE ORIENTÉE SERVICE

2.1 Introduction

2.2 Définitions et principes de la SOA

2.3 Service web

2.4 Service web sémantique

2.5 Services web d'accès aux données

2.6 Conclusion

2.1 Introduction

Les dernières années ont été marquées par l'évolution rapide des systèmes d'information qui sont devenus de plus en plus distribués, complexes et coûteux en termes de gestion. Cette évolution du monde informatique a entraîné le développement de nouvelles architectures et technologies facilitant le déploiement des applications et leurs interactions. Parmi ces architectures, nous pouvons citer l'architecture orientée objet, l'architecture orientée composant et l'architecture orientée services.

L'architecture orientée objet (Taylor, 1997) trouve son origine dans les années 1960 en particulier dans le domaine de la simulation avec la diffusion du langage Simula qui introduit la notion de classes et donc d'objets. Dans le cadre de cette architecture, des nouvelles technologies sont développées telles que DCOM, RMI et CORBA. L'architecture orientée composant étend et améliore les principes de l'approche orientée objet, elle consiste à déterminer les éléments architecturaux d'un système, c'est à dire les composants qui définissent les traitements métiers, les connecteurs qui décrivent les communications entre ces composants et la configuration qui représente la topologie des liens entre les composants. Cependant, cette architecture impose l'hétérogénéité technique des composants (i.e., hétérogénéité des langages de programmation : JEE, .NET ; hétérogénéité des protocoles de communication : IIOP, RMI; etc.). Ce type d'hétérogénéité empêche les différents types d'applications telles que les applications web (e.g: PHP, ASP, JSP, etc.), et les applications mobiles (e.g: Android, IOS, etc.) d'interagir avec les composants logiciels. Ainsi, la forte dépendance qui existe entre les composants au-sein d'une application empêche leur évolution et leur maintenabilité (Malki, 2015). Ces lacunes ont donné naissance à une nouvelle architecture, l'architecture orientée service (SOA), plus agile et plus flexible permettant aux applications d'être indépendantes, autonomes et facilement intégrables. L'architecture SOA est une extension de l'architecture objet et de l'architecture composant, elle vise à apporter une grande flexibilité dans la maintenance et l'évolution des systèmes, et garantir un haut degré d'interopérabilité entre systèmes hétérogènes (Papazoglou, et al., 2008).

Dans ce chapitre, nous décrivons dans un premier temps les concepts de base de l'architecture orientée service. Ensuite, nous présentons les services web qui représentent une des implantations les plus répandues de l'architecture orientée service. Nous terminons ce chapitre par la définition des services web

Rapport-Gratuit.com

sémantique ainsi des services web d'accès aux données, un type particulier des services web.

2.2 Définitions et principes de la SOA

L'Architecture Orientée-Service (en anglais, Service-Oriented Architecture - SOA) (Stal, 2002) est un paradigme architectural présentant un intérêt particulier pour les technologies de l'information et le domaine de l'entreprise. Cette approche a été développée par diverses entreprises de recherche en informatique telles qu'IBM, BEA, Oracle ou Microsoft qui ont chacune leur propre concept et leur propre définition de cette architecture. Bien que toutes les définitions de la littérature semblent similaires, elles comportent des divergences. Voici quelques définitions de l'architecture SOA:

> *« Service-oriented Architecture (SOA) is an architectural style that supports service-orientation. » (The Open Group, 2007).*

> *« A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed. Those connections are Web Services » (Barry, 2003).*

D'après les définitions précédentes, l'architecture orientée service est construite autour de la notion de service, qui est matérialisé par un composant logiciel assurant une ou plusieurs fonctionnalités et accessible via son interface. Un service est toujours accompagné d'une description fournissant aux applications clientes les informations nécessaires à son utilisation.

Les composants de base de l'architecture SOA sont : le ***fournisseur de service***, le ***registre de service*** et le ***client de service***. Les interactions entre ces trois composants se résument en trois actions principales : la ***publication***, la ***découverte*** et l'***invocation***. Le fournisseur de service a pour fonction de déployer un service sur un serveur et de générer une description de ce service. Cette dernière précise à la fois les opérations disponibles et leur mode d'invocation. Cette description est publiée dans le registre de service, aussi appelé annuaire (étape de publication). Les consommateurs de services peuvent découvrir les services disponibles répondant à leurs besoins en lançant une recherche sur l'annuaire (étape de découverte). Les consommateurs peuvent ensuite utiliser la description du service afin d'établir une connexion avec le fournisseur et invoquer

les opérations du service souhaité (étape de découverte). Les composants de base de SOA et leurs interactions sont illustrées par la **Figure 1**.

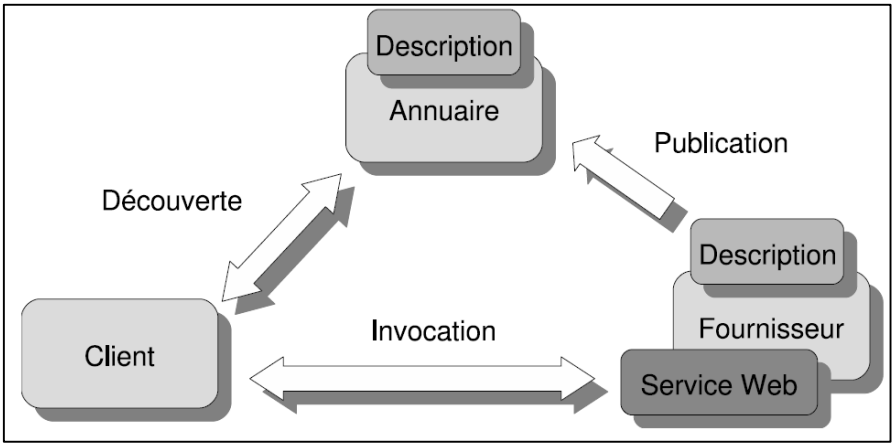

Figure 1. Interactions au sein de l'architecture SOA (Mrissa, 2007).

L'architecture orientée service se base sur les principes suivants (Erl, 2008) :

- *Contrat de service standardisé* : Le contrat de service définit un accord entre le fournisseur et le consommateur, composé des éléments suivants: i) l'interface du service qui contient les opérations, leurs paramètres d'entrée et de sortie et les contraintes sur les entrées ; ii) une description informelle des opérations sous formes de règles et contraintes d'utilisation du service (e.g., volume des données échangées) ; iii) un niveau de service (QoS et SLA) précisant les engagements du service (e.g., temps de réponse maximum attendu, plages horaires d'accessibilité, le temps de reprise après interruption, les procédures mises en œuvre en cas de panne, les procédures de prise en charge du support).
- *Faible couplage entre les services* : Les services maintiennent une relation minimisant les dépendances. Ils n'ont plus besoin d'un système de middleware réparti commun pour communiquer, mais seulement de protocoles et technologies de communication interopérables sur Internet.
- *Abstraction de service* : En dehors des différentes descriptions dans le contrat de service, les services cachent leur logique au monde extérieur. Le contrat de service ne doit contenir que les informations essentielles à son invocation.
- *Services réutilisables* : La logique est divisée en différents services avec comme objectif de promouvoir la réutilisation. Les services peuvent être ainsi partagés parmi différents domaines.
- *Services autonomes* : Les services contrôlent la logique d'exécution qu'ils encapsulent. Plus ce contrôle est fort, plus l'exécution d'un service est prédictible.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

- *Services dépourvus d'état* : Les services minimisent la consommation de ressources en déléguant la gestion des informations d'état quand cela est nécessaire.
- *Services découvrables* : La description des services est complétée par un ensemble de métadonnées permettant leur découverte et leur interprétation de manière efficace et appropriée.
- *Services composables* : Les services sont conçus de manière à participer à des compositions de services. Les services peuvent être orchestrés pour améliorer l'agilité du processus métier implémenté.

L'architecture SOA repose sur un ensemble de langages standardisés et de protocoles, selon quatre couches de fonctionnalités, à savoir :

- La **couche transport**, qui permet de véhiculer les messages échangés entre services à travers le réseau.
- La **couche publication**, qui permet la centralisation, le stockage et la diffusion des descriptions de services.
- La **couche description**, qui regroupe les détails nécessaires à l'invocation des services dans un document.
- La **couche communication**, qui assure la structuration et l'échange uniformes des messages. Cette couche spécifie les protocoles et le mode d'échange des messages entre le service web et ses clients.

L'architecture SOA possède plusieurs implantations. Dans le cadre de cette thèse, nous nous intéressons aux services Web, qui représentent l'implantation la plus largement répandue dans le monde industriel.

2.3 Service web

Le service web est défini comme étant l'élément principal de l'architecture SOA. Plusieurs définitions ont été proposées dans la littérature. Cependant, la première définition d'IBM et la deuxième définition de W3C constituent des définitions de référence.

> *« Web services are a new breed of web applications. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions that can be anything from simple requests to complicated business processes.» (Wahli, et al., 2002).*

Cette définition affirme que les services web sont des applications modulaires autonomes qui peuvent être publiées et invoquées à travers le web.

Les services web proposent des interfaces bien définies qui décrivent les fonctionnalités fournies.

> *« A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. » (Booth, et al., 2004).*

Cette deuxième définition met en valeur les avantages principaux d'un service web, à savoir le couplage faible et l'interopérabilité qui sont assurés par l'utilisation des normes actuelles du web d'un côté et par l'utilisation de langages et de protocoles indépendants des plateformes d'implantation de l'autre côté. Autour de cette définition, la pile standard de protocoles des services web est représentée par la **Figure 2**.

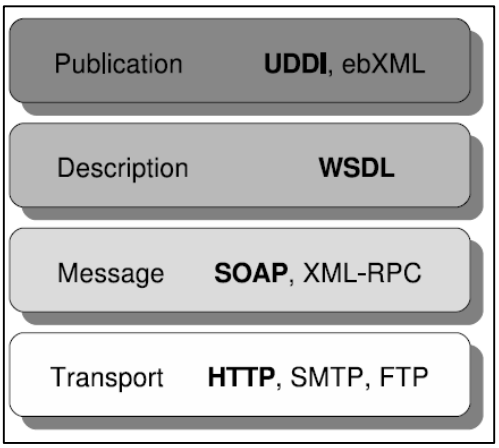

Figure 2 . Couches de fonctionnalités et protocoles/langages correspondants (Mrissa, 2007).

- La **couche transport**, qui s'intéresse aux protocoles de transport de bas niveau permettant de transporter les requêtes et les réponses échangées entre services. Le protocole le plus utilisé est le HTTP (HyperText Transfert Protocol) (Fielding, et al., 1999), mais d'autres implémentations peuvent utiliser le protocole SMTP (Simple Mail Transfer Protocol) ou le protocole FTP (File Transfer Protocol).
- La **couche publication**, qui repose sur le protocole UDDI (Universal Description, Discovery, and Integration) (OASIS UDDI Specification Technical Committee, 2007), qui assure le regroupement, le stockage et la diffusion des descriptions de services Web. Citons aussi ebXML (OASIS and

UNCEFACT, 2007), une autre solution alternative fournissant les mêmes fonctionnalités.

- La **couche description**, est prise en charge par le langage WSDL (Web Service Description Language) (Christensen, et al., 2001), un langage de description à base de XML (Bray, et al., 2008) utilisé pour décrire les fonctionnalités fournies par le service Web, les messages reçus et envoyés pour chaque fonctionnalité, ainsi que le protocole adopté pour la communication. Les types des données contenues dans les messages sont décrits à l'aide du langage XML Schema (W3C XML Core Working Group, 2004).
- La **couche communication**, qui utilise des protocoles reposant sur le langage XML, car sa syntaxe unique résout les problèmes d'interopérabilité entre plates-formes. Actuellement, XML-Remote Procedure Call (XML-RPC) et son dérivé Simple Object Access Protocol (SOAP) (Box, et al., 2007) sont les deux protocoles utilisés pour cette couche, SOAP étant le protocole prédominant.

Nous rappelons dans la suite les trois principaux standards, basés sur le langage XML, sur lesquels reposent les services web :

- **SOAP** (Simple Object Access Protocol) (Box, et al., 2007), est un protocole standard de communication. SOAP est standardisé par le W3C et il définit un protocole léger d'échange de données structurées dans un environnement totalement distribué et hétérogène, tout en résolvant les conflits syntaxiques et techniques. Il permet d'invoquer des services indépendamment de la plate-forme d'exécution. Il a aussi l'intérêt de pouvoir être employé dans tous les types de communication : synchrones ou asynchrones, point à point ou multipoint. Pour la négociation et la transmission des messages, il s'appuie sur d'autres protocoles de la couche de transport, notamment HTTP.
- **WSDL** (Web Service Description Language) (Christensen, et al., 2001), est un langage de description standard qui offre une grammaire décrivant l'interface des services web d'une manière générique. Il indique comment utiliser le service Web et comment interagir avec lui. WSDL permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyée.
- **UDDI** (Universal Description, Discovery, and Integration) (OASIS UDDI Specification Technical Committee, 2007), est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services Web aux clients. Les fournisseurs disposent d'un schéma de description permettant

de publier des données concernant leurs activités, la liste des services qu'ils offrent et les détails techniques sur chaque service. La spécification UDDI offre aussi une API aux applications clientes, pour consulter et extraire des données concernant un service et/ou son fournisseur. Un registre UDDI est décomposé en pages blanches, jaunes et vertes. Les pages blanches contiennent des données sur les fournisseurs des services. Les pages jaunes contiennent des données concernant l'activité ou le service métier des fournisseurs. Finalement, les pages vertes décrivent les données techniques de chaque service publié.

La **Figure 3** montre un scénario classique d'utilisation de ces trois standards (SOAP, WSDL, UDDI) dans le cadre d'une architecture orientée service.

1. Les fournisseurs de services web publient les descriptions de leurs services Web (les fichiers WSDL) dans un annuaire UDDI.
2. Les clients de services web interrogent l'annuaire UDDI afin d'obtenir une liste de services web répondant à leurs exigences.
3. Les clients des services web récupèrent, depuis l'annuaire UDDI, les fichiers WSDL des services web candidats.
4. Les clients des services web envoient des messages SOAP aux fournisseurs conformes aux fichiers WSDL pour invoquer les services web des fournisseurs.
5. Finalement, les fournisseurs de services web répondent par des messages SOAP conformes aux fichiers WSDL.

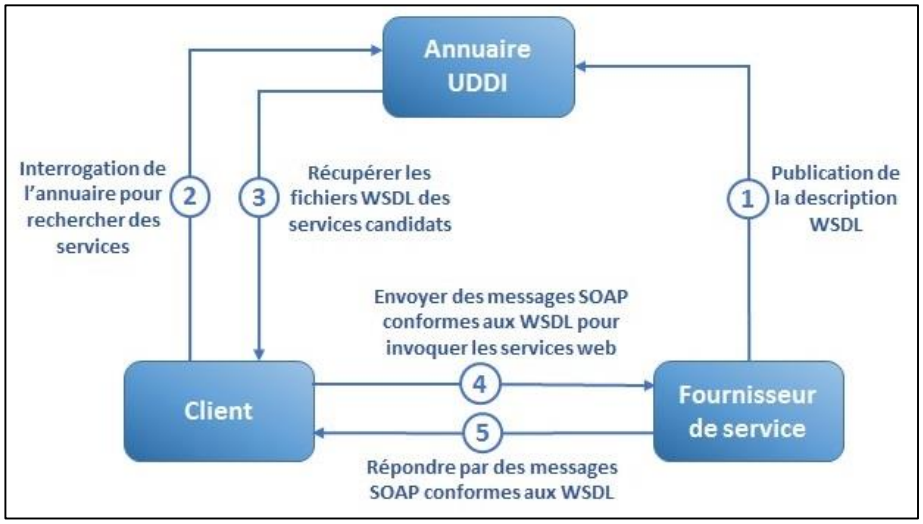

Figure 3 : Architecture et standards des services web.

2.4 Service web sémantique

La dimension sémantique reste encore ignorée par la pile standard de protocoles des services Web. Lors de la réalisation d'une composition de service par exemple, des conflits sémantiques apparaissent quand les services sont décrits avec des vocabulaires différents, ou lorsque les termes utilisés diffèrent d'un service web à un autre. Ces conflits et ces hétérogénéités sémantiques peuvent générer des dysfonctionnements aux moments de la découverte, de la composition et même de l'invocation des services web.

La réponse à de telles exigences d'interopérabilité sémantique a été apportée par les services web sémantiques qui combinent le web sémantique et la technologie des services web (voir la **Figure 4**). Des approches sémantiques, étudiées dans le chapitre 3, ont été alors proposées. Elles ont pour objectif de résoudre ces problèmes de sémantique rencontrés par les services web. Ces approches intègrent la sémantique dans les descriptions des services web, qui deviennent des services web sémantiques. Une description de service web sémantique repose sur un vocabulaire commun décrit dans une ontologie, qui est définie comme « une conceptualisation partagée des connaissances d'un domaine » (Gruber, 1995). Une ontologie décrit le vocabulaire d'un domaine de connaissance, incluant les concepts utilisés dans le domaine et les relations entre ces concepts. Elle représente donc un schéma conceptuel, qui tente de désigner une description rigoureuse et exhaustive d'un domaine.

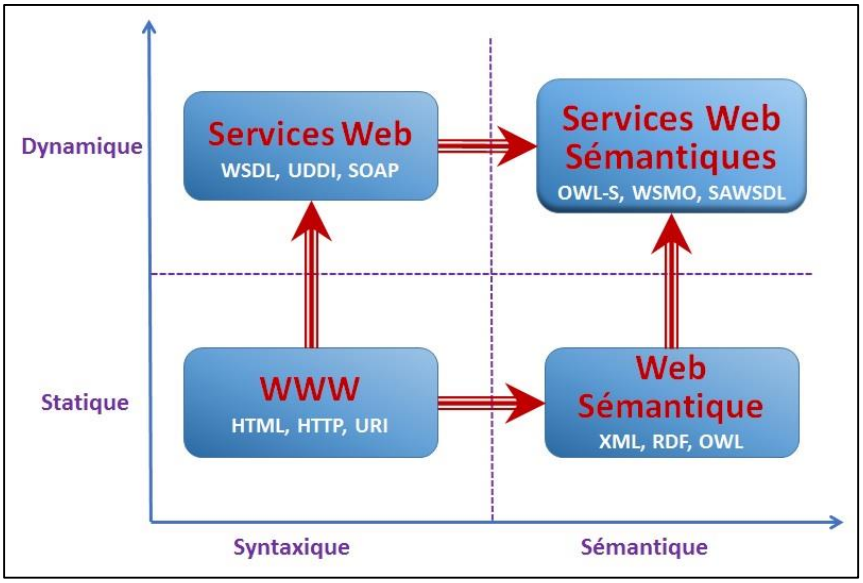

Figure 4 : Web services sémantiques.

De manière générale, l'objectif des services web sémantiques est de créer des services web dont les éléments (propriétés, fonctionnalités, interfaces, etc.) sont décrits de manière non ambiguë et exploitable par des machines. La sémantique exprimée permettra l'automatisation des différentes tâches de cycle de vie de services web (description, publication, découverte, sélection et composition).

2.5 Services web d'accès aux données

Les entreprises d'aujourd'hui se sont orientées vers une architecture orientée services pour la publication et le partage des données sur le web. L'accès à ces sources de données est effectué via des services web, ce qui fournit une méthode uniforme, interopérable et indépendante des plateformes pour l'interaction avec les différentes sources de données hétérogènes. Ce type de services est connu sous le nom de services DaaS (Data as a Service) ou les services web d'accès aux données. Les services DaaS sont définis comme suit:

> *« An information service (i.e., data service) provides a simplified, integrated view of real-time, high-quality information about a specific business entity, such as a customer or product. It can be provided by middleware or packaged as an individual software component. The information that it provides comes from a diverse set of information resources, including operational systems, operational data stores, data warehouses, content repositories, collaboration stores, and even streaming sources in advanced cases » (Mike Gilpin, 2007)*

Un service DaaS est un nouveau type de service web qui se place entre les applications SOA et les sources de données hétérogènes. Un service DaaS, à la différence d'un service web traditionnel connu sous le nom de services SaaS (Software as a Service), accède aux sources de données uniquement en mode lecture (Interrogation) et ne change en aucun cas leurs états. Ces services DaaS sont appelés services sans état (*stateless* services). L'exécution d'un service SaaS produit des effets pouvant changer l'état des sources de données. Par exemple, un service de vente de livres a comme effet de débiter la carte de crédit du client et d'expédier le livre à l'adresse du client. Les services DaaS sont des requêtes d'accès aux sources de données via des services web. Par exemple, un service DaaS dans le domaine de la météorologie peut retourner les conditions climatiques d'une zone bien spécifique. Lorsqu'un tel service est exécuté, il requiert des données en entrée et retourne à l'utilisateur des données en sortie.

L'objectif des services DaaS est de fournir un accès homogène aux sources de données distribuées et hétérogènes. Les caractéristiques principales des services DaaS sont :

- *Hétérogénéité* : L'adoption de l'architecture SOA et les services DaaS décharge les développeurs d'applications SOA des problèmes d'hétérogénéité. Effectivement, dans le monde des services web toutes les sources de données sont décrites à l'aide du standard WSDL et invoquées via des messages SOAP (donc avoir la même interface). En plus, les données sont au format XML et décrites en XML schéma.
- *Réutilisabilité et agilité* : La valeur ajoutée de l'architecture SOA au développement d'application est la réutilisabilité et l'agilité. Les services DaaS peuvent être utilisés et réutilisés dans plusieurs processus métiers. Cela simplifie largement le développement et la maintenance des applications de type SOA, impose une utilisation standard des données et introduit des fonctionnalités maniables pour utiliser les données dans le cas des processus dynamiques et en temps réel.
- *Faible couplage* : Dans un système d'intégration de données basé sur le paradigme SOA, les consommateurs de données (utilisateurs ou programmes) n'ont pas besoin au préalable de prendre connaissance des sources de données. Ils ont seulement besoin d'opérer avec les services DaaS qui fournissent une couche d'abstraction entre les consommateurs et les sources de données. En fait, dans un système d'intégration de données basé sur le paradigme SOA, les utilisateurs des données sont entièrement indépendants des sources de données et de leurs localisations et origines (base de données, fichiers systèmes, application d'une autre entreprise, etc.).

La **Figure 5** présente l'architecture des services DaaS (Carey, et al., 2012). Cette architecture est composée de trois couches : la couche *Données*, la couche *Services* et la couche *Mapping*.

La couche *Données* couvre toutes sources de données pouvant être accessibles via des services DaaS. Ces sources de données sont décrites par différents modèles de représentation (e.g. modèle relationnel, XML, WSDL, etc.) et interrogées par différents langages (e.g. SQL, XPath/XQuery, SOAP, etc.).

La couche *Services* est décrite par un ensemble d'opérations permettant l'utilisation des services DaaS afin d'accéder aux différentes sources de données. Deux types d'opérations sont définies : les opérations de type *fonction* et les opérations de type *requête*. Les opérations de type *fonction* sont similaires aux opérations d'un service web classique, dans lesquelles les utilisateurs sont complètement liés à la signature de l'opération, et à ce qui est retournée par cette

dernière. Par contre, les opérations de type *requête* permettent aux utilisateurs de poser leurs propres requêtes, et d'interroger les sources de données de la façon qu'ils désirent. Les deux types d'opérations (*fonction* et *requête*) sont définis et formulés à travers le modèle de service (i.e., schéma de service) appelé : *modèle externe*. Ce dernier modèle est un modèle commun qui décrit de façon uniforme et globale les sources de données accessibles via des services DaaS. Il peut être comparé au schéma global dans un système d'intégration de données classique (e.g. approche médiatrice, approche fédératrice, etc).

La couche *Mapping* définit les correspondances entre les modèles des données de la couche *Données* et le modèle externe de la couche *Services*. Il existe de types de mappings, les mappings déclaratifs, et les mappings procéduraux. Les mappings déclaratifs sont souvent similaires aux vues d'une base de données relationnelle. Ils sont utilisés pour définir les mappings des sources de données dont le langage d'interrogation est déclaratif (e.g. SQL pour le relationnel, XQuery pour XML, SPARQL pour RDF/RDFs, etc). Les mappings procéduraux sont utilisés pour les données qui sont fournies par des services DaaS ou par des applications spécifiques (e.g. CRM, ERP, etc).

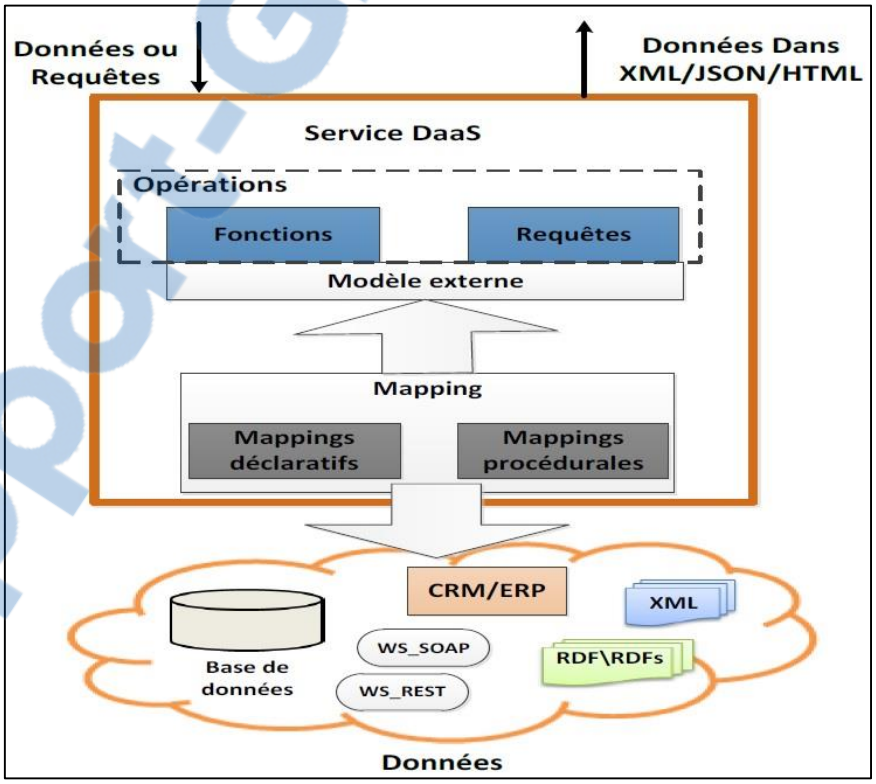

Figure 5 : Architecture des services DaaS (Malki, 2015).

Rapport-Gratuit.com

2.6 Conclusion

Dans ce chapitre, nous avons présenté les principaux concepts autour de l'architecture SOA, ainsi que son implantation la plus répandue qui est les services web. Nous avons également présenté les services web sémantiques et les services web d'accès aux données. Les services web sémantiques ont pour objectif de combler certaines lacunes des services web traditionnels liées à la prise en compte de l'aspect sémantique. Cette sémantique permet l'automatisation des différentes tâches de cycle de vie de services web (description, publication, découverte, sélection et composition).

Dans le chapitre suivant, nous décrivons les approches et les travaux autour de la description des services web. La description est la première phase indispensable dans le cycle de vie d'un système basé sur une architecture SOA

CHAPITRE 3

DESCRIPTION DE SERVICES WEB

3.1 Introduction

3.2 Description syntaxique

3.3 Description sémantique

A. Approches basées sur des langages sémantiques

OWL-S

WSMO

B. Approches de description à base d'annotations

WSDL-S

USDL

SAWSDL

3.4 Etude comparative des approches de description

Adaptabilité / Réutilisabilité

Extensibilité / Flexibilité

Explicité / Expressivité

3.5 Conclusion

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

3.1 Introduction

L'une des caractéristiques principales des architectures SOA est la réutilisabilité. Cette réutilisabilité est conditionnée par le fait que chaque service utilisé dans un système doit être, au préalable, décrit par son fournisseur. La description d'un service web, une étape importante dans son cycle de vie, est nécessaire pour sa publication par son fournisseur et sa sélection ultérieure par des clients.

La description d'un service consiste en la définition d'une interface exposant les opérations accomplies par le service et à lier chaque opération à sa réalisation. De nombreuses approches ont été développées dans l'objectif de décrire les services web. Nous présentons dans ce chapitre un état de l'art sur ces approches de description de services web. Nous les classifions en approches syntaxiques et approches sémantiques. L'étude du standard de description de services web (WSDL), une description considérée comme purement syntaxique, fait l'objet de la deuxième section de ce chapitre, tandis que les approches proposant une description sémantique de services web sont exposés dans la troisième section. Nous citons dans la quatrième section quelques travaux de description des services web qui se basent sur les approches décrites dans la section précédente. Une étude comparative des différentes approches de description de services web est présentée à la fin de ce chapitre.

3.2 Description syntaxique

Le langage WSDL (*Web Service Description Language*) est devenu le standard actuel pour la définition et la description des services. L'importance que nous donnons à l'étude de WSDL s'explique par le fait que WSDL est au cœur de la technologie des services web. Il existe deux versions standardisées, par le W3C, de WSDL : WSDL 1.1 (Christensen, et al., 2001) et WSDL 2.0 (Chinnici, et al., 2007).

Des applications, sur le Web ou dans un réseau, peuvent communiquer grâce aux services. Ces communications ont pu être automatisées grâce au standard WSDL et son approche de description contenant les données nécessaires aux échanges entre les applications. Pour ce faire, WSDL fournit, à la fois, un format XML et un modèle pour décrire un service à travers une interface présentant un ensemble d'opérations et leurs paramètres d'entrée et de sortie respectifs.

Dans WSDL, il existe une séparation entre deux niveaux indépendants : un premier niveau de description abstraite et un deuxième niveau de description concrète. Le *niveau abstrait* regroupe les informations pouvant être réutilisées (non spécifique à un service), tandis que le *niveau concret* est constitué de la description des protocoles d'accès au service Web (information particulière à un service). Le niveau abstrait est utilisé principalement lors du processus de sélection, tandis que le niveau concret est seulement utilisé lors de l'invocation des méthodes du service Web (Lopez-Velasco, 2009).

Le **niveau abstrait** décrit principalement les messages envoyés et reçus par le service, ainsi les données échangées lors de l'invocation du service. Ce niveau contient les éléments d'information suivants :

- **Les types de données :** décrivent les types de données utilisés dans les messages échangés par le service. Ces types de données sont définis par un schéma XML, ils peuvent être simples (tels que les entiers, les chaines de caractères) ou complexes (tels que les tableaux, les enregistrements).
- **Les messages :** cet élément définit les messages échangés par le service. La description d'un message contient le nom de l'élément en paramètre (élément d'entrée ou de sortie selon le type du message) et son type.
- **Les opérations :** une opération est une description abstraite d'une action accomplie par le service. Chaque opération est identifiée par son nom.

Le **niveau concret** décrit la manière dont le client accède à un service, il précise les détails sur le moyen d'échange des messages (comment et vers où les messages sont envoyés sur le réseau). Les informations décrites dans ce niveau sont les suivantes :

- **Le protocole de communication :** définit le protocole de communication à utiliser pour l'appel des méthodes du service. Le protocole de communication peut être différent pour chaque opération du service décrit.
- **Les ports d'accès au service :** l'accès au service est défini par une collection de ports d'accès qui représente la localisation du service (URL). Un même service peut être accessible sur plusieurs ports différents.

Comme nous l'avons déjà mentionné, le langage WSDL a deux versions standardisées : WSDL 1.1 et WSDL 2.0. Nous choisissons d'étudier la deuxième version du standard. Les exemples et les travaux étudiés dans la suite de ce mémoire se basent sur cette deuxième version du standard (WSDL 2.0). La structure générale d'un document WSDL 2.0 est composée d'un élément racine (*description*) et de quatre sous-éléments obligatoires (*types*, *interface*, *binding* et *services*). La **Figure 6** illustre cette structure et montre le lien entre les éléments

XML et les éléments d'information des deux niveaux de description de WSDL (niveau abstrait et niveau concret).

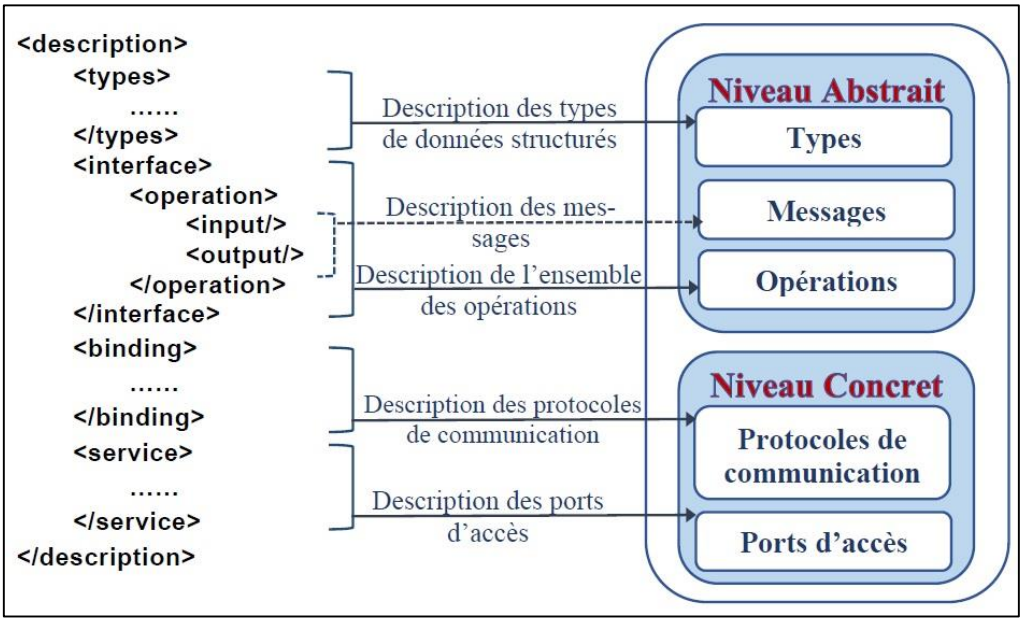

Figure 6 : Structures générale d'un document WSDL 2.0.

Dans la suite de cette section, nous illustrons l'utilisation de chaque élément XML composant un document WSDL 2.0 par un exemple de service web. Le service web exemple (**Figure 7**) est nommé *GlobalWeather* et il est proposé par le fournisseur de service web *webserviceX.net*. Ce service web renvoi, en précisant le pays et la ville, un ensemble de données concernant la météorologie (température, visibilité, vitesse du vent, etc.).

L'élément **"description"** est l'élément racine d'un document WSDL 2.0, il est utilisé afin de déclarer les espaces de noms (lignes 3 à 6) utilisés tout au long du document.

L'élément **"types"** est un conteneur définissant, à travers un schéma XML, les types des données des messages échangés par le service (lignes 7 à 22). Les données peuvent être soit simples soit complexes (par exemple, l'élément *GetWeather* de la ligne 9 est composé d'un élément simple *CityName* et d'un autre élément simple *CountryName*).

L'élément **"interface"** décrit l'ensemble des opérations fournies par le service web. Une opération (élément *operation*) est une description abstraite d'une action accomplie par le service. Les messages sont définis au niveau de l'opération et sont deux types : soit des messages reçus par le service (élément *input*), soit des messages que le service envoie au client (élément *output*). L'interface *GlobalWeatherInterface* (lignes 23 à 32) du service exemple est

composée de deux opérations (*GetWeather* et *GetCitiesByCountry*), chaque opération contient deux messages échangés entre le service et le client. Par exemple, l'opération *GetWeather* (lignes 24 à 27) est composée du message d'entrée *GetWeather* et du message de sortie *GetWeatherResponse*.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <wsdl:description
3     xmlns:tns="http://www.webserviceX.NET"
4     xmlns:s="http://www.w3.org/2001/XMLSchema"
5     targetNamespace="http://www.webserviceX.NET"
6     xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
7    <wsdl:types>
8      <s:schema targetNamespace="http://www.webserviceX.NET">
9        <s:element name="GetWeather">
10         <s:complexType>
11           <s:sequence>
12             <s:element minOccurs="0" name="CityName" type="s:string" />
13             <s:element minOccurs="0" name="CountryName" type="s:string" />
14           </s:sequence>
15         </s:complexType>
16       </s:element>
17       <s:element name="GetWeatherResponse"> ... </s:element>
18       <s:element name="GetCitiesByCountry"> ... </s:element>
19       <s:element name="GetCitiesByCountryResponse"> ... </s:element>
20       <s:element name="string" nillable="true" type="s:string" />
21     </s:schema>
22   </wsdl:types>
23   <wsdl:interface name="GlobalWeatherInterface">
24     <wsdl:operation name="GetWeather">
25       <wsdl:input element="tns:GetWeather" />
26       <wsdl:output element="tns:GetWeatherResponse" />
27     </wsdl:operation>
28     <wsdl:operation name="GetCitiesByCountry">
29       <wsdl:input element="tns:GetCitiesByCountry" />
30       <wsdl:output element="tns:GetCitiesByCountryResponse" />
31     </wsdl:operation>
32   </wsdl:interface>
33   <wsdl:binding name="GlobalWeatherSoap" interface="tns:GlobalWeatherInterface">
34     <wsdl:operation ref="GetWeather"
35                     soapAction="http://www.webserviceX.NET/GetWeather" />
36    <wsdl:operation ref="GetCitiesByCountry"
37                     soapAction="http://www.webserviceX.NET/GetCitiesByCountry" />
38   </wsdl:binding>
39   <wsdl:service name="GlobalWeather" interface="tns:GlobalWeatherInterface">
40     <wsdl:endpoint name="GlobalWeatherSoapEndpoint" binding="tns:GlobalWeatherSoap"
41                    address="http://www.webservicex.net/globalweather.asmx"  />
42   </wsdl:service>
43 </wsdl:description>
```

Figure 7 : Description du service web *GlobalWeather* en WSDL 2.0.

L'élément **"binding"** (lignes 33 à 38) définit les protocoles de communication et fournit les détails nécessaires pour accéder au service. Pour les deux opérations du service web exemple, le protocole de communication à utiliser est SOAP.

L'élément **"service"** (lignes 39 à 42) précise la localisation du service Web. Pour chaque interface décrite, un élément service lui est associé. Le sous-élément

endpoint définit un port d'accès en référençant l'élément binding associé et en déclarant l'URL localisant le service (avec l'attribut *address*). Ceci permet qu'une interface d'un service possède plus d'une localisation (*i.e.* plus d'un élément *endpoint*) pour répondre aux problèmes d'indisponibilité.

3.3 Description sémantique

Le standard WSDL, quelle que soit la version utilisée (1.1 ou 2.0), présente des lacunes de précision dans la description d'un service web. Les lacunes de cette description syntaxique sont liées au niveau d'expressivité faible qui consiste seulement à énumérer les opérations et à décrire les types des paramètres d'entrées et de sorties utilisés. Cette description reste insuffisante et ne caractérise pas la sémantique des fonctionnalités accomplies par le service. Cependant, le manque de la sémantique dans WSDL empêche la découverte automatique et par conséquent l'invocation et la composition automatiques de services.

Pour pallier le manque de la sémantique, la communauté du web sémantique propose de rajouter une couche au-dessus de WSDL complétant la description syntaxique par des précisions sémantiques. L'ajout de la sémantique aux services web a donné naissance aux services web sémantiques (SWS). L'objectif des services web sémantiques est de faciliter les tâches liées à leur utilisation, telles que la découverte, la sélection, l'orchestration et l'invocation, par le biais de leurs descriptions qui rendent la sémantique explicite et compréhensible par les machines. De nombreuses approches ont été proposées afin de décrire sémantiquement les services web. Leur description sémantique peut être faite de deux manières. La première consiste à développer un langage de description sémantique décrivant les services web ainsi que leur sémantique dans un seul bloc. La deuxième manière consiste à annoter les approches existantes avec de l'information sémantique. Nous classifions donc les approches de description sémantique des services web en deux grandes classes : la première classe d'approches concerne les langages de description sémantique, telles que OWL-S et WSMO. La deuxième classe d'approches concerne les approches de description à base d'annotations, telles que WSDL-S, SAWSDL et USDL.

A. Approches basées sur des langages sémantiques

Certaines approches se basent sur des langages sémantiques afin de décrire les services web. Nous décrivons ci-après les deux langages sémantiques de services les plus connus à savoir : OWL-S et WSMO.

OWL-S

Le langage OWL-S (*Ontology Web Language for Services*) (Martin, et al., 2004) est une ontologie OWL de haut niveau construite spécialement pour la description sémantique des services web. Il a été intégré au consortium W3C en 2004 et il est compatible avec des formats de description syntaxique tels que WSDL. L'objectif principal du langage OWL-S est de résoudre les ambiguïtés et de rendre la description d'un service compréhensible et interprétable par une machine.

Une description OWL-S se compose de trois éléments : le *service profile*, le *service model* et le *service grounding* (**Figure 8**). Ces trois éléments permettent de décrire l'aspect fonctionnel et non fonctionnel du service, de décrire la composition de services et de relier cette description avec l'interface WSDL.

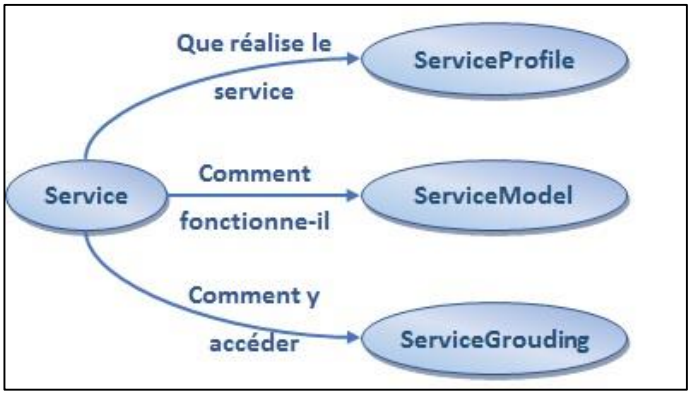

Figure 8 : Les éléments de haut niveau de OWL-S.

Le *Service Profile* (que réalise le service ?) permet la description, la publication et la découverte des services en spécifiant des propriétés fonctionnelles (comme les entrées, les sorties, les préconditions et les post conditions des opérations) et des propriétés non fonctionnelles (comme la réputation, la catégorie et la qualité du service d'une manière générale). Le *Service Profile* est utilisé à la fois par les fournisseurs pour publier leurs services et par les clients pour spécifier leurs besoins.

Le *Service Model* (comment fonctionne-il ?) décrit comment le service fonctionne et comment un client peut interagir avec le service. Dans cet élément du langage OWL-S, le service est modélisé comme un processus (Process). Un processus est une spécification de la manière dont le client peut interagir avec le service. Il est composé d'un ensemble d'information : son nom, les participants à son exécution (un client simple ou d'autres services Web), ce qu'il fait, les conditions d'utilisation et ses effets, son résultat et ses paramètres (entrée, sortie). Trois types de processus existent: atomique, simple et composé. Un *processus atomique* représente le niveau le plus fin pour un processus et correspond à une

action que le service peut effectuer en une seule interaction avec le client. Un *processus simple* est utilisé pour fournir une vue d'un processus atomique ou une représentation simplifiée d'un processus composé. Un *processus composé* est décomposable en d'autres processus (composés ou non) ; sa décomposition peut être spécifiée en utilisant un ensemble de structures de contrôles tels que : *Sequence*, *Split*, *If-Then-Else, etc*.

Le *Service Grounding* (comment y accéder ?) permet de concrétiser la définition du processus donnée dans le *Service Model*. Il décrit les moyens d'accès au service en spécifiant le protocole et les numéros de port de communication, les formats de messages, l'encodage des paramètres et d'autres détails utilisés au moment de l'invocation de services.

Le *Service Profile* et le *Service Model* sont considérés comme des représentations abstraites, tandis que le *Service Grounding* est une représentation concrète. Afin d'illustrer la description sémantique en OWL-S, nous reprenons l'exemple du service web *GlobalWeather* (précédemment décrit en WSDL). La **Figure 9** présente un extrait de la description OWL-S de ce service. Elle est composée du profil du service (*GlobalWeatherProfile*), du processus du service (*GlobalWeatherProcess*) et du grounding du service (*WsdlGrounding*).

```
1  <service:Service rdf:ID="GlobalWeatherService">
2    <service:presents rdf:resource="&my_profile;#GlobalWeatherProfile"/>
3    <service:describedBy rdf:resource="&my_process;#GlobalWeatherProcess"/>
4    <service:supports rdf:resource="&my_grounding;#WsdlGrounding"/>
5  </service:Service>
```

Figure 9 : Description du service web GlobalWeather en OWL-S.

La **Figure 10** présente le profil (*Service Profile*) du service *GlobalWeather*. Tout d'abord, un identifiant est affecté à ce profil (ligne 1). On précise ensuite le service et le processus auquel appartient ce profil (lignes 2 et 3). Le nom et la description textuelle du service sont donnés dans les lignes 4 à 7. Ensuite, les coordonnées nécessaires pour contacter le fournisseur du service sont précisées dans l'élément *contactInformation* (lignes 8 à 10). Enfin, une description fonctionnelle en termes de IOPEs est mentionnée dans les lignes 11 à 14.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

```
1  <profileHierarchy:GlobalWeather rdf:ID="GlobalWeatherProfile">
2    <service:presentedBy rdf:resource="&my_service;#GlobalWeatherService"/>
3    <profile:has_process rdf:resource="&my_process;#GlobalWeatherProcess"/>
4    <profile:serviceName> GlobalWeatherService </profile:serviceName>
5    <profile:textDescription>
6      Récupérer les informations météorologiques des villes à travers le monde
7    </profile:textDescription>
8    <profile:contactInformation>
9      Les informations nécessaires pour contacter le fournisseur du service
10   </profile:contactInformation>
11   <profile:hasInput ... />
12   <profile:hasPrecondition ... />
13   <profile:hasResult ... />
14   <profile:hasOutput ... />
15 </profileHierarchy:GlobalWeather>
```

Figure 10 : Le profil du service web GlobalWeather.

La **Figure 11** présente le processus (*Service Model*) du service *GlobalWeather*. Ce dernier est composé de deux processus atomiques : *GetWeather* (lignes 2 à 24) et *GetCitiesByCountry* (lignes 25 à 40). Chaque processus atomique est décrit par ses entrées et ses sorties.

```
1  <rdf:RDF>
2    <process:AtomicProcess rdf:ID="GlobalWeatherSoap_GetWeather">
3      <process:hasInput>
4        <process:Input rdf:ID="CityName">
5          <process:parameterType rdf:datatype="&xsd;#anyURI">
6            http://.../Weather.owl#CityNameType
7          </process:parameterType>
8        </process:Input>
9      </process:hasInput>
10     <process:hasInput>
11       <process:Input rdf:ID="CountryName">
16     </process:hasInput>
17     <process:hasOutput>
18       <process:Output rdf:ID="CityWeather">
23     </process:hasOutput>
24   </process:AtomicProcess>
25   <process:AtomicProcess rdf:ID="GlobalWeatherSoap_GetCitiesByCountry">
26     <process:hasInput>
27       <process:Input rdf:ID="CountryName">
32     </process:hasInput>
33     <process:hasOutput>
34       <process:Output rdf:ID="CitiesName">
39     </process:hasOutput>
40   </process:AtomicProcess>
41 </rdf:RDF>
```

Figure 11 : Le processus du service web GlobalWeather.

La **Figure 12** illustre l'accès au service *GlobalWeather* par le biais de son *Service Grounding*. Ce service possède deux accès *GlobalWeatherSoap_GetWeather_Grounding* (lignes 8 à 27) et *GlobalWeatherSoap_GetCitiesByCountry_Grounding* (lignes 28 à 47). Ces deux points d'accès correspondent aux deux opérations du service : *GetWeather* et *GetCitiesByCountry*. Chaque point d'accès est lié au processus atomique

Rapport-Gratuit.com

correspondant dans le *Service Model*. Cette liaison est concrétisée en spécifiant des références vers les éléments du document WSDL.

```
1  <rdf:RDF>
2    ...
3    <grounding:WsdlGrounding rdf:ID="WsdlGrounding">
4      <service:supportedBy rdf:resource="GlobalWeatherService"/>
5      <grounding:hasAtomicProcessGrounding rdf:resource="#GlobalWeatherSoap_GetWeather_Grounding"/>
6      <grounding:hasAtomicProcessGrounding rdf:resource="#GlobalWeatherSoap_GetCitiesByCountry_Grounding"/>
7    </grounding:WsdlGrounding>
8    <grounding:WsdlAtomicProcessGrounding rdf:ID="GlobalWeatherSoap_GetWeather_Grounding">
9      <!-- La localisation du document WSDL -->
10     <grounding:wsdlDocument rdf:datatype="&xsd;#anyURI">
11       http://.../GlobalWeatherGrounding.wsdl
12     </grounding:wsdlDocument>
13     <!-- Grounding du Processus atomique GlobalWeatherSoap_GetWeather -->
14     <grounding:owlsProcess rdf:resource="http://.../GlobalWeatherProcess.owl#GlobalWeatherSoap_GetWeather"/>
15     <!-- Les opérations WSDL correspondantes -->
16     <grounding:wsdlOperation>
17       <grounding:WsdlOperationRef>
18         <grounding:portType rdf:datatype="&xsd;#anyURI">...</grounding:portType>
19         <grounding:operation rdf:datatype="&xsd;#anyURI">GetWeather</grounding:operation>
20       </grounding:WsdlOperationRef>
21     </grounding:wsdlOperation>
22     <grounding:wsdlInputMessage rdf:datatype="&xsd;#anyURI">...</grounding:wsdlInputMessage>
23     <grounding:wsdlInput>...</grounding:wsdlInput>
24     <grounding:wsdlOutputMessage rdf:datatype="&xsd;#anyURI">...</grounding:wsdlOutputMessage>
25     <grounding:wsdlOutput>...</grounding:wsdlOutput>
26     <grounding:wsdlReference rdf:datatype="&xsd;#anyURI">...</grounding:wsdlReference>
27   </grounding:WsdlAtomicProcessGrounding>
28   <grounding:WsdlAtomicProcessGrounding rdf:ID="GlobalWeatherSoap_GetCitiesByCountry_Grounding">
48 </rdf:RDF>
```

Figure 12 : Le Service Grounding du service web GlobalWeather.

WSMO

L'ontologie WSMO (*Web Service Modeling Ontology*) (Roman, et al., 2005) (De Bruijn, et al., 2005) est une spécification similaire à OWL-S. WSMO est initiée par l'ESSI WSMO workgroup (ESSI WSMO working group, 2004) et elle est basée sur le WSMF (Web Service Modeling Framework) (Fensel, et al., 2002) qui spécifie les éléments principaux pour décrire les services web sémantiques. Le but de WSMO est d'automatiser les tâches liées aux services web (découverte, sélection, composition, médiation et invocation). WSMO est organisée autour de quatre éléments principaux : Les ontologies, les objectifs, les services web et les médiateurs, comme le montre la **Figure 13**.

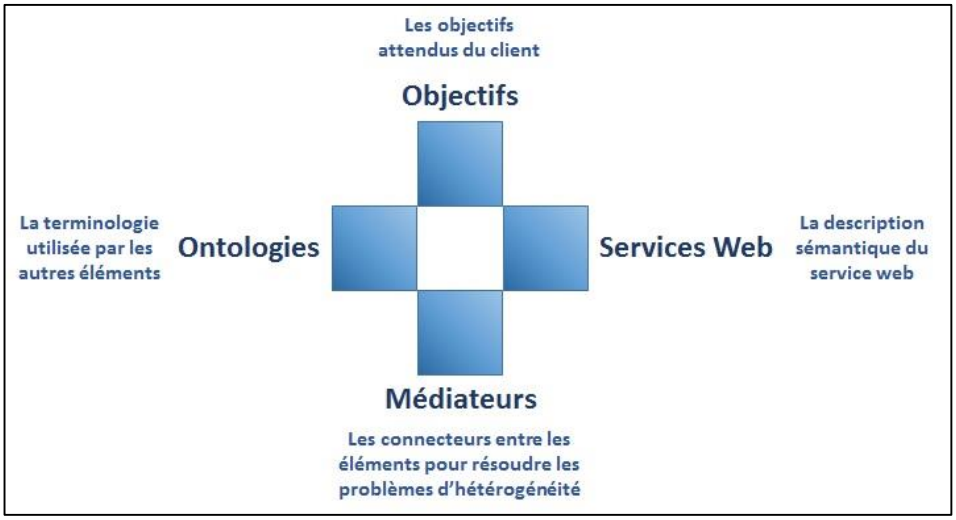

Figure 13 : Les éléments fondamentaux de WSMO.

- **Les ontologies** fournissent la terminologie utilisée par les autres éléments WSMO, afin de décrire le vocabulaire du domaine de connaissance en termes de concepts, relations, fonctions, instances et axiomes.
- **Les services web** définissent la description sémantique des services web, en spécifiant les propriétés fonctionnelles et non fonctionnelles, ainsi d'autres aspects importants pour leurs interactions. Tous ces aspects des services web sont décrits en utilisant la terminologie définie par les ontologies.
- **Les objectifs** représentent les souhaits de l'utilisateur attendus en utilisant le service. Un objectif décrit la fonctionnalité, les entrées/sorties, les préconditions et les postconditions d'un service Web.
- **Les médiateurs** permettent de résoudre les problèmes d'incompatibilités entre les différents éléments WSMO. Les incompatibilités sont à trois niveaux : données, processus et protocoles. Les incompatibilités au niveau données sont signalées lorsque les services web utilisent différents terminologies. Les incompatibilités au niveau processus surgissent au moment de la combinaison des services web. Les incompatibilités au niveau protocole se présentent au moment de l'établissement de la communication entre les services web.

Le langage WSML (Web Service Modeling Language) (De Bruijn, et al., 2006) est utilisé pour décrire formellement tous les éléments WSMO. Afin d'illustrer la description sémantique en WSMO, nous reprenons l'exemple du service web *GlobalWeather* (précédemment décrit en WSDL et OWL-S). La **Figure 14** présente un extrait de la description WSMO de ce service au moyen du langage WSML.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

```
1  namespace   <<http://www.wsmo.org/TR/d2/v1.3/>>
2     dc:   <<http://purl.org/dc/elements/1.1#>>
3     wt:   <<http://localhost/ontologies/weather-ont.owl#>>
4     targetnamespace:  <<http://www.wsmo.org/TR/d2/v1.3/>>
5
6  webservice
7     nonFunctionalProperties
8        dc:title hasValue "GlobalWeather"
9        dc:description hasValue "web service for getting the cities weather."
10    endNonFunctionalProperties
11
12    capability
13       precondition
14          axiom _#
15             nonFunctionalProperties
16                dc:description hasValue "the input has to be a CountryName."
17             endNonFunctionalProperties
18          definedBy
19             ?CountryName memberof wt:CountryName
20
21       assumption
22          definedBy
23             ...
24
25       postcondition
26          axiom _#
27             nonFunctionalProperties
28                dc:description hasValue "the output of the service is a CityWeather."
29             endNonFunctionalProperties
30          definedBy
31             ?CityWeather memberof wt:CityWeather
32
33       effect
34          definedBy
35             ...
36    interface
37       nonFunctionalProperties
38                dc:description hasValue "describes the Interface of Web Service."
39       endNonFunctionalProperties
40       ...
```

Figure 14 : Extrait du service Web GlobalWeather décrit en WSML dans WSMO.

B. Approches de description à base d'annotations

Contrairement aux approches précédentes (OWL-S et WSMO), plusieurs travaux proposent d'étendre les standards existants de description de services web. Ces nouvelles approches n'ont pas été conçues pour remplacer une ontologie ou un langage d'ontologie mais par contre elles prévoient des mécanismes par lesquels des concepts ontologiques définis en dehors des documents WSDL, peuvent être référencés pour annoter sémantiquement des descriptions WSDL.

Plus précisément, l'annotation sémantique consiste à enrichir et à compléter la description syntaxique d'un service. Elle permet de mettre en correspondance les éléments de la description syntaxique avec des éléments d'ordre sémantique

(faisant référence à des ontologies). Nous présentons dans la suite, trois approches principales de description à base d'annotation sémantique, à savoir WSDL-S, USDL et SAWSDL.

WSDL-S

Le standard WSDL des services web fonctionne à un niveau syntaxique simple car il ne dispose pas de la sémantique nécessaire pour représenter de façon signifiante les paramètres des services web. Ceci représente un problème récurrent pour les utilisateurs des services web. Pour apporter une première solution à ce problème et en se basant sur le premier travail proposé par le laboratoire LSDIS de l'université de Georgia, le W3C a développé le langage WSDL-S (*Web Services Description Language - Semantic*) (Akkiraju, et al., 2005) qui propose des mécanismes avec lesquels les annotations sémantiques peuvent être ajoutées à des éléments WSDL.

L'objectif principal de WSDL-S est de fournir un système d'annotation sémantique compatible avec les langages existants. Le système d'annotation de WSDL-S se base sur des mécanismes permettant de référencer des concepts des modèles sémantiques définis à l'extérieur du document WSDL. Plus concrètement, le WSDL-S étend le WSDL en rajoutant trois éléments (*category*, *precondition* et *effect*) et deux attributs (*modelReference* et *schemaMapping*). Les éléments introduits permettent de rajouter des nouvelles informations, qui n'étaient pas prises en compte dans WSDL, comme la catégorie d'une interface, les préconditions et les effets d'une opération. Tandis que les attributs rajoutés permettent de référencer des concepts dans des ontologies de référence. Nous illustrons dans la suite les annotations WSDL-S par l'exemple de l'interface *PurchaseOrder* présenté dans (Akkiraju, et al., 2005). La **Figure 15** présente un extrait de cet exemple.

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <definitions ...
3    xmlns:wssem="http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/examples/purchaseOrder.wsdl"
4    xmlns:POOntology="http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/ontologies/PurchaseOrder.owl"
5    xmlns:Rosetta="http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/ontologies/rosetta.owl">
6      <types>
29     <interface name="PurchaseOrder">
30             <!--Category is added as an extensible element of an interface-->
31             <wssem:category name="Electronics"
32                             taxonomyURI="http://www.naics.com/" taxonomyCode="443112" />
33             <operation name="processPurchaseOrder" pattern="wsdl:in-out"
34                     wssem:modelReference="Rosetta:RequestPurchaseOrder" >
35              <input   messageLabel ="processPurchaseOrderRequest"
36                     element="tns:processPurchaseOrderRequest"/>
37              <output  messageLabel ="processPurchaseOrderResponse"
38                     element="processPurchaseOrderResponse"/>
39             <!--Precondition and effect are added as extensible elements on an operation-->
40             <wssem:precondition name="ExistingAcctPrecond"
41                                 wssem:modelReference="POOntology#AccountExists"/>
42             <wssem:effect name="ItemReservedEffect"
43                           wssem:modelReference="POOntology#ItemReserved"/>
44             </operation>
45     </interface>
46   </definitions>
```

Figure 15 : Exemple WSDL-S.

L'élément **category**, sous élément de l'élément *interface*, indique la catégorie d'un service, une information qui peut être utilisée lors de la publication du service dans un registre (comme UDDI). Dans l'exemple, la catégorie de l'interface *PurchaseOrder* est *Electronics* (ligne 31) de la taxonomie disponible à l'adresse "http://www.naics.com/".

L'élément **precondition**, sous élément de l'élément *operation*, indique l'ensemble des assertions qui doivent être vérifiées avant l'invocation de l'opération. Dans l'exemple, pour l'opération *processPurchaseOrder*, l'élément *precondition* (ligne 40) indique que pour passer une commande à une compagnie, il faut avoir un compte existant avec cette compagnie.

L'élément **effect**, sous élément de l'élément *operation*, indique le résultat de l'exécution de l'opération. Dans l'exemple, pour l'opération *processPurchaseOrder*, l'élément *effect* (ligne 42) indique qu'après l'exécution de cette opération un article est réservé pour la commande effectuée.

L'attribut **modelReference** peut être ajouté aux éléments WSDL suivants : *complexType*, *element* et *operation*, ainsi qu'aux éléments de l'extension WSDL-S : *precondition* et *effect*. Il définit une association avec l'un de ces éléments et un concept d'ontologie. Dans l'exemple, pour l'opération *processPurchaseOrder*, l'attribut *modelReference* (ligne 34) établit le lien avec le concept *RequestPurchaseOrder* de l'ontologie *Rosetta*.

L'attribut **schemaMapping** peut être ajouté aux éléments WSDL suivants : *element* et *complexType* (types de données XML schéma). Il permet de gérer les conflits structurels entre les types de données XML schéma et leurs correspondants concepts de modèle sémantique.

USDL

Le langage USDL (*Universal Service-Semantics Description Language*) (BANSAL, et al., 2005) est une autre approche de description sémantique des services web, elle est basée à la fois sur le standard WSDL et le langage OWL. Le WSDL est utilisé pour la description syntaxique du service web, tandis que le langage OWL est utilisé pour exprimer la sémantique du service en se basant sur l'ontologie OWL WordNet (Miller, 1995).

Cette approche est plus pratique et plus souple que les anciennes approches parce que la description des services d'un côté est créée plus facilement par les humains et de l'autre côté elle est traitée plus facilement par les ordinateurs. Le langage USDL décrit formellement et sémantiquement les services web. Au niveau syntaxique, le WSDL décrit les types, les paramètres et les messages des services. Cette description syntaxique est complétée par une description sémantique en utilisant l'ontologie OWL WordNet, qu'est considérée comme une représentation commune des concepts du monde réel. Les relations fournies par cette ontologie telles que l'hyponymie, l'antonymie et la synonymie sont utilisées pour donner du sens aux éléments WSDL.

USDL propose d'enrichir la description syntaxique (WSDL) par une description sémantique. Cet enrichissement est réalisé en décrivant trois éléments principaux de la description USDL : *Concept*, *Affects* et *Conditions*. La **Figure 16** illustre ces trois éléments de description USDL.

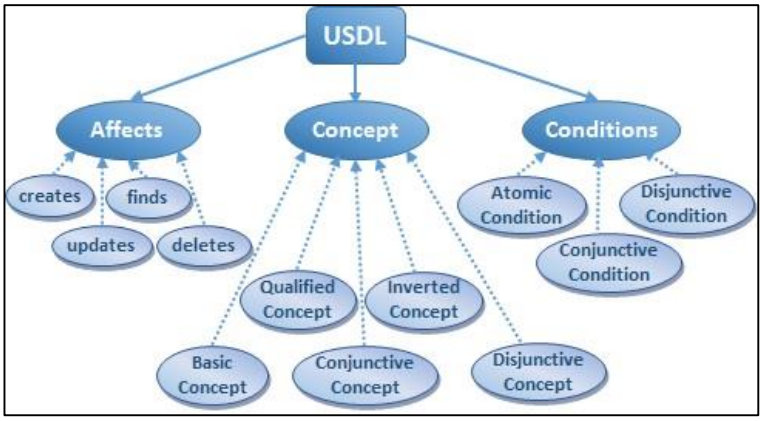

Figure 16 : Principaux éléments de la description USDL.

- L'élément *Concept* est une classe générique pour modéliser des concepts du monde réel. Cette classe est utilisée pour définir la sémantique des éléments WSDL et elle propose de les associer à des concepts basiques, des concepts qualifiés, des concepts inversés, des concepts conjonctifs et des concepts disjonctifs de l'ontologie WordNet.
- L'élément *Affects* est une classe générique pour décrire des effets du service sur le monde réel suite à une opération de création, de mise à jour, de suppression ou de recherche.
- L'élément *Conditions* est une classe générique pour décrire des contraintes pour le service. Les conditions sont représentées sous l'une des trois formes suivantes : atomique, conjonctive ou disjonctive.

SAWSDL

Le SAWSDL (*Semantic Annotations for Web Services Description Language and XML Schema*) (Farrell, et al., 2007), recommandé par W3C en août 2007, est un langage sémantique de description de service web. Le SAWSDL est le successeur de WSDL-S, il est évolutif et compatible avec les standards des services web existants et plus spécifiquement avec WSDL.

L'objectif de SAWSDL est d'ajouter de la sémantique à la description WSDL des services et des schémas XML. Pour cela, il fournit des mécanismes permettant d'annoter sémantiquement d'une part les types de données, les opérations, les entrées et les sorties de WSDL et d'autre part, il ajoute des éléments pour spécifier les préconditions, les effets et les catégories des services web (M'BARECK, 2008). Ces mécanismes d'annotation sont possibles grâce à l'extensibilité de WSDL 2.0 et ils sont indépendants de tout langage de représentation de sémantique.

SAWSDL propose des extensions à WSDL similaires à celles proposées par WSDL-S. Il propose deux sortes d'annotations sémantiques : une première pour associer un élément WSDL ou XML schéma à un concept sémantique (représentée par l'attribut **modelReference**) et une seconde pour spécifier les correspondances entre les éléments XML et les données sémantiques (représentée par les attributs **liftingSchemaMapping** et **loweringSchemaMapping**). La **Figure 17** illustre l'approche SAWSDL et son principe d'annotation.

L'attribut **modelReference** prend comme valeur une ou plusieurs URI, chaque URI pointe vers un concept d'un modèle sémantique. Il est utilisé pour annoter d'une part les éléments XML schéma (définitions des types, les déclarations d'éléments et d'attributs) et de l'autre part les éléments WSDL (interfaces, opérations et messages d'erreur). Cependant, la recommandation

SAWSDL considère que l'annotation de ces éléments par l'attribut **modelReference** apporte un sens particulier.

- L'annotation des éléments XML schéma (*xs:complexType, xs:simpleType, xs:element* et *xs:attribute*) permet de donner du sens aux structures de données utilisées pour définir les paramètres des messages échangés.
- L'annotation de l'interface (*wsdl:interface*) apporte une définition sémantique de l'interface en spécifiant des aspects comportementaux, elle peut aussi apporter une classification selon certains modèles.
- L'annotation de l'opération (*wsdl:operation*) fournit une description très détaillée de l'opération, précise ses effets et ses aspects comportementaux.
- L'annotation du message d'erreur (*wsdl:fault*), qui représente un message d'erreur lors de l'appel d'une méthode de l'interface, donne un sens au message d'erreur.

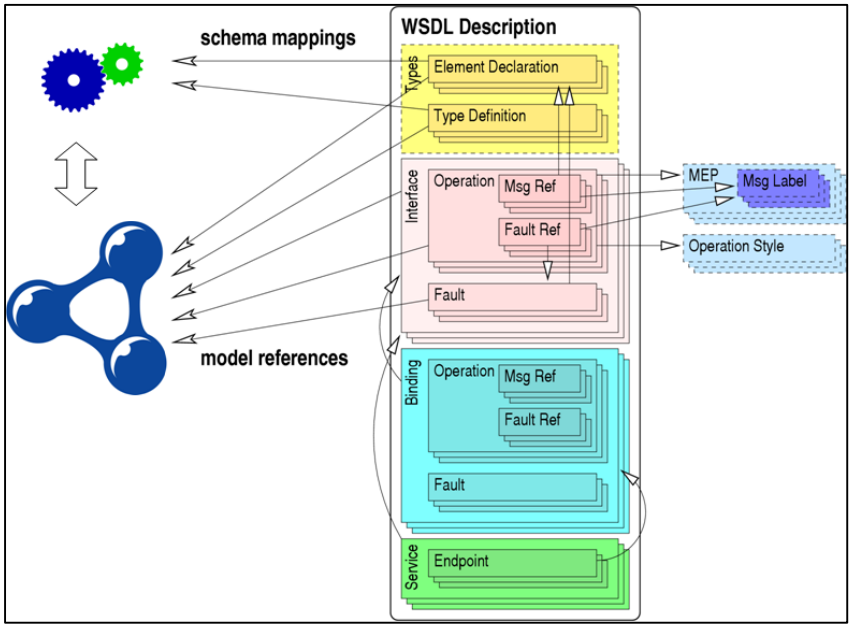

Figure 17 : Approche SAWSDL (Kopecky, 2007).

Les deux autres attributs **liftingSchemaMapping** et **loweringSchemaMapping** sont ajoutés aux éléments de schéma XML et sont utilisés pour spécifier la correspondance (dit Mapping) entre ces éléments XML et les concepts sémantiques. Le premier attribut associe aux éléments XML un modèle sémantique. Inversement, le deuxième attribut associe au modèle sémantique une structure XML. Ces mappings sont utiles, lors de l'invocation du service, lorsque la structure de données (schéma XML) ne correspond pas de manière intuitive à la structure du modèle sémantique (ontologie). L'utilisation de ces deux attributs n'est pas illustrée ici car nous nous ne préoccupons pas dans notre travail à la cohérence structurelle des données.

Afin d'illustrer l'utilisation des annotations SAWSDL, nous reprenons l'exemple du service web *GlobalWeather*. La **Figure 18** présente un extrait de la description SAWSDL de ce service.

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <wsdl:description
3      xmlns:tns="http://www.webserviceX.NET"
4      xmlns:s="http://www.w3.org/2001/XMLSchema"
5      targetNamespace="http://www.webserviceX.NET"
6      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl"
7      xmlns:sawsdl="http://www.w3.org/ns/sawsdl">
8   <wsdl:types>
9     <s:schema targetNamespace="http://www.webserviceX.NET">
10      <s:element name="GetWeather"
11       sawsdl:modelReference="http://localhost/weather-ont.owl#Location">
12        <s:complexType>
13          <s:sequence>
14            <s:element minOccurs="0" name="CityName" type="s:string" />
15            <s:element minOccurs="0" name="CountryName" type="s:string" />
16          </s:sequence>
17        </s:complexType>
18      </s:element>
19      <s:element name="GetWeatherResponse"
20       sawsdl:modelReference="http://localhost/weather-ont.owl#WeatherObservation">
23      <s:element name="GetCitiesByCountry"> ... </s:element>
24      <s:element name="GetCitiesByCountryResponse"> ... </s:element>
25      <s:element name="string" nillable="true" type="s:string" />
26    </s:schema>
27  </wsdl:types>
28  <wsdl:interface name="GlobalWeatherInterface">
29    <wsdl:operation name="GetWeather">
30      <wsdl:input element="tns:GetWeather" />
31      <wsdl:output element="tns:GetWeatherResponse" />
32    </wsdl:operation>
33    <wsdl:operation name="GetCitiesByCountry">
37  </wsdl:interface>
38  <wsdl:binding name="GlobalWeatherSoap" interface="tns:GlobalWeatherInterface">
44  <wsdl:service name="GlobalWeather" interface="tns:GlobalWeatherInterface">
48  </wsdl:description>
```

Figure 18 : Description SAWSDL du service GlobalWeather.

Nous illustrons dans cet exemple l'annotation sémantique de la structure de données des entrées et des sorties de l'opération *GetWeather* (lignes 29 à 32) du service web. L'élément *GetWeather* (ligne 10) contient les informations de la ville et du pays, il est annoté par le concept *Location* (ligne 11) de l'ontologie *weather-ont*. Tandis que l'élément *GetWeatherResponse* (ligne 19), qui englobe un ensemble d'informations concernant les observations météorologiques (telles que la température, l'humidité et la vitesse du vent), est annoté par le concept *WeatherObservation* (ligne 20) de la même ontologie.

Le schéma illustré dans la **Figure 19** présente et résume notre classification des approches de description de services web.

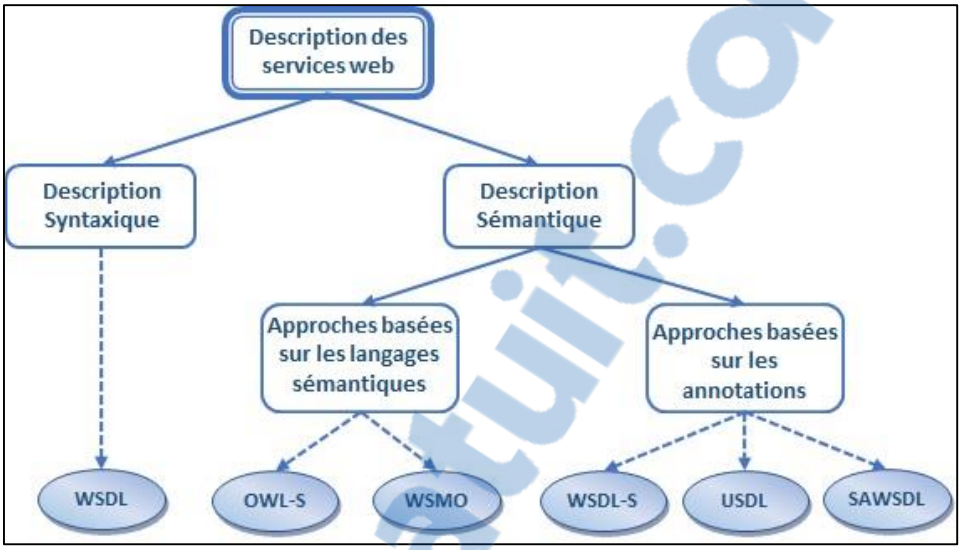

Figure 19 : Classification des approches de description de services web.

3.4 Travaux de description des services web

Nous présentons dans cette section quelques travaux de description des services web. La majorité de ces travaux se basent sur les approches de description décrites auparavant.

DIANE Service Description, 2005

Dans ce travail, la description des services est spécifiée par le langage DSD (DIANE Service Description) (Klein, et al., 2005). Ce langage n'est pas basé sur l'une des ontologies existantes ou un langage de programmation logique comme la logique de description ou la logique floue souvent utilisées, mais il est basé sur un nouveau langage d'ontologie et un mécanisme de raisonnement spécialisés pour la découverte de service connu sous le nom *Diane Elements* (DE).

L'approche DSD utilise des constructeurs spécialisés fournis par le langage DE pour décrire les services. Le langage DE a été créé en combinant les concepts des autres langages de représentation de connaissance : les langages orienté cadre (frame languages), la logique de description, la logique floue et la logique modale. L'idée principale de conception de DSD est de se baser sur un simple langage et l'étendre par les concepts des autres langages afin de décrire toutes les caractéristiques des services. Pour cela, le langage DE est composé de deux parties : (1) un simple langage d'ontologie à base de cadre pour décrire le domaine

Rapport-Gratuit.com

(appelé DE-I) et (2) un petit ensemble d'extensions supplémentaires à base de logique qui sont exclusivement nécessaires pour représenter les fonctionnalités des services (appelé DE-II). Le travail proposé fournit une architecture globale pour la description de services Web sémantiques, avec un fort support de raisonnement, qui emprunte de nombreux avantages apportés par les approches OWL-S et WSMO.

Travail proposé par (Martin, et al., 2007)

Les auteurs de ce travail (Martin, et al., 2007) montrent comment réaliser des correspondances entre les constructeurs OWL-S et les annotations SAWSDL. Ils expliquent quelles sont les constructeurs OWL-S qui peuvent être appropriés à une utilisation avec les diverses annotations SAWSDL. Ils fournissent ainsi la logique et les directives de leur utilisation. L'idée principale de ce travail est de continuer à utiliser les constructeurs OWL-S et d'adopter une perspective basée sur les annotations SAWSDL. Pour cela, les auteurs donnent une série de recommandations pour l'utilisation des constructeurs OWL-S comme référence de l'attribut *modelReference* de chaque élément dans WSDL. En d'autres termes, comment l'attribut *modelReference* de chaque élément WSDL peut faire référence à des constructeurs OWL-S.

WSMO-Lite, 2007

WSMO-Lite (Vitvar, et al., 2007) (Vitvar, et al., 2008) (Fensel, et al., 2010) se présente à la fois comme une extension des annotations SAWSDL ainsi qu'une ontologie légère RDFS. L'ontologie, inspirée du modèle WSMO, permet de classifier les fonctionnalités du service, d'exprimer des propriétés non-fonctionnelles comme des valeurs constantes de paramètres de qualité de service ainsi que des préconditions et des effets. Ces derniers sont exprimés dans le format de règles RIF préconisé par le W3C.

WSMO-Lite définit une ontologie pour la sémantique des services, utilisée directement dans SAWSDL pour annoter les services. Deux types d'annotations sont utilisés dans WSMO-Lite, les annotations de référence et les annotations de transformation. Une annotation de référence relie n'importe quel élément WSDL avec un concept sémantique de WSMO-Lite (équivalente à l'attribut de SAWSDL *modelReference*). Une annotation de transformation spécifie une transformation de données appelé levage d'un élément XML schéma à un élément de l'ontologie et une transformation inverse (de l'ontologie à XML) (équivalent aux attributs SAWSDL *liftingSchemaMapping* et *loweringSchemaMapping*).

WSMO-Lite définit quatre types de descriptions sémantiques de service : modèle d'information, description fonctionnelle, description non fonctionnelle et

description comportemental. Le modèle d'information définit le sens des informations échangées avec le service. Ce modèle est utilisé par les autres descriptions et il généralement appliqué lors de la médiation des données par le biais des techniques de fusion ou d'alignement d'ontologies. La sémantique fonctionnelle est une description statique des fonctionnalités du service, c'est-à-dire que peut le service offrir à ses clients lorsqu'elle est invoquée (ce que fait le service ?). La troisième description, non-fonctionnelle, définit les détails spécifiques à l'environnement d'implémentation ou d'exécution du service, comme le prix, l'emplacement ou la qualité du service. La dernière description, comportemental, spécifie le comportement privé et publique du service comme par exemple le protocole suivi par un client lors de l'utilisation du service (comment interagir avec le service ?).

Le langage YASA4WSDL, 2008

Les auteurs de ce travail (Chabeb, et al., 2008) (Chabeb, et al., 2009) proposent une extension du langage SAWSDL pour la description des services web. Cette extension vient de combler un manque au niveau de la spécification de la nature de l'annotation sémantique dans SAWSDL. La description YASA4WSDL (Yet Another Semantic Annotation 4 WSDL) est enrichie par des références portées sur des concepts techniques de services tels que précondition, effet, opération, etc. Ils utilisent une ontologie technique qui décrit la sémantique des concepts d'un service Web, et une ontologie de domaine qui décrit les concepts d'un domaine métier particulier pour les services web. L'ontologie technique de descripteurs de services web a été définie en intégrant les concepts du méta-modèle WSDL, ceux de l'ontologie OWL-S et WSMO. L'intégration de ces ontologies est réalisée avec différentes techniques d'appariement de concepts. L'ontologie résultante offre une couverture sémantique large des concepts spécifiques aux services Web.

La plateforme PASiS, 2011

Similairement au travail précèdent, les auteurs de la plateforme PASiS (Publishing And Searching for intentional Services) (Aljoumaa, et al., 2011) proposent une extension de SAWSDL pour les services intentionnels. Le Modèle intentionnel de Services MiS partage avec les approches sémantiques la volonté d'enrichir la description des services web et de rencontrer ainsi les attentes des agents métiers. En effet, MiS a été proposé pour la modélisation et la description des services en termes intentionnels, en portant toute la notion d'un service au niveau des objectifs métiers de l'agent. La plateforme PASiS est basée sur l'utilisation de trois ontologies. La première est l'ontologie de services

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

intentionnels qui contient les concepts définissant les éléments du service intentionnel (service, service atomique, agrégat, but, verbe, objet...). La deuxième est l'ontologie de verbes (Urrego, 2005) qui représente des concepts sémantiques et syntaxiques relatifs aux verbes. La troisième, appelée Ontologie des produits, contient la sémantique des termes dans un domaine métier.

Travail proposé par (Brut, et al., 2011)

Les auteurs de ce travail (Brut, et al., 2011) présentent une solution basée sur l'ontologie WSMO pour décrire de manière sémantique l'interface des services web en charge de l'indexation multimédia. La solution considère la possibilité de combiner plusieurs services afin d'obtenir des descripteurs plus riches pour l'indexation et la recherche d'images par le contenu. L'idée a été d'utiliser un format générique XML qui couvre les formats des métadonnées multimédias existants pour décrire la fonctionnalité d'un tel service Web en termes des métadonnées qu'il fournit suite à l'indexation.

Travail proposé par (Christensen, et al., 2015)

Un autre travail lié à la fois à OWL-S et WSDL-S (le prédécesseur de SAWSDL) est présenté dans (Christensen, et al., 2015). Dans ce travail, les auteurs conçoivent et implémentent un annuaire sémantique UDDI, dénoté Sem-UDDI, qui fournit des fonctionnalités pour la publication et la recherche des services web sémantiques et non sémantiques. L'annuaire a été développé pour être capable de gérer les deux standards OWL-S et WSDL-S afin de décrire sémantiquement les services web. Pour comparer les besoins de recherche sémantique avec les services web sémantiques publiés, un ensemble de règles pour le calcul d'un score d'appariement a été développé.

3.5 Etude comparative des approches de description

Dans cette section, nous présentons une étude comparative des différentes approches de description des services web. Nous avons classifié ces approches en deux grandes catégories : des approches dites syntaxiques et des approches dites sémantiques. Nous nous intéressons dans notre étude à comparer les trois approches les plus utilisés dans la description sémantique des services web, à savoir OWL-S, WSMO et SAWSDL. Nous identifions trois critères de comparaison afin d'évaluer les différentes approches de description sémantique des services web :

1. **Adaptabilité / Réutilisabilité :** ce critère mesure le degré d'adaptabilité et de réutilisabilité de l'approche par rapport aux standards et outils de description déjà disponibles.
2. **Extensibilité / Flexibilité :** ce critère mesure le niveau de dépendance de l'approche à un langage de description sémantique.
3. **Explicité / Expressivité :** ce critère donne le niveau d'explicité et de précision de la sémantique de l'approche de description utilisée. En d'autres termes, on étudie les moyens offerts par les approches afin de décrire les services web.

Nous utilisons dans notre étude ces critères pour évaluer et examiner les approches de description des services web. Pour cela, nous identifions dans le tableau suivant (**Tableau 2**) les symboles d'évaluation et leurs interprétations.

Symboles	Interprétations
○	**ne répond pas au critère**
◐	**répond partiellement au critère**
●	**répond parfaitement au critère**

Tableau 2 : Symboles d'évaluation.

Dans la suite, nous présentons notre étude des approches de description des services web par rapport aux critères identifiés ci-dessus.

Adaptabilité / Réutilisabilité

L'adaptation et la réutilisation sont souvent liées. Une approche facilement adaptable favorise une meilleure réutilisation. En termes d'adaptation, deux classes d'approches de description existent. Soit on utilise un langage d'ontologie approprié pour décrire les services web et leurs sémantiques dans la même structure tels que OWL-S et WSMO, soit on utilise des mécanismes d'annotation pour décrire des modèles de description existants tels que SAWSDL. La deuxième classe d'approches facilite l'adaptation des descriptions déjà développées aux annotations et aux solutions proposées.

En termes de réutilisation d'outils, SAWSDL nécessite peu d'efforts de la part des développeurs familiarisés à WSDL contrairement à OWL-S et WSMO. Beaucoup d'outils sont développés dans la cadre du langage WSDL qui nécessite une légère adaptation pour une meilleure réutilisation avec le langage SAWSDL. Par contre, les outils pour les deux autres langages s'avèrent plus difficile à développer, surtout WSMO qui se base sur un langage peu utilisé (le langage WSML).

Extensibilité / Flexibilité

L'extensibilité d'une approche accroitre sa flexibilité. Les approches de description sémantique des services web utilisent des ontologies de domaine pour ajouter des concepts sémantiques à leurs descriptions. Un concept sémantique est utilisé pour annoter une partie de la description du service web par le concept métier correspondant.

Les deux approches OWL-S et WSMO sont des approches dites fermées, car elles permettent d'une part d'annoter les services web par des langages de représentation sémantique spécifiques, OWL pour OWL-S et WSML pour WSMO. D'autre part, elles ne spécifient qu'un ensemble défini mais très limité de concepts qui ne sont pas facilement extensibles. En revanche, l'approche SAWSDL est une approche dite ouverte car elle permet d'utiliser tous types d'ontologies grâce à la séparation entre les mécanismes d'annotation sémantique et la représentation de la description sémantique. Avec une telle approche, les développeurs ont assez de flexibilité pour choisir leurs langages de représentation sémantique.

Explicité / Expressivité

Les approches de description de services web se diffèrent du principe de description sémantique utilisé. Par conséquent, la sémantique ne serait pas autant explicite et précise pour les différentes approches. Les éléments décrits varient d'une approche à l'autre, certaines sont plus expressive sur certains aspects que d'autres. Par exemple, dans certaines approches, on n'explicite même pas que les annotations soient des effets ou des conditions et ceci relève de l'ambiguïté.

Les deux approches OWL-S et WSMO spécifient plus d'éléments à annoter que dans SAWSDL. Dans ce dernier, les annotations sémantiques concernent les parties *type*, *interface*, *operation*, *input* et *output*. Tandis que, les annotations sémantiques dans OWL-S et WSMO décrivent, d'une part, de la même façon la partie opération par les éléments *input*, *output*, *precondition* et *effect*. D'autres parts, plus d'éléments sont spécifiques à chacune des deux : *postcondition* et *assumption* pour WSMO, et *result* et *process* pour OWL-S.

Le tableau présenté ci-dessous (**Tableau 3**) fournit un récapitulatif de notre étude de comparaison.

Approches / Critères	OWL-S	WSMO	SAWSDL
Adaptabilité / Réutilisabilité	◐	◐	●
Extensibilité / Flexibilité	○	○	●
Explicité / Expressivité	●	●	◐
Total	◐ ○ ●	◐ ○ ●	● ● ◐

Tableau 3 : Récapitulatif de l'étude comparative des approches de description.

3.6 Conclusion

Dans ce chapitre, nous avons présenté les approches les plus utilisées dans la littérature pour décrire les services web. Après la présentation du WSDL, modèle omniprésent de description syntaxique, nous avons mis en évidence la nécessité de compléter la description syntaxique par une description sémantique. À cette fin, nous avons revu les approches de description sémantique des services web que nous avons classé en deux grandes catégories. Nous avons présenté dans la première classe les langages de description sémantique, telles que OWL-S et WSMO. Après, dans la deuxième classe, nous avons présenté les approches de description à base d'annotations, telles que WSDL-S, SAWSDL et USDL. Ensuite, nous avons énuméré quelques travaux de description des services web qui se basent sur les langages et les approches de description décrites avant. Finalement, nous avons conclu ce chapitre par une étude comparative des principales approches de description sémantique des services web.

La description est la première phase indispensable dans le cycle de vie d'un système basé sur une architecture SOA. Une description étoffée des services web est un élément clé pour l'automatisation de la découverte et de l'appariement de services. Plusieurs approches de découverte de service web ont été formalisées dans la littérature. Nous en discuterons dans le chapitre suivant.

CHAPITRE 4

DÉCOUVERTE DE SERVICES WEB

4.1 Introduction

4.2 Les approches non logiques

AASDU

DSD-matchmaker

iMatcher1

4.3 Les approches logiques

WSC

ALS

GR

PCEM

4.4 Les approche hybrides

OWLS-MX

FC-Match

WSMO-MX

SAWSDL-MX

4.5 Etude comparative des approches de découverte

Langage de description

Type d'appariement

Eléments d'appariement

Degrés d'appariement

Niveau d'automatisation

4.6 Conclusion

4.1 Introduction

Le Web a beaucoup évolué ces dernières années grâce à l'adoption croissante des services web comme technologie d'implantation de l'architecture SOA, et par la suite l'évolution du web vers un web sémantique. S'ajoute à cela, l'évolution des contraintes des clients qui sont devenus plus exigeants et cherchent à réutiliser des services.

Conformément à l'architecture SOA, les descriptions des services web sont publiées dans des registres spécialement conçus à cet effet (par exemple, UDDI). Au début, la découverte est faite au niveau du registre UDDI, elle est basée essentiellement sur la recherche syntaxique des descriptions WSDL des services Web. Mais avec le développement des technologies du Web sémantique, les techniques de découverte sont devenues essentiellement sémantiques. Cette sémantique est apportée grâce aux ontologies, une des technologies importantes du Web sémantique. Ainsi, des agents logiciels peuvent être développés afin de raisonner sur ces ontologies rendant la découverte des services Web dynamique et automatique.

Malgré que le domaine de découverte de services web soit assez récent, il a été abordé dans un grand nombre de projets et travaux de recherche. Bien que toutes les définitions apparaissant dans la littérature se ressemblent, elles comportent, chacune, des divergences. Voici quelques définitions de la découverte des services:

> *"Web service discovery is the act of locating a machine-processable description of a Web service that may have been previously unknown and that meets certain functional criteria. It involves matching a set of criteria with a set of Web service descriptions. The goal is to find an appropriate Web service." (W3C Working Group Note, 2004).*
>
> *"Localisation automatique des services répondant à une requête utilisateur." (Keller, et al., 2004).*
>
> *"Le processus qui prend en entrée une requête utilisateur et retourne une liste de ressources ou services pouvant combler éventuellement le besoin décrit." (Toma, et al., 2005).*

D'après les définitions précédentes, le principe de la découverte est simple : identifier les services qui peuvent répondre à une requête. Pour cela, il faut identifier un degré de similitude entre les différents concepts sémantiques qui

décrivent le service requis (la requête) et ceux des services offerts (les services publiés). Ce mécanisme est appelé : Appariement (Matching en Anglais).

Dans la littérature, les approches de découverte ont été classées selon plusieurs critères :

- **Architecture :** ce critère concerne la manière de stockage et de localisation des descriptions de services sur un réseau. On trouve soit des approches centralisées (basées sur un seul annuaire) ou des approches distribuées (basées sur la fédération d'annuaires).
- **Automatisation :** ce critère concerne le degré d'intervention humaine dans la découverte des services. Les approches sont classées en trois catégories : approches manuelles, approches semi-automatiques et approches automatisées.
- **Appariement :** ce critère concerne la méthode d'appariement (matching) à appliquer dans le calcul de la similarité au moment de la sélection des services. Les approches sont regroupées en trois classes : approches non logiques (dites aussi algébriques), approches logiques (dites aussi déductives) ou approches hybrides.

Dans notre travail de thèse, nous adoptons le classement selon le troisième critère : la méthode d'appariement utilisée. Nous présentons dans la suite de ce chapitre le principe et les travaux les plus connus de chacune de trois classes d'approches (non logique, logiques et hybrides). Une étude comparative des différentes approches de découverte de services web est présentée à la fin de ce chapitre.

4.2 Les approches non logiques

Ces approches sont aussi appelées des approches algébriques. Elles trouvent leurs origines dans les domaines de l'algèbre et la recherche d'information (RI). Le matching utilisé dans ces approches se base sur des mécanismes syntaxiques, structurels et numériques. Nous citons par exemple le calcul du degré de similarité de graphes structurés, le calcul des fréquences des termes, le calcul de distance entre concepts, etc.

Nous citons ci-après quelques approches de découverte non logiques, les plus citées dans la littérature, à savoir : AASDU, DSD-matchmaker et iMatcher1.

AASDU

L'approche AASDU (Agent Approach for Service Discovery and Utilization) (Palathingal, et al., 2004) est une approche multi-agents proposée pour la

découverte de services web, et elle est basée sur les standards des services web : UDDI, SOAP, WSDL et XML. La découverte des services dans cette approche est réalisée par des agents logiciels afin de mieux répondre aux requêtes des utilisateurs. L'approche AASDU est composée de quatre modules (**Figure 20**) :

1) **Une interface utilisateur graphique (GUI) :** cette interface graphique permet à l'utilisateur d'interagir avec le système. La requête est exprimée sous forme de chaine de caractère et envoyée au deuxième module.
2) **Un agent analyseur de requête (QAA) :** ce module s'occupe de l'analyse de la requête en extrayant les mots-clés pertinents qui seront utilisés par la suite dans le processus de découverte.
3) **Un système de référence multi-agents :** ce module permet de référencer les agents selon leur expertise. Chaque agent a juste connaissance des services relatifs à son domaine d'expertise, il n'est pas nécessaire qu'il ait connaissance de tous les services publiés dans les registres distribués. Dans cette approche, chaque agent a un profil déterminant ses intérêts et son expertise. L'expertise de l'agent est représentée par un vecteur de mots-clés tel que chaque mot clé représente un domaine donné. Pour chaque mot-clé, un score est assigné indiquant le degré d'expertise de l'agent dans ce domaine. De plus, chaque agent a une liste d'agents voisins (Neighbor list) qui indique les domaines d'expertise de ses voisins. Lorsqu'un agent vient rejoindre le système, un ensemble de voisins lui est assigné de façon aléatoire.
4) **Un module de services :** ce module permet aux fournisseurs de services de réaliser les trois fonctions suivantes : (i) la publication des descriptions de services, (ii) la négociation permettant la sélection de service et (iii) la composition dont le rôle est d'invoquer l'un des services issus de l'étape de sélection ou d'invoquer un service similaire lors de la défaillance du service sélectionné en premier. Un agent est assigné pour chacune des trois fonctions précédentes.

Rapport-Gratuit.com

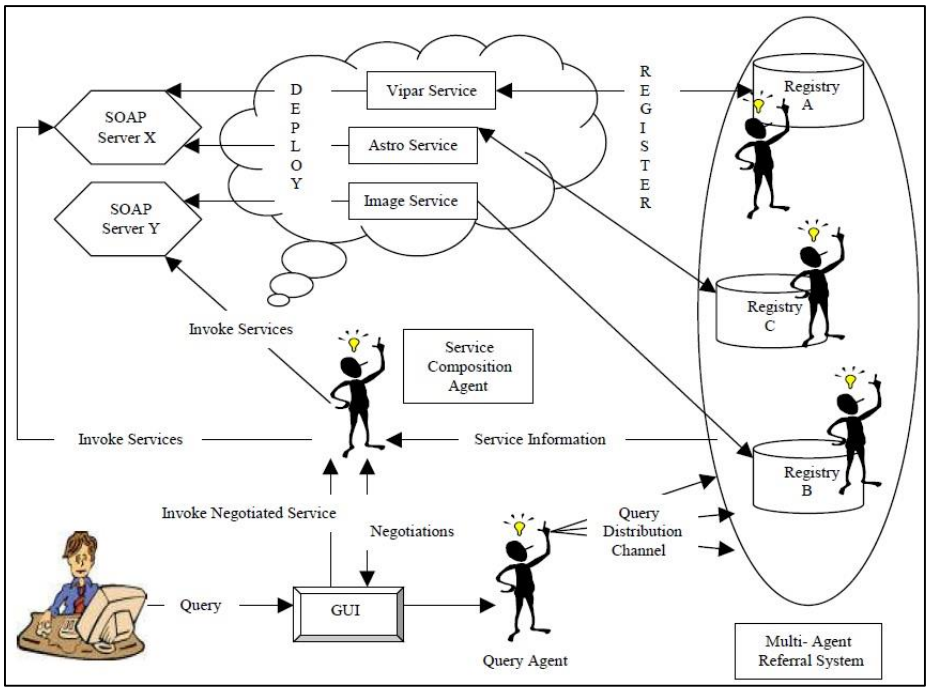

Figure 20 : Architecture de l'approche AASDU (Palathingal, et al., 2004).

Les quatre modules de l'approche AASDU s'enchainent afin d'identifier les services qui peuvent répondre à une requête. Tout d'abord, la requête est exprimée, via l'interface GUI, sous forme d'une chaîne de caractères. Elle est ensuite envoyée au deuxième module (QAA) qui réalise un appariement syntaxique fondé sur une simple variante de la technique TFIDF (Term Frenquency Inverse Document Frequency) pour indexer la requête et trier les résultats. Une table d'indexation contenant la liste des mots-clés dans un domaine particulier, et des vecteurs d'agents, issus du troisième module, contenant la valeur de chaque mot-clé dans la table d'indexation sont créés. L'analyse de la requête, dans le module QAA, commence par une racinisation (stemming en anglais) pour extraire les mots-clés pertinents. Un vecteur de requête est créé en se basant sur les termes dans la table d'indexation, où chaque scalaire du vecteur prend les valeurs basées sur le nombre d'occurrences d'un terme indexé dans le vecteur de la requête. Dans la phase suivante, chaque mot est comparé avec les vecteurs agents. La technique TFIDF est utilisée pour calculer la similarité du terme avec chaque vecteur agent. Si la similarité dépasse un seuil, l'agent est sélectionné en lui associant avec le terme recherché. Les agents sélectionnés transmettent ensuite les paramètres des services appartenant à leur domaine d'expertise, à un agent de composition. Ce dernier invoque l'un des services candidats selon le choix de l'utilisateur ou compose certains parmi eux pour répondre à sa requête.

DSD-matchmaker

L'approche DSD-matchmaker (Küster, et al., 2007a) (Küster, et al., 2007b) est une approche non logique basée sur un nouveau langage d'ontologie et un mécanisme de raisonnement spécialisés pour la découverte de service connu sous le nom *Diane Elements (DE)*. Le processus d'appariement dans cette approche est basé sur l'appariement de graphes entre les descriptions des deux services (service offert et service requête) spécifiées dans le langage DSD (DIANE Service Description) que nous avons présenté dans la **section 3.4** (page **55**) du chapitre précèdent. Il compare récursivement les nœuds de ces deux graphes et calcul un degré d'appariement entre ces deux nœuds. Le degré d'appariement de deux nœuds est une agrégation du degré d'appariement des deux types des concepts sémantiques représentés par les deux nœuds et du degré d'appariement des propriétés de ces deux concepts.

Le DSD-matchmaker procède en plusieurs étapes dans lesquelles il effectue des estimations sur les services offerts. D'abord, il détermine les valeurs concrètes des entrées (inputs) à utiliser pour l'exécution d'estimation. Cela permet également de recueillir des informations pour savoir si une estimation d'un service offert est prometteuse, c'est-à-dire si la valeur fournie par cette opération est utile pour décider à quel niveau ce service offert correspond à une requête donnée. Dans une deuxième étape, le DSD-matchmaker exécute seulement les estimations prometteuses et met à jour les descriptions des services offerts à l'aide d'informations collectées dynamiquement. Finalement, dans une troisième étape, l'appariement final est effectué sur la base des descriptions mises à jour.

iMatcher1

iMatcher1 (Klusch, 2008) est un système de découverte non logique qui interroge un ensemble de profils de services web décrits en OWL-S. Ces derniers sont stockés sous forme de graphes RDF sérialisés dans des une base de données RDF. Le processus d'appariement de ce système est basé sur le calcul de quatre métriques, tirées du domaine de la recherche d'information, de similarité syntaxique : *TFIDF* (Term Frequency-Inverse Document Frequency), la *distance de Levenshtein*, la *mesure cosinus* et la *mesure de divergence de Jensen-Shannon*. Les résultats sont classés en fonction des scores de ces mesures et d'un seuil défini par l'utilisateur.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

4.3 Les approches logiques

Ces approches sont aussi appelées des approches déductives, elles sont basées sur la logique. La description des services et des requêtes est spécifiée par des langages fondés sur des formalismes logiques, telles que la logique de premier ordre et la logique de description. Les approches logiques utilisent des concepts ontologiques et des règles logiques. Le matching utilisé dans ces approches se calcule de différentes façons en fonction de la sémantique des éléments des descriptions à apparier. Il existe principalement trois types d'appariement :

- **IO-matching :** Les éléments à apparier, dans ce type d'appariement, sont limités aux entrées (inputs) et aux sorties (outputs) des services et des requêtes.
- **PE-matching :** Les éléments à apparier, dans ce type d'appariement, sont limités aux préconditions (P) et aux effets (E) des services et des requêtes.
- **IOPE-matching :** Les éléments à apparier, dans ce type d'appariement, sont les inputs (I), les outputs (O), les pré-conditions (P) et les effets (E) des services et des requêtes.

Nous mentionnons ci-après quelques approches de découverte logiques, basées sur ces types d'appariement, à savoir : WSC, ALS, GR et PCEM.

WSC

L'approche WSC (Web Service Capabilities) (Paolucci, et al., 2002) est l'une des premières approches sémantiques de découverte de service. Dans cette approche, la description des services est basée sur l'ontologie DAML (DARPA Agent Markup Language) (Hendler, et al., 2000) et l'appariement adopté est de type IO-matching. Elle prend en compte que les entrées et les sorties du service durant le processus d'appariement. Un service répond parfaitement à une requête lorsque les deux conditions suivantes sont satisfaites : (1) toutes les sorties de la requête correspondent à toutes les sorties du service, et (2) toutes les entrées de la requête correspondent à toutes les entrées du service.

Le degré d'appariement entre deux entrées ou deux sorties dépend de la relation qui existe entre les concepts associés aux entrées et aux sorties. Les auteurs de cette approche proposent quatre degrés d'appariement : *EXACT*, *PLUG-IN*, *SUBSUMES* et *FAIL*. La **Figure 21** représente l'algorithme des règles d'affectation des degrés d'appariement des sorties. Soient **S** un service, **Q** une requête, **outS** et **outQ** leurs sorties respectives. L'appariement entre les sorties du service **S** et les sorties de la requête **Q** est égal à :

- *EXACT* lorsque elles sont équivalentes ou lorsque la sortie de Q est une sous classe de la sortie de S (lignes 2 et 3).
- *PLUG-IN* lorsque la sortie de S est un concept plus générique que la sortie de Q (ligne 4).
- *SUBSUMES* lorsque la sortie de Q est un concept plus générique que la sortie de S (ligne 5).
- *FAIL* lorsque aucune relation hiérarchique est identifiée entre les sorties du service S et la requête Q (ligne 6).

```
1   degreeOfMatch (outQ, outS)
2      if outS=outQ then return exact
3      if outQ subclassOf outS then return exact
4      if outS subsumes outQ then return plugIn
5      if outQ subsumes outS then return subsumes
6      otherwise fail
```

Figure 21 : Règles d'affectation des degrés d'appariement des sorties.

Par ailleurs, le processus de sélection des services web découverts dans WSC se base sur la sélection du service possédant le meilleur score au niveau des sorties. En cas où les scores d'appariement des sorties de ces services sont égaux, une comparaison de leurs entrées est effectuée pour sélectionner le meilleur. La **Figure 22** présente l'algorithme de tri des appariements de deux services (S_1 et S_2). Soient match1et match2 les scores d'appariement de chaque service avec la requête. L'algorithme sélectionne l'un des deux services sur la base des règles suivantes :

- Le premier service (S_1) est sélectionné si son score d'appariement de ses sorties est plus grand que le score d'appariement des sorties du deuxième service (ligne 2).
- Le premier service (S_1) est sélectionné si son score d'appariement de ses sorties est égal au score d'appariement des sorties du deuxième service et si son score d'appariement de ses entrées est plus grand que le score d'appariement des entrées du deuxième service (lignes 3 et 4).
- Les deux services (S_1 et S_2) sont sélectionnés s'ils ont le même score d'appariement de leurs sorties et le même score d'appariement de leurs entrées (lignes 5 et 6).

```
1   sortRule (match1, match2)
2      if match1.output > match2.output then match1 > match2
3      if match1.output = match2.output
4         &  match1.input > match2.input then match1 > match2
5      if match1.output = match2.output
6         &  match1.input = match2.input then match1 = match2
```

Figure 22 : Règles de tri des scores d'appariement.

ALS

L'approche ALS (Automatic Location of Services) (Keller, et al., 2005) propose un modèle conceptuel de localisation (découverte) sémantique de services. Ce modèle inclut : la réutilisation des objectifs prédéfinis (predefined Goals), la découverte de services abstraits pertinents et l'élaboration de contrats de services concrets pour répondre à l'objectif de la requête. Le type d'appariement adopté dans cette approche est de type IOPE-matching.

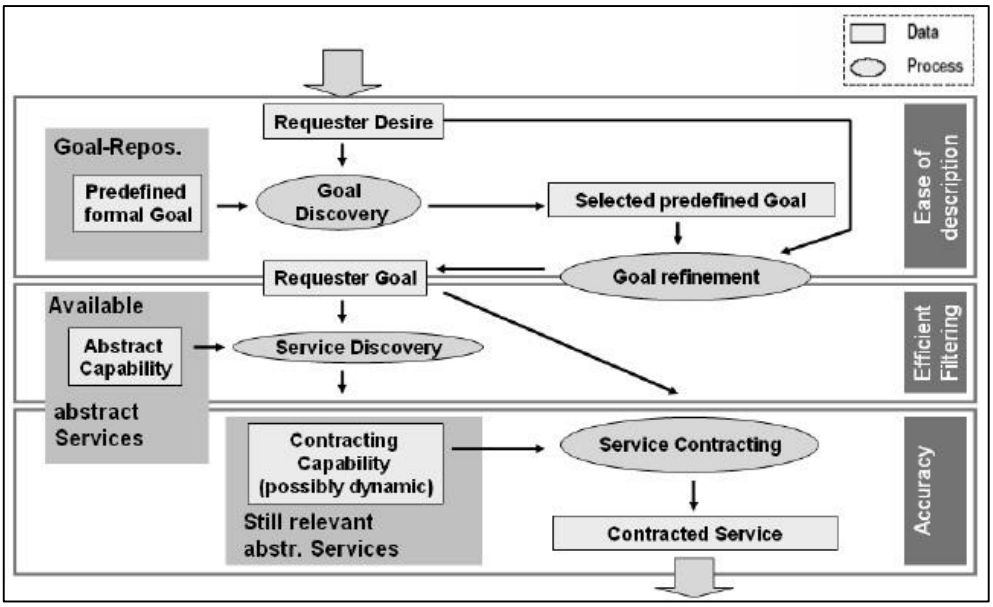

Figure 23 : Un modèle conceptuel pour le processus de découverte dans l'approche ALS (Keller, et al., 2005).

Le processus de découverte (**Figure 23**) de l'approche ALS est composé de quatre sous processus (phases) :

1) **Découverte de l'objectif (Goal Discovery) :** a pour but de localiser, parmi l'ensemble des objectifs prédéfinis, l'objectif prédéfini (predefined Goal) qui répond au besoin de l'utilisateur. L'objectif prédéfini sélectionné est une abstraction du besoin de l'utilisateur en objectif générique et réutilisable.
2) **Raffinement de l'objectif (Goal Refinement) :** l'objectif prédéfini sélectionné dans la première phase est raffiné afin de refléter au mieux le besoin de l'utilisateur. Le résultat de cette phase est un objectif formalisé.
3) **Découverte de service (Service Discovery) :** l'objectif sélectionné dans la phase précédente sera comparé avec les descriptions abstraites de services afin d'identifier les services candidats possibles.

4) **Contractualisation de service (Service Contracting) :** les services abstraits sélectionnés dans la phase précédente sont vérifiés s'ils sont capables de fournir des services concrets adéquats qui répondent à l'objectif de l'utilisateur.

Cette approche propose cinq degrés d'appariement appelés "degrés d'appariement intentionnel" permettant de comparer un service abstrait avec une requête. Ils sont définis comme suit :

- ***Match*** si le service satisfait complètement la requête.
- ***ParMatch*** si le service satisfait partiellement la requête. Des services abstraits additionnels seraient nécessaires pour satisfaire complétement la requête.
- ***PossMatch*** si le service peut satisfaire complètement la requête, tout en donnant une description plus détaillée de ce service au moment de la contractualisation.
- ***PossParMatch*** si le service peut satisfaire partiellement la requête, tout en donnant une description plus détaillée de ce service au moment de la contractualisation.
- ***NoMatch*** si aucun des cas précédents n'est identifié, pas de correspondance entre le service et la requête.

GR

L'approche GR (Graded Relevance) (Küster, et al., 2008) propose une extension de l'approche précédente (l'approche ALS), elle introduit une échelle de pertinence personnalisée pour le domaine d'appariement de services web sémantiques. Cette échelle de pertinence s'appuie sur les relations d'appariement de l'approche ALS et deux nouveaux degrés d'appariement proposés dans le cadre de cette nouvelle approche. Les deux nouveaux degrés d'appariement permettent de prendre en compte les services exclus par les relations précédentes de l'approche ALS, ils sont définis comme suit :

- ***RelationMatch*** signifie que le service offert ne peut pas satisfaire la requête directement mais peut fournir des fonctionnalités connexes. Un tel service peut alors être utilisé en combinaison avec d'autres. A titre d'exemple, un service proposant la réservation des chambres d'hôtels par le biais des codes d'hôtels peut être retenu par la fonction *RelationMatch* pour un utilisateur qui souhaite réserver une chambre en fournissant le nom des villes. Un service tiers peut alors produire un code d'hôtel sur la base d'un nom de ville.
- ***ExcessMatch*** signifie que le service offert est en mesure de satisfaire la requête mais son utilisation pourrait se traduire par des effets supplémentaires indésirables non demandés par le client. Cette fonction permet par exemple d'informer un client qui souhaite seulement acheter un téléphone par le biais

d'un service, que ce service lie l'achat de téléphones à un contrat téléphonique. Par exemple, un service qui permet à un client d'acheter seulement un téléphone, va lui proposer en plus un contrat téléphonique lié à son achat.

PCEM

L'approche PCEM (Pre-Conditions and Effects Matchmaker) (Botelho, et al., 2008) est une approche d'appariement de services web utilisée au sein du projet CASCOM (Intelligent Service Coordination in the Semantic Web) (Schumacher, et al., 2008). Le type d'appariement adopté dans cette approche est de type PE-matching, il compare les préconditions et les effets des services. Dans cette approche, les préconditions et les effets sont transformés en prédicats logiques en s'appuyant sur le langage PROLOG pour déterminer le degré d'appariement.

L'approche PCEM (**Figure 24**) est composée de trois modules :

1) **Moteur de composant (Component Engine) :** ce module contrôle les deux autres modules et détermine le degré d'appariement final entre la requête et chacun des services disponibles.
2) **Traitement de langage (Languages Processing) :** ce module s'occupe d'une part de la conversion des ontologies OWL vers des ontologies Prolog, et d'autre part de la conversion des préconditions et des effets en OWLS vers des préconditions et des effets en Prolog.
3) **Appariement des préconditions et des effets (Preconditions and Effects Matchmaking) :** étant donné que les préconditions et les effets sont en Prolog, le module bénéficie directement des avantages du langage en termes d'appariement et de raisonnement. Ce module effectue deux types d'appariement : un appariement exact et un appariement à base de raisonnement. Un appariement exact entre deux services est vérifié si les préconditions du premier service correspondent aux préconditions du deuxième service et si les effets du premier service correspondent aux effets du deuxième service. L'appariement à base de raisonnement est plus complexe que l'appariement exact, il utilise les règles d'inférence générales (règles de déduction) et les règles d'inférence spécifiques de domaine. Toutes les règles d'inférence générales sont applicables à toutes sortes d'effets et de préconditions. Les règles d'inférence spécifiques de domaine sont applicables seulement à certaines préconditions ou certains effets.

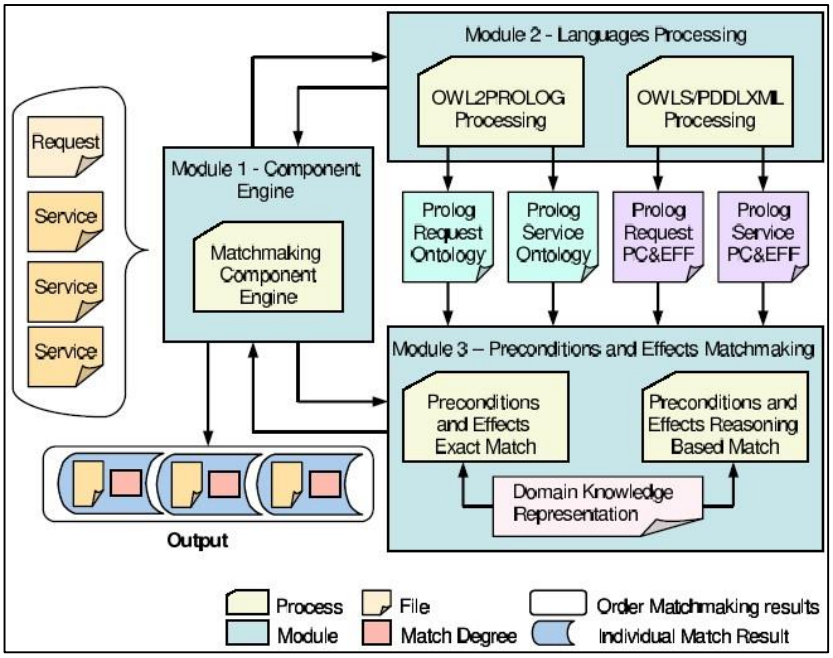

Figure 24 : Architecture de l'approche PCEM (Botelho, et al., 2008).

4.4 Les approche hybrides

Les approches hybrides sont des approches qui combinent les techniques logiques avec les techniques non logiques. L'idée de ces approches est de bénéficier des avantages de chacune des deux techniques toute en remédiant leurs limites. Plusieurs travaux adoptent des techniques hybrides, nous décrivons ci-après quelques-uns de ces travaux, les plus citées dans la littérature, à savoir : OWLS-MX, FC-Match, WSMO-MX et SAWSDL-MX.

OWLS-MX

L'approche OWLS-MX (Klusch, et al., 2006) est un matchmaker sémantique hybride, il combine le raisonnement logique et les techniques de recherche d'information pour réaliser un IO-matching entre des profils de services OWL-S.

OWLS-MX propose cinq degrés d'appariement: EXACT, PLUG-IN, SUBSUMES, SUBSUMED-BY and NEAREST-NEIGHBOR. Les trois premiers sont basés uniquement sur la logique, tandis que les deux derniers (subsumé par et le plus proche voisin) sont hybrides et comprennent des calculs supplémentaires pour mesurer la similarité syntaxique.

Dans OWLS-MX, les degrés d'appariement sont ordonnés en tenant compte de la tolérance d'appariement sémantique. Les auteurs précisent que plus les données en sorties offertes (services) sont génériques par rapport à celles requises

(requête), plus l'appariement sémantique est tolérant. L'ordre proposé est le suivant, allant du plus haut au plus bas degré :

Exact; Plug-In; Subsumes; Subsumed-By; Logic-based Fail; Nearest-neighbor; Fail.

La figure suivante (**Figure 25**) présente les définitions formelles des degrés d'appariement de l'approche OWLS-MX.

$LSC(C)$ the set of least specific concepts (direct children) C' of C, i.e. C' is immediate sub-concept of C $LGC(C)$ the set of least generic concepts (direct parents) C' of C, i.e., C' is immediate super-concept of C $Sim_{IR}(A, B) \in [0, 1]$ the numeric degree of syntactic similarity between strings A and B according to chosen IR metric $\alpha \in [0, 1]$ given syntactic similarity threshold $\doteq$ and $\dot{\geq}$ denote terminological concept equivalence and subsumption, respectively.	
Exact match	Service S EXACTLY matches request R $\Leftrightarrow \forall$ IN$_S$ $\exists$ IN$_R$: IN$_S \doteq$ IN$_R \wedge \forall$ OUT$_R$ $\exists$ OUT$_S$: OUT$_R \doteq$ OUT$_S$.
Plug-in match	Service S PLUGS INTO request R $\Leftrightarrow \forall$ IN$_S$ $\exists$ IN$_R$: IN$_S \dot{\geq}$ IN$_R \wedge \forall$ OUT$_R$ $\exists$ OUT$_S$: OUT$_S \in$ LSC(OUT$_R$).
Subsumes match	Request R SUBSUMES service S $\Leftrightarrow \forall$ IN$_S$ $\exists$ IN$_R$: IN$_S \dot{\geq}$ IN$_R \wedge \forall$ OUT$_R$ $\exists$ OUT$_S$: OUT$_R \dot{\geq}$ OUT$_S$.
Subsumed-by match	Request R is SUBSUMED BY service S $\Leftrightarrow \forall$ IN$_S$ $\exists$ IN$_R$: IN$_S \dot{\geq}$ IN$_R \wedge$ $\forall$ OUT$_R$ $\exists$ OUT$_S$: (OUT$_S \doteq$ OUT$_R \vee$ OUT$_S \in$ LGC(OUT$_R$)) $\wedge$ SIM$_{IR}$(S, R) $\geq \alpha$.
Logic-based fail	Service S fails to match with request R according to the above logic-based semantic filter criteria.
Nearest-neighbor match	Service S is NEAREST NEIGHBOR of request R $\Leftrightarrow \forall$ IN$_S$ $\exists$ IN$_R$: IN$_S \geq$ IN$_R \wedge \forall$ OUT$_R$ $\exists$ OUT$_S$: OUT$_R \dot{\geq}$ OUT$_S$ $\vee$ SIM$_{IR}$(S, R) $\geq \alpha$.
Fail	Service S does not match with request R according to any of the above filters.

Figure 25 : Les degrés d'appariement dans OWLS-MX (Klusch, et al., 2006).

En réponse à une requête, l'algorithme de matching retourne un ensemble ordonné de services jugés pertinents à qui sont associés un degré d'appariement et une valeur de similarité syntaxique par rapport à la requête. L'utilisateur spécifie le degré d'appariement souhaité et le seuil de la valeur de similarité syntaxique.

L'algorithme ne calcule pas uniquement l'appariement logique des entrées et des sorties mais il calcule en parallèle les similarités syntaxiques entre les concepts des entrées et des sorties. L'échec d'un appariement logique au niveau du raisonneur logique de l'approche est toléré si et seulement si la similarité syntaxique correspondante entre le concept du service en cours et de la requête reste supérieure au seuil de similarité souhaité.

Plusieurs variantes de l'approche OWLS-MX ont été implémentées : OWLS-M1, OWLS-M2, OWLS-M3 et OWLS-M4. Chacune d'entre elles utilise les mêmes degrés logiques mais des métriques de similarité syntaxique différentes (Loss-of-Information, Extended Jaccard, Cosine et Jensen-Shannon). Une autre variante OWLS-M0 ne réalise que de l'appariement sémantique logique.

FC-Match

L'approche FC-Match (Bianchini, et al., 2006) est une approche hybride d'appariement sémantique dont les concepts des services sont spécifiés en OWL-

DL (Web Ontology Language Description Logics). Les concepts utilisés pour décrire un service apportent des informations sur les éléments : *Category*, *Operation*, *Input* et *Output*.

Dans FC-Match, un service S est défini comme étant une conjonction logique d'expressions existentielles (avec le quantificateur ∃) de rôles. Chaque rôle correspond à un paramètre du profil sélectionné, le rôle est décrit par des concepts. Soit C_1 le concept associé au paramètre *catégorie* du profil de S, soit C_2 le concept associé au paramètre *opération*, soit C_3 le concept associé au paramètre *input*, soit C_4 le concept associé au paramètre *output*. Le concept global associé au profil du service S se note comme suit :

$$S = \exists hasCategory(C_1) \wedge \exists hasOperation(C_2) \wedge \exists hasInput(C_3) \wedge \exists hasOutput(C_4)$$

Cette approche opte également pour un appariement sémantique logique et un appariement algébrique fondé sur le calcul de similarité syntaxique. Le calcul de la valeur globale d'appariement dans FC-Match est effectué en agrégeant deux éléments :

- le résultat de l'appariement fondé sur la subsomption logique des concepts entre les deux profils du service et de la requête.
- la valeur du coefficient de similarité syntaxique entre les termes des concepts apparents dans les profils en respectant les relations terminologiques spécifiées dans le thesaurus WordNet (Fellbaum, 1998).

L'agrégation en appariement sémantique inclut les paramètres fonctionnels et non fonctionnels d'un profil OWL-S :

- les paramètres fonctionnels : les inputs et les outputs,
- les paramètres non-fonctionnels : à travers *hasCategory* et *hasOperation*.

WSMO-MX

L'approche WSMO-MX (Kaufer, et al., 2006) est une approche hybride d'appariement sémantique des services spécifiés en WSML (Web Service Modeling Language) (De Bruijn, et al., 2006) ; le langage utilisé pour décrire formellement tous les éléments WSMO. WSMO-MX applique, d'une manière similaire à OWLS-MX, plusieurs filtres pour la recherche des services WSMO qui sont sémantiquement pertinents pour une requête donnée contenant l'objectif à satisfaire. Dans WSMO, la requête d'un utilisateur recherchant un service est décrite par la structure *goal* (objectif). Un médiateur (structure Mediator) intervient durant le processus de découverte pour pré-lier les services web aux objectifs (*goals*).

Par ailleurs, l'appariement sémantique dans WSMO-MX porte essentiellement sur les préconditions et les postconditions des services et de la

Rapport-Gratuit.com

requête. WSMO-MX propose sept degrés d'appariement, présentés dans le **Tableau 4** qui illustre aussi leur ordre, leur symbole et leur signification par rapport aux états de préconditions et postconditions. Chaque degré d'appariement est calculé par l'agrégation des valeurs de : (1) l'appariement basé sur les ontologies, (2) l'appariement des relations et des contraintes logiques et (3) l'appariement syntaxique.

Soient ***G*** l'objectif à satisfaire de la requête (le service requis) et ***W*** le service WSMO (le service offert). Les degrés sont cités en ordre décroissant de leur valeur d'appariement, allant de l'équivalence à l'échec. Les degrés *Equivalence*, *Plugin*, *Inverse-plugin*, *Intersection* et *fail* sont d'ordre logique tandis que *Fuzzy similarity* et *Neutral* sont d'ordre non logique. L'équivalence est en haut des degrés, notée *Equivalence,* qui correspond à un appariement exact. Ensuite viennent les deux degrés *Plugin* et *Inverse-plugin* qui sont définis différemment par rapport à un état de précondition ou de postcondition. Si les préconditions de ***G*** impliquent les préconditions de ***W*** et si les postconditions de ***W*** impliquent les postconditions de ***G***, on parle de la relation *Plugin*. Sinon la relation est *Inverse-plugin*. Puis, on retrouve le degré *Intersection* qui a lieu quand ***G*** et ***W*** ont au moins un élément en commun, suivi du degré *Fuzzy similarity* (similarité floue) qui se réfère à un appariement sémantique non logique comme la similarité syntaxique. WSMO-MX identifie en avant dernier le degré *Neutral* (neutre) qui signifie pas d'appariement ni d'échec entre ***G*** et ***W***, et en dernier ordre le degré *fail* qui interprète un échec total de l'appariement.

Ordre	Symbole	Degré d'appariement	Pré	Post
1	$\equiv$	Equivalence	$G = W$	
2	$\sqsubseteq$	Plugin	$G \subseteq W$	$W \subseteq G$
3	$\sqsupseteq$	Inverse-plugin	$G \supseteq W$	$W \supseteq G$
4	$\sqcap$	Intersection	$G \cap W \neq \emptyset$	
5	$\sim$	Fuzzy similarity	$G \sim W$	
6	$\circ$	Neutral	By derivative specific definition	
7	$\perp$	Disjunction (fail)	$G \cap W = \emptyset$	

Tableau 4 : Les degrés d'appariement dans WSMO-MX (Kaufer, et al., 2006).

WSMO-MX sélectionne les états de précondition et de postcondition de chaque service offert à partir de sa base de connaissances, ensuite il les apparie un par un avec les états de la requête (service requis). Dans le cas où il n'existe pas de préconditions, le résultat d'appariement de cet élément est fixé à *Equivalent* par défaut. Dans ces cas de défauts (absences) d'annotations, WSMO-MX propose d'utiliser des stratégies spécifiques appelées "stratégies de défauts".

Après le traitement des résultats, l'algorithme d'appariement renvoie un vecteur de valeurs d'appariements agrégées, avec des annotations sur les résultats du processus d'appariement. Ces annotations sont considérées comme une sorte de retour explicatif à l'utilisateur. Ce retour essaie, en cas de résultats d'appariement insuffisants, de faciliter le raffinement itératif de l'objectif *Goal* de la part de l'utilisateur.

SAWSDL-MX

L'approche SAWSDL-MX (Klusch, et al., 2008) est une approche hybride d'appariement sémantique des services décrits en SAWSDL. Ce matchmaker est inspiré des deux autres matchmakers OWLS-MX et WSMO-MX. Le processus de découverte dans SAWSDL-MX est basé à la fois sur une approche d'appariement logique (raisonnement par subsomption) et sur une approche d'appariement syntaxique (techniques de recherche d'information).

Le SAWSDL-MX défini les degrés d'appariement logiques comme suit :

- **"Exact" :** On dit que le service S "Match exactly" la requête R, si les concepts entrées/sorties du service S sont équivalents des concepts entrées/sorties de la requête R.
- **"Plug-in" :** On dit que le service S "Plugs into" la requête R, si **(i)** les concepts d'entrées du service S sont plus génériques (en termes d'hiérarchie de concepts) des concepts d'entrées de la requête R et **(ii)** les concepts de sorties du service S sont des sous classes directes des concepts de sorties de la requête R.
- **"Subsumes" :** On dit que le service S "Subsumes" la requête R, si **(i)** les concepts d'entrées du service S sont plus génériques (en termes d'hiérarchie de concepts) des concepts d'entrées de la requête R et **(ii)** les concepts de sorties du service S plus spécifiques des concepts de sorties de la requête R.
- **"Subsumed-by" :** On dit que le service S "is Subsumed by" la requête R, si **(i)** les concepts d'entrées du service S sont plus génériques (en termes d'hiérarchie de concepts) des concepts d'entrées de la requête R et **(ii)** les concepts de sorties du service S sont des supers classes directes des concepts de sorties de la requête R**.**
- **"Fail" :** si aucun des degrés d'appariement précédents ne peut être déterminé.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

L'appariement adopté dans SAWSDL-MX couvre les éléments WSDL suivants, rappelons qu'une description SAWSDL est une extension de WSDL :

- L'élément *interface* pour des services décrits en WSDL 2.0 et l'élément *portType* pour des services décrit en WSDL 1.1,
- Les éléments *operation*, *input*, *output* pour des services décrits en WSDL 1.1 ou en WSDL 2.0.

Pour l'appariement des interfaces, le matchmaker effectue des appariements sur des graphes bipartis avec :

- des nœuds représentant les opérations,
- des arcs dont les valeurs sont calculées à partir des degrés d'appariement des opérations.

Pour l'appariement des opérations, le matchmaker se base sur :

- une approche logique qui utilise les annotations sémantiques spécifiées par les attributs *modelReference*,
- des approches syntaxiques qui utilisent à la fois les mêmes mesures de similarités adoptées dans OWLS-MX (Loss-of-Information, Extended Jaccard, Cosine et Jensen-Shannon) et d'autres mesures de similarité textuelles comme celles fournies par le projet *SimPack* (elles sont aussi utilisées dans le matchmaker iMatcher). Deux vecteurs de mots-clés pondérés sont générés pour chaque opération de service: un vecteur pour les entrées et un autre pour les sorties, ces vecteurs sont comparés avec les vecteurs de la requête en utilisant les mesures de similarité citées ci-dessus.
- des approches hybrides :
  - "Compensative": le matchmaker utilise la similarité syntaxique lorsqu'aucun degré d'appariement logique ne peut être associé à un service,
  - "Intégrative": pour remédier aux problèmes liés aux appariements faux positifs. Il ne faut pas uniquement prendre en compte la similarité syntaxique lorsqu'un appariement logique échoue, mais plutôt la considérer comme une contrainte conjonctive à chaque degré d'appariement logique.

Dans SAWSDL-MX, les degrés d'appariement peuvent être distingués en deux catégories :

- Les degrés d'appariement calculés par l'approche logique: *Exact*, *Plug-in*, *Subsumes* et *Subsumed-by*,
- Les degrés d'appariement calculés par l'approche hybride: *Hybride Subsumed-by* (il a besoin de calcul de similarité syntaxique additionnelle), *Nearest-neighbour* (pour compenser les appariements faux négatifs).

Pour illustrer le principe d'appariement dans SAWSDL-MX, nous considérons le service et la requête présentés dans la **Figure 26**. Chaque opération de la requête (RO_1, RO_2) est comparée avec chaque opération du service (O_1, O_2, O_3). Un appariement existe entre la première opération de la requête (RO_1) et la première opération du service (O_1) avec un degré *exact*. Un autre appariement existe entre la deuxième opération de la requête (RO_2) et la troisième opération du service (O_3) avec un degré *plug-in*. De ce fait, le résultat de l'appariement des opérations est : $\{<RO_1, O_1>, <RO_2, O_3>\}$. Pour déterminer le degré d'appariement global entre le service et la requête, une agrégation des degrés d'appariement des opérations doit être calculée. Le SAWSDL-MX dans sa première version (1.0) opte pour la plus petite valeur des degrés d'appariement pour garantir un minimum de similarité entre le service et la requête. Ceci est appelé le principe d'agrégation "Valeur minimale" (notée Min). Dans l'exemple, l'agrégation des deux degrés *exact* et *plug-in*, donne lieu au degré final d'appariement *plug-in* entre le service et la requête. D'autres possibilités par exemple est de fusionner les résultats d'appariement des opérations en se basant sur leurs moyennes de similarité syntaxique et de faire participer l'utilisateur dans cette opération d'appariement.

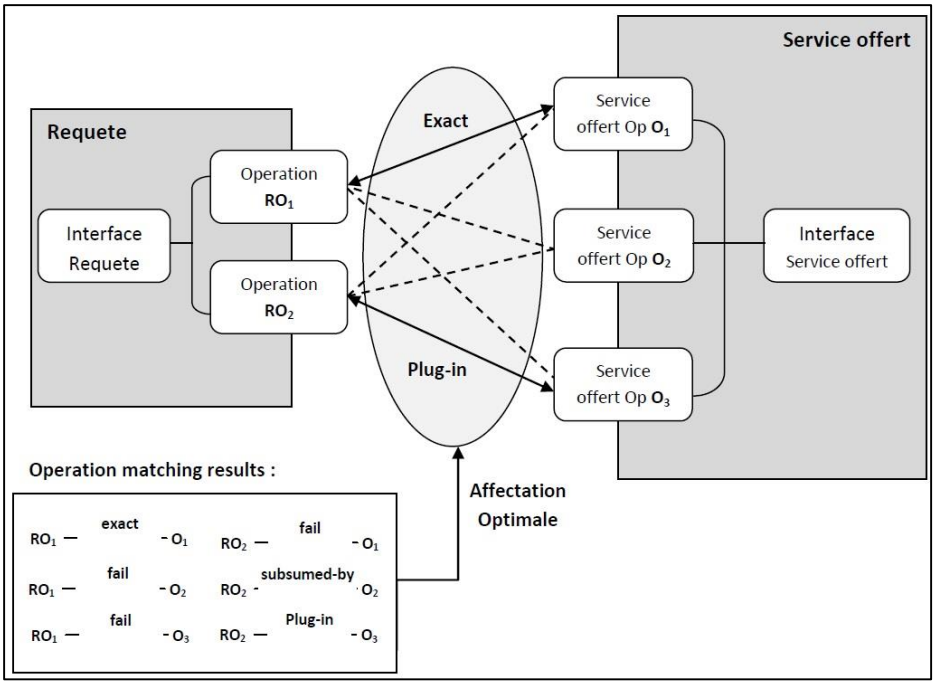

Figure 26 : Exemple d'appariement dans SAWSDL-MX. (Klusch, et al., 2008)

Il est important de noter que la première version de SAWSDL-MX (1.0) se concentre sur les annotations sémantiques de la signature d'opérations et ne prend pas en compte la structure arborescente du document WSDL. Notons aussi que la combinaison des filtres logiques et mesures textuelles de SAWSDL-MX 1.0 est

figée et n'est pas adaptative, en plus le choix du meilleur seuil de la similarité textuelle est fastidieux et dépend de la requête et des services disponibles. Toutes ces limitations ont poussé les auteurs à créer une nouvelle version : SAWSDL-MX 2.0 (Klusch, et al., 2009). Cette dernière applique trois types de similarité sémantique : logique, textuelle et structurelle. En particulier, la similarité structurelle est calculée par l'outil WSDL-Analyzer (Zinnikus, et al., 2006) en se basant sur la comparaison textuelle, lexicale et structurelle des arbres XML des fichiers WSDL.

La **Figure 27** montre un large aperçu de l'architecture globale de l'approche SAWSDL-MX. Principalement, il se compose de cinq éléments: SAWSDL Matching Engine, Service Registry, Ontology Handlers, Local Matchmaker Ontology et Similarity Measures. Du point de vue des fournisseurs de services, SAWSDL-MX permet l'enregistrement des services SAWSDL dans la base de registre. Pour les utilisateurs, SAWSDL-MX fournit une interface pour envoyer des requêtes au moyen d'un document SAWSDL précisant les détails du service souhaité. Après le processus de découverte, le moteur d'appariement de SAWSDL-MX (SAWSDL Matching Engine) retourne une liste triée des services qui répondent à la requête.

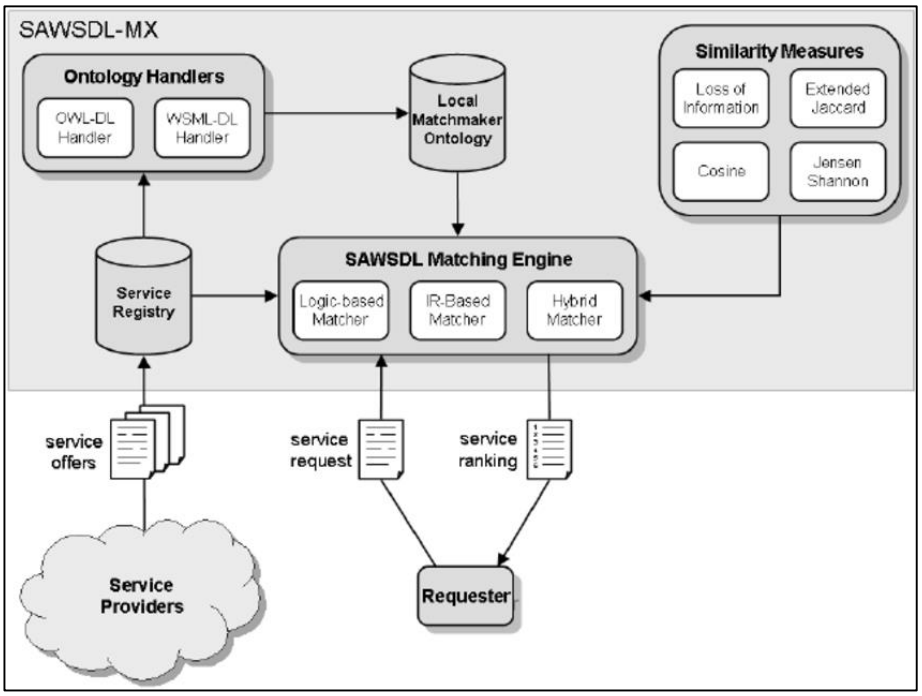

Figure 27 : L'architecture de SAWSDL-MX. (Klusch, et al., 2008)

Plusieurs variantes de l'approche SAWSDL-MX ont été implémentées : SAWSDL-M0, SAWSDL-M1, SAWSDL-M2, SAWSDL-M3 et SAWSDL-M4. La première variante SAWSDL-M0 ne réalise que de l'appariement sémantique

logique. Les quatre autres variantes, comme dans OWLS-MX, utilisent les mêmes degrés logiques cités auparavant mais des métriques de similarité syntaxique différentes (Loss-of-Information, Extended Jaccard, Cosine et Jensen-Shannon).

Le schéma présenté dans la **Figure 28** illustre et résume notre classification des approches de découverte de services web.

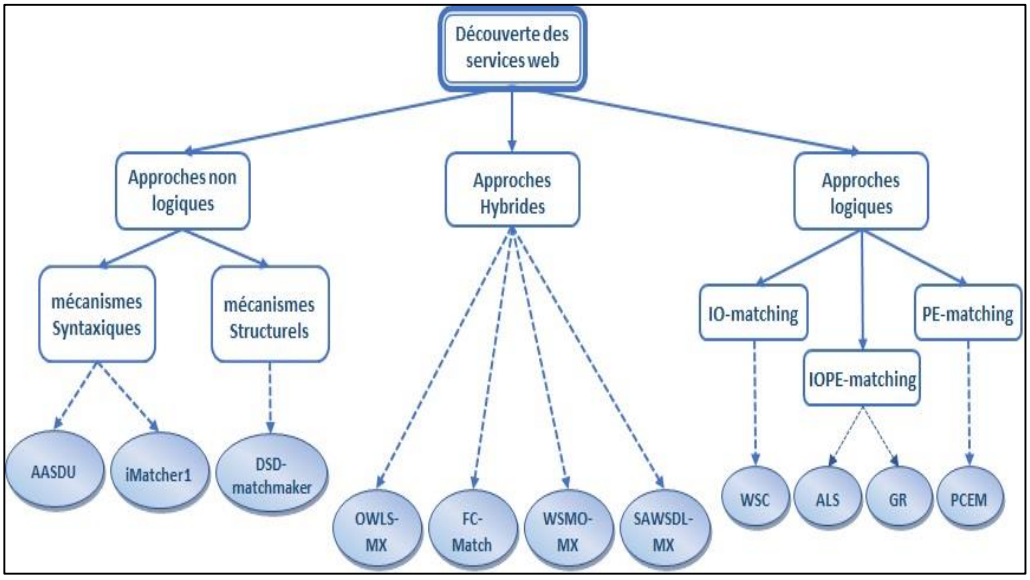

Figure 28 : Classification des approches de découverte de services web.

4.5 Etude comparative des approches de découverte

Dans cette section, nous présentons une étude comparative des différentes approches de découverte des services web étudiées dans ce chapitre. Nous avons classifié ces approches en trois grandes catégories : des approches non logiques, des approches logiques et des approches hybrides. Nous identifions cinq critères de comparaison afin d'évaluer les différentes approches de découverte des services web :

1. **Langage de description :** ce critère renseigne le langage et le formalisme utilisé pour représenter les services web. La plupart des approches utilisent des ontologies comme modèles de description sémantique des services web. Ce critère est primordial dans l'évaluation d'une approche. Une approche est dite fermée si elle dépend strictement d'un type particulier d'ontologie. Par contre, une approche est dite ouverte si elle ne dépend pas de types particuliers d'ontologies. Ceci offre plus de richesse dans la description et favorise la réutilisation des ontologies existantes.

2. **Type d'appariement :** ce critère s'intéresse à la technique utilisée lors du processus d'appariement. Ce processus peut être syntaxique, sémantique ou les deux à la fois (hybride). Une approche est dite profonde et intelligente si elle ne se limite pas au niveau syntaxique, elle doit aussi couvrir les aspects sémantiques.
3. **Elément d'appariement :** ce critère précise les éléments à apparier entre la description des services web et la requête. Cette information est importante parce qu'elle permet de mesurer la qualité de l'appariement fait tout en dépendant des éléments appariés. Une approche est dite pertinente si le nombre des éléments utiles à apparier est important. Toutefois, il faut souligner que ce n'est pas le grand nombre d'éléments appariés en lui-même qui importe, mais c'est plutôt le nombre d'éléments utiles qui fait la différence, vu l'information qu'ils peuvent apporter.
4. **Degré d'appariement :** ce critère définit le niveau de similitude entre les éléments appariés. Certaines approches considèrent qu'un élément s'apparie avec son correspondant seulement si le niveau de leur similitude est fort (exact). D'autres approches proposent une échelle de degrés de similitude (Plug-in, Subsumes, etc.). Les éléments à apparier ne doivent pas être les mêmes, mais par exemple l'un peut couvrir ou être une instance de l'autre. Une approche est dite flexible si elle propose, en cas d'absence de services web qui répondent exactement à une requête, d'autres services qui peuvent répondre partiellement à la requête. Toutefois, il faut souligner que ce n'est pas le nombre de degrés en lui-même qui importe, mais c'est plutôt le nombre de degrés significatifs qui fait la différence, vu la précision qu'ils peuvent apporter au processus d'appariement.
5. **Niveau d'automatisation :** ce critère mesure le niveau d'intervention humaine dans le processus de découverte. Une approche de découverte peut être totalement automatique (pas d'intervention humaine) ou semi-automatique (avec l'intervention humaine). Pour une approche, l'intervention humaine dans son processus de découverte peut présenter sa limite. Ainsi, pour que l'interprétation humaine soit pertinente, il faut un bon niveau d'expertise. Une approche semi-automatique peut couter plus chère, en termes de temps et d'argents, qu'une approche totalement automatique bien élaborée.

Dans la suite, nous présentons notre étude des approches de découverte des services web par rapport aux cinq critères identifiés ci-dessus. Le **Tableau 5** présente un récapitulatif de notre étude de comparaison. Ce tableau comparatif permet en fait de déterminer jusqu'à quel point chacune de ces approches serait

en mesure d'être qualifiée d'ouverte, de profonde, de pertinente, de flexible et d'automatique, et ce, en évaluant ses caractéristiques.

Langage de description

Comme il est indiqué dans le tableau comparatif au niveau de la deuxième colonne, on note que la plupart des approches utilisent des ontologies comme modèles de description sémantique des services web. On cite par exemple les approches fermées qui dépendent d'un type particulier d'ontologies comme : l'approche WSC qui dépend de l'ontologie DAML, les approches iMatcher1, OWLS-MX et FC-Match qui dépendent de OWL-S et l'approche WSMO-MX qui dépend de WSMO. Tandis que l'approche SAWSDL-MX, décrite en WSDL, ne dépend d'aucun type d'ontologie. Elle est qualifiée d'approche ouverte parce que les annotations sémantique dans cette approche peuvent provenir de différents types d'ontologies.

Type d'appariement

Comme il est indiqué dans le tableau comparatif au niveau de la troisième colonne, on note que la plupart des approches ont un appariement de type syntaxique, sémantique ou les deux. D'une façon générale, nous remarquons que les approches non logiques restent fortement liées aux techniques de recherche d'information (par exemple, AASDU et iMatcher1) et d'appariement de graphes (par exemple, DSD-matchmaker). Par ailleurs, les approches logiques utilisent un appariement sémantique soit de type IO-matching (l'approche WSC), soit de type IOPE-matching (ALS et GR) ou de type PE-matching (PCEM). Tandis que les approches hybrides utilisent des mécanismes syntaxiques et sémantiques pour remédier aux limites des approches précédentes.

Elément d'appariement

Comme il est indiqué dans le tableau comparatif au niveau de la quatrième colonne, on note que l'appariement dans certaines approches (WSC, OWLS-MX) se base sur les inputs/outputs. D'autres approches jugent que les inputs/outputs ne sont pas suffisants pour apparier deux descriptions de services. Dans ce sens, nous citons les approches ALS et GR qui ont proposé d'ajouter un appariement au niveau de services lui-même (Goals). Nous citons aussi les approches FC-Match et SAWSDL-MX qui proposent de rajouter des éléments, en plus des inputs/outputs, à leurs processus d'appariement : *Category* & *Operation* pour l'approche FC-Match et *Interface* & *Operation* pour l'approche SAWSDL-MX.

Degré d'appariement

Comme il est indiqué dans le tableau comparatif au niveau de la cinquième colonne, on note l'absence d'une échelle de degrés d'appariement (comme dans AASDU, DSD-matchmaker, iMatcher1, PCEM et FC-Match) ou une échelle incomplète et insuffisante pour décortiquer les différents cas d'appariement (comme dans l'approche WSC). Par contre, d'autres approches (ALS, GR, OWLS-MX, WSMO-MX et SAWSDL-MX) proposent des degrés d'appariement plus précis pour prendre en compte le plus possible de relations sémantiques.

Niveau d'automatisation

Le niveau d'automatisation est spécifié au niveau de la dernière colonne du tableau comparatif. Bien que l'intervention humaine dans un processus d'appariement semi-automatique soit considérée souvent comme une limite surtout quand elle coûte cher en termes de temps, elle est exigée dans les approches DSD-matchmaker, iMatcher1, OWLS-MX, FC-Match, WSMO-MX et SAWSDL-MX. Dans ces approches, une des deux interventions humaines est requise :

- Le choix du seuil à partir duquel un appariement est jugé réussi,
- La sélection du service jugé comme le plus proche de la requête. La sélection est précédée d'une vérification des résultats de la découverte.

D'autres approches (AASDU, WSC, ALS, GR et PCEM) optent pour un appariement automatique afin d'éviter l'intervention de l'utilisateur et accélérer le processus de découverte.

	Critères / Approches	Langage de description	Type d'appariement	Elément d'appariement	Degré d'appariement	Niveau d'automatisation
Approches non logiques	**AASDU**	WSDL	Syntaxique TFIDF	Agents logiciels	Pas d'échelle de degrés	Automatique
	DSD-matchmaker	DSD	Syntaxique Appariement de graphes	Nœuds de graphes	Pas d'échelle de degrés	Semi-automatique Sélection des services
	iMatcher1	OWL-S	Syntaxique	Profil du service	Pas d'échelle de degrés	Semi-automatique Choix du seuil Sélection des services
Approches logiques	**WSC**	DAML	Sémantique IO-matching	Input & Output	Echelle : Exact, Plug-In, Subsumes et Fail	Automatique
	ALS	Logique (Goals + Axiomes)	Sémantique IOPE-matching	Input, Output et Goals	Echelle : Match, PossMatch, ParMatch, PossParMatch et NoMatch	Automatique
	GR	Logique (Goals + Axiomes)	Sémantique IOPE-matching	Input, Output et Goals	Echelle : Match, PossMatch, ParMatch, PossParMatch, Relation-Match, Excess-Match et NoMatch	Automatique
	PCEM	Logique (Prolog)	Sémantique PE-matching	Préconditions et Effets	Pas d'échelle de degrés	Automatique
Approches hybrides	**OWLS-MX**	OWL-S	Syntaxique & Sémantique (IO-matching)	Input & Output	Echelle : Exact, Plug-In, Subsumes, Subsumed-By, Logic-based Fail, Nearest-neighbor et Fail.	Semi-automatique Choix du seuil Sélection des services
	FC-Match	OWL-S & OWL-DL	Syntaxique & Sémantique	Category, Operation, Input et Output	Pas d'échelle de degrés	Semi-automatique Choix du seuil Sélection des services
	WSMO-MX	WSML	Syntaxique & Sémantique	Goals, Préconditions et Postconditions	Echelle : Equivalence, Plugin, Inverse-plugin, Intersection, Fuzzy similarity, Neutral et Fail	Semi-automatique Sélection des services
	SAWSDL-MX	WSDL étendu	Syntaxique & Sémantique	Interface (PortType), Input, Output et Operation	Echelle : Exact, Plug-in, Subsumes, Subsumed-by, Nearest-neighbour et Fail	Semi-automatique Choix du seuil

Tableau 5 : Récapitulatif de l'étude comparative des approches de découverte.

Support-Gratuit.com

4.6 Conclusion

Dans ce chapitre, nous avons présenté les approches les plus utilisées dans la littérature de la découverte et de l'appariement de services web. Nous avons groupé ces approches en trois grandes classes : non logique, logique et hybride. Les approches hybrides combinent les techniques des approches non logiques et logiques tout en bénéficiant de leurs avantages et en remédiant de leurs limites. Nous avons terminé ce chapitre par une étude comparative des différentes approches de découverte et d'appariement de services web.

Nous remarquons que les approches d'appariement les plus citées et les plus élaborées sont celles basées sur la logique (pure ou hybride) tels que SAWSDL-MX, OWLS-MX et WSMO-MX. Ces dernières approches utilisent les langages de description de services les plus connus et répondus, à savoir SAWSDL, OWL-S et WSMO. Un intérêt particulier pour SAWSDL que nous avons dégagé dans le chapitre de description se confirme vu que l'approche de SAWSDL-MX se base et combine les deux autres approches WSMO-MX et OWLS-MX.

Dans les chapitres suivants, nous allons décrire notre approche qui utilise les technologies du web sémantique et se base sur les approches de description et d'appariement pour offrir une solution permettant la recherche de contenus multimédias. Notre nouvelle approche doit, à la fois, tirer profit des avantages de ses approches et combler leurs lacunes.

DEUXIEME PARTIE

CONTRIBUTIONS

Chapitre 5 *L'approche SeSaM*

Chapitre 6 *Le langage SA4MaaS*

Chapitre 7 *Le matchmaker MaaS-MX*

Chapitre 8 *Prototype et évaluation*

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

CHAPITRE 5

L'APPROCHE SESAM

5.1 Introduction

5.2 Rappel de la problématique

5.3 L'approche SeSaM

5.4 Présentation des principales notions de l'approche SeSaM

5.4.1 Les ontologies

L'ontologie métier

L'ontologie multimédia

5.4.2 Les services web multimédias

Motivations

Les services MaaS

5.4.3 La requête utilisateur

5.5 Conclusion

5.1 Introduction

Dans les systèmes d'intégration de données à base de services web sémantiques, les sources de données sont accessibles par des services web de type DaaS (Data as a Service). Ces derniers services ne sont pas adaptés aux sources de données multimédias sur le web. Ces sources de données multimédias distribuées posent de nouveaux problèmes à cause de leurs volumétries, leurs diversités des interfaces, leurs formats de représentation, leurs localisations, etc. En plus, le besoin d'intégrer la sémantique dans les systèmes de recherches d'information multimédia devient de plus en plus fort.

Dans ce chapitre nous présentons l'approche **SeSaM** (**Se**mantic **Sea**rch of **M**ultimedia content) qui est une approche de recherche d'information multimédia basée sur les services web. Cette approche repose sur la définition d'un nouveau type de services permettant l'accès à des sources de données multimédia, que nous avons baptisé services MaaS (Multimedia as a Service). SeSaM est fondée sur un processus en deux étapes : description et appariement des services MaaS pour répondre aux requêtes de l'utilisateur dans le contexte de systèmes de données distribués et hétérogènes.

Ce chapitre est organisé comme suit. Tout d'abord, dans la section 5.2, nous rappelons la problématique et les objectifs de la thèse en nous appuyant sur un scénario exemple. Ensuite, dans la section 5.3, nous présentons une vue globale de l'approche SeSaM. La section 5.4 est consacrée à la définition des principales notions de cette approche. Un résumé de l'approche est rappelé en conclusion de ce chapitre (section 5.5).

5.2 Rappel de la problématique

Avant de présenter notre approche, nous allons rappeler la problématique de la thèse à l'aide d'un scénario exemple. Nous avons choisi le domaine médical comme domaine d'application, il permet d'illustrer concrètement les challenges et les objectifs de la thèse. Cependant, il est important de noter que les perspectives d'application de notre travail sont beaucoup plus vastes et concernent tous les domaines dans lesquels l'utilisation (l'implication) des services web multimédias peut être envisagée.

Nous avons étendu l'exemple de motivation présenté dans l'introduction de ce mémoire (section **1.1**, page **16**). Un étudiant en médecine, en l'occurrence l'étudiant Bob voudrait approfondir ses notions sur les maladies cancéreuses et

plus spécifiquement le cancer du poumon, il cherche exactement à récupérer tout type d'information sur différentes méthodes de diagnostic. Il veut avoir comme résultat toutes les vidéos fournies par «le centre de cancérologie» au format «mpeg». Supposons qu'il a à sa disposition les services web présentés dans le **Tableau 6**. Les services web disponibles permettent d'accéder à des ressources de données de nature différente réparties sur trois sites : le centre d'imagerie médical, le centre de cancérologie et la faculté de médecine. Les deux premiers sites matérialisent l'expérience professionnelle et pratique, tandis que le troisième site matérialise l'expérience pédagogique et théorique. Les services du premier site, le centre d'imagerie médical, retournent des rapports et des résultats d'actes d'imagerie (radios, IRM, etc.). Les services du deuxième site, le centre de cancérologie, retournent des rapports et résultats des différents actes médicaux pratiqués sur des sujets sains et malades (analyses biologiques et génétiques, diagnostiques, traitements, etc.). Les services du troisième site, la faculté de médecine, retournent des documents pédagogiques variés (cours, expériences, conférences, articles scientifiques, etc.).

Bob peut utiliser les services web du **Tableau 6** pour répondre à ses besoins. Une première lecture des fonctionnalités des services disponibles montre que seulement trois services S_2, S_3 et S_9 sont plus proches de sa demande, tandis que les autres services n'ont pas la même sémantique en terme de fonctionnalités de ces trois services (par exemple le service S_1 retourne des images sur les différents traitements du cancer du poumon). Les trois services (S_2, S_3 et S_9) retournent les différentes méthodes de diagnostic de cancer du poumon, sachant d'une part que le scanner thoracique est un type d'examen imagerie qui permet de diagnostiquer un patient (service S_2) et d'autre part que les cancers bronchiques non à petites cellules sont des types de cancer de poumon (service S_9). Ces trois services ont une sémantique similaire en termes de fonctionnalités mais ils retournent différents types et formats de fichiers avec des fournisseurs différents.

D'une manière intuitive Bob procède de la manière suivante pour répondre à ses besoins. Dans un premier temps, il compare les fonctionnalités de chaque service avec les fonctionnalités de la requête. Donc, il élimine les services S_1, S_4, S_5, S_6, S_7, et S_8 et il garde les services S_2, S_3 et S_9 parce que leurs sémantiques correspondent à la sémantique de sa requête. Dans un deuxième temps, il conserve uniquement le service S_2 et S_9 car leurs propriétés multimédias (type, format et fournisseur des fichiers) correspondent parfaitement aux propriétés multimédias de sa requête.

Services	Fonctionnalités	Fournisseur	Format
S_1	Retourne des images visualisant les différents traitements possibles du cancer de poumon	Centre d'imagerie médical	Jpeg
S_2	Retourne des vidéos qui montrent les différentes méthodes de diagnostic du cancer de poumon	Centre de cancérologie	Mpeg
S_3	Retourne des fichiers audios discutant les résultats d'un scanner thoracique d'un patient atteint d'un cancer de poumon	Faculté de médecine	Wav
S_4	Retourne des vidéos qui expliquent les différents symptômes du cancer de poumon	Centre de cancérologie	Mpeg
S_5	Retourne des images visualisant les différentes causes du cancer de poumon	Centre d'imagerie médical	Jpeg
S_6	Retourne des fichiers audios des cours sur les traitements possibles du cancer de la prostate	Faculté de médecine	mp4
S_7	Retourne des images illustrant les différents traitements possibles pour les cancers bronchiques non à petites cellules	Centre d'imagerie médical	Png
S_8	Retourne des vidéos de cours informant la liste des tests et examens à faire en cas d'apparition des symptômes du cancer du poumon	Faculté de médecine	mp4
S_9	Retourne des vidéos montrant les différents examens à réaliser pour les cancers bronchiques non à petites cellules	Centre de cancérologie	mpeg

Tableau 6 : Les services MaaS du scénario exemple.

A l'aide de cet exemple (**Figure 29**), nous illustrons les principaux challenges qui doivent être traités dans cette thèse. Nous résumons ces challenges par les quatre questions suivantes :

1. **Quelles sont les étapes nécessaires à suivre pour répondre à la requête de Bob ?**

Notre premier challenge dans cette thèse est de connaitre les étapes nécessaires à suivre pour répondre à la requête de Bob dans le contexte d'un système hétérogène et distribué contenant des données multimédia. La réponse à cette question est donnée dans la suite de ce chapitre par la proposition de l'approche SeSaM (**Se**mantic **Sea**rch of **M**ultimedia content).

2. **Que sont ces services d'accès aux données multimédia ?**

Le noyau de notre travail de thèse se fonde sur la définition d'un nouveau type de services accédant aux données multimédias. Le challenge ici est comment

définir ces services d'accès aux données multimédias. En quoi sont-ils différents des services d'accès aux données traditionnels, les services DaaS (Data as a Service). La réponse à ce challenge est donnée dans la suite de ce chapitre par la définition d'un nouveau type de services web, que nous appelons "services MaaS" (Multimedia as a Services).

3. Comment représenter les services MaaS avec leurs différentes vues sémantiques possibles ?

Après la définition de ce nouveau type de services, nous nous intéressons à la manière de le décrire et le représenter avec ses différentes vues sémantiques. De nombreux services peuvent avoir la même sémantique en termes de fonctionnalités, mais complètement différentes en termes de propriétés multimédias. Par exemple, les services S_2 et S_3 ont la même sémantique pour les fonctionnalités, ils ont la même entrée (cancer de poumon) et la même sortie (méthode de diagnostique), mais ils sont différents sur le plan des propriétés multimédias (type, fournisseur et format). Le challenge ici est de savoir comment représenter et décrire les services multimédia pour renforcer l'expressivité et la précision de l'approche proposée. La réponse à ce challenge est donnée dans le **chapitre 6** par une proposition de langage de description des services MaaS, que nous appelons le langage **SA4MaaS**.

4. Comment apparier les services MaaS avec la requête de Bob ?

La prochaine étape, après la définition des services MaaS et du langage SA4MaaS, est d'identifier les services pertinents répondant à la requête de Bob. Un tel mécanisme est appelé appariement (matching en anglais). Le challenge ici est de montrer comment apparier les services MaaS avec la requête de Bob prenant en compte leurs descriptions enrichies en utilisant le langage SA4MaaS. Bob devrait se rendre compte que les deux services S_2 et S_9 satisfont ses besoins fonctionnels et multimédias en même temps. Pour répondre à ce challenge, nous avons proposé dans le **chapitre 7** le matchmaker **MaaS-MX**.

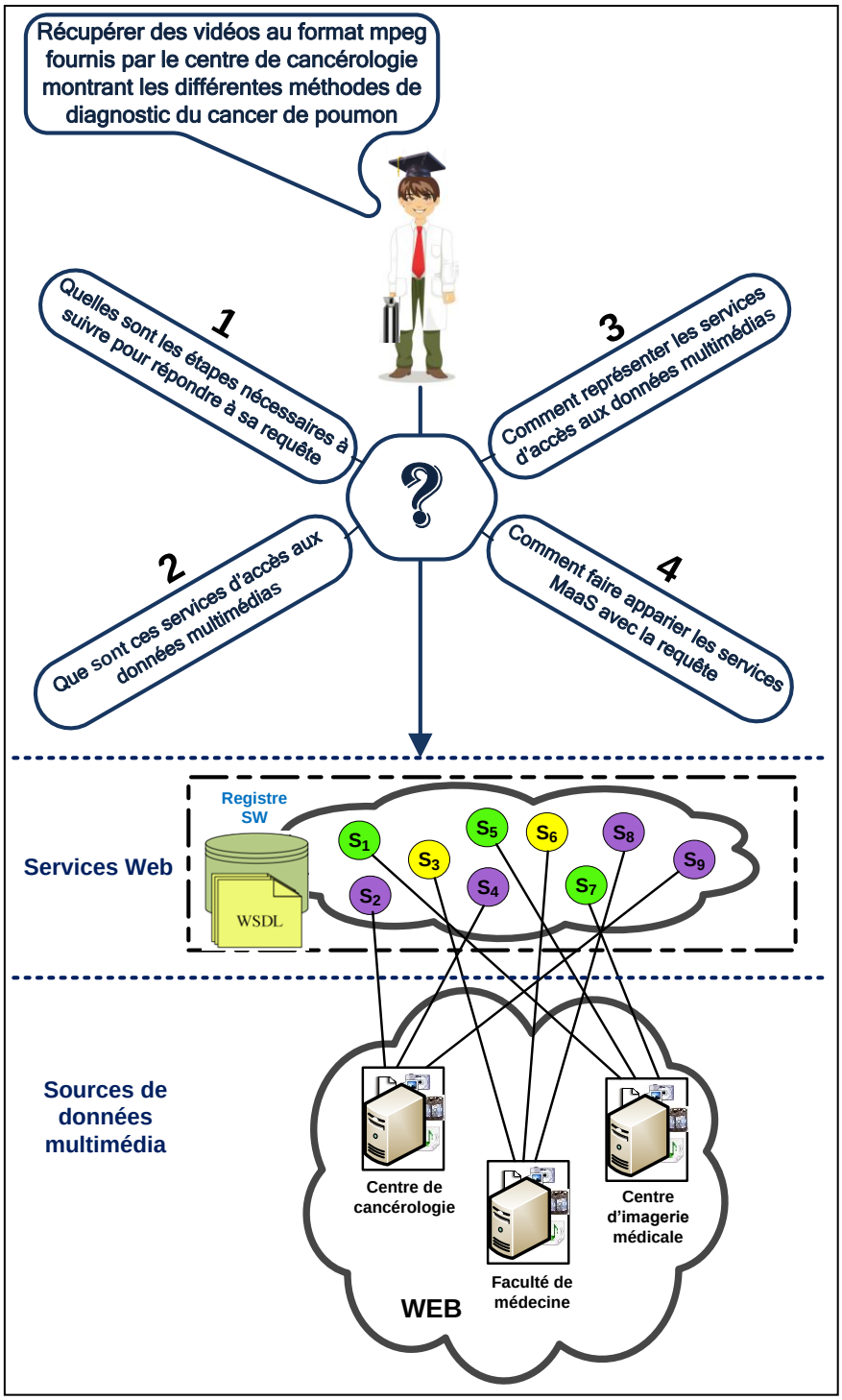

Figure 29 : Scénario de motivation.

5.3 L'approche SeSaM

Nous présentons dans cette section une vue générale de notre approche SeSaM (**Se**mantic **Sea**rch of **M**ultimedia content) qui permet de rechercher des contenus multimédias en se basant sur une architecture orientée services web. Le fonctionnement de notre approche repose sur un processus en deux étapes : description, et appariement. Dans cette approche full services les sources de données sont interrogées par un ensemble de services web dont le but est de trouver une solution (une combinaison de services web MaaS) qui répond au mieux à la requête de l'utilisateur. La **Figure 30** présente notre approche des services MaaS.

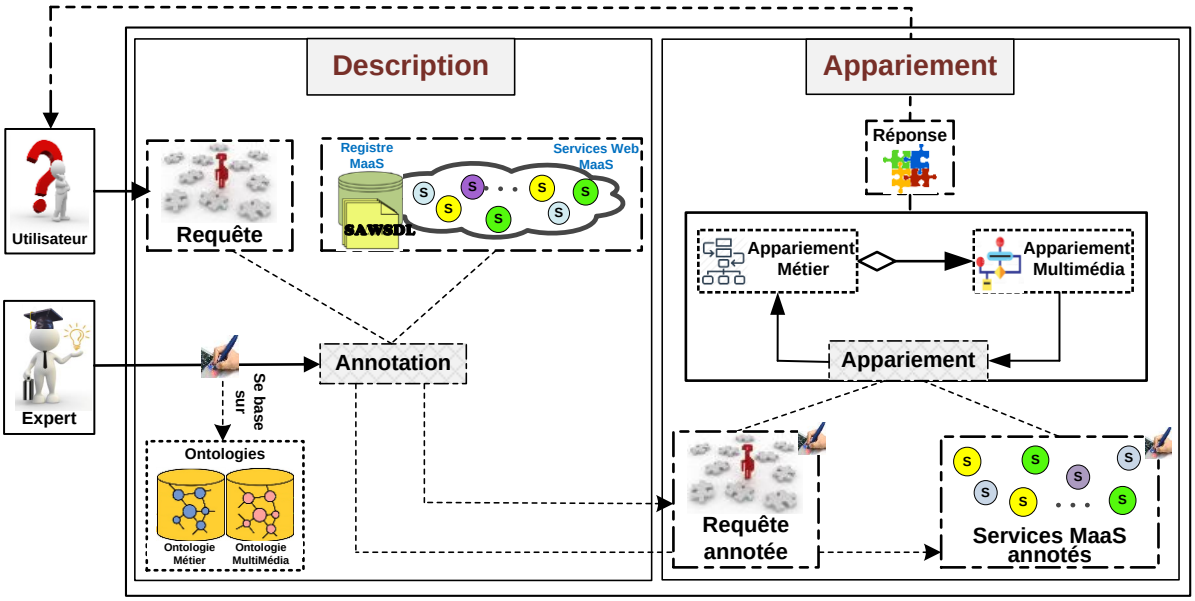

Figure 30 : Vue globale de l'approche SeSaM.

Les deux étapes de notre approche sont :

1) **Etape de description,** le but de cette étape est de décrire sémantiquement les services MaaS accédant aux ressources multimédias disponibles sur le web. Pour la description de nos services MaaS, nous nous sommes intéressés à étendre les approches de description de services web sémantique, plus particulièrement le langage SAWSDL.

   Le principe général de notre extension est l'intégration de la sémantique des aspects multimédias dans la description des services MaaS. Par conséquent, nous avons proposé d'utiliser deux types ontologies de

nature différente dans la tâche de description. Une première ontologie dite « ontologie métier » contenant des concepts définissant la sémantique métier des services MaaS. Une deuxième ontologie dite « ontologie multimédia » contenant des concepts définissant la sémantique des annotations multimédias des services MaaS. La requête de l'utilisateur est décrite en se basant sur les mêmes ontologies (métier et multimédia) utilisées dans la description des services MaaS. Le **chapitre 6** présente en détail notre proposition de description de services MaaS.

2) **Etape d'appariement,** consiste à identifier les services MaaS pertinents afin de répondre à la requête de l'utilisateur. Cet affinement est fait en appliquant notre approche d'appariement (défini dans le **chapitre 7**) entre la requête et les services MaaS candidats.

   Dans le cadre de notre travail, nous supposons que les services MaaS ainsi que la requête sont annotés en utilisant les mêmes ontologies (métier et multimédia). C'est-à-dire qu'il n'y a pas d'appariement si les concepts de la requête et ceux des services MaaS proviennent d'ontologies différentes. Cette problématique, liée au domaine des ontologies, dépasse le domaine des services web et ne sera pas traitée dans le cadre de cette thèse. Nous partons de l'hypothèse que l'appariement se fait entre deux concepts d'une même ontologie. Ceci n'empêche pas que la description des services MaaS puisse utiliser plusieurs ontologies.

   Les concepts d'entrée et de sortie des services MaaS et de la requête sont extraits de leurs fichiers SAWSDL. Notre mécanisme de matching se déroule en deux étapes. La première étape dite « appariement métier » consiste à comparer la description métier des services MaaS candidats et de la requête. La deuxième étape, qui est exécutée si seulement si l'étape d'appariement métier a réussi, est dite « appariement multimédia » consiste à comparer les descriptions multimédias des services MaaS et de la requête. Le **chapitre 7** présente en détail notre proposition d'appariement de services MaaS.

Rapport-Gratuit.com

5.4 Présentation des principales notions de l'approche SeSaM

Nous présentons dans cette section les principales entités de l'approche SeSaM, à savoir : les ontologies, les services web multimédias et la requête.

5.4.1 Les ontologies

Les ontologies sont communes aux fonctions de description et de recherche. Le rôle de ces ontologies est de disposer d'un vocabulaire commun pour spécifier les descripteurs des services MaaS et de la requête utilisateur. Ces ontologies définissent une terminologie réutilisable et partageable par ceux qui fournissent les services et par ceux qui les recherchent.

Formellement, nous avons repris la définition proposée par (Maedche, et al., 2002) qui définit une ontologie par un 6-tuple :

$$\mathcal{O} = (C, P, A, H^C, prop, att) \text{ où}$$

- C est un ensemble de concepts,
- P est un ensemble de relations,
- A est un ensemble d'attributs,
- H^C est une relation dans $(C \times C)$, représentant une hiérarchie des concepts. Par exemple, soit $c_1, c_2 \in C$, $c_1 H^C c_2$ signifie que c_1 est un sous concept de c_2,
- *prop* est une fonction qui relie un concept avec un autre concept, chaque fonction *prop* a son domaine et co-domaine (range) dans C.
  Par exemple, soit $c_1, c_2 \in C$, $prop(P)=(c_1,c_2)$ (notée aussi $c_1 P c_2$) signifie que c_1 est relié à c_2 par P,
- *att* est une fonction qui relie un concept avec un attribut, chaque fonction *att* a un domaine dans C et un co-domaine dans A.

Notre approche est basée sur l'utilisation de deux types d'ontologies lors de l'annotation sémantique des services multimédias. La première est l'ontologie métier qui contient les concepts définissant les termes dans un domaine métier. La deuxième ontologie est l'ontologie multimédia qui contient les concepts définissant les annotations des objets multimédias.

L'ontologie métier

Elle définit un vocabulaire qui décrit un domaine d'application particulier comme une ontologie du domaine médical ou touristique. On suppose que les ontologies

du domaine d'application sont disponibles et qu'elles sont réutilisables; nous pouvons les utiliser en les important directement. Dans ce travail, et pour les besoins d'illustrations de l'approche SeSaM, nous avons développé une ontologie dans le domaine médical, plus précisément une ontologie des maladies cancéreuses. Cette ontologie, que nous avons baptisé *cancerOnto*, est utilisée pour annoter les services MaaS du scénario exemple (section 5.2). Un extrait de cette ontologie est présenté dans la **Figure 33** (c.f. **chapitre 6**).

L'ontologie multimédia

La disponibilité de grandes masses d'objets multimédias implique forcément la nécessité des systèmes de recherche efficaces et performants qui facilitent le stockage et la recherche non seulement des données textuelles mais aussi des données de type image, vidéo et audio. Une éventuelle approche peut se baser sur l'annotation sémantique du contenu multimédia pour être sémantiquement décrit et interprété aussi bien par des agents humains (utilisateurs) et des agents automatiques. Par conséquent, il y a un grand besoin d'annoter les contenus multimédia afin d'améliorer l'interprétation et le raisonnement des agents pour une recherche efficace.

L'expression des connaissances multimédia par le biais des ontologies augmente la précision des systèmes de recherche d'information multimédia (Suárez-Figueroa, et al., 2013). En plus, les ontologies ont le potentiel d'améliorer l'interopérabilité des différentes applications qui produisent et utilisent les annotations multimédias. Par conséquent, les ontologies jouent un rôle important dans les systèmes multimédia tout en échangeant la sémantique de leurs contenus entre les systèmes d'informations distribués.

Dans la dernière décennie, d'importants travaux de recherche se sont focalisés sur la construction et la mise en œuvre des ontologies multimédias. Les auteurs (Suárez-Figueroa, et al., 2013) comparent les ontologies les plus connus dans le domaine du multimédia. L'étude comparative est faite sur la base de 16 ontologies qui sont classées en quatre catégories : (1) des ontologies dédiées pour décrire des objets multimédia en général ; (2) des ontologies décrivant les images et les formes comme des éléments visuels pour représenter les images ; (3) des ontologies décrivant des objets visuels en général ; et (4) des ontologies de musique.

Nous nous sommes intéressés dans nos travaux de thèse aux ontologies de la première catégorie qui sont considérées comme des ontologies génériques pour le domaine du multimédia. Les ontologies les plus importantes de cette catégorie sont :

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

- **L'ontologie COMM** (Core Ontology for MultiMedia)[1] (Arndt, et al., 2007), une ontologie avec une conception modulaire qui facilite son extensibilité et son intégration avec d'autres ontologies. L'ontologie COMM est une ontologie OWL formalisant les descriptions du standard MPEG-7. Elle couvre certains outils de description de bas niveau (structurelle, localisation, etc.). En outre, elle fournit les moyens d'inclure des aspects d'analyse dans les annotations, tels que des informations sur l'algorithme et les paramètres correspondants utilisés dans l'extraction d'une description donnée. Cette ontologie est utilisée dans le domaine industriel.
- **L'ontologie M3O** (Multimedia Metadata Ontology)[2] (Saathoff, et al., 2009), qui est destinée aux présentations multimédias sur le web. Le M3O introduit des patrons de conception d'ontologie de base pour les annotations et la décomposition des contenus multimédias. On distingue clairement l'information véhiculée par les éléments multimédias et leur réalisation. Cette ontologie fournit l'infrastructure pour représenter des annotations sémantiques de haut niveau avec des connaissances de base aussi bien que l'annotation avec les éléments de bas niveau extraits des contenus multimédias. Les annotations M3O sont en RDF et peuvent être incorporées dans des présentations multimédias de type SMIL (Synchronized Multimedia Integration Language). Cette ontologie fournit des règles d'appariement avec d'autres ontologies comme : l'ontologie COMM, l'ontologie des ressources multimédias et le standard des métadonnées d'images EXIF.
- **L'ontologie des ressources multimédias** (Ontology for Media Resources)[3] (Lee, et al., 2012), est développée par le groupe MAWG[4] (Media Annotation Working Group) du W3C afin de fournir un vocabulaire de base décrivant les ressources multimédias publiées sur le web. Une ressource multimédia est

[1] http://multimedia.semanticweb.org/COMM/ontology/
[2] https://www.kd.informatik.uni-kiel.de/en/research/ontologies/multimedia-metadata-ontology-m3o
[3] http://www.w3.org/TR/mediaont-10/
[4] https://www.w3.org/2008/WebVideo/Annotations/

définie comme toute ressource physique ou logique qui peut être identifié à l'aide d'un identifiant uniforme de ressource (URI), et qui est lié à un ou plusieurs types de contenu multimédia. L'ontologie des ressources multimédias définie un ensemble de règles d'appariement syntaxique et sémantique avec une variété de formats et standards multimédia.

Afin de représenter la sémantique multimédia des services MaaS, nous avons choisi d'utiliser l'ontologie des ressources multimédias. Notre choix est justifié par les éléments suivants :

- L'ontologie est une recommandation W3C, qui garantit la compatibilité avec les autres standards et technologies du web ;
- L'ontologie fournit des règles d'appariement entre un grand nombre de formats de métadonnées multimédia (Dublin Core, LOM 2.1, ID3, MPEG-7, EXIF, DIG35, etc.), qui facilitent l'interopérabilité ;
- L'ontologie vise à intégrer des ressources de données multimédia disponibles sur le web, qui correspond parfaitement au contexte initié par cette thèse ;
- L'ontologie est accompagnée par une API, qui fournit un accès uniforme à ses éléments.
- L'ontologie est bien documentée, qui profite à la compréhension et l'utilisation de l'ontologie.
- En outre, cette ontologie couvre tous les aspects multimédias, c'est la plus générale pour décrire des objets multimédia.

Un extrait de l'ontologie des ressources multimédias est présenté dans la **Figure 34** (c.f. **chapitre 6**).

5.4.2 Les services web multimédias

Nous présentons dans cette section les services web multimédias. Cette section se décompose en deux parties. La première partie explique nos motivations par la présentation des principaux travaux autour des services web multimédias. La seconde partie donne notre définition des services web multimédia, que nous avons baptisé MaaS (Multimedia as a Service).

Motivations

Une grande partie des travaux autour des services web multimédias se concentrent sur la matérialisation des traitements de contenus multimédias sous forme de services web. Dans le domaine de la recherche d'images, à titre d'exemple, les services web sont utilisés pour des opérations de traitement d'images. Dans un premier travail (Fronckowiak, et al., 2008), les auteurs ont développé une

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

application qui utilise un ensemble de services web d'Amazon (AWS) pour charger et traiter les images. Dans un autre travail (Almeida, et al., 2009), les auteurs ont proposé à leur tour un algorithme de déconvolution d'images accessible via un portail web en utilisant les services web.

Une autre catégorie des travaux utilisent les services web pour l'indexation et la recherche d'images par le contenu. Les auteurs de (Brut, et al., 2011) présentent une solution basée sur l'ontologie WSMO pour décrire de manière sémantique l'interface des services Web en charge de l'indexation multimédia. La solution considère la possibilité de combiner plusieurs services afin d'obtenir des descripteurs plus riches. L'idée a été d'utiliser un format générique XML qui couvre les formats des métadonnées multimédias existants pour décrire la fonctionnalité d'un tel service Web en termes des métadonnées qu'il fournit suite à l'indexation. De même, les auteurs de (Giro-i-Nieto, et al., 2010) ont conçu une architecture d'un système de recherche d'image (CBIR: Content-Based Image Retrieval) basée sur les services web. La solution présentée est composée de deux parties : une interface utilisateur graphique pour la formulation de la requête et un moteur de recherche explorant la base d'images afin de calculer la similarité entre les descripteurs visuels des images de la base avec les descripteurs visuels de la requête.

D'autres travaux utilisent les services web afin de réaliser une opération ou un traitement bien spécifique, nous citons, par exemple, le travail de (Wagner, et al., 2004) qui s'intéresse à améliorer l'accès universel au contenu multimédia dans le contexte des appareils mobiles en utilisant des services web et des concepts de web sémantique basés sur l'ontologie OWL-S.

Dans un autre travail (Rosaci, et al., 2013), les auteurs ont proposé un système de recommandation basé sur l'architecture agent capable de calculer les recommandations des services web multimédia basé à la fois sur le profil de l'utilisateur et l'appareil exploité afin de répondre plus précisément aux préférences et aux intérêts de l'utilisateur. Toutefois, ce travail se concentre principalement sur une analyse expérimentale et il manque de bases théoriques.

A l'issu de l'étude de ces travaux présentés autour des services web multimédias, nous avons constaté que la majorité des travaux traitent séparément chaque type de données multimédia et qu'il existe peu de travaux qui s'intéressent à plusieurs types de données en même temps. Notre état de l'art a révélé aussi qu'il y a assez peu de travaux qui se basent sur les services web multimédia, pour répondre à des requêtes utilisateurs dans le contexte des données hétérogènes et distribuées, comme nous l'envisageons. Notre idée clé consiste alors à proposer une approche, basée sur les services web multimédias, capable de réaliser des

médiations syntaxiques et sémantiques afin de répondre aux besoins de l'utilisateur.

Les services MaaS

La définition et la découverte d'un nouveau type de service adapté à l'accès aux données multimédia dans le contexte de systèmes de données distribués et hétérogènes est le noyau de nos travaux de recherche. L'état de l'art fournit de nombreux outils formels pour gérer les services d'accès aux données (au sens des bases de données relationnelles), on parle de DaaS (Data as a Service) ou service fournisseur de données (Vaculín, et al., 2008) (Carey, et al., 2012) mais on manque manifestement d'outils pour gérer des services web multimédias. C'est à ce verrou que nous consacrons ce travail de thèse qui se situe à la croisée des communautés RI (Recherche d'Information) et Services.

Dans notre approche, les services web multimédias que nous avons baptisé services MaaS acronyme de Multimedia as a Services, sont des services web particuliers de type DaaS car ils accèdent à des données multimédias. En d'autres termes, les services MaaS sont des services fournisseurs qui traitent les données multimédias. Un service MaaS peut retourner un ou plusieurs fichiers images, vidéos ou audios. La représentation sémantique des services MaaS couvre deux types de connaissances : des connaissances métiers qui représentent la logique métier des services et des connaissances multimédias qui symbolisent les propriétés multimédias des services.

Nous modélisons les services MaaS par leurs entrées et sorties. Nous avons étendu le travail de (Vaculín, et al., 2008), qui définit un service comme un couple d'entrées et de sorties, par l'ajout d'une part des concepts ontologiques métiers annotant les entrées et les sorties des services et d'autre part des concepts ontologiques multimédias annotant les données fournies par les services.

Un service web MaaS est un triplet :

$$MaaS = (S_tD, S_mD, MD) \text{ où}$$

- S_tD est la description syntaxique du service,
- S_mD est la description sémantique du service,
- MD est la description multimédia du service.

$$S_tD = (Name, Text, I, O) \text{ où}$$

- ***Name*** est le nom du service,
- ***Text*** est la description textuelle du service,

- ***I*** est l'ensemble des entrées du service, I={(?v,T) | ?v ∈ Var, T ∈ Type},
- ***O*** est l'ensemble des sorties du service, O={(?v,T) | ?v ∈ Var, T ∈ Type},

Var est l'ensemble des noms d'entrées et de sorties,

Type est le type des entrées et des sorties qui peut être soit un type de données XML de base, un type simple ou un type complexe défini dans un XML Schema (XSD).

$$S_mD = (\mathcal{O}, IC, OC) \text{ où}$$

- $\mathcal{O}$ est l'ontologie métier utilisée pour annoter les entrées et les sorties des services,
- ***IC*** est l'ensemble des concepts annotant les entrées du service, $IC \subset C$,
- ***OC*** est l'ensemble des concepts annotant les sorties du service, $OC \subset C$.

$$MD = (\mathcal{O}_m, MC) \text{ où}$$

- $\mathcal{O}_m$ est l'ontologie multimédia utilisée pour annoter les données fournies par les services,
- ***MC*** est l'ensemble des concepts multimédias annotant les objets fournis par les services. Nous avons formulé à partir de ces concepts une requête SPARQL permettant de représenter la partie multimédia du service MaaS (nous expliquons ce choix de représentation plus loin dans ce mémoire). Le ***MC*** est donc représenté comme suit :

$$MC = (I_{md}, O_{md}, R_{md}) \text{ où}$$

- I_{md} représente les entrées de la requête SPARQL,
- O_{md} représente les entrées de la requête SPARQL,
- R_{md} représente les relations sémantiques qui peuvent existaient entre les entrées et les sorties de la requête SPARQL.

5.4.3 La requête utilisateur

Dans SeSaM, la recherche commence à partir d'un besoin utilisateur exprimé par une requête. La satisfaction d'un besoin peut nécessiter plusieurs services. Chaque requête est aussi représentée comme un service MaaS. En effet, la requête Q est définie par un triplet :

$$Q = (S_tD_Q, S_mD_Q, MD_Q) \text{ où}$$

- S_tD_Q est la description syntaxique de la requête,
- S_mD_Q est la description sémantique de la requête,
- MD_Q est la description multimédia de la requête.

$$S_tD_Q = (Name_Q, Text_Q, I_Q, O_Q) \text{ où}$$

- $Name_Q$ est le nom de la requête,
- $Text_Q$ est la description textuelle de la requête,
- I_Q est l'ensemble des entrées de la requête, I={(?v,T) | ?v ∈ Var, T ∈ Type},
- O_Q est l'ensemble des sorties de la requête, O={(?v,T) | ?v ∈ Var, T ∈ Type},

Var est l'ensemble des noms d'entrées et de sorties,

Type est le type des entrées et des sorties qui peut être soit un type de données XML de base, un type simple ou un type complexe défini dans un XML Schema (XSD).

$$S_mD_Q = (\mathcal{O}, IC_Q, OC_Q) \text{ où}$$

- $\mathcal{O}$ est l'ontologie métier utilisée pour annoter les entrées et les sorties de la requête,
- IC_Q est l'ensemble des concepts annotant les entrées de la requête, $IC \subset C$,
- OC_Q est l'ensemble des concepts annotant les sorties de la requête, $OC \subset C$.

$$MD_Q = (\mathcal{O}_m, MC_Q) \text{ où}$$

- $\mathcal{O}_m$ est l'ontologie multimédia utilisée pour annoter les données demandées par de la requête,
- MC_Q est l'ensemble des concepts multimédias annotant les objets demandés par de la requête. Ces concepts, comme nous l'avons dit dans la formalisation des services MaaS, forment une requête SPARQL permettant de représenter la partie multimédia de la requête. Le MC_Q est donc représenté comme suit :

$$MC_Q = (I_{mdq}, O_{mdq}, R_{mdq}) \text{ où}$$

- I_{mdq} représente les entrées de la requête SPARQL,
- O_{mdq} représente les entrées de la requête SPARQL,

- $\boldsymbol{R_{mdq}}$ représente les relations sémantiques qui peuvent existaient entre les entrées et les sorties de la requête SPARQL.

Nous rappelons que les ontologies (métier et multimédia) utilisées pour annoter la requête sont les mêmes utilisées lors de l'annotation des services MaaS.

5.5 Conclusion

Dans ce chapitre, nous avons présenté SeSaM comme une approche de recherche d'information multimédia en s'appuyant sur une architecture SOA, plus précisément la technologie des services web. SeSaM est fondée sur la définition d'un nouveau type de services web, que nous avons appelé les services MaaS, permettant un accès efficace et homogène à des sources de données multimédia sur le web. Cette approche est motivée par un scénario exemple dans le domaine médical.

Le fonctionnement de notre approche est basé sur un processus en deux étapes principales: description et appariement des services MaaS. Nous avons décrit sommairement les deux étapes et nous avons défini formellement les principales notions de SeSaM : les ontologies, les services MaaS et la requête utilisateur. Les deux étapes de notre approche sont détaillées dans la suite de ce mémoire. L'étape de description est détaillée dans le chapitre suivant (c**hapitre 6**) par la définition d'un nouveau langage de description sémantique des services MaaS, le langage **SA4MaaS**. L'étape d'appariement est détaillée dans le **chapitre 7** par la présentation d'un nouveau matchmaker pour l'appariement des services MaaS, le matchmaker **MaaS-MX**. Dans le chapitre suivant, nous présentons en détails le langage SA4MaaS, le langage de description sémantique des services MaaS.

CHAPITRE 6

LE LANGAGE SA4MAAS (DESCRIPTION SÉMANTIQUE DES SERVICES MAAS)

6.1 Introduction

6.2 Motivations de la contribution

6.3 Le langage de description sémantique SA4MaaS

6.4 Exemple illustratif de description sémantique SA4MaaS

6.5 Conclusion

Rapport-Gratuit.com

6.1 Introduction

La description des services web est basée sur des standards tels que le langage WSDL. Ce standard présente des limites de précision et par conséquent un niveau d'expressivité faible pour la description d'un service web. Le manque de sémantique dans WSDL est un frein à une découverte performante et automatique, ainsi il empêche l'invocation et la composition automatiques de services. Le chapitre **3** (page **38**) dans la partie de l'état de l'art a permis d'étudier les principales approches de description des services web. De nombreux langages et approches ont été développés dans l'objectif de décrire sémantiquement les services web pour pallier le manque de sémantique du langage WSDL. Certains efforts tels que WSDL-S et SAWSDL rajoutent des annotations sémantiques aux éléments WSDL afin de répondre à ce manque.

Dans notre travail, nous nous intéressons à étendre les approches de description existantes. Ce chapitre présente notre langage SA4MaaS (Semantic Annotation for MaaS services), qui est une extension de SAWSDL pour la description sémantique des services MaaS. Le principe général de cette contribution est l'intégration, en plus de la sémantique métier, de la sémantique de l'information multimédia dans la description des services MaaS. Par conséquent, cette description se base sur l'utilisation de deux types d'ontologies. Une première ontologie dite « ontologie métier » contient des concepts définissant la sémantique métier des services MaaS (e.g : médical, éduction, tourisme, etc.) et une deuxième ontologie dite « ontologie multimédia » contient des concepts définissant l'ensemble des propriétés et annotations multimédia des services (e.g : format, localisation, information de création, etc.). Ce principe permet à notre approche d'avoir plus de précision et d'expressivité au niveau de la description des services MaaS.

Par la suite, nous expliquons tout d'abord dans la section 6.2 les motivations de cette contribution. La section 6.3 présente le principe et les avantages de notre extension, le langage de description sémantique des services multimédias SA4MaaS. Ensuite, nous présentons dans la section 6.4 un exemple illustratif afin de mieux comprendre les éléments de notre contribution. Un résumé de la contribution est rappelé en conclusion de ce chapitre.

6.2 Motivations de la contribution

Le manque de sémantique est un des principaux reproches faits au standard WSDL. Pour pallier le manque de sémantique de WSDL, de nombreux langages et approches ont été développés dans l'objectif de décrire sémantiquement les services web en ajoutant des concepts sémantiques à leurs descriptions. Nous distinguons, comme nous l'avons déjà présenté dans la section **3.3** (chapitre **3**, page **42**), deux grandes classes de ces approches. Les approches de la première classe utilisent des ontologies de haut niveau telles que OWL-S ou WSMO, évitant ainsi les problèmes d'hétérogénéité sémantique qui peuvent survenir. Les approches de la deuxième classe reposent sur l'ajout d'annotations, tels que WSDL-S, SAWSDL et USDL.

A l'issue de l'étude comparative des principales approches de description sémantique (OWL-S, WSMO et SAWSDL) et par rapport aux critères de comparaison définis dans la section **3.5** (chapitre **3**, page **58**), nous constatons que malgré l'ancienneté et la maturité des approches comme OWL-S et WSMO, elles restent des approches peu flexibles. Ce sont des approches dites "**approches fermées**" car d'une part elles se basent chacune sur un langage de spécification d'ontologie : OWL pour OWL-S et WSML pour WSMO, et d'autre part elles ne spécifient qu'un ensemble très limité de concepts qui sont difficilement extensibles. Par contre, les approches de la deuxième classe qui se basent sur le mécanisme d'annotation sémantique, principalement SAWSDL, restent des approches indépendantes du langage de représentation sémantique grâce à la séparation entre les mécanismes d'annotation sémantique et la représentation de la description sémantique. Ces dernières approches sont dites "**approches ouvertes**" car elles permettent d'utiliser tous types d'ontologies et offrent aux développeurs assez de flexibilité pour choisir leurs langages de représentation sémantique favoris, ou pour réutiliser leurs propres ontologies afin d'annoter les descriptions des services.

Un autre avantage de SAWSDL par rapport au OWL-S et WSMO est par rapport à la réutilisation d'outils, SAWSDL nécessite peu d'efforts de la part des développeurs familiarisés à WSDL contrairement à OWL-S et WSMO. Beaucoup d'outils sont développés dans la cadre du langage WSDL qui nécessite une légère adaptation pour une meilleure réutilisation avec le langage SAWSDL. Par contre, les outils pour les deux autres langages (OWL-S et WSMO) s'avèrent plus difficile à développer, surtout WSMO qui se base sur un langage peu utilisé (le langage WSML).

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Dans cette contribution, nous nous intéressons à la description sémantique des services MaaS. Les annotations sémantiques de ces services MaaS sont de deux types et sont de natures différentes. Le premier type concerne les annotations métiers des services et le deuxième type concerne les annotations multimédias des services. L'idée est d'étendre les approches de description existantes (telles que celles que nous avons étudiées dans l'état de l'art) pour proposer une description adaptée aux services MaaS. Nous constatons que chacune de ces approches possède des forces et des faiblesses. Selon les avantages du langage SAWSDL cités en dessus, par rapport aux OWL-S et WSMO, nous jugeons qu'il est le plus proche et le plus adapté pour décrire les services MaaS. Néanmoins, le langage SAWSDL reste insuffisant pour prendre en compte les spécificités de ces services MaaS et présente un inconvénient majeur au niveau d'explicité et d'expressivité de ses éléments annotés. SAWSDL ne permet pas de spécifier la nature de l'annotation sémantique apportée aux éléments WSDL, elle est purement fonctionnelle et dans aucun cas on peut spécifier des annotations non fonctionnelles. Par conséquent, nous avons choisi d'étendre le langage SAWSDL pour décrire les services MaaS. Cette extension vient combler le manque au niveau de la spécification de la nature de l'annotation sémantique dans SAWSDL par rapport aux services MaaS. La solution se base sur l'ajout des annotations sémantiques multimédias des services MaaS au mécanisme d'annotation du langage SAWSDL.

Après avoir présenté les motivations de notre contribution en termes de description de services MaaS, nous présentons dans la section suivante le langage de description sémantique SA4MaaS pour les services MaaS.

6.3 Le langage de description sémantique SA4MaaS

Dans cette section, nous présentons notre nouvelle extension du langage SAWSDL pour les services MaaS. SAWSDL fournit des mécanismes pour référencer des concepts de modèles définis à l'extérieur du document WSDL. Cela se fait grâce à trois attributs d'extensions. Le premier est *modelReference* et permet d'associer un concept d'une ontologie à un élément WSDL ou XML Schema. Les éléments concernés par cette annotation sont les interfaces, les opérations, les entrées/sorties et les types XML simples. Les deux autres attributs sont *liftingSchemaMapping* et *loweringSchemaMapping* qui permettent de spécifier les correspondances (mapping) entre les données sémantiques et les éléments XML. Le *liftingSchemaMapping* associe aux éléments XML un modèle sémantique. Inversement, le *loweringSchemaMapping* associe au modèle sémantique une structure XML. Ces mappings sont utiles, lors de l'invocation du

service, lorsque la structure de données (schéma XML) ne correspond pas de manière intuitive à la structure du modèle sémantique (ontologie). L'utilisation de ces deux attributs n'est pas illustrée ici car nous nous ne préoccupons pas dans notre travail à la cohérence structurelle des données.

Cependant, nous nous intéressons dans ce travail à l'attribut *modelReference* de SAWSDL qui permet d'associer aux éléments WSDL des concepts d'ontologies de domaine. Cet attribut ne permet en aucun cas de spécifier les informations et les métadonnées des objets multimédias fournis par nos services MaaS. La particularité et la spécificité de ces services MaaS, nous forcent à distinguer les informations métiers des informations multimédias des services. Pour répondre à cet objectif, nous avons étendu le langage SAWSDL en ajoutant à son mécanisme d'annotation un nouvel attribut appelé *multimediaConcept*, qui permet d'ajouter un nouveau niveau d'annotation lié aux aspects multimédias des services MaaS. L'attribut *multimediaConcept* contient une requête SPARQL formulée à partir des métadonnées des objets multimédias fournis par les services MaaS. Nous avons baptisé cette extension SA4MaaS (Semantic Annotation For MaaS services). L'idée principale de SA4MaaS est d'étendre le langage SAWSDL afin d'améliorer l'expressivité de la description des services MaaS.

Le principe général du langage SA4MaaS est de mettre ensemble dans la même structure, tout en gardant la séparation, les annotations sémantiques métiers et les annotations sémantiques multimédias des services. Pour cette raison, la description sémantique des services MaaS se base sur l'utilisation de deux types d'ontologies. Une première ontologie dite « ontologie métier » qui contient des concepts définissant les sémantiques métiers des services MaaS et une deuxième ontologie dite « ontologie multimédia » qui contient des concepts définissant un ensemble d'annotation pour décrire les aspects multimédias des services (e.g. format, localisation, informations de création, etc.). Ce principe permet plus d'expressivité au niveau de la description des services MaaS.

La description sémantique des services MaaS, en s'appuyant sur notre langage SA4MaaS, est structurée en trois niveaux comme le montre la **Figure 31**. Chaque niveau correspond à un élément de la définition formelle des services MaaS (**MaaS = (S_tD, S_mD, MD)**) donnée dans le chapitre précédent (page **101**) :

- Description **syntaxique** représentée par les éléments du standard WSDL, qui correspond à l'élément $\mathbf{S_tD}$ de la définition de services MaaS.
- Description **métier** représentée par un ensemble d'annotations basées sur des ontologies de domaine pour décrire l'aspect métier des services MaaS. Cette description est donnée en utilisant l'attribut *modelReference* du langage

SAWSDL, elle correspond à l'élément $\mathbf{S_mD}$ de la définition de services MaaS. L'attribut *modelReference* contient des références vers des concepts ontologiques métiers.

– Description **multimédia** représentée par un ensemble d'annotations basées sur des ontologies multimédia pour décrire l'aspect multimédia des services MaaS. Cette description est donnée en utilisant le nouvel attribut *multimediaConcept*, elle correspond à l'élément **MD** de la définition de services MaaS. La description multimédia est définie, via l'attribut *multimediaConcept*, par une requête SPARQL contenant des concepts de l'ontologie multimédia qui représentent les métadonnées des objets multimédias fournis par les MaaS.

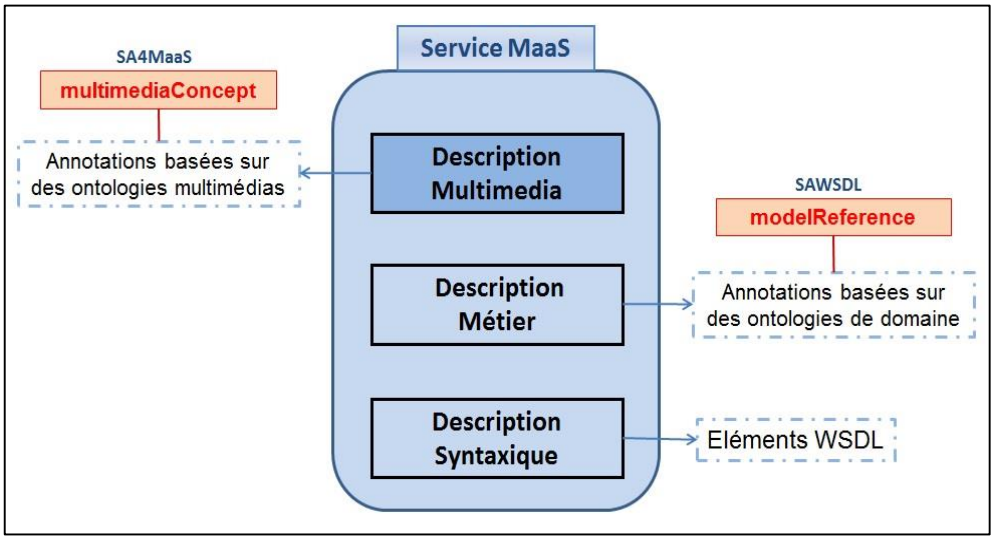

Figure 31 : La description sémantique dans SA4MaaS.

Nous avons choisi de représenter la description multimédia en SPARQL pour les raisons suivantes :

– SPARQL est un langage de requêtes qui permet d'exploiter le volet sémantique de notre approche. Il nous offre la possibilité d'obtenir exactement les données souhaitées. Il permet d'être aussi générique que précis dans notre processus de recherche.

– SPARQL joue le rôle d'une API universelle d'accès aux données. Il permet d'accéder avec aisance aux données et documents stockés sur le web (le contexte de notre travail), nous n'avons pas besoin de connaître a priori la structure et le contenu des données pour pouvoir les interroger.

– SPARQL permet de prendre en compte les relations sémantiques qui peuvent exister entre les concepts représentant les métadonnées multimédias des services MaaS.

– L'utilisation de SPARQL dans notre approche permet d'apporter de la précision et d'avoir de meilleures performances à notre mécanisme d'appariement que nous détaillons dans le chapitre suivant.

L'ajout du nouvel attribut *multimediaConcept* de notre extension au système d'annotation SAWSDL implique la modification de son XSD (XML schema) afin qu'il accepte ce nouveau attribut. Dans la **Figure 32**, nous présentons le XSD de notre extension SA4MaaS. L'attribut ajouté est encadré dans la ligne 12 de la figure.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xs:schema targetNamespace="http://localhost/maas/ns/sa4Maas/"
3             xmlns="http://localhost/maas/ns/sa4Maas/"
4             xmlns:xs="http://www.w3.org/2001/XMLSchema"
5             xmlns:wsdl="http://www.w3.org/ns/wsdl">
6
7    <xs:simpleType name="listOfAnyURI">
8     <xs:list itemType="xs:anyURI"/>
9    </xs:simpleType>
10
11   <xs:attribute name="modelReference" type="listOfAnyURI"/>
12   <xs:attribute name="multimediaConcept" type="listOfAnyURI"/>
13   <xs:attribute name="liftingSchemaMapping" type="listOfAnyURI"/>
14   <xs:attribute name="loweringSchemaMapping" type="listOfAnyURI"/>
15
16   <xs:element name="attrExtensions">
17     <xs:complexType>
18       <xs:annotation>
19         <xs:documentation>This element is for use in WSDL 1.1 only.
20                           It does not apply to WSDL 2.0 documents. Use in
21                           WSDL 2.0 documents is invalid.
22         </xs:documentation>
23       </xs:annotation>
24       <xs:anyAttribute namespace="##any" processContents="lax"/>
25     </xs:complexType>
26   </xs:element>
27 </xs:schema>
```

Figure 32 : XSD du langage SA4MaaS.

La description des services MaaS est enrichie par des références portées sur des concepts multimédias tels que *type*, *format*, *creation*, etc. Le langage SA4MaaS prétend qu'introduisant l'attribut *multimediaConcept* rend les descriptions SAWSDL plus expressives et significatives pour la recherche d'information multimédia. La recherche d'une donnée multimédia dans notre système est basée soit sur les annotations métiers, soit sur les annotations multimédias ou les deux simultanément. Les avantages offerts par notre proposition (extension) sont :

– L'adoption des standards et la réutilisation des outils de services web déjà disponibles. Notre extension SA4MaaS se base sur SAWSDL et par conséquent sur WSDL qui est le langage le plus familier de la communauté des développeurs des services web. Le SAWSDL se distingue aussi par la

flexibilité de ses éléments, lui permettant de s'étendre et de s'adapter à de nouvelles extensions.

- Une description de services qui ne dépend pas d'un langage de description d'ontologie en particulier pour décrire les aspects sémantiques des services web. Cette indépendance offerte par SA4MaaS permet aux développeurs des services web de choisir leurs langages de description d'ontologies (par exemple, UML ou OWL) afin d'annoter leurs services.
- Une description sémantique autant explicite que précise et qui couvre les éléments métiers et multimédia des services. SA4MaaS renforce l'expressivité de la description sémantique des services MaaS, en ajoutant aux annotations métiers des annotations multimédias via l'attribut *multimediaConcept*.

Afin de bien comprendre les éléments de notre proposition que nous apportons aux problèmes de description de services MaaS, nous présentons dans la section suivante un exemple illustratif.

6.4 Exemple illustratif de description sémantique SA4MaaS

Dans cette section, nous présentons un exemple illustratif afin de mieux comprendre le principe de description sémantique des services MaaS par le biais du langage SA4MaaS. Nous poursuivons notre illustration avec le même exemple présenté au début de ce mémoire et que nous avons étoffé dans le chapitre précédent (page **89**). Nous décrivons tout d'abord comment annoter les services MaaS de l'exemple (**Tableau 6**) et la requête de l'utilisateur. Nous présentons ensuite un extrait de description d'un service MaaS annoté par le langage SA4MaaS.

Nous avons défini les services MaaS comme un couple d'entrées et de sorties annotées, d'une part, par des concepts ontologiques métiers et, d'autre part, par des concepts ontologiques multimédias. Par conséquent, la description sémantique des services MaaS se basent, comme nous l'avons déjà présenté, sur l'utilisation de deux types d'ontologies : une ontologie métier et une ontologie multimédia.

L'ontologie métier définit un vocabulaire décrivant un domaine d'application particulier. Pour notre exemple illustratif, nous avons développé une ontologie dans le domaine médical, plus précisément une ontologie des maladies cancéreuses que nous avons baptisé *cancerOnto*. Un extrait de cette ontologie est présenté dans la **Figure 33** qui illustre ses principaux concepts et

leurs relations hiérarchiques. Ces dernières sont des liens de type « is-a » représentées par des flèches en pointillé. Par exemple, un cancer de poumon (*Lung_Cancer*) est un type de cancer (*Cancer*), la chimiothérapie (*Chemotherapy*) est un traitement pour le cancer de poumon (*LC_Treatment*), etc. Un autre type de relation peut exister entre les concepts, ce sont les relations « Object properties » dans le jargon des ontologies. Par exemple, un cancer de poumon (*Lung_Cancer*) est causé (*has_causes*) soit par le tabac (*Smoking*), la pollution (*Pollution*) ou des facteurs génétiques (*Genetics*).

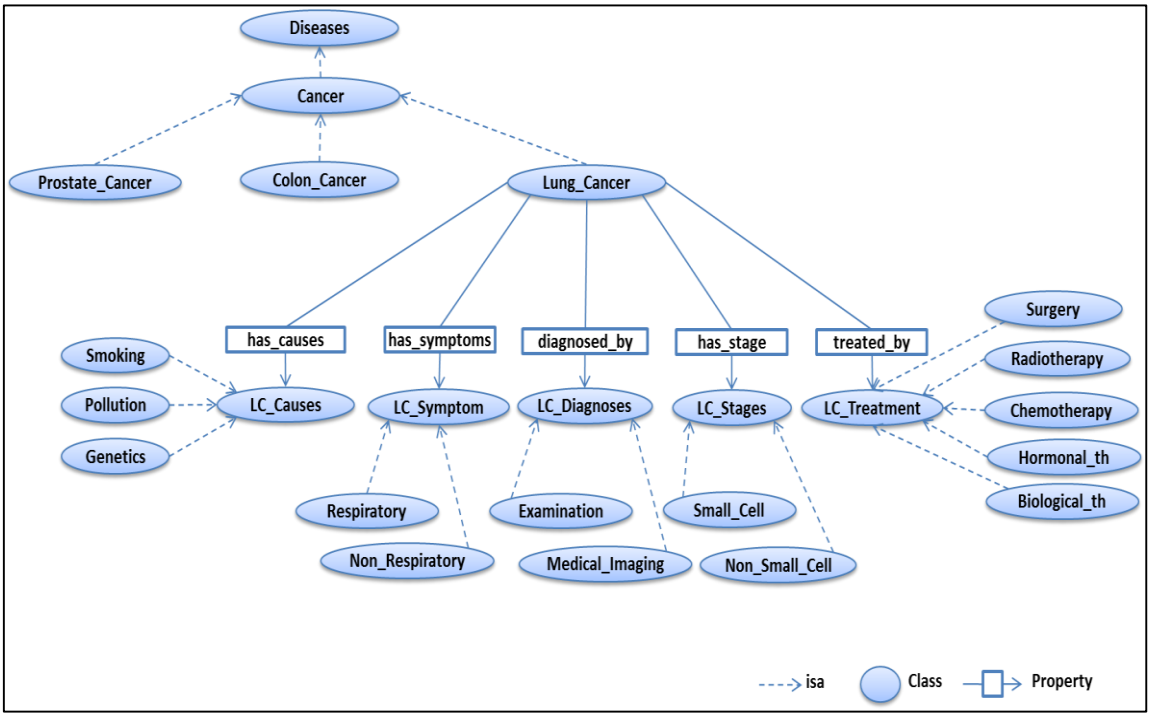

Figure 33 : L'ontologie métier *cancerOnto*.

Le deuxième type d'ontologie, l'ontologie multimédia, permet de représenter la sémantique multimédia des services MaaS. Cette ontologie a pour objectif d'organiser et de définir les métadonnées des contenus multimédias fournis par les services MaaS. Plusieurs travaux de recherche se sont focalisés sur la construction et la mise en œuvre des ontologies multimédias. Nous avons présenté dans le chapitre précédent (page **97**) une étude comparative des ontologies les plus importantes dans le domaine du multimédia. Nous avons choisi d'utiliser l'ontologie des ressources multimédias du W3C, qui est considérée comme l'ontologie la plus générale pour décrire les objets multimédia. Cette ontologie fournit un vocabulaire de base décrivant les ressources multimédias publiées sur le web. Une ressource multimédia est définie comme toute ressource physique ou logique qui peut être identifié à l'aide d'un identifiant uniforme de ressource (URI), et qui est lié à un ou plusieurs types de contenu multimédia.

La figure suivante (**Figure 34**) présente un extrait de l'ontologie des ressources multimédia. L'ontologie est représentée en termes de concepts et de liens entre eux. Chaque concept de l'ontologie caractérise une classe (ellipse) et possède des relations (rectangle) avec soit les autres classes (*Object properties*) ou des propriétés de type de données (*Datatype properties*). La classe principale de cette ontologie est « *MediaResource* » qui définit une ressource multimédia matérialisant soit une image, une vidéo ou un audio (ces relations sont représentées par des liens de type « is-a »). Cette ressource multimédia possède des informations d'identification, de localisation, de création, de publication, de description du contenu, de droit d'auteur et d'autres propriétés techniques (format, compression, langue, etc.).

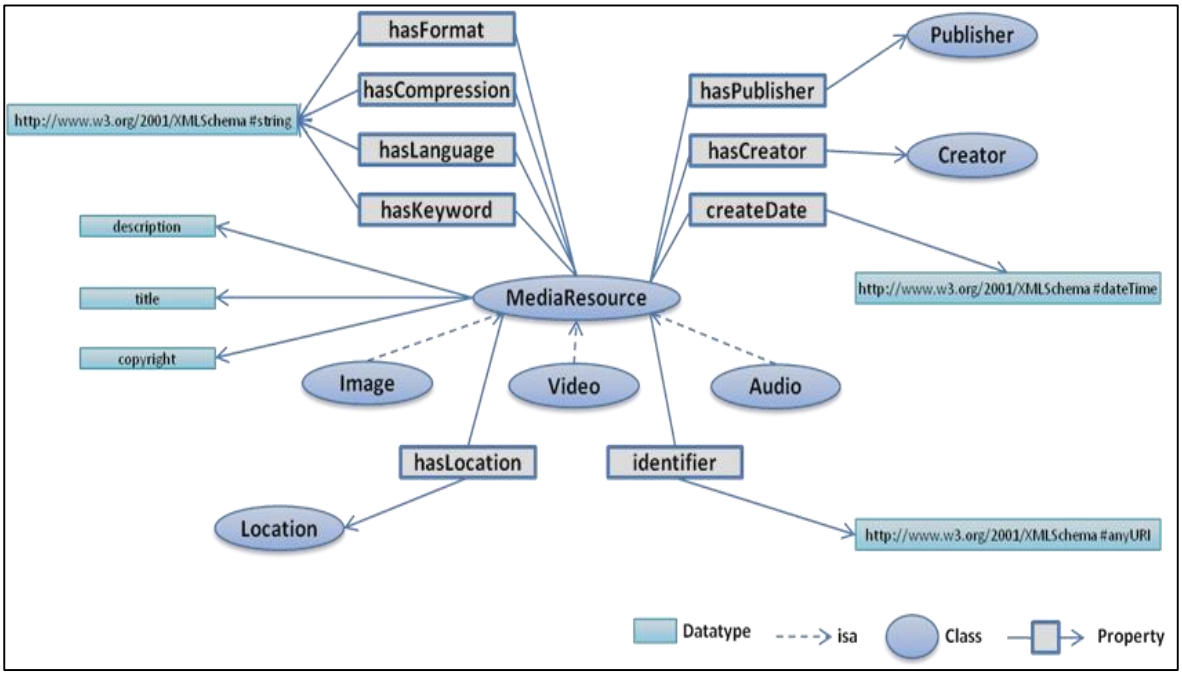

Figure 34 : Extrait de l'ontologie des ressources multimédias.

Le **Tableau 7** présente le résultat de la phase de description sémantique des services MaaS de l'exemple en annotant leurs entrées et sorties par des concepts ontologiques des deux ontologies en dessus. Nous supposons que ces annotations sont faites d'une manière manuelle par un expert du domaine. La troisième et la quatrième colonne du tableau représentent les concepts métiers annotant respectivement les entrées et les sorties des services. Tandis que, la dernière colonne représente les concepts multimédias annotant les objets multimédias fournis par les services MaaS. Ces derniers concepts forment la requête SPARQL qui représente la partie multimédia des services MaaS.

Services	Fonctionnalités	Annotations métiers		Annotations multimédias
		Entrées	Sorties	
S_1	Retourne des images visualisant les différents triatements possibles du cancer de poumon	Lung_Cancer	LC_ Treatment	**MediaResource :** Image **Publisher :** Centre d'imagerie médical **Format :** jpeg
S_2	Retourne des vidéos qui montrent les différentes méthodes de diagnostic du cancer de poumon	Lung_Cancer	LC_ Diagnoses	**MediaResource :** Video **Publisher :** Centre de cancérologie **Format :** mpeg
S_3	Retourne des fichiers audios discutant les résultats d'un scanner thoracique d'un patient atteint d'un cancer de poumon	Lung_Cancer	Medical_ Imaging	**MediaResource :** Audio **Publisher :** Faculté de médecine **Format :** wav
S_4	Retourne des vidéos qui expliquent les différents symptômes du cancer de poumon	Lung_Cancer	LC_ Symptom	**MediaResource :** Video **Publisher :** Centre de cancérologie **Format :** mpeg
S_5	Retourne des images visualisant les différentes causes du cancer de poumon	Lung_Cancer	LC_ Causes	**MediaResource :** Image **Publisher :** Centre d'imagerie médical **Format :** jpeg
S_6	Retourne des fichiers audios des cours sur les traitements possibles du cancer de la prostate	Prostate_Cancer	PC_ Treatment	**MediaResource :** Audio **Publisher :** Faculté de médecine **Format :** mp4
S_7	Retourne des images illustrant les différents traitements possibles pour les cancers bronchiques non à petites cellules	Non_Small_cell	LC_ Treatment	**MediaResource :** Image **Publisher :** Centre d'imagerie médical **Format :** png
S_8	Retourne des vidéos de cours informant la liste des tests et examens à faire en cas d'apparition des symptômes du cancer du poumon	LC_ Symptom	LC_ Diagnoses	**MediaResource :** Video **Publisher :** Faculté de médecine **Format :** mp4
S_9	Retourne des vidéos montrant les différents examens à réaliser pour les cancers bronchiques non à petites cellules	Non_Small_cell	Examination	**MediaResource :** Video **Publisher :** Centre de cancérologie **Format :** mpeg

Tableau 7 : Les services MaaS annotés du scénario exemple.

ppor-t-Gratuit.com

Rappelons la requête de notre utilisateur Bob, il cherche à approfondir ses notions sur les maladies cancéreuses et plus spécifiquement le cancer du poumon, il cherche exactement à récupérer tout type d'information sur ses différentes méthodes de diagnostic. Il veut avoir comme résultat toutes les vidéos fournies par «le centre de cancérologie» au format «mpeg». Le **Tableau 8** présente le résultat de l'annotation sémantique des entrées et des sorties de la requête par des concepts ontologiques des deux ontologies décrites auparavant : l'ontologie métier (**Figure 33**) et l'ontologie multimédia (**Figure 34**).

Requête	Concepts d'entrées	Concepts de sorties	Concepts multimédia
Q	Lung_cancer	LC_Diagnoses	**MediaResource :** Video **Publisher :** Centre de cancérologie **Format :** mpeg

Tableau 8 : La requête annotée.

Nous présentons dans la suite le fichier SAWSDL de l'un des services MaaS du **Tableau 7**. Nous avons choisi le service S_2 comme service exemple (**Figure 35**). Ce service retourne des vidéos qui montrent les différentes méthodes de diagnostic du cancer de poumon. Ces vidéos sont publiées par le « centre de cancérologie » au format « mpeg ». Le service S_2 est en conséquence annoté, d'une part, par « *Lung_Cancer* » pour l'entrée et « *LC_Diagnoses* » pour la sortie qui sont des concepts de l'ontologie métier et, d'autre part, par « *Video* », « *Publisher* » et « *Format* » qui sont des concepts de l'ontologie multimédia.

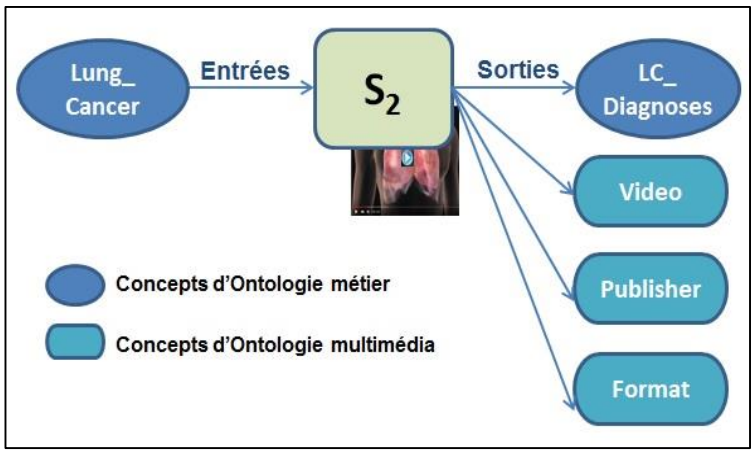

Figure 35 : Service MaaS exemple.

La **Figure 36** montre un extrait de la description sémantique du service S_2 en s'appuyant sur notre langage SA4MaaS. Comme pour SAWSDL, une description SA4MaaS est composée, en plus de l'élément racine, de quatre parties

principales : types (lignes 12 à 30), interface(s) (lignes 31 à 39), binding(s) (lignes 40 à 42) et service (lignes 43 à 48).

- L'élément racine, l'élément « description », est utilisé afin de déclarer les espaces de nom (lignes 2 à 11) utilisés tout au long du document. Les espaces de noms ont été introduits en XML afin de pouvoir mélanger plusieurs vocabulaires au sein d'un même document. Par exemple, la ligne 10 fait référence à l'espace de nom qui définit le vocabulaire de l'ontologie multimédia. Les éléments de ce vocabulaire peuvent être utilisés en les précédant par le préfixe « *ma-ont* ».
- La première partie, l'élément « *types* », définit les types des données utilisés par le service. Par exemple, la structure de données des éléments « *causes* », « *symptoms* », « *diagnoses* », « *stages* » et « *treatment* » est définie comme étant de type *string*.
- La deuxième partie, l'élément « *interface* », décrit les messages envoyés et reçus par le service via des éléments « *operation* ». Chaque élément « *operation* » est une séquence d'entrée « *input* » et de sortie « *output* ». Dans notre exemple, l'interface « *LungCancerDiagnosesServiceInterface* » (lignes 31 à 39) est composée de l'opération « *getVideosDiagnosesLC* ». L'entrée de cette opération est annotée sémantiquement, via l'attribut *modelReference*, par le concept métier « *LungCancer* » (ligne 34) de l'ontologie du domaine *cancerOnto* (**Figure 33**). Par contre, la sortie de l'opération est annotée sémantiquement par les deux attributs *modelReference* et *multimediaConcept*. Ils indiquent respectivement le concept métier « *LC_Diagnoses* » (ligne 36) correspondant à l'annotation métier et la requête SPARQL (ligne 37) correspondant à l'annotation multimédia du service.
- La troisième partie, l'élément « *binding* » (lignes 40 à 42), décrit le format de message concret et le protocole de transmission des données pour les opérations de l'interface « *LungCancerDiagnosesServiceInterface* ».
- La quatrième partie, l'élément « *service* » (lignes 43 à 48), définit un ensemble de « *endpoint* » pour l'interface de notre service. Ces « *endpoint* » spécifient les détails d'implémentation nécessaires pour accéder au service.

La **Figure 37** présente la requête SPARQL du service S_2 qui correspond à sa description multimedia. Cette requête définie les concepts multimédias représentant les métadonnées des vidéos fournis par le service S_2. Les deux premières lignes (lignes 1 et 2) représentent les espaces de noms utilisés dans la requête. La requête (lignes 3 à 11) permet de récupèrer l'identificateur (x) et le titre (y) des vidéos fournis par le service MaaS. Ces vidéos ont été publiées par le « *Centre de cancérologie* » au format « *mpeg* ».

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <wsdl:description
3       targetNamespace="http://localhost/maas/LungCancerServices/"
4       xmlns="http://schemas.xmlsoap.org/wsdl/"
5       xmlns:wsdl="http://www.w3.org/ns/wsdl"
6       xmlns:xs="http://www.w3.org/2001/XMLSchema"
7       xmlns:sawsdl="http://www.w3.org/ns/sawsdl"
8       xmlns:sa4Maas=" http://localhost/maas/ns/sa4Maas"
9       xmlns:tns="http://localhost/maas/LungCancerServices/"
10      xmlns:ma-ont="http://www.w3.org/ns/ma-ont"
11      name="LungCancerDiagnosesWebService">
12    <wsdl:types>
13      <xs:schema targetNamespace="http://localhost/maas/LungCancerServices/">
14        <xs:element name="causes" type="xs:string"/>
15        <xs:element name="symptoms" type="xs:string" />
16        <xs:element name="diagnoses" type="xs:string" />
17        <xs:element name="stages" type="xs:string" />
18        <xs:element name="treatment" type="xs:string" />
19        <xs:complexType name="cancer"> <!-- Type Cancer -->
20          <xs:all minOccurs="0">
21            <xs:element ref="causes"/>
22            <xs:element ref="symptoms" />
23            <xs:element ref="diagnoses" />
24            <xs:element ref="stages" />
25            <xs:element ref="treatment" />
26          </xs:all>
27        </xs:complexType>
28        <xs:element name="lungCancer" type="cancer" />
29      </xs:schema>
30    </wsdl:types>
31    <wsdl:interface name="LungCancerDiagnosesServiceInterface">
32      <wsdl:operation name="getVideosDiagnosesLC">
33        <wsdl:input element="tns:lungCancer"
34          sawsdl:modelReference="http://localhost/canceronto#Lung_Cancer"/>
35        <wsdl:output element="tns:diagnoses"
36          sawsdl:modelReference="http://localhost/canceronto#LC_Diagnoses"
37          sa4Maas:multimediaConcept="Select..." />
38      </wsdl:operation>
39    </wsdl:interface>
40    <wsdl:binding name="LungCancerDiagnosesServiceSOAPBinding"
41      interface="tns:LungCancerDiagnosesServiceInterface"> ...
42    </wsdl:binding>
43    <wsdl:service name="LungCancerDiagnosesWebService"
44      interface="tns:LungCancerDiagnosesServiceInterface">
45       <wsdl:endpoint name="LungCancerDiagnosesServiceEndpoint"
46         binding="tns:LungCancerDiagnosesServiceSOAPBinding"
47         address="http://localhost/LungCancerDiagnosesWebService/" />
48    </wsdl:service>
49  </wsdl:description>
```

Requête SPARQL presentée dans la Figure 37

Figure 36 : Exemple d'un service MaaS présenté dans SA4MaaS.

```
1   PREFIX rdf:"http://www.w3.org/1999/02/22-rdf-syntax-ns"
2   PREFIX ma-ont:"http://www.w3.org/ns/ma-ont"
3   SELECT  ?x, ?y
4   WHERE{
5      ?M    rdf:type              ma-ont:Video.
6      ?M    ma-ont:identifier     ?x.
7      ?M    ma-ont:title          ?y.
8      ?M    ma-ont:hasPublisher   ?P.
9      ?P    rdf:type              ma-ont:Publisher.
10     ?P    ma-ont:hasName        "Centre de cancérologie".
11     ?M    ma-ont:hasFormat      "mpeg".}
```

Figure 37 : Requête SPARQL pour le service S_2.

6.5 Conclusion

Dans ce chapitre, nous avons présenté notre contribution de description sémantique des services MaaS. Dans la première partie du chapitre, nous avons défini le langage SA4MaaS qu'est une extension du langage SAWSDL pour les services MaaS. Un nouvel attribut (*multimediaConcept*) a été ajouté au mécanisme d'annotation de SAWSDL pour définir la description multimédia des services MaaS. Notre extension repose sur l'utilisation de deux types d'ontologies. Une première ontologie, l'ontologie métier, contient des concepts définissant la sémantique métier des services MaaS. Une deuxième ontologie, l'ontologie multimédia, contient des concepts définissant les métadonnées des objets multimédias fournis par les services MaaS. Ces derniers concepts sont formulés en SPARQL pour représenter la description multimédia des services. Ce principe permet à SA4MaaS d'avoir plus de précision et d'expressivité au niveau de la description par rapport aux autres approches de description des services, notamment SAWSDL.

La deuxième partie de ce chapitre était dédiée à un exemple illustratif pour mieux comprendre les éléments de notre contribution. Le chapitre suivant décrit notre matchmaker MaaS-MX, notre méthode d'appariement sémantique entre des descriptions SA4MaaS utilisée pour améliorer la découverte sémantique des services MaaS.

CHAPITRE 7

LE MATCHMAKER MAAS-MX (APPARIEMENT SÉMANTIQUE DES SERVICES MAAS)

7.1 Introduction

7.2 Motivations de la contribution

7.3 Le matchmaker MaaS-MX

7.3.1 Appariement métier

7.3.2 Appariement multimédia

7.4 Exemple illustratif d'appariement sémantique MaaS-MX

7.5 Conclusion

7.1 Introduction

La découverte des services web cherche à identifier les services qui peuvent répondre à une requête donnée en procédant par appariement (Matching en Anglais) des éléments de la requête avec ceux des services disponibles. Ce mécanisme d'appariement consiste à identifier un degré de similitude entre les différents concepts sémantiques qui décrivent le service requis (la requête) et ceux des services offerts (les services publiés). Le chapitre **4** (page **63**) dans la partie de l'état de l'art a permis d'étudier les principales approches d'appariement des services web. De nombreux langages et approches ont été développés dans l'objectif de trouver la liste des services répondant à une requête utilisateur.

Dans notre travail, nous nous intéressons à l'extension des approches d'appariement existantes pour le cas de nos services MaaS décrits par le langage SA4MaaS. Rappelons qu'une description SA4MaaS comporte deux types de description : une description métier et une description multimédia. Nous présentons dans ce chapitre un nouveau matchmaker MaaS-MX (MaaS services Matchmaker) adapté au modèle de description des services MaaS (le langage SA4MaaS). Par conséquent, notre matchmaker est composé de deux étapes primordiales : appariement métier et appariement multimédia. L'appariement métier consiste à comparer la description métier des services et de la requête, tandis que l'appariement multimédia compare la description multimédia des services et de la requête.

Ce chapitre est organisé comme suit. La section 7.2 explique les motivations de proposer un nouveau matchmaker. La section 7.3 décrit le principe et les étapes de notre matchmaker MaaS-MX, le matchmaker sémantique des services MaaS. Ensuite, nous présentons dans la section 7.4 un exemple illustratif afin de mieux comprendre les éléments de notre contribution. La section 7.5 conclut ce chapitre.

7.2 Motivations de la contribution

La découverte des services web a été abordée dans un grand nombre de travaux de recherche. L'élément principal d'un processus de découverte est le mécanisme d'appariement (matching). Ce mécanisme reçoit en entrée la requête et un ensemble de services web et il retourne comme résultat les services web pertinents, classés selon leur degré d'appariement. Nous avons classé ces travaux selon leurs méthodes d'appariement, comme nous l'avons déjà présenté dans l'état de l'art (chapitre **4**, page **63**), en trois grandes classes :

- Les approches non logiques, qui sont basées sur des mécanismes syntaxiques, structurels ou numériques. Comme par exemple le calcul du degré de similarité de graphes structurés, le calcul des fréquences des termes, le calcul de distance entre concepts, les techniques de recherche d'information, le datamining, etc.
- Les approches logiques, qui sont basées sur des mécanismes déductifs fondés sur des formalismes logiques, telles que la logique de premier ordre et la logique de description.
- Les approches hybrides, qui combinent les mécanismes non logiques et logiques.

L'inconvénient majeur des approches non logiques est souvent d'ordre linguistique, terminologique ou statistique. Ainsi, le grand problème de ces approches réside dans le choix des techniques de matching et leurs algorithmes d'optimisation. Des résultats de découverte non pertinents peuvent être causés par plusieurs problématiques bien connues dans ce domaine, telles que les mauvaises et les fausses interprétations de la syntaxe, une morphologie inadaptée des termes, la richesse sémantique d'une langue, des hypothèses de description non respectées, etc.

Les approches logiques possèdent à leurs tours quelques inconvénients majeurs, que nous résumons par les points suivants :

- La complexité des techniques de matching qui affaiblit les chances de passage à l'échelle.
- Le niveau d'intervention de l'utilisateur dans le processus d'appariement qui peut jouer sur le coût d'une approche. Une approche semi-automatique peut couter plus chère, en termes de temps et d'argent, qu'une approche totalement automatique bien élaborée.
- La pertinence de l'interprétation humaine des résultats d'appariement. Le niveau d'expertise joue un rôle primordial dans l'efficacité du processus d'appariement. Un mauvais choix du seuil par exemple conduit forcément à un échec d'appariement.

Les approches hybrides cherchent à remédier à certaines limites des deux approches non logiques et logiques en combinant leurs techniques et leurs mécanismes. L'idée de ces combinaisons hybrides est de réussir là où chacune de ces deux approches a échoué. Les experts dans le domaine de découverte affirment que l'appariement hybride, en combinant l'appariement logique et non logique, donnerait de meilleures performances contrairement à l'utilisation séparée des deux techniques.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

La découverte repose sur la structure du modèle de description des services web et de la requête utilisateur. Dans le cadre de notre approche, les services MaaS et la requête utilisateur sont décrits par le langage SA4MaaS d'où la nécessité de définir une nouvelle approche de découverte adaptée à ce nouveau modèle de description. Vu les avantages des approches hybrides, nous nous sommes inspirés de ces approches pour définir notre nouvelle approche de découverte des services MaaS (le matchmaker MaaS-MX).

A l'issue de l'étude comparative des principales approches de découverte des services web et par rapport aux critères de comparaison définis dans la section **4.5** (chapitre **4**, page **81**), nous constatons que le formalisme (le langage de description) et la technique (le type d'appariement) utilisés pour l'appariement des services web sont deux critères cruciaux mais non exclusifs pour l'évaluation des approches de découverte des services web. D'autres critères, à savoir les éléments et les degrés d'appariement ainsi le niveau d'intervention humaine sont aussi des critères importants pour évaluer une approche de découverte de services web. En analysant ces critères, nous pouvons conclure qu'une bonne approche de découverte doit :

- Opter pour une approche ouverte qui ne dépend pas d'un langage ou une ontologie bien spécifique, tout en essayant de s'aligner avec les standards W3C. C'est le cas de notre approche qui étend SAWSDL (basé sur le langage WSDL) et peut être utilisée avec n'importe quel type d'ontologie. Les annotations sémantiques dans notre approche proviennent de deux types d'ontologies : une ontologie métier qui dépend du domaine métier utilisé et une ontologie multimédia pour annoter les contenus multimédias fournis par les services MaaS.
- Opter pour un appariement hybride pour ne pas écarter d'éventuels résultats et profiter des avantages de chaque type d'appariement. Nous avons proposé dans notre approche : (1) un appariement logique basé sur une approche sémantique de type IO-matching pour la partie métier des services MaaS, et (2) un appariement hybride basé à la fois sur une approche sémantique et syntaxique pour la partie multimédia des services MaaS.
- Opter pour un appariement flexible et en même temps précis afin de répondre au mieux aux besoins de l'utilisateur. Nous avons proposé une échelle plus large de degrés de similarité pour prendre en compte le plus possible de relations sémantiques. Nous nous sommes basés sur les degrés d'appariement proposés dans l'état de l'art et nous les avons étendus par l'ajout de deux nouveaux degrés : "Has-Relation" et "Has-same-Hierarchy" (que nous verrons plus tard) pour avoir plus de souplesse et découvrir d'éventuels services pouvant répondre à la requête en dehors de degrés d'appariement classiques

(Exact, Subsumes, Subsumes-by et Fail). Nous avons représenté ces degrés d'appariement par des scores numériques afin d'avoir plus de flexibilité et de précision.

– Opter pour un appariement totalement automatique ou semi-automatique pour avoir des résultats plus pertinents. Dans le cas de notre approche, l'intervention humaine est exigée seulement au moment du choix du seuil à partir duquel un appariement est jugé réussi.

Après avoir présenté les motivations de notre contribution en termes de découverte de services MaaS, nous présentons dans la section suivante le matchmaker sémantique MaaS-MX pour les services MaaS.

7.3 Le matchmaker MaaS-MX

Dans cette section, nous présentons notre nouveau matchmaker MaaS-MX pour les services MaaS. L'appariement adopté dans MaaS-MX repose sur la structure du modèle de description des services MaaS. Rappelons qu'un service MaaS est défini par un couple d'entrées et de sorties annotées et il est décrit par le langage SA4MaaS qu'est une extension de SAWSDL. Une description SA4MaaS est composée, comme nous l'avons déjà présenté, de deux types de descriptions : une description métier définissant la sémantique métier des services et une description multimédia définissant la sémantique multimédia des services. La description métier est définie via l'attribut *modelReference*, tandis que la description multimédia est définie via l'attribut *multimediaConcept* du langage SA4MaaS.

Par conséquent, le processus d'appariement dans notre matchmaker MaaS-MX (**Figure 38**) se déroule en deux étapes successives. La première étape dite « appariement métier » consiste à comparer la description métier des services MaaS candidats et de la requête. La deuxième étape, qui est exécutée si seulement si l'étape d'appariement métier a réussi, est dite « appariement multimédia » consiste à comparer les descriptions multimédias des services MaaS et de la requête. En d'autres termes, on ne procède pas à l'appariement multimédia si l'appariement métier a échoué.

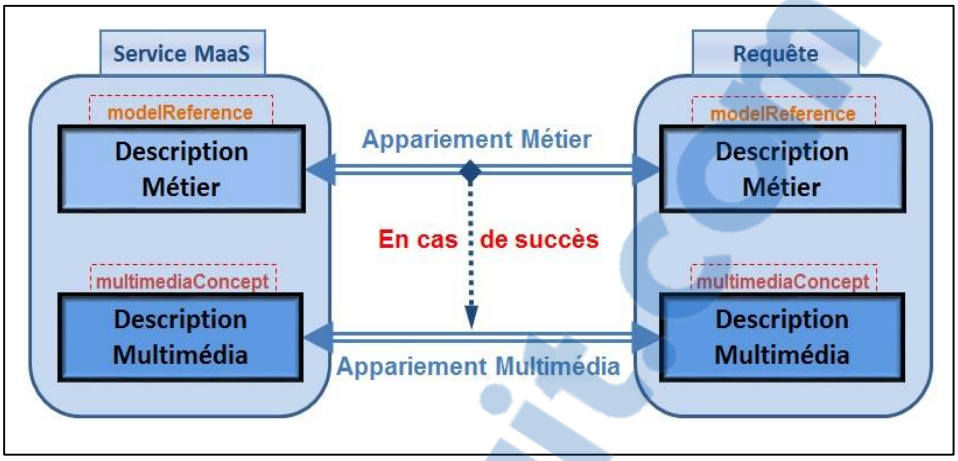

Figure 38 : Le matchmaker MaaS-MX.

L'approche MaaS-MX est une approche hybride d'appariement sémantique des services MaaS décrits en langage SA4MaaS. Ce matchmaker est inspiré des matchmakers de l'état de l'art, spécialement SAWSDL-MX, OWLS-MX et WSMO-MX. La découverte dans notre approche est basée à la fois sur :

- Une approche d'appariement logique (appariement sémantique de type IO-matching pour la partie métier des services MaaS).
- Une approche d'appariement sémantique et syntaxique (techniques de recherche d'information) pour la partie multimédia des services MaaS.

Nous présentons dans la suite les deux étapes de notre matchmaker MaaS-MX, à savoir l'appariement métier et l'appariement multimédia.

7.3.1 Appariement métier

Cette première étape se focalise sur l'identification des services MaaS pertinents répondant à la requête de l'utilisateur tout en se basant sur leurs descriptions métiers. Cela s'effectue en appliquant notre mécanisme (décrit ci-dessous) entre les services MaaS candidats et la requête. Rappelons que dans le cadre de cette thèse, nous avons supposé que les services MaaS ainsi que la requête sont annotés en utilisant les mêmes ontologies (métier et multimédia). C'est-à-dire qu'il n'y a pas d'appariement si les concepts de la requête et ceux des services MaaS proviennent d'ontologies différentes. Cette problématique, liée au domaine des ontologies, dépasse le domaine des services web et ne sera pas traitée dans le cadre de cette thèse. Nous partons donc de l'hypothèse que l'appariement se fait entre deux concepts d'une même ontologie.

L'appariement métier de MaaS-MX est basé sur un IO-matching, qu'est l'un des trois principaux types d'appariement des approches logiques. Les

Rapport-Gratuit.com

éléments à apparier, dans ce type d'appariement, sont limités aux entrées (inputs) et aux sorties (outputs) des services et de la requête. Nous avons choisi ce type d'appariement parce qu'il correspond parfaitement à la structure des services MaaS (couple d'entrées et de sorties). Notre processus d'appariement métier consiste tout simplement à comparer les concepts métiers annotant, d'une part, les entrées des services et de la requête et, d'autre part, les sorties des services et de la requête. Nous supposons que les entrées et les sorties sont annotées chacune par un seul concept. La similarité entre deux concepts (soit $c_1, c_2 \in C$, C : est l'ensemble des concepts de l'ontologie métier) est évaluée par un degré d'appariement. Les degrés d'appariement utilisés dans notre matchmaker sont :

- "Exact" : si le premier concept (c_1) et le deuxième concept (c_2) sont les mêmes (ou équivalents),
- "Subsumes" : si le premier concept (c_1) est un sous concept du deuxième concept (c_2),
- "Subsumes-by" : si le deuxième concept (c_2) est un sous concept du premier concept (c_1),
- "Has-Relation" : si les deux concepts (c_1 et c_2) sont liés par une relation,
- "Has-same-Hierarchy" : si les deux concepts (c_1 et c_2) appartiennent à la hiérarchie du même concept,
- "Unknown" : si l'un des concepts (c_1 ou c_2) n'est pas spécifié,
- "Fail" : si aucune relation ne peut être déterminée entre les deux concepts.

Nous avons étendu les travaux de l'état de l'art sur l'appariement des services par l'ajout de deux nouveaux degrés : "Has-Relation" et "Has-same-Hierarchy". Nous estimons que ces deux derniers degrés apportent plus de précision aux relations qui peuvent exister entre les concepts annotant les entrées et les sorties des services MaaS.

Nous associons à ces degrés d'appariement des valeurs numériques dans [0,1] (**Tableau 9**), qui représentent des degrés de similarité, permettant le calcul de la fonction de similarité globale (dénotée **SIM**) entre une requête et un service MaaS. Pour calculer **SIM**, nous comparons les concepts d'entrées et de sorties de la requête avec les concepts d'entrées et de sorties des services MaaS. Nous considérons que les entrées et les sorties des services MaaS (ou de la requête) ont la même importance dans notre processus d'appariement. De ce fait, la fonction de similarité **SIM** est calculée en se basant à part égale sur la similarité des entrées (dénoté $\mathbf{SIM_I}$) et la similarité des sorties (dénoté $\mathbf{SIM_O}$), la fonction de similarité globale est donnée par la formule suivante :

$$SIM = (SIM_I + SIM_O) / 2$$

Les deux similarités (**SIM_I** et **SIM_O**) représentent les degrés de similarité calculés entre un concept d'entrée (de sortie) d'un service et un concept d'entrée (de sortie) de la requête utilisateur.

Degré d'appariement	Exact	Subsumes / Subsumes-by	Has-Relation	Has-same-Hierarchy	Unknown	Fail
Degré de similarité	1	4/5	3/5	2/5	1/5	0

Tableau 9 : Les degrés de similarité dans MaaS-MX.

Le degré de similarité est une fonction notée *Similarity* dans notre algorithme ci-dessous, elle est définie par :

$$Similarity : C \times C \mapsto [0,1]$$

Cette fonction permet d'associer la similarité entre deux concepts c_1 $(c_1 \in C)$ et c_2 $(c_2 \in C)$ à un degré de similarité $Similarity(c_1, c_2) \in [0,1]$.

Un degré de similarité égal à 1 signifie que l'appariement entre les deux concepts est exact ou équivalent, par contre un degré de similarité égal à 0 signifie un échec d'appariement. Nous considérons que les deux degrés d'appariement "Subsumes" et "Subsumes-by" ont le même degré de similarité qui est égal à 0,8.

Pour cette étape, nous avons développé un algorithme (**Figure 39**) permettant de comparer la description métier des services et de la requête. Le principe général de l'algorithme est de calculer la similarité entre les services MaaS candidats $\{S_1, S_2,..., S_n\}$ et la requête Q afin de retourner la liste des services MaaS pertinents selon un seuil fixé par l'utilisateur. Cet algorithme se déroule comme suit :

- On calcule la similarité (SIM_I) entre le concept d'entrée du premier service S_1 et le concept d'entrée de la requête Q (ligne 9),
- On calcule la similarité (SIM_O) entre le concept de sortie du premier service S_1 et le concept de sortie de la requête Q (ligne 10),
- La fonction de similarité globale est calculée, $SIM = (SIM_I + SIM_O) / 2$ (ligne 11),
- On teste si la mesure de similarité *SIM* du premier service S_1 dépasse un seuil θ (θ est une valeur numérique choisie par l'utilisateur, θ ∈ [0,1]), alors on retient le service (lignes 12 à 14).
- Le même traitement est répété pour tous les services MaaS restants $\{S_2, S_3,..., S_n\}$ (lignes 8 à 15) dont le but de retourner à la fin la liste des services pertinents *R* (ligne 16).

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

```
1  Algorithm 1: Appariement métier
2      Inputs: Requête Q, ensemble S {s1,s2, ..., sn} de services MaaS candidats, Seuil θ
3      Outputs: ensemble R {s1,s2, ..., sm} de services MaaS pertinents
4      /* la fonction Similarity retourne le degré de similarité entre deux concepts. */
5      /* la fonction SIM retourne le degré de similarité global. */
6  Begin
7     R← ∅
8     for each service si in S do
9        SIMI ← Similarity(si.ic,Q.ic)
10       SIMO ← Similarity(si.oc,Q.oc)
11       SIM ← (SIMI + SIMO)/2
12          if ( SIM >= θ ) then
13             R←R∪{si}
14          end if
15    end for
16    return R
17 End
```

Figure 39 : Algorithme d'appariement métier.

7.3.2 Appariement multimédia

Dans cette deuxième étape de notre matchmaker MaaS-MX, nous comparons la description multimédia des services MaaS pertinents issus de la première étape (appariement métier) avec la description multimédia de la requête. Cette deuxième étape, l'appariement multimédia, n'est pas effectuée si l'étape de l'appariement métier a échoué. C'est-à-dire que si l'appariement métier ne retourne pas au moins un service MaaS pertinent, on arrête le processus d'appariement et on informe l'utilisateur qu'aucun des services MaaS disponibles ne répond à sa requête. Rappelons que dans le cadre de cette thèse, nous avons représenté la description multimédia par une requête SPARQL qui contient des concepts de l'ontologie multimédia modélisant les métadonnées des objets multimédias fournis par les services MaaS.

Le mécanisme de l'appariement métier, basé sur un processus IO-matching, consiste à comparer les concepts métiers annotant les entrées et les sorties des services avec les concepts métiers annotant les entrées et les sorties de la requête. Ce type d'appariement qui se fait entre deux concepts métiers n'est pas adapté à l'appariement multimédia car il ne permet pas de comparer plusieurs concepts et leurs relations sémantiques qui peuvent exister entre ces concepts représentant les métadonnées des objets multimédias. Sachant que ces concepts et leurs relations sémantiques sont formulés par une requête SPARQL pour décrire la partie

multimédia des services MaaS, nous proposons pour cette deuxième étape un nouveau mécanisme d'appariement basé sur la comparaison de la requête SPARQL représentant la description multimédia d'un service MaaS candidat avec la requête SPARQL représentant la description multimédia de la requête de l'utilisateur. Dans une requête SPARQL, les relations sémantiques entre les concepts ontologiques sont spécifiées par des "Object properties".

Pour cette étape d'appariement multimédia, nous avons développé un algorithme (**Figure 40**) permettant de comparer la description multimédia des services et de la requête. L'idée principale de l'algorithme est de comparer la requête SPARQL de la requête utilisateur Q avec les requêtes SPARQL des services MaaS candidats issus de l'appariement métier $\{S_1, S_2,..., S_m\}$. Nous nous sommes basés sur le principe d'inclusion et d'équivalence des requêtes (Query Containment and equivalence) (Halevy, 2001) pour permettre la comparaison entre les deux requêtes SPARQL représentant la partie multimédia d'un service MaaS candidat ($S_i.MD$) et la partie multimédia de la requête ($Q.MD$). On dit que $S_i.MD$ est incluse dans $Q.MD$, noté par $S_i.MD \subseteq Q.MD$, si seulement si le résultat fourni par le service candidat S_i est un sous ensemble du résultat fourni par la requête Q. On dit que $S_i.MD$ est équivalente à $Q.MD$, noté par $S_i.MD \equiv Q.MD$ si $S_i.MD \subseteq Q.MD$ et $Q.MD \subseteq S_i.MD$.

```
1  Algorithm 2: Appariement multimédia
2      Inputs: Requête Q, ensemble R {s1,s2, ..., sm} de services MaaS pertinents issus de
3      l'appariement métier
4      Outputs: ensemble T {s1,s2, ..., sk} de services MaaS qui répondent à Q
5      /* la fonction MDMatching retourne True si la partie multimédia du service
6      correspond à la partie multimédia de la requête. */
7  Begin
8      T← ∅
9      for each service si in S do
10         if (MDMatching(si.MD, Q.MD)) then
11                 T ← T∪{si}
12         end if
13     end for
14     return T
15 End
```

Figure 40 : Algorithme d'appariement multimédia.

Le mappage d'inclusion (containment mapping en anglais) a été utilisé dans le domaine d'intégration de données comme une technique nécessaire et suffisante pour tester l'inclusion des requêtes (Chandra, et al., 1977). Nous nous s'inspirons de ce principe pour vérifier l'inclusion entre la requête SPARQL représentant la partie multimédia de la requête utilisateur Q et la requête SPARQL

représentant la partie multimédia d'un service MaaS candidat S_i. On dit que la partie multimédia de la requête utilisateur (Q.MD) contient la partie multimédia d'un service MaaS (S_i.MD) si seulement s'il existe un mappage d'inclusion entre la requête utilisateur et le service. Un mapping β entre Q.MD et S_i.MD est un mappage d'inclusion s'il vérifie les deux conditions suivantes :

1) β mappe chaque classe (concept) de la requête utilisateur Q avec une classe du service MaaS candidat S_i.
2) Toutes les propriétés d'objet (objectproperties) de la requête utilisateur Q sont maintenues entre les classes mappées du service MaaS candidat S_i.

Ce principe d'inclusion de requête via le mappage d'inclusion est vérifié dans notre algorithme par la fonction *MDMatching()* (ligne 10) qui retourne *True* dans le cas où la partie multimédia du service candidat s'apparier (match) avec la partie multimédia de la requête et *False* dans le cas contraire. On réitère l'opération pour tous les services MaaS candidats et à la fin de l'algorithme on garde parmi ces services que les services qui ont un appariement multimédia avec la requête.

Dans la **Figure 41** nous illustrons le fonctionnement de la fonction *MDMatching()*. Cette fonction est divisée en deux grandes parties qui correspondent aux deux conditions à vérifier pour avoir mappage d'inclusion. La première partie compare les classes (concepts) de la requête *Q* avec les classes des services MaaS candidats. Tandis que la deuxième partie compare les propriétés d'objet (object property) de la requête Q avec les propriétés d'objet des services MaaS candidats.

La première partie (lignes 9 à 17) de la fonction *MDMatching()*, qui implémente la première condition de mappage d'inclusion, permet de vérifier via la fonction *classeNodeMapping()* (ligne 12) l'appariement de chaque classe de la requête *Q* avec les classes de l'un des services MaaS candidats. En effet, on compare la première classe de la requête *Q* (C_Q) avec les classes du service candidat (C_{Si}). Si on trouve un appariement, on refait le traitement pour les autres classes de la requête *Q*. Sinon on arrête la comparaison et on conclut qu'il n y a pas d'appariement entre les classes de la requête *Q* et les classes du service MaaS candidat.

La deuxième partie (lignes 19 à 27) de la fonction *MDMatching()*, qui implémente la deuxième condition de mappage d'inclusion, permet de vérifier via la fonction *objectPropertyMapping()* (ligne 22) l'appariement de chaque propriété d'objet (object property) de la requête *Q* avec les propriétés d'objet de l'un des services MaaS candidats. En effet, on compare la première propriété d'objet de la requête *Q* (OP_Q) avec les propriétés d'objet du service candidat

(OP_{Si}). Si on trouve un appariement, on refait le traitement pour les autres propriétés d'objet de la requête *Q*. Sinon on arrête la comparaison et on conclut qu'il n y a pas d'appariement entre les propriétés d'objet de la requête *Q* et les propriétés d'objet du service MaaS candidat.

```
1  Sub-Algorithm 1: MDMatching(Si.md, Q.md)
2        Inputs: Multimedia part Q.md of Query Q, Multimedia part Si.md of MaaS services Si
3        Outputs: Boolean value isMatched (Match or no match)
4  Begin
5        classNode CQ in Q.MD;
6        objectProperty OPQ in Q.MD;
7        isMatched ← false;
8        CQ.first();
9        do
10             classMatch ← false;
11             for each class node CSi in  Si.MD do
12                 if (classNodeMapping(CQ,CSi)) then
13                     classMatch← true;
14                     break;
15             end for
16             CQ.next();
17       while (CQ.hasnext() and classMatch)
18       OPQ.first();
19       do
20             objPropertyMatch← false;
21             for each object property OPSi in  Si.MD do
22                 if (objectPropertyMapping(OPQ,OPSi)) then
23                     objPropertyMatch← true;
24                     break;
25             end for
26             OPQ.next();
27       while (OPQ.hasnext() and objPropertyMatch)
28       if (classMatch and objPropertyMatch) then
29             isMatched← true;
30       return isMatched;
31 End
```

Figure 41 : Algorithme de la fonction MDMatching.

La fonction *MDMatching()* retourne *True* dans le cas où on trouve un double appariement (lignes 28 & 29) : un premier appariement entre les classes de la requête *Q* et les classes du service MaaS candidat, et un deuxième appariement entre les propriétés d'objet de la requête *Q* et les propriétés d'objet du service

MaaS candidat. Si ce double appariement est vérifiée, on rajoute le service MaaS candidat à la liste des services MaaS qui satisfait la requête à la fois sur sa partie métier et sa partie multimédia (en même temps).

Afin de bien comprendre le fonctionnement de notre matchmaker MaaS-MX que nous proposons pour la découverte des services MaaS, nous présentons dans la section suivante un exemple illustratif.

7.4 Exemple illustratif d'appariement sémantique MaaS-MX

Dans cette section, nous présentons un exemple illustratif afin d'expliciter le principe d'appariement de notre matchmaker MaaS-MX. Nous continuons notre illustration avec le même exemple présenté au début de ce mémoire et que nous avons étoffé dans le chapitre **5** (page **89**). Rappelons que les services MaaS ainsi que la requête de cet exemple sont déjà annotés en utilisant notre langage de description sémantique SA4MaaS. Le résultat de ces annotations est illustré, dans le chapitre précédent, par le **Tableau 7** pour les services MaaS et par le **Tableau 8** pour la requête.

Comme nous l'avons déjà présenté dans la section précédente, notre matchmaker MaaS-MX est composé de deux étapes successives : appariement métier et appariement multimédia. Commençons par l'appariement métier qui consiste à calculer la similarité (**SIM**) entre la requête utilisateur **Q** et les services MaaS disponibles $\{S_1, S_2,..., S_n\}$. Le calcul de cette similarité se base à part égale sur : (1) le calcul de la similarité d'entrée ($\mathbf{SIM_I}$) entre les concepts annotant les entrées de la requête **Q** et les concepts annotant les entrées des services MaaS disponibles $\{S_1, S_2,..., S_n\}$ (2) et le calcul de la similarité de sortie ($\mathbf{SIM_O}$) entre les concepts annotant les sorties de la requête **Q** et les concepts annotant les sorties des services MaaS disponibles $\{S_1, S_2,..., S_n\}$. Nous illustrons dans le **Tableau 10** le résultat de ces calculs. La quatrième et la sixième colonne du tableau représentent respectivement la similarité d'entrée ($\mathbf{SIM_I}$) et la similarité de sortie ($\mathbf{SIM_O}$). L'avant dernière colonne représente la similarité globale ($\mathbf{SIM = (SIM_I + SIM_O)/2}$) entre chaque service MaaS de l'exemple et la requête.

Services	Fonctionnalités	Concepts d'entrées	SIM_I	Concepts de sorties	SIM_O	SIM	Annotations multimédias
S_1	Retourne des images visualisant les différents triatements possibles du cancer de poumon	Lung_Cancer	**1**	LC_Treatment	**0**	**0.5**	**MediaResource :** Image **Publisher :** Centre d'imagerie médical **Format :** jpeg
S_2	Retourne des vidéos qui montrent les différentes méthodes de diagnostic du cancer de poumon	Lung_Cancer	**1**	LC_Diagnoses	**1**	**1**	**MediaResource :** Video **Publisher :** Centre de cancérologie **Format :** mpeg
S_3	Retourne des fichiers audios discutant les résultats d'un scanner thoracique d'un patient atteint d'un cancer de poumon	Lung_Cancer	**1**	Medical_ Imaging	**0.8**	**0.9**	**MediaResource :** Audio **Publisher :** Faculté de médecine **Format :** wav
S_4	Retourne des vidéos qui expliquent les différents symptômes du cancer de poumon	Lung_Cancer	**1**	LC_Symptom	**0**	**0.5**	**MediaResource :** Video **Publisher :** Centre de cancérologie **Format :** mpeg
S_5	Retourne des images visualisant les différentes causes du cancer de poumon	Lung_Cancer	**1**	LC_Causes	**0**	**0.5**	**MediaResource :** Image **Publisher :** Centre d'imagerie médical **Format :** jpeg
S_6	Retourne des fichiers audios des cours sur les traitements possibles du cancer de la prostate	Prostate_Cancer	**0.4**	PC_Treatment	**0**	**0.2**	**MediaResource :** Audio **Publisher :** Faculté de médecine **Format :** mp4
S_7	Retourne des images illustrant les différents traitements possibles pour les cancers bronchiques non à petites cellules	Non_Small_cell	**0.8**	LC_Treatment	**0**	**0.4**	**MediaResource :** Image **Publisher :** Centre d'imagerie médical **Format :** png
S_8	Retourne des vidéos de cours informant la liste des tests et examens à faire en cas d'apparition des symptômes du cancer du poumon	LC_ Symptom	**0.6**	LC_Diagnoses	**1**	**0.8**	**MediaResource :** Video **Publisher :** Faculté de médecine **Format :** mp4
S_9	Retourne des vidéos montrant les différents examens à réaliser pour les cancers bronchiques non à petites cellules	Non_Small_cell	**0.8**	Examination	**0.8**	**0.8**	**MediaResource :** Video **Publisher :** Centre de cancérologie **Format :** mpeg

Tableau 10 : Le calcul des similarités des services MaaS du scénario exemple.

Pour bien comprendre le principe de ces calculs, prenant par exemple le service S_3. Le calcul de sa similarité se fait en trois étapes :

- La similarité d'entrée de S_3 égale à **1** (**$SIM_I(S_3) = 1$**) parce que le degré d'appariement entre le concept d'entrée du service **(S_3.ic = "Lung_Cancer")** et le concept d'entrée de la requête de l'utilisateur **(Q.ic = "Lung_Cancer")** est « Exact », équivalent à **1** : degré de similarité (**Similarity(S_3.ic,Q.ic) = 1**).
- La similarité de sortie de S_3 égale à **0.8** (**$SIM_O(S_3) = 0.8$**) parce que le degré d'appariement entre le concept d'entrée du service **(S_3.oc = "Medical_Imaging")** et le concept d'entrée de la requête de l'utilisateur **(Q.oc = "LC_Diagnoses")** est « Subsumes », équivalent à **0.8** : degré de similarité (**Similarity(S_3.oc,Q.oc) = 0.8**).
- La similarité globale égale à **0.9** (**SIM = (1 + 0.8)/2**).

Après le calcul de ces mesures de similarité, nous comparons la similarité globale (**SIM**) de chaque service MaaS avec le seuil fixé par l'utilisateur. Supposons que le seuil fixé par l'utilisateur, dans cet exemple, est de 0.7 (θ=0.7). Cela veut dire que l'utilisateur veut garder les services qui ont une mesure de similarité supérieure ou égale à 0.7. En d'autres termes, si la similarité du service en cours de test dépasse le seuil fixé ($SIM_{SI} \geq 0.7$) alors on retient ce service dans la liste des services MaaS pertinents. En conséquence, les services retenus dans cette première étape d'appariement (appariement métier) sont :

$$\textbf{Appariement métier} = \{S_2, S_3, S_8, S_9\}$$

Passons maintenant à la deuxième étape d'appariement de notre matchmaker MaaS-MX, qui est l'appariement multimédia. Rappelons que pendant cette étape, nous vérifions la correspondance entre la partie multimédia des services MaaS retenus par l'appariement métier et la partie multimédia de la requête de l'utilisateur. La comparaison entre ces parties multimédias (représentées par des requêtes SPARQL) est basée sur le principe d'inclusion des requêtes que nous avons implémenté dans notre algorithme d'appariement multimédia (**Figure 40**). Le résultat de cette deuxième étape d'appariement est :

$$\textbf{Appariement multimédia} = \{S_2, S_9\}$$

Les services de cette dernière liste correspondent aux services qui satisfont la requête de l'utilisateur à la fois dans sa partie métier et sa partie multimédia.

7.5 Conclusion

Dans ce chapitre, nous avons présenté notre contribution de découverte sémantique des services MaaS : le matchmaker MaaS-MX. Dans la première partie du chapitre, nous avons décrit le matchmaker MaaS-MX qui est composé de deux étapes successives : appariement métier et appariement multimédia. L'appariement multimédia n'est pas réalisé si l'appariement métier a échoué. Dans l'appariement métier on compare la description métier des services MaaS et de la requête, tandis que dans l'appariement multimédia on compare la description multimédia des services MaaS et de la requête. Nous avons écrit des algorithmes permettant de valider le principe de fonctionnement de deux étapes d'appariement de notre matchmaker.

Dans la deuxième partie du chapitre, nous avons exposé un exemple illustratif pour mieux comprendre le fonctionnement de notre matchmaker ainsi pour montrer l'intérêt apporté par l'appariement multimédia en termes de précision à notre contribution.

Rapport-Gratuit.com

CHAPITRE 8

PROTOTYPE ET ÉVALUATION

8.1 Introduction

8.2 Prototype

8.2.1 Module de description

8.2.12 Module d'appariement

8.3 Evaluation

8.3.1 Les mesures de performance

8.3.2 Types d'évaluation

8.3.3 Scénarios d'évaluation

8.4 Conclusion

8.1 Introduction

Ce chapitre est consacré à la validation de notre approche. Pour cela nous avons procédé à un développement d'un prototype, une expérimentation de notre système et une évaluation de ses performances. Nous avons expérimenté notre approche dans deux domaines différents : la médecine et le tourisme.

Nous décrivons, en premier lieu, notre prototype nommé DAMaS (**D**escription **a**nd **M**atching of M**a**aS services **S**ystem) en présentant son architecture, ses différents modules et ses interfaces graphiques. Nous abordons ensuite notre évaluation et nos expérimentations menées et nous discutons les résultats obtenus.

8.2 Prototype

Nous avons développé un prototype intitulé DAMaS (**D**escription **a**nd **M**atching of M**a**aS services **S**ystem). Ce prototype est implémenté en Java (JDK 1.8) sous l'environnement NetBeans 8.0.2. Il est composé principalement de deux modules : le module de description et le module d'appariement. La **Figure 42** illustre l'architecture du système DAMaS.

8.2.1 Module de description

Ce module est accessible uniquement par l'expert. Il permet la description sémantique des services MaaS en s'appuyant sur notre langage SA4MaaS présenté dans le chapitre **6**. Nous rappelons que le langage SA4MaaS utilise deux types d'ontologies : une ontologie métier et une ontologie multimédia pour l'annotation sémantique des différents services MaaS et de la requête utilisateur.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

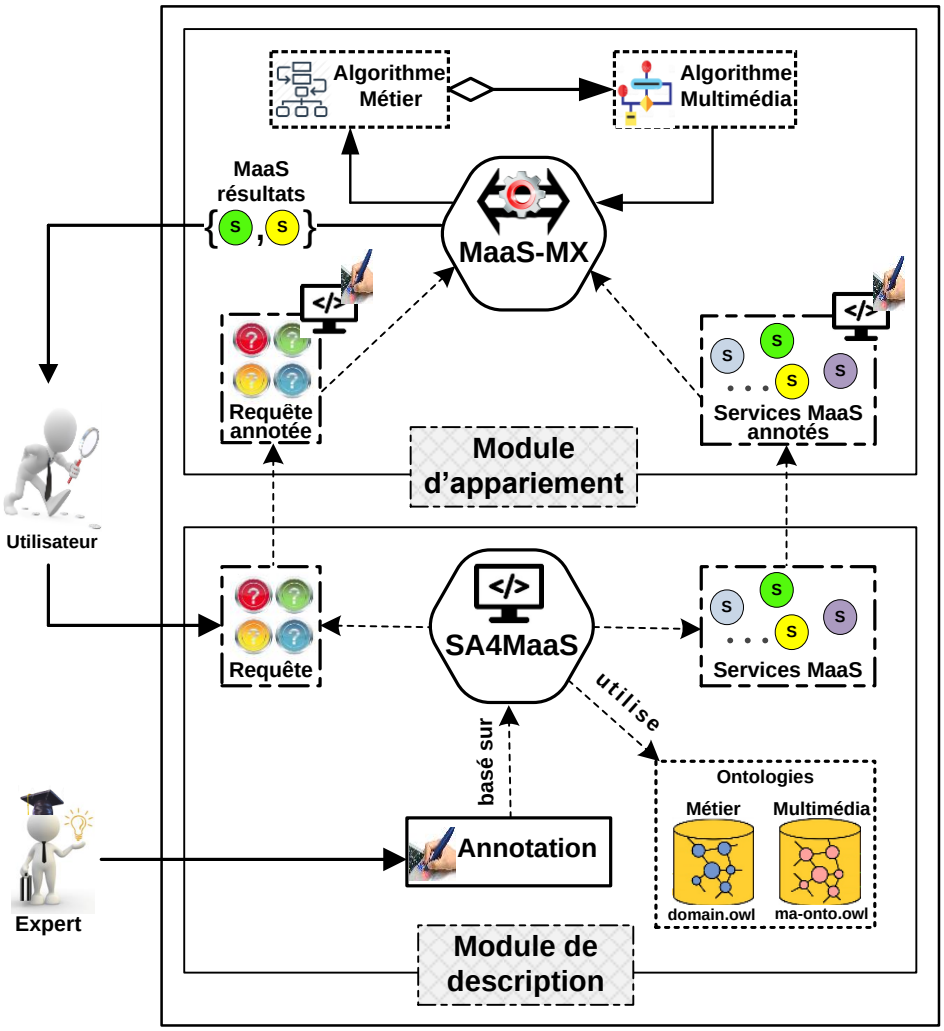

Figure 42 : Architecture du système DAMaS.

Les figures **Figure 43** et **Figure 44** présentent les interfaces du module de description. L'expert choisi d'abord le service MaaS à annoter et les fichiers owl des deux ontologies (métier et multimédia) utilisées dans la description sémantique des services MaaS (**Figure 43**). Ensuite, il associe les concepts métiers et multimédias aux entrées et sorties du service MaaS en cours d'annotation (**Figure 44**). L'expert répète cette opération pour les autres services MaaS disponibles et la requête de l'utilisateur. A la fin de cette opération d'annotation, des fichiers SA4MaaS seront générés par le système DAMaS des différents services MaaS et de la requête utilisateur. Un exemple de ce fichier est illustré par la **Figure 36** présentée dans le chapitre **6**.

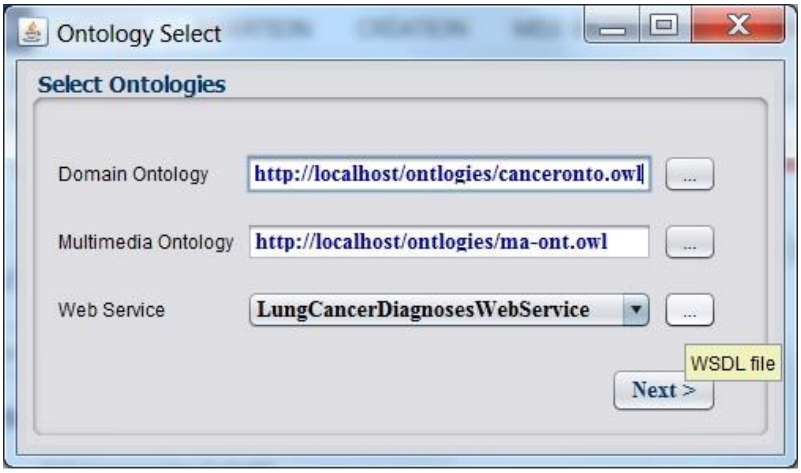

Figure 43 : La préparation de l'annotation sémantique dans DAMaS.

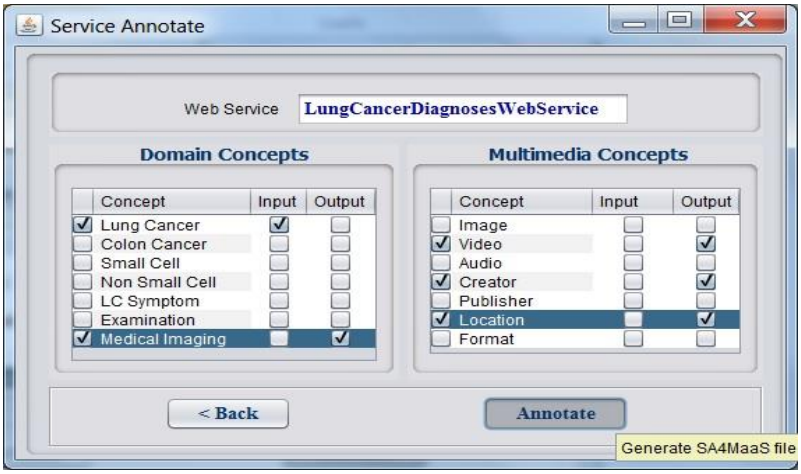

Figure 44 : L'annotation métier et multimédia dans DAMaS.

8.2.2 Module d'appariement

Ce module permet la comparaison des annotations, issues du premier module, des services MaaS avec celles de la requête de l'utilisateur. Cette comparaison est faite sur la base des fichiers SA4MaaS générés par le module de description. Ce deuxième module implémente les deux algorithmes d'appariement proposés par notre matchmaker MaaS-MX. Le premier algorithme compare les annotations métiers des services MaaS avec celles de la requête, il fournit à la fin de cette comparaison les similarités d'entrées et de sorties de chaque service MaaS avec la requête. Sur la base de ces similarités calculées et du seuil défini par l'utilisateur, le système DAMaS filtre la liste des services MaaS disponibles pour ne garder que les services les plus proches du besoin de l'utilisateur. Le deuxième algorithme de notre système compare les annotations multimédias (représentées

par des requêtes SPARQL) de la requête utilisateur avec celles des services MaaS filtrés par le premier module. Il vérifie d'abord si les concepts (classes) de la requête SPARQL, représentant les services MaaS, correspondent bien aux concepts de la requête SPARQL, représentant la requête utilisateur. Ensuite, il vérifie la correspondance entre les relations (objectproperties) des requêtes SPARQL, représentant les services MaaS, avec les relations de la requête SPARQL, représentant la requête utilisateur. Le système DAMaS fournit à travers ce module les services MaaS qui répondent aux besoins de l'utilisateur sur le plan à la fois métier et multimédia. La **Figure 45** montre l'interface principale du module d'appariement du système DAMaS.

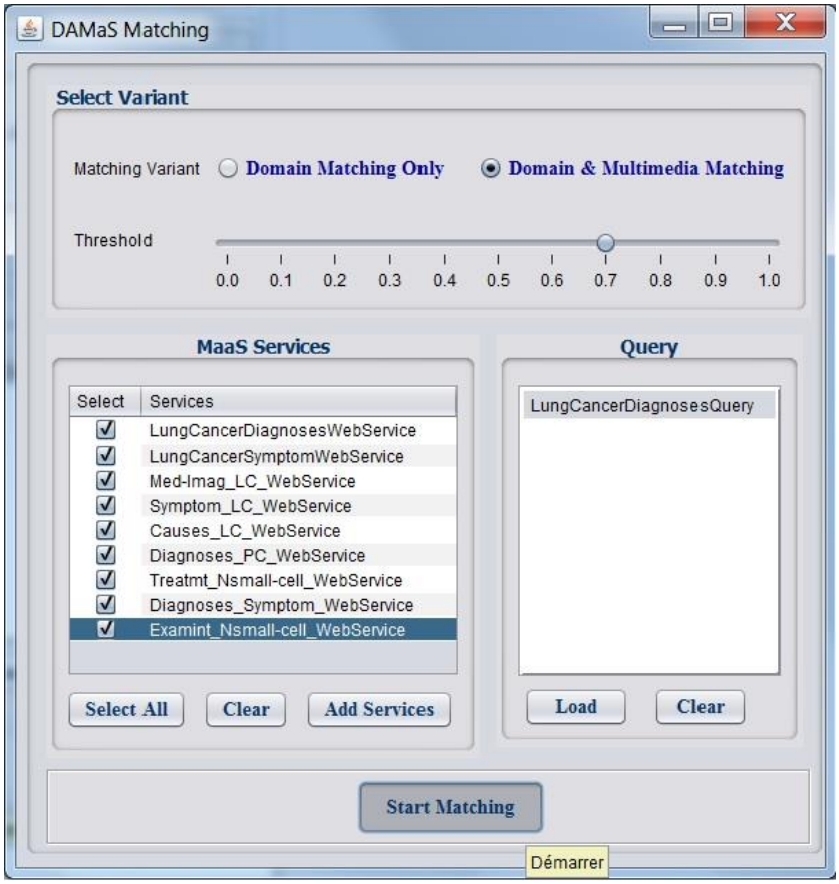

Figure 45 : L'appariement dans le système DAMaS.

8.3 Evaluation

Pour évaluer les performances de notre système, nous avons utilisé les trois mesures de performance les plus connues, à savoir : rappel, précision et F-Mesure. Dans cette section, nous commençons par définir ces mesures avant de les utiliser dans notre évaluation.

8.3.1 Les mesures de performance

Rappel

Le rappel est défini par le nombre (A) des résultats pertinents retrouvés par le système au regard du nombre total (B) des résultats pertinents. (B) représente le nombre des résultats pertinents retrouvés (appelés aussi résultats vrais positifs) et non retrouvés (appelés aussi résultats faux négatifs) par le système. Le rappel est ainsi défini par la formule suivante :

$$R = \frac{A}{B}$$

En fait, lorsque l'utilisateur interroge la base de services, il souhaite obtenir toutes les solutions qui pourraient répondre à sa requête. Le taux de rappel devrait alors être élevé. Par contre, si le système possède de nombreux services pertinents et la plupart d'entre eux ne sont pas fournis par le système, le taux du silence serait élevé. Le silence s'oppose alors au rappel (rappel + silence = 1).

Précision

La précision est définie par le nombre (A) des résultats pertinents retrouvés par le système par rapport au nombre total (C) des résultats. (C) indique le nombre des résultats pertinents et non pertinents retrouvés par le système. Les résultats non pertinents retrouvés par le système sont appelés aussi résultats faux positifs. La précision est ainsi définie par la formule suivante :

$$P = \frac{A}{C}$$

En fait, lorsque l'utilisateur interroge la base de services, il souhaite que les solutions proposées en réponse à sa requête soient pertinentes. Les services non pertinents retournés constituent du bruit. La précision s'oppose alors au bruit (précision + bruit = 1). Si elle est élevée, cela signifie que peu de solutions non pertinentes sont proposées par le système et que ce dernier peut être considéré comme "précis".

F-Mesure

La F-mesure est une mesure populaire qui combine la précision et le rappel pour tester l'exactitude du système. Elle est calculée par la formule suivante :

$$F = \frac{2.R.P}{R + P}$$

8.3.2 Types d'évaluation

Afin de valider notre approche, nous avons réalisé deux types d'évaluations :

- Dans le premier, nous avons comparé les deux variantes de notre approche qui se basent sur les deux types d'appariement. La première variante de l'approche concerne l'utilisation de l'appariement métier sans l'appariement multimédia. La deuxième variante de l'approche concerne l'utilisation des deux types d'appariements (métier et multimédia). L'objectif de cette évaluation est de mesurer l'impact et la valeur ajoutée de l'appariement multimédia sur notre approche.
- Dans le second, nous avons comparé les deux variantes de notre approche avec l'approche SAWSDL-MX. Nous avons choisi l'approche SAWSDL-MX, parmi les approches existantes de l'appariement, parce que nous la considérons comme la solution la plus proche de notre approche (rappelons que le langage de description de nos services MaaS, le langage SA4MaaS, est une extension du langage SAWSDL). Ainsi, le SAWSDL-MX est le matchmaker hybride le plus connu et le plus complet car il est inspiré des deux autres matchmakers les plus largement répandus, à savoir OWLS-MX et WSMO-MX. L'objectif de cette évaluation est de mesurer les performances de notre approche par rapport à celles des autres approches de l'état de l'art. Plusieurs variantes de SAWSDL-MX sont implémentées dans la littérature, comme nous l'avons déjà cité dans l'état de l'art, mais aucune d'elles ne correspond parfaitement à la structure de nos services MaaS. Dans notre système DAMaS, nous avons adapté et implémenté le principe de base d'appariement de l'approche SAWSDL-MX (la variante SAWSDL-M0) pour permettre sa comparaison avec notre approche. Dans notre implémentation de SAWSDL-MX, nous nous sommes focalisés sur le calcul de similarité entre les services et la requête en nous basant sur les différents degrés d'appariement logiques proposés par l'approche (*Exact*, *Plug-in*, *Subsumes, Subsumed-by* et *Fail*). Nous avons attribué à ces degrés de matching, selon leurs importances, les mêmes degrés de similarité utilisés dans notre approche, pour permettre la comparaison avec SAWSDL-MX. Ce deuxième type d'évaluation a été expérimenté seulement sur notre deuxième scénario relatif au tourisme, que nous présentons dans la section suivante.

8.3.2 Scénarios d'évaluation

Nous avons expérimenté notre système dans deux domaines différents : la médecine et le tourisme. Dans un premier temps, nous évaluons le système DAMaS sur le scénario médical présenté tout au long de ce mémoire. Dans un deuxième temps, nous évaluons le système DAMaS sur un nouveau scénario dans le domaine de tourisme.

Scénario Médical

Nous évaluons notre système DAMaS sur le scénario présenté tout au long de ce mémoire en calculant les trois mesures de performance citées dans la section précédente. Rappelons que les neuf services MaaS et la requête de ce scénario sont déjà annotés en utilisant notre langage de description sémantique SA4MaaS. Le résultat de ces annotations est illustré, dans le chapitre **6**, par le **Tableau 7** pour les services MaaS et par le **Tableau 8** pour la requête. Ainsi, le **Tableau 10** du chapitre précédent présente le résultat de calcul des similarités des services MaaS par rapport à la requête du scénario. Rappelons que le seuil fixé par l'utilisateur, dans ce scénario, était de 0.7 (θ=0.7). Par conséquent, le résultat logique (raisonnement humain) qu'on devrait avoir est représenté par la liste suivante des services : $\{S_2, S_9\}$. Les services de cette liste représentent les services relevant de la requête de l'utilisateur, qui permet le calcul des différentes mesures de performance de notre système.

Dans notre premier type d'évaluation, nous comparons les deux variantes de notre approche, la variante 1 représente l'approche sans l'utilisation de l'appariement multimédia tandis que la variante 2 représente l'approche avec l'utilisation des deux types d'appariement. Le **Tableau 11** et le **Tableau 12** présentent les résultats de rappel, précision et F-Mesure de ces deux variantes sur le scénario médical en variant le seuil θ à 0.5, 0.6, 0.7, 0.8, 0.9 et 1.

Les figures suivantes (**Figure 46**, **Figure 47** et **Figure 48**) montrent la représentation graphique des mesures de performance (rappel, précision et F-Mesure) des deux variantes de notre système.

Mesures / Seuil θ	Rappel	Précision	F-Mesure
θ = 0.5	0.95	0.29	0.44
θ = 0.6	0.90	0.40	0.55
θ = 0.7	0.80	0.45	0.58
θ = 0.8	0.68	0.50	0.58
θ = 0.9	0.45	0.55	0.49
θ = 1	0.35	0.62	0.45

Tableau 11 : Rappel, Précision et F-Mesure de la variante 1.

Mesures / Seuil θ	Rappel	Précision	F-Mesure
θ = 0.5	1	0.60	0.75
θ = 0.6	0.90	0.77	0.83
θ = 0.7	0.82	0.92	0.87
θ = 0.8	0.72	1	0.84
θ = 0.9	0.50	1	0.67
θ = 1	0.37	1	0.54

Tableau 12 : Rappel, Précision et F-Mesure de la variante 2.

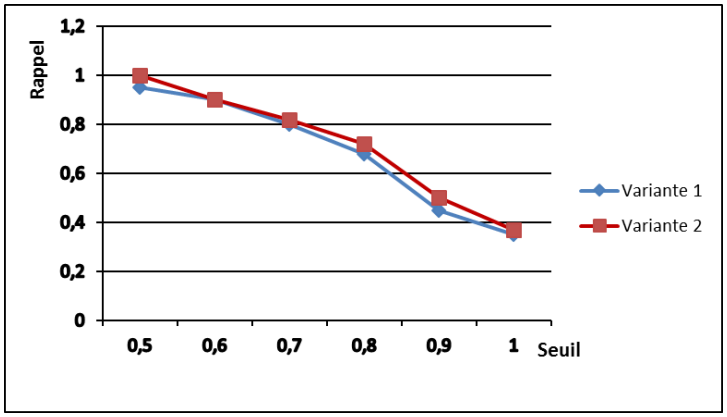

Figure 46 : Le graphe de rappel des deux variantes de DAMaS.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

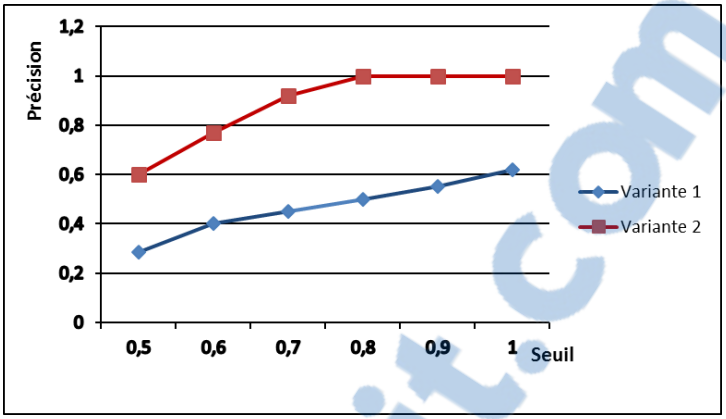

Figure 47 : Le graphe de précision des deux variantes de DAMaS.

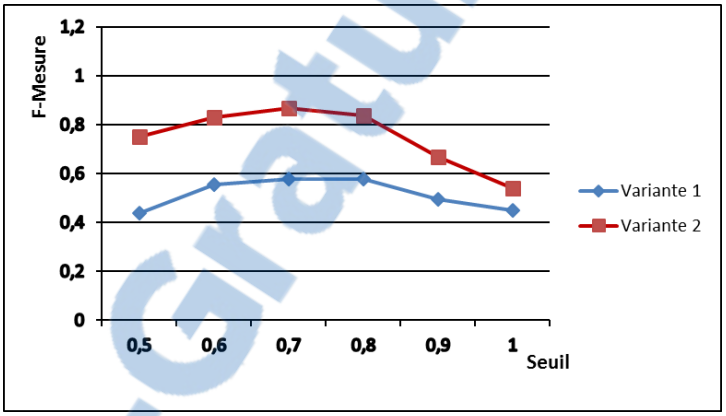

Figure 48 : Le graphe de F-Mesure des deux variantes de DAMaS.

A partir de ces graphes, on remarque que les deux variantes ont des performances similaires en termes de rappel. Par contre, il y a une large supériorité de la deuxième variante par rapport à la première en termes de précision. Cette situation est causée par l'élimination des résultats non pertinents récupérés par le système lorsqu'on utilise les deux appariements ensemble (métier et multimédia). Ces observations sont confirmées par le graphe de F-Mesure présenté dans la **Figure 48** qui montre que la deuxième variante offre des meilleures performances. Les résultats montrent que l'intervalle de seuil [0.7, 0.8] est un bon compromis pour le rappel et la précision du système.

Scénario Tourisme

Nous évaluons notre système DAMaS sur un scénario dans le domaine de tourisme en montrant le cycle de vie complet de notre approche.

Supposons que le touriste Jacques est intéressé par voir les images des différentes attractions d'une région ou d'une ville donnée. Supposons que ces images sont publiées par la compagnie "Tripadvisor" au format "jpeg" et que les services MaaS, présentés dans le **Tableau 14**, sont accessible à Jacques. Les

RapportGratuit.com

services MaaS dans le domaine du tourisme, permettent d'accéder à des ressources de données multimédias de natures différentes et publiées par différents fournisseurs.

Le système DAMaS permet à Jacques de vérifier si des services MaaS disponibles répondent à ses besoins. Dans son module de description, les services MaaS sont annotés en se basant sur les deux types d'ontologies : métier et multimédia. Pour ce scénario, nous avons développé une ontologie métier dans le domaine de tourisme que nous avons baptisé *TourismOnto*. Un extrait de cette ontologie est présenté dans la **Figure 49** qui illustre ses principaux concepts et relations. Pour l'ontologie multimédia, nous avons utilisé l'ontologie des ressources multimédia pour annoter les aspects multimédias des services MaaS. Un extrait de cette ontologie est déjà illustré dans la **Figure 34** du chapitre 6. La troisième, la cinquième et la huitième colonne du **Tableau 14** présentent le résultat de la phase de description sémantique des services MaaS du scénario. Elles représentent respectivement les annotations métiers des entrées, les annotations métiers des sorties et les annotations multimédias des objets multimédias fournis par les services MaaS. Ces dernières annotations forment la requête SPARQL qui représente la partie multimédia des services MaaS.

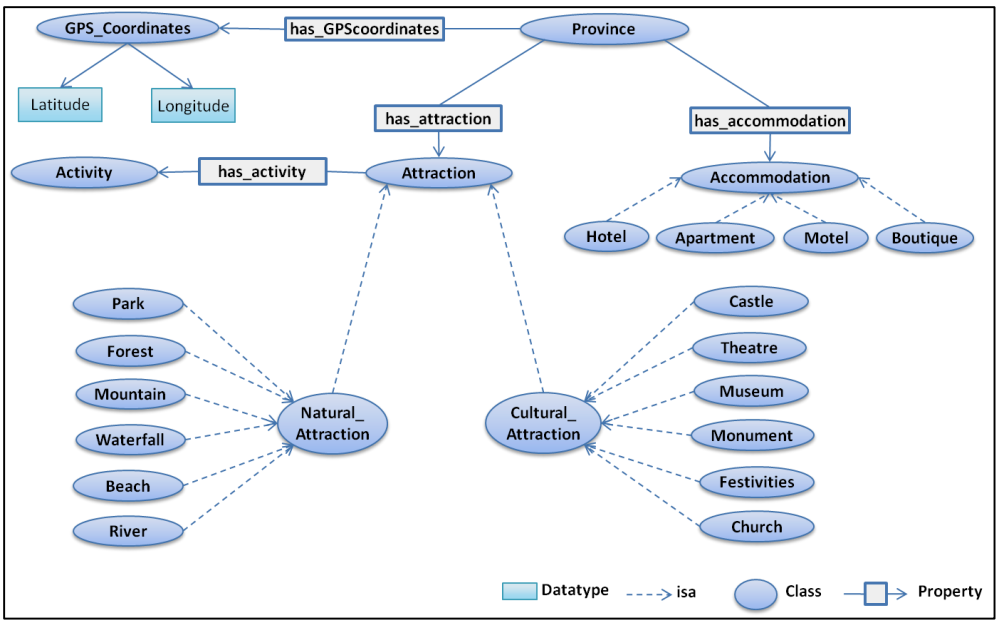

Figure 49 : L'ontologie métier TourismOnto.

De la même façon, la requête de l'utilisateur Jacques est annotée en se basant sur les deux types d'ontologies (métier et multimédia). Le résultat de cette annotation est présenté au **Tableau 13**.

Requête	Concepts d'entrées	Concepts de sorties	Concepts multimédia
Q	Province	Attraction	**MediaResource :** Image **Publisher :** Tripadvisor **Format :** jpeg

Tableau 13 : La requête annotée du scenario de tourisme.

Dans le module d'appariement de DAMaS, les annotations métiers et multimédias des services MaaS sont comparées avec les annotations métiers et multimédias de la requête. Dans un premier temps, les similarités d'entrées et de sorties de chaque service MaaS sont calculées par l'algorithme métier du module d'appariement. La quatrième, la sixième et la septième colonne du **Tableau 14** représentent respectivement les similarités d'entrées, les similarités des sorties et les similarités globales des services MaaS par rapport à la requête de Jacques. La similarité globale de chaque service est comparée avec le seuil de l'utilisateur. Supposons que Jacques a fixé un seuil de 0.7 (θ=0.7), c'est-à-dire il ne veut garder que les services qui ont une mesure de similarité globale supérieure ou égale à 0.7. En conséquence, le résultat logique (raisonnement humain) que l'on devrait avoir est représenté par la liste suivante des services : $\{S_1, S_4, S_6, S_7, S_{10}\}$.

Le système DAMaS, après l'exécution de l'algorithme d'appariement métier, retient les services suivants : $\{S_1, S_2, S_3, S_4, S_7, S_{10}\}$. Après, l'algorithme d'appariement multimédia (en se basant sur le principe d'inclusion) filtre cette dernière liste des services pour ne garder à la fin que les services qui satisfont la requête à la fois dans sa partie métier et sa partie multimédia. En conséquence, les services retenus par le système DAMaS après l'exécution de l'algorithme multimédia sont : $\{S_1, S_4, S_7, S_{10}\}$. Les services de cette dernière liste correspondent aux services qui satisfont la requête de Jacques à la fois dans sa partie métier et sa partie multimédia.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Services	Fonctionnalités	Concepts d'entrées	SIM_I	Concepts de sorties	SIM_O	SIM	Annotations multimédias
S_1	Retourne des images illustrant les différentes montagnes d'une province donnée	Province	**1**	Mountains	**0.4**	**0.7**	**MediaResource :** Image **Publisher :** Tripadvisor **Format :** jpeg
S_2	Retourne des vidéos qui montrent les différentes montagnes d'une province donnée	Province	**1**	Mountains	**0.4**	**0.7**	**MediaResource :** Video **Publisher :** Tripadvisor **Format :** mpeg
S_3	Retourne des fichiers audios discutant les différentes attractions naturelles d'une province donnée	Province	**1**	Natural_ Attractions	**0.8**	**0.9**	**MediaResource :** Audio **Publisher :** Youtube **Format :** wav
S_4	Retourne des images qui visualisent les différentes attractions culturelles d'une zone limitée par leurs coordonnées GPS	GPSCoordinates	**0.6**	Cultural_ Attractions	**0.8**	**0.7**	**MediaResource :** Image **Publisher :** Tripadvisor **Format :** jpeg
S_5	Retourne des images visualisant les différents hotels d'une région limitée par leurs coordonnées GPS	GPSCoordinates	**0.6**	Hotels	**0**	**0.3**	**MediaResource :** Image **Publisher :** Booking **Format :** jpeg
S_6	Retourne des images des musées d'une région limitée par leurs coordonnées GPS	GPSCoordinates	**0.6**	Museums	**0.4**	**0.5**	**MediaResource :** Image **Publisher :** Tripadvisor **Format :** jpeg
S_7	Retourne des images illustrant les différentes cascades d'une province donnée	Province	**1**	Waterfalls	**0.4**	**0.7**	**MediaResource :** Image **Publisher :** Tripadvisor **Format :** jpeg
S_8	Retourne des images qui montrent les différents hébergements possibles d'une province donnée	Province	**1**	Accommodations	**0**	**0.5**	**MediaResource :** Image **Publisher :** Tripadvisor **Format :** jpeg
S_9	Retourne des images des différentes boutiques d'une province donnée	Province	**1**	Boutique	**0**	**0.5**	**MediaResource :** Image **Publisher :** Booking **Format :** jpeg
S_{10}	Retourne des images des différentes attractions culturelles d'une province donnée	Province	**1**	Cultural_ Attractions	**0.8**	**0.9**	**MediaResource :** Image **Publisher :** Tripadvisor **Format :** jpeg
S_{11}	Retourne des images qui visualisent les différentes activités touristiques d'une zone limitée par leurs coordonnées GPS	GPSCoordinates	**0.6**	Activity	**0.6**	**0.6**	**MediaResource :** Image **Publisher :** Tripadvisor **Format :** jpeg

Tableau 14 : Les services MaaS du scenario de tourisme.

L'évaluation de notre système DAMaS avec le scénario de tourisme, consiste à comparer dans un premier temps, comme nous l'avons fait avec le scénario médical, les deux variantes de notre approche. Dans un deuxième temps, nous comparons les deux variantes de notre approche avec l'approche SAWSDL-MX (plus exactement sa variante SAWSDL-M0). La première variante de notre approche représente l'approche sans l'utilisation de l'appariement multimédia et la deuxième variante représente l'approche avec l'utilisation des deux types d'appariement (métier et multimédia). Cette évaluation offre un double avantage : d'une part tester notre système sur une nouvelle base de service dans un autre domaine afin de valider et confirmer les premiers résultats obtenus dans le premier scénario, et d'autre part comparer notre système avec une approche répandue de l'état de l'art (en l'occurrence SAWSDL-MX). Le **Tableau 15**, le **Tableau 16** et le **Tableau 17** présentent les résultats de Rappel, Précision et F-Mesure des deux variantes de notre approche et de l'approche SAWSDL-MX sur le scénario de tourisme en variant le seuil de l'utilisateur θ à 0.5, 0.6, 0.7, 0.8, 0.9 et 1.

Mesures / Seuil θ	Rappel	Précision	F-Mesure
θ = 0.5	0.95	0.50	0.65
θ = 0.6	0.80	0.57	0.66
θ = 0.7	0.64	0.60	0.61
θ = 0.8	0.40	0.55	0.46
θ = 0.9	0.30	0.53	0.38
θ = 1	0.20	0.56	0.29

Tableau 15 : Rappel, Précision et F-Mesure de la variante 1 (Tourisme).

Mesures / Seuil θ	Rappel	Précision	F-Mesure
θ = 0.5	1	0.71	0.83
θ = 0.6	0.80	0.85	0.82
θ = 0.7	0.68	1	0.80
θ = 0.8	0.35	1	0.51
θ = 0.9	0.28	1	0.43
θ = 1	0.23	1	0.37

Tableau 16 : Rappel, Précision et F-Mesure de la variante 2 (Tourisme).

Mesures / Seuil θ	Rappel	Précision	F-Mesure
θ = 0.5	0.72	0.63	0.67
θ = 0.6	0.62	0.60	0.61
θ = 0.7	0.45	0.55	0.49
θ = 0.8	0.25	0.50	0.34
θ = 0.9	0.05	0.05	0.05
θ = 1	0	0	0

Tableau 17 : Rappel, Précision et F-Mesure de SAWSDL-M0 (Tourisme).

Les figures **Figure 50**, **Figure 51** et **Figure 52** montrent la représentation graphique des mesures de performance (rappel, précision et F-Mesure) des deux variantes de notre système et de la variante SAWSDL-M0 de l'approche SAWSDL-MX sur le scénario de tourisme.

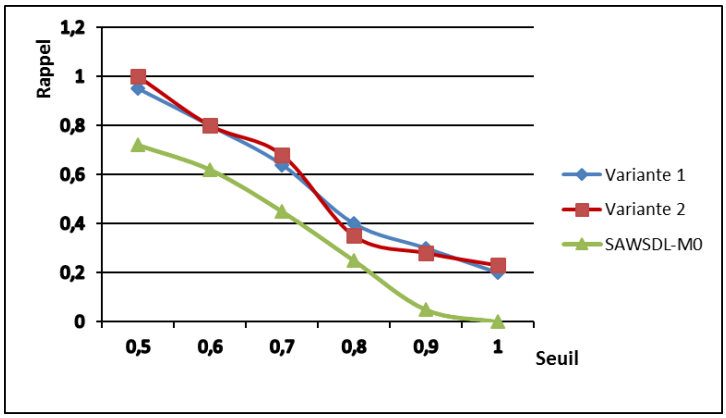

Figure 50 : Le graphe de rappel des deux variantes de DAMaS (Tourisme).

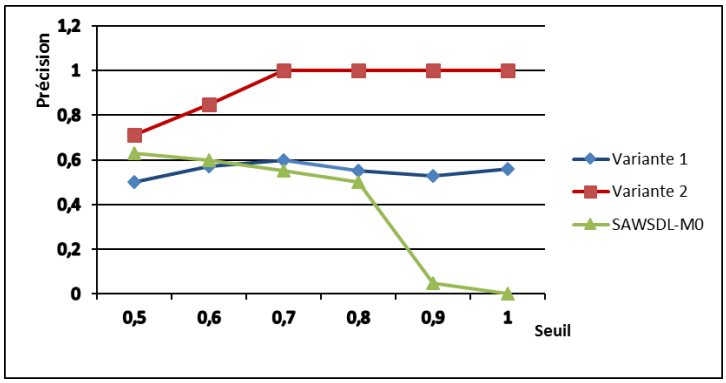

Figure 51 : Le graphe de précision des deux variantes de DAMaS (Tourisme).

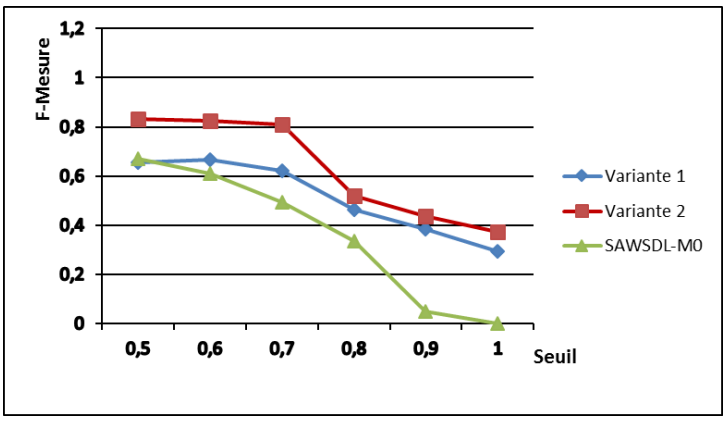

Figure 52 : Le graphe de F-Mesure des deux variantes de DAMaS (Tourisme).

A partir de ces graphes, nous confirmons les résultats obtenus lors de notre évaluation du système sur le scénario médical. Nous remarquons aussi que les deux variantes ont des performances similaires en termes de rappel et une large supériorité de la deuxième variante par rapport à la première en termes de précision. Cette situation, comme nous l'avons déjà précisé, est causée par l'élimination des résultats non pertinents récupérés par le système lorsqu'on utilise les deux appariements ensemble (métier et multimédia). Ces observations

sont aussi confirmées par le graphe de F-Mesure présenté dans la **Figure 52** qui montre que la deuxième variante offre des meilleures performances en termes de rappel et de précision.

Concernant la comparaison de notre approche avec l'approche SAWSDL-MX, nous remarquons dans les figures précédentes une large supériorité de notre approche (variante 1 & 2) par rapport à l'approche SAWSDL-MX en termes de rappel et de précision. Cela est justifié par les points suivants :

- Le mécanisme d'appariement de SAWSDL-MX calcule en même moment le degré d'appariement entre les concepts d'entrées et de sorties avec les concepts d'entrées et de sorties de la requête. Par contre, nous proposons dans notre mécanisme d'appariement pour le calcul de similarité globale de réaliser deux appariements indépendants : un appariement pour les entrées pour calculer la similarité d'entrée et un autre appariement pour les sorties pour calculer la similarité de sortie. Ce mécanisme de séparation entre le calcul de similarité d'entrée et de sortie rend l'appariement plus flexible et permet d'avoir des résultats plus larges que ceux de SAWSDL-MX.
- Les filtres logiques dans SAWSDL-MX sont trop rigides et trop strictes par rapport à nos filtres logiques. Nous avons rajouté des nouveaux filtres logiques dans notre approche permettant de couvrir plus de relations possibles entre les concepts d'entrées et de sorties des services et de la requête utilisateur. Par exemple, le service S_4 qui est rejeté par SAWSDL-MX, est accepté par notre approche avec un degré logique *Has_Relation* entre le concept d'entrée du service et celui de la requête. Cette relation permet d'avoir une similarité globale de 0.7 au lieu de 0 (dans SAWSDL-MX) pour le service S_4 qui est considéré comme pertinent pour la requête de l'utilisateur.

8.4 Conclusion

Dans ce chapitre, nous avons présenté le prototype DAMaS qui nous a permis de valider nos propositions de description et de découverte sémantique des services web multimédias (services MaaS). L'architecture proposée se compose de deux modules : module de description et module d'appariement. Le premier module qui se base sur notre langage SA4MaaS (proposé en chapitre 6) permet la description et l'annotation sémantique des services MaaS. Le deuxième module, qui concerne la découverte et l'appariement sémantique des services MaaS, repose sur notre matchmaker MaaS-MX (proposé en chapitre 7) et implémente ses deux algorithmes proposés : algorithme métier et algorithme multimédia.

Nous avons également effectué un ensemble d'expérimentation à travers deux scénarios relevant de deux domaines différents : la médecine et le tourisme. Les résultats obtenus montrent l'intérêt d'étendre les langages de description et de découverte des services web actuels, par l'ajout des annotations multimédias et des mécanismes d'appariement appropriés, pour accroitre et améliorer les performances des systèmes de recherche d'information multimédia surtout en termes de précision.

CHAPITRE 9

CONCLUSION ET PERSPECTIVES

9.1 Résumé des contributions

9.2 Perspectives

9.1 Résumé des contributions

Avec l'immense prolifération des sources de données multimédias sur le web et le besoin d'intégrer la sémantique dans les systèmes de recherche d'information, les utilisateurs d'aujourd'hui (du web) ont besoin de nouveaux moyens et mécanismes pour accéder d'une manière efficace, transparente et appropriée à des données multimédia qui se situent dans un environnement ouvert et distribué. La technologie des services web, l'implantation la plus répandue de l'architecture orientée service (architecture SOA), est utilisée dans le cadre de nos travaux de recherche comme solution à ces problèmes d'intégration et d'interopérabilité de données. Les travaux présentés dans cette thèse s'inscrivent dans ce cadre-là.

Dans la première partie de cette thèse, nous avons présenté la technologie des services web et ses principes, et mis en évidence les services web sémantiques et les services web d'accès aux données (services DaaS). Nous avons aussi présenté les travaux d'état de l'art autour de la description et la découverte des services web, les deux étapes clés dans leur cycle de vie. Nous avons regroupés les travaux de description en deux grandes classes : syntaxique et sémantique. La classe sémantique est à leur tour divisée en deux types de travaux : les travaux de description basés sur les langages sémantiques (OWL-S et WSMO) et les travaux de description à base d'annotations (WSDL-S, USDL et SAWSDL). Par ailleurs, les travaux de découverte sont regroupés en trois grandes classes : non logique, logique et hybride.

L'étude de l'état de l'art nous a permis de mieux comprendre la problématique d'un point de vue système d'annotation et appariement sémantiques. Afin d'améliorer la description, l'appariement et la découverte de services web, il y a des manques à combler d'une part au niveau de la spécification de la nature de l'annotation sémantique dans certaines approches, et d'autre part, dans la prise en charge de différentes facettes de la sémantique dans les mécanismes de l'appariement. Ces manques engendrent des imprécisions lors de l'appariement et de la découverte.

Pour pallier ces manques, nous avons proposé l'approche **SeSaM** (**Se**mantic **Se**arch of **M**ultimedia content) pour la recherche de contenus multimédias en se basant sur une architecture services web (cette contribution est décrite dans le chapitre 5). SeSaM repose sur la définition d'un nouveau type de services web accédant à des contenus multimédias. Nous avons baptisé ce nouveau type de service, les services MaaS (acronyme de Multimédia as a Services). La représentation sémantique des services MaaS couvre deux types de connaissances : des connaissances métiers qui représentent la logique métier des

Rapport-Gratuit.com

services et des connaissances multimédias qui symbolisent les propriétés multimédias des services. Dans l'approche SeSaM, les sources de données sont interrogées par un ensemble de services MaaS dont le but est de trouver une solution (une combinaison de services MaaS) qui répond au mieux à la requête de l'utilisateur.

Le fonctionnement de l'approche SeSaM se base sur un processus en deux étapes principales : description et filtrage. SeSaM s'appuie sur des concepts abordés dans la littérature traitant la description et la découverte de services mais elle propose de les adapter de façon à prendre en charge les aspects métiers et multimédias des services MaaS. La description et la découverte des services MaaS sont détaillées respectivement dans les chapitres 6 et 7 et font l'objet de la deuxième et la troisième contribution de cette thèse.

Notre seconde contribution (présentée dans le Chapitre 6) a été de définir le langage **SA4MaaS** (Semantic Annotation for MaaS services) pour la description sémantique des services MaaS. Le langage WSDL est une recommandation W3C pour la description des services web. Etant donné que ce langage se limite à une description syntaxique des services web, de nombreux travaux ont été proposés dans la littérature dont l'objectif d'intégrer la sémantique dans la description des services. Nous citons les plus connus, à savoir OWL-S, WSMO et SAWSDL. L'étude comparative de ces travaux menée dans le chapitre 4, nous a permis de constater que le langage SAWSDL présente plusieurs avantages par rapport aux OWL-S et WSMO. De ce fait, SA4MaaS étend SAWSDL pour décrire sémantiquement les services MaaS.

Le principe général de notre extension est l'intégration, en plus de la sémantique métier, de la sémantique des aspects multimédias dans la description des services MaaS. Par conséquent, SA4MaaS se base sur l'utilisation de deux types d'ontologies de nature différente dans la tâche de description. Une première ontologie dite « ontologie métier » contenant des concepts définissant la sémantique métier des services MaaS (e.g : médical, éduction, tourisme, etc.). Une deuxième ontologie dite « ontologie multimédia » contenant des concepts définissant la sémantique des annotations multimédias des services MaaS (e.g : format, location, information de création, etc.). Ce principe permet à notre approche d'avoir plus de précision et d'expressivité au niveau de la description des services MaaS.

Notre troisième contribution (présentée dans le Chapitre 7) est le matchmaker **MaaS-MX** (MaaS services Matchmaker) pour la découverte des services MaaS. L'élément principal d'un processus de découverte est le mécanisme d'appariement (matching). Ce mécanisme reçoit en entrée la requête

et un ensemble de services web et il retourne comme résultat les services web pertinents, classés selon leurs degrés d'appariement. Plusieurs approches de découverte ont été proposées dans la littérature dont l'objectif d'identifier les services web pertinents. Nous avons classé ces approches selon leurs méthodes d'appariement en trois grandes classes : non logique, logique et hybride. L'étude comparative de ces approches menée dans le chapitre 5, nous a permis de constater que les approches hybrides présente plusieurs avantages par rapport aux approches logiques et non logiques. Ces approches hybrides cherchent à remédier à certaines limites des deux approches non logiques et logiques en combinant leurs techniques et leurs mécanismes.

Vu les avantages des approches hybrides, nous nous sommes inspirés de ces approches pour définir notre nouvelle approche de découverte des services MaaS (le matchmaker MaaS-MX). La découverte repose sur la structure du modèle de description des services web et de la requête utilisateur. Dans le cadre de l'approche SeSaM, les services MaaS et la requête utilisateur sont décrits par le langage SA4MaaS d'où la nécessité de définir une nouvelle approche de découverte adaptée à ce nouveau modèle de description. Par conséquent, notre matchmaker est composé de deux étapes primordiales : appariement métier et appariement multimédia. L'appariement métier consiste à comparer la description métier des services et de la requête, tandis que l'appariement multimédia compare la description multimédia des services et de la requête. Nous avons implémenté deux algorithmes pour le matchmaker MaaS-MX, le premier pour l'appariement métier et le deuxième pour l'appariement multimédia.

Notre quatrième contribution (présentée dans le Chapitre 8) est l'implémentation et l'évaluation de notre approche. SeSaM a été évaluée dans deux domaines différents : médical et tourisme. Les résultats ont montré que l'intégration de la sémantique multimédia dans la description et la découverte des services web multimédia améliore les performances d'un système de recherche d'information multimédia.

9.2 Perspectives

A la fin de cette thèse, nous dégageons un ensemble de perspectives futures permettant d'améliorer et d'enrichir nos différentes propositions.

Une première perspective consiste à étendre l'approche SeSaM en lui ajoutant une phase de restitution. Cette phase a pour objectif de restituer à l'utilisateur ce que les services MaaS fournissent comme données à la fin de la phase d'appariement. Nous pensons à la construction des clusters homogènes afin de fournir à l'utilisateur des résultats classés et cohérents. Deux types de

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

clustering peuvent être envisagés : clustering **structurel** et/ou clustering **sémantique**. On peut soit choisir l'un des deux types, soit les enchainer pour mettre les services MaaS semblables ensemble. Pour le premier type de clustering, il s'agit tout simplement de créer des types de clusters selon le type de données retournées par les services MaaS (Image, Vidéo ou Audio). Pour le deuxième type de clustering, les services MaaS sont divisés en clusters selon la sémantique des services en se basant sur leur similarité sémantique. La similarité entre deux services MaaS est calculée en appliquant notre même mécanisme de matching sémantique entre les concepts métiers et multimédias annotant ces deux services. Les services à comparer sont les services résultats de la phase de filtrage (appariement), qui sont les services les plus proches de la requête utilisateur.

Une deuxième perspective concerne la catégorisation des services MaaS. Cette catégorisation permettra d'affecter automatiquement les services MaaS à des catégories prédéfinies ($MaaS_{Image}$, $MaaS_{Video}$, $MaaS_{Audio}$) en se basant sur l'ontologie multimédia pour détecter le type de MaaS. Nous pensons que cette catégorisation facilitera la présentation des données multimédias fournies par les services MaaS.

Une troisième perspective de mes travaux est liée à l'annotation des services MaaS. Nous avons supposé que les annotations métiers et multimédias sont faites par un utilisateur expert. La qualité de ces annotations dépend grandement de l'expertise de l'utilisateur. Nous pensons que l'association de cette expertise à des traitements semi-automatiques facilite cette tâche et représente un gain de temps. C'est pourquoi nous souhaitons assouplir cette annotation manuelle en offrant à l'utilisateur expert une annotation assistée pour améliorer sa tâche en termes de temps et de productivité. L'idée est d'assister l'utilisateur expert tout au long sa tâche d'annotation en lui proposant des annotations préalablement définies. Par exemple, on peut lui proposer les annotations les plus fréquentes utilisées ou les dernières annotations utilisées.

Une quatrième perspective est liée au passage à l'échelle. Notre collection de services actuelle est composée d'une dizaine de services MaaS soit dans le domaine médical ou tourisme. Etant donné que nous avons défini un nouveau pattern de services (services MaaS), il était difficile de trouver une collection importante de services qui colle bien avec notre modèle de description de services MaaS. Nous souhaitons évaluer la scalabilité sur une collection de services plus large pour s'assurer que les résultats restent satisfaisants. Nous pensons à adapter des collections de services réelles (comme SAWSDL-TC) à notre modèle de description de services MaaS pour faire les tests.

Liste des publications

- **Journaux internationaux avec comité de lecture et indexés par Thomson**

  Sid-Ahmed-Djallal Midouni, Youssef Amghar, Azeddine Chikh: Multimedia Data Retrieving based on SOA Architecture. J. Web Eng. 15(3&4): 339-360 (2016). (Impact Factor= 0,622).

  http://www.rintonpress.com/journals/jweonline.html#v15n34

- **Journaux internationaux avec comité de lecture et indexés par Scopus**

  Sid-Ahmed-Djallal Midouni, Youssef Amghar, Azeddine Chikh: A semantic annotation for MaaS services: the SA4MaaS approach. IJMSO 11(2): 82-92 (2016).

  http://www.inderscience.com/info/inarticletoc.php?jcode=ijmso&year=2016&vol=11&issue=2

- **Conférences internationales avec comité de lecture**

  MIDOUNI, Sid Ahmed Djallal, AMGHAR, Youssef, et CHIKH, Azeddine. A Full Service Approach for Multimedia Content Retrieval. In: Model and Data Engineering. Springer International Publishing, 2014. p. 125-137.

  A.K.Y Settouti, T.Taleb, S.A.D.Midouni, SPARQL based Full-Text Search Added Filter Function to Jena, World Symposium on Computer Applications & Research (WSCAR 2014), Sousse-Tunisia, January 2014.

  F.Hadjila, M-A Chikh, S.A.D.Midouni, M.Benazzouz, Planification des services web sémantique en utilisant l'indexation conceptuelle des entrées sorties, 5ième International Symposium on Distance Education, Tunis, Août 2009.

  S.A.D.Midouni, J.Darmont, F.Bentayeb, "Approche de modélisation multidimensionnelle des données complexes : Application aux données médicales", 5èmes Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 09), Montpellier, Juin 2009; Revues des Nouvelles Technologies de l'Information, Vol. B-5, Cépaduès Editions, Toulouse, 155-166.

  S.A.D.Midouni, A.Chikh, "Modèle multidimensionnel des données complexes, cas des activités d'apprentissage dans un dispositif de formation en ligne", Environnements d'Apprentissage pour les Communautés de Pratique (EACP'08), Colloque International, Tlemcen, Novembre 2008.

Bibliographie

Akkiraju, Rama, et al. 2005. **Web Service Semantics - WSDL-S.** ***World Wide Web Consortium.*** **[En ligne] November 2005. http://www.w3.org/Submission/WSDL-S/.**

Aljoumaa, Kadan, Assar, Saïd et Souveyet, Carine. 2011. **Publishing Intentional Services using extended semantic annotation.** ***Research Challenges in Information Science (RCIS), 2011 Fifth International Conference on.*** **IEEE, 2011, pp. 1-9.**

Almeida, Francisco, et al. 2009. **IDEWEP: Web service for astronomical parallel image deconvolution.** ***Journal of Network and Computer Applications.*** **Elsevier, 2009, Vol. 32, 1, pp. 293-313.**

Arndt, Richard, et al. 2007. **COMM: designing a well-founded multimedia ontology for the web.** ***6th International Semantic Web Conference ISWC'2007.*** **Springer Berlin Heidelberg, 2007, pp. 30-43.**

BANSAL, Ajay, et al. 2005. **A universal service-semantics description language.** ***Third IEEE European Conference On Web Services (ECOWS).*** **2005, pp. 214-225.**

Barry, Douglas K. 2003. ***Web services, service-oriented architectures, and cloud computing.*** **[éd.] Morgan Kaufmann. 2003.**

Berners-Lee, Tim, et al. 2001. **The semantic web.** ***Scientific american.*** **New York, NY, USA:, 2001, Vol. 284, 5, pp. 28-37.**

Bianchini, Devis, et al. 2006. **Semantic-enriched service discovery.** ***22nd International Conference on Data Engineering Workshops (ICDEW'06).*** **IEEE, 2006.**

Booth, David, et al. 2004. **Web Services Architecture.** ***W3C Working Group Note.*** **[En ligne] Février 2004. https://www.w3.org/TR/ws-arch/.**

Botelho, Luis, et al. 2008. **Service discovery. [éd.] Michael Schumacher, Heikki Helin et Heiko Schuldt.** ***CASCOM: Intelligent Service Coordination in the Semantic Web.*** **Birkhäuser Basel, 2008, pp. 205-233.**

Box, Don, et al. 2007. **Simple Object Access Protocol (SOAP) 1.2.** ***W3C Recommendation (Second Edition).*** **[En ligne] 27 Avril 2007. https://www.w3.org/TR/soap/.**

Bray, Tim, et al. 2008. **Extensible Markup Language (XML).** ***W3C Recommendation.*** **[En ligne] 28 Novembre 2008. https://www.w3.org/TR/xml/.**

Brut, Mihaela, et al. 2011. **Adapting Indexation to the Content, Context and Queries Characteristics in Distributed Multimedia Systems.** ***Signal-Image Technology and Internet-Based Systems (SITIS), 2011 Seventh International Conference on.*** **IEEE, 2011, pp. 118-125.**

Carey, Michael J, Onose, Nicola et Petropoulos, Michalis. 2012. **Data services.** ***Communications of the ACM.*** **ACM, 2012, Vol. 55, 6, pp. 86-97.**

—. 2012. **Data Services.** ***Communications of the ACM.*** **2012, Vol. 55, 6, pp. 86-97.**

Chabeb, Yassin et Tata, Samir. 2008. **Yet another semantic annotation for WSDL.** ***IADIS International Conference WWW/Internet.*** **2008, pp. 437-441.**

Chabeb, Yassin, Tata, Samir et Belaïd, Djamel. 2009. **Toward an integrated ontology for web services.** ***Internet and Web Applications and Services, 2009. ICIW'09. Fourth International Conference on.*** **IEEE, 2009, pp. 462-467.**

Chandra, Ashok K et Merlin, Philip M. 1977. **Optimal implementation of conjunctive queries in relational data bases.** ***Proceedings of the ninth annual ACM symposium on Theory of computing.*** **ACM, 1977, pp. 77-90.**

Chinnici, Roberto, et al. 2007. **Web services description language (wsdl) version 2.0.** ***World Wide Web Consortium.*** **[En ligne] June 2007. http://www.w3.org/TR/wsdl20/.**

Christensen, Erik, et al. 2001. **Web Services Description Language (WSDL) 1.1.** ***World Wide Web Consortium.*** **[En ligne] March 2001. http://www.w3.org/TR/wsdl.**

Christensen, Kim, Olesen, Thorbjørn Højgaard et Thomsen, Lone Leth. 2015. **Matching semantically described web services using ontologies.** ***Information Technology And Control.*** **2015, Vol. 35, 3.**

De Bruijn, Jos, et al. 2006. ***The web service modeling language WSML: an overview.*** **s.l. : Springer, 2006.**

De Bruijn, Jos, et al. 2005. **Web service modeling ontology (WSMO).** ***World Wide Web Consortium.*** **[En ligne] June 2005. http://www.w3.org/Submission/WSMO/.**

Erl, Thomas. 2008. ***SOA Principles of Service Design.*** **s.l. : Prentice Hall Upper Saddle River, 2008. Vol. 1.**

ESSI WSMO working group. 2004. **Web Service Modeling Ontology (WSMO). [En ligne] 2004. http://www.wsmo.org.**

Farrell, Joel et Lausen, Holger. 2007. **Semantic Annotations for WSDL and XML Schema.** ***World Wide Web Consortium.*** **[En ligne] Août 2007. http://www.w3.org/TR/sawsdl/.**

Fellbaum, Christiane. 1998. **WordNet. Wiley Online Library, 1998.**

Fensel, Dieter et Bussler, Christoph. 2002. **The web service modeling framework WSMF.** ***Electronic Commerce Research and Applications.*** **Elsevier, 2002, Vol. 1, 2, pp. 113-137.**

Fensel, Dieter, et al. 2010. **WSMO-Lite: Lightweight semantic descriptions for services on the web. W3C Member Submission, 2010, Vol. 23.**

Fielding, Roy, et al. 1999. **Hypertext transfer protocol--HTTP/1.1.** ***W3C recommandation.*** **[En ligne] Juin 1999. https://www.w3.org/Protocols/.**

Fronckowiak, John et Myer, T. 2008. **Processing Images with Amazon Web Services.** ***Sep.*** **2008, Vol. 21, p. 8.**

Giro-i-Nieto, Xavier, et al. 2010. **System architecture of a web service for content-based image retrieval.** ***Proceedings of the ACM International Conference on Image and Video Retrieval.*** **ACM, 2010, pp. 358-365.**

Gruber, Thomas R. 1995. **Toward principles for the design of ontologies used for knowledge sharing.** ***International journal of human-computer studies.*** **Elsevier, 1995, Vol. 43, 5, pp. 907-928.**

Halevy, Alon Y. 2001. **Answering queries using views: A survey. [éd.] Springer.** ***The VLDB Journal.*** **2001, Vol. 10, 4, pp. 270-294.**

Hendler, James et McGuinness, Deborah L. 2000. **The DARPA agent markup language. 2000, Vol. 15, 6, pp. 63-76.**

Kaufer, Frank et Klusch, Matthias. 2006. **WSMO-MX: A Logic Programming Based Hybrid Service Matchmaker.** ***2006 European Conference on Web Services (ECOWS'06).*** **IEEE, 2006, pp. 161-170.**

Keller, Uwe, et al. 2005. **Automatic location of services.** ***The Semantic Web : Research and Applications.*** **Springer Berlin Heidelberg, 2005, pp. 1-16.**

Keller, Uwe, et al. 2004. **Wsmo web service discovery.** ***WSML Working Draft.*** **November 2004, Vol. 5.**

Klein, Michael, Konig-Ries, Birgitta et Mussig, Michael. 2005. **What is needed for semantic service descriptions? A proposal for suitable language constructs.** ***International Journal of Web and Grid Services.*** **Inderscience Publishers, 2005, Vol. 1, 3-4, pp. 328-364.**

Klusch, Matthias et Kapahnke, Patrick. 2008. **Semantic web service selection with SAWSDL-MX. [éd.] Ruben Lara, Tommaso Di Noia et Ioan Toma.** ***2nd International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMRR).*** **CEUR-WS.org, 2008, Vol. 416, pp. 14-28.**

Klusch, Matthias, Fries, Benedikt et Sycara, Katia. 2006. **Automated semantic web service discovery with OWLS-MX.** ***Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems.*** **ACM, 2006, pp. 915-922.**

Klusch, Matthias, Kapahnke, Patrick et Zinnikus, Ingo. 2009. **Hybrid adaptive web service selection with SAWSDL-MX and WSDL-Analyzer.** ***European Semantic Web Conference.*** **Springer, 2009, pp. 550-564.**

Klusch, Matthine. 2008. **Semantic web service coordination.** ***CASCOM: Intelligent Service Coordination in the Semantic Web.*** **Springer, 2008, pp. 59-104.**

Kopecky, Jacek. 2007. **Semantic Annotations for WSDL and XML Schema.** ***W3C track of WWW2007 Conference.*** **Banff, Canada : s.n., May 2007.**

Küster, Ulrich et König-Ries, Birgitta.

—. 2008. **Evaluating semantic web service matchmaking effectiveness based on graded relevance.** ***The 7th International Semantic Web Conference.*** **October 2008, p. 35.**

Küster, Ulrich, et al. 2007b. **Diane: A matchmaking-centered framework for automated service discovery, composition, binding, and invocation on the web.** ***International Journal of Electronic Commerce.*** **2007b, Vol. 12, 2, pp. 41-68.**

Küster, Ulrich, et al. 2007a. **DIANE: an integrated approach to automated service discovery, matchmaking and composition.** ***Proceedings of the 16th international conference on World Wide Web.*** **ACM, 2007a, pp. 1033-1042.**

Lee, WonSuk, et al. 2012. **Ontology for Media Resources 1.0. [En ligne] February 2012. https://www.w3.org/TR/mediaont-10/.**

Lopez-Velasco, Céline. 2009. ***Sélection et composition de services Web pour la génération d'applications adaptées au contexte d'utilisation.*** **UNIVERSITE JOSEPH FOURIER. 2009. Thèse de doctorat.**

M'BARECK, Nomane OULD AHMED. 2008. ***Une approche sémantique pour la description, l'abstraction et l'interconnexion de workflows dans un contexte interorganisationnel.*** **Télécom SudParis & Universite d'Evry-Val d'Essonne. 2008. Thèse de doctorat.**

Maedche, Alexander et Zacharias, Valentin. 2002. **Clustering ontology-based metadata in the semantic web.** ***European Conference on Principles of Data Mining and Knowledge Discovery.*** **Springer, 2002, pp. 348-360.**

Malki, Abdelhamid. 2015. ***Modélisation sémantique du cloud computing: vers une composition de services DaaS à sémantique incertaine.*** **Université Djillali Liabes de Sidi Bel Abbes, Algérie / Université Claude Bernard-Lyon I, France. Lyon1 : s.n., 2015. Thèse de doctorat.**

Martin, David, et al. 2004. **OWL-S: Semantic markup for web services.** ***W3C member submission.*** **2004, Vol. 22, pp. 2007-04.**

Martin, David, Paolucci, Massimo et Wagner, Matthias. 2007. **Towards semantic annotations of web services: Owl-s from the sawsdl perspective.** ***Proceedings of the OWL-S Experiences and Future Developments Workshop at ESWC.*** **Citeseer, 2007.**

Mike Gilpin, Noel Yuhanna, Katie Smillie, Gene Leganza, Randy Heffner et Jost Hoppermann. 2007. ***Information-As-A-Service: What's Behind This Hot New Trend?*** **s.l. : Forrester Research, 2007.**

Miller, George A. 1995. **WordNet: a lexical database for English.** ***Communications of the ACM.*** **ACM, 1995, Vol. 38, 11, pp. 39-41.**

Mrissa, Michael. 2007. ***Médiation Sémantique Orientée contexte pour la composition de services Web.*** **Université Clause Bernard Lyon1. 2007. Thèse de doctorat.**

OASIS and UNCEFACT. 2007. **EBXML | Online Community for ebXML (Electronic Business Using XML) standards.** ***ebXML XML.org.*** **[En ligne] Janvier 2007. http://ebxml.xml.org.**

OASIS UDDI Specification Technical Committee. 2007. **UDDI | Online community for the Universal Description, Discovery, and Integration OASIS Standard.** ***UDDI XML.org.*** **[En ligne] Janvier 2007. http://uddi.xml.org.**

Palathingal, Paul et Chandra, Sandeep. 2004. **Agent approach for service discovery and utilization.** ***System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on.*** **IEEE, 2004.**

Paolucci, Massimo, et al. 2002. **Semantic matching of web services capabilities.** ***The Semantic Web - ISWC 2002.*** **Springer Berlin Heidelberg, 2002, pp. 333-347.**

Papazoglou, Michael P, et al. 2008. **Service-oriented computing: a research roadmap.** ***International Journal of Cooperative Information Systems.*** **World Scientific, 2008, Vol. 17, 02, pp. 223-255.**

Roman, et al. 2005. **Web Service Modeling Ontology.** ***Applied ontology.*** **IOS Press, 2005, Vol. 1, 1, pp. 77-106.**

Rosaci, Domenico et Sarné, Giuseppe ML. 2013. **Recommending multimedia web services in a multi-device environment. Elsevier, 2013, Vol. 38, 2, pp. 198-212.**

Saathoff, Carsten et Scherp, Ansgar. 2009. **M3O: the multimedia metadata ontology.** ***Proceedings of the Workshop on Semantic Multimedia Database Technologies, 10th International Workshop of the Multimedia Metadata Community (SeMuDaTe 2009), Graz, Austria.*** **Citeseer, 2009.**

Schumacher, Michael, Helin, Heikki et Schuldt, Heiko. 2008. **CASCOM: intelligent service coordination in the semantic web. Springer Science & Business Media, 2008.**

Stal, Michael. 2002. **Web services: beyond component-based computing.** ***Communications of the ACM.*** **ACM, 2002, Vol. 45, 10, pp. 71-76.**

Suárez-Figueroa, Mari Carmen, Atemezing, Ghislain Auguste et Corcho, Oscar. 2013. **The landscape of multimedia ontologies in the last decade.** ***Multimedia tools and applications.*** **Springer, 2013, Vol. 62, 2, pp. 377-399.**

Taylor, David A. 1997. ***Object technology: a manager's guide.*** **s.l. : Addison-Wesley Longman Publishing Co., Inc., 1997.**

The Open Group. 2007. **What Is SOA? [En ligne] 2007. https://www.opengroup.org/soa/source-book/soa/soa.htm.**

Toma, Ioan, et al. 2005. **An Evaluation of Discovery approaches in Grid and Web services Environments.** ***NODe/GSEM.*** **2005, pp. 233-247.**

Urrego, GS. 2005. ***Approches linguistiques pour l'ingénierie de besoins fonctionnels et non-fonctionnels.*** **Centre de Recherche en Informatique, University of Paris 1 La Sorbonne. 2005. Thèse de doctorat.**

Vaculín, Roman, et al. 2008. **Modeling and discovery of data providing services.** ***Web Services, 2008. ICWS'08. IEEE International Conference on.*** **IEEE, 2008, pp. 54-61.**

Vitvar, Tomas, et al. 2008. **Wsmo-lite annotations for web services.** ***European Semantic Web Conference.*** **Springer, 2008, pp. 674-689.**

Vitvar, Tomas, et al. 2007. **Wsmo-lite: Lightweight semantic descriptions for services on the web.** ***Web Services, 2007. ECOWS'07. Fifth European Conference on.*** **IEEE, 2007, pp. 77-86.**

W3C Working Group Note. 2004. **Web Services Architecture. [En ligne] February 2004. https://www.w3.org/TR/ws-arch/.**

W3C XML Core Working Group. 2004. **W3C XML Schema. [En ligne] Octobre 2004. https://www.w3.org/XML/Schema.**

Wagner, Matthias et Kellerer, Wolfgang. 2004. **Web services selection for distributed composition of multimedia content.** ***Proceedings of the 12th annual ACM international conference on Multimedia.*** **ACM, 2004, pp. 104-107.**

Wahli, Ueli, et al. 2002. ***Web services wizardry with WebSphere studio application developer.*** **[éd.] IBM Corp. 2002.**

Zinnikus, Ingo, Rupp, Hans-Jörg et Fischer, Klaus. 2006. **Detecting similarities between web service interfaces: The WSDL Analyzer.** ***Interoperability for Enterprise Software and Applications: Proceedings of the Workshops and the Doctorial Symposium of the Second IFAC/IFIP I-ESA International Conference: EI2N, WSI, IS-TSPQ 2006.*** **2006, pp. 145-156.**

Rapport-Gratuit.com

Abstract

Multimedia data sources from various fields (medical, tourism, trade, art and culture, etc.) became essential on the web. Accessing multimedia data in distributed systems poses new challenges due to many system parameters: volume, diversity of interfaces, representation format, location, etc. In addition, the growing needs of users and applications to incorporate semantics in the information retrieval pose new issues. To take into account this new complexity, we are interested in our research of data integration solutions based on web services.

In this thesis, we propose an approach-oriented service for the semantic search of multimedia content. We called this approach SeSaM (**Se**mantic **Sea**rch of **M**ultimedia content). SeSaM is based on the definition of a new pattern of services to access multimedia content, which is the MaaS services (Multimedia as a Services). It is based on a two-phase process: description and discovery of MaaS services. As for the MaaS services description, we have defined the SA4MaaS language (Semantic Annotation for MaaS services), which is an extension of SAWSDL (W3C recommendation). The main idea of this language is the integration, in addition to business domain semantic, of multimedia information semantics in the MaaS services description.

As for the MaaS service discovery, we have proposed a new matchmaker MaaS-MX (MaaS services Matchmaker) adapted to the MaaS services description model. MaaS-MX is composed of two essential steps: domain matching and multimedia matching. Domain matching consists in comparing the business domain description of MaaS services and the query, whereas multimedia matching compares the multimedia description of MaaS services and the query.

The approach has been implemented and evaluated in two different domains: medical and tourism. The results indicate that using both domain and multimedia matching considerably improves the performance of multimedia data retrieving systems.

Keywords: Semantic web services, Information retrieval, Multimedia Data, Web service description, Web service discovery.

Résumé

Les sources de données multimédias provenant de divers domaines (médical, tourisme, commerce, art et culture, etc.) sont devenues incontournables sur le web. L'accès à ces sources multimédias dans les systèmes distribués pose de nouveaux problèmes en raison de nombreux paramètres : volumétrie, diversité des interfaces, format de représentation, localisation, etc. En outre, l'exigence de plus en plus forte des utilisateurs et des applications à vouloir intégrer la sémantique dans la recherche d'information pose de nouvelles questions à résoudre. Pour prendre en compte cette nouvelle complexité, nous nous intéressons dans notre travail de recherche aux solutions d'intégration de données basées sur les services web.

Dans cette thèse, nous proposons une approche orientée service pour la recherche sémantique de contenus multimédia. Nous avons appelé cette approche SeSaM (**Se**mantic **Sea**rch of **M**ultimedia content). SeSaM repose sur la définition d'un nouveau type de services accédant aux contenus multimédias, qui est les services MaaS (Multimedia as a Services). Elle est basée sur un processus en deux phases : description et découverte des services MaaS. En ce qui concerne la description de services MaaS, nous avons défini le langage SA4MaaS (Semantic Annotation for MaaS services), qui est une extension de SAWSDL (recommandation W3C). L'idée principale de ce langage est l'intégration, en plus de la sémantique métier, de la sémantique de l'information multimédia dans la description des services MaaS.

En ce qui concerne la découverte de services MaaS, nous avons proposé un nouveau matchmaker MaaS-MX (MaaS services Matchmaker) adapté au modèle de description des MaaS. MaaS-MX est composé de deux étapes primordiales : appariement métier et appariement multimédia. L'appariement métier consiste à comparer la description métier des services et de la requête, tandis que l'appariement multimédia compare la description multimédia des services et de la requête.

L'approche a été prototypée et évaluée dans deux domaines différents : médical et tourisme. Les résultats indiquent que l'utilisation de l'appariement métier et l'appariement multimédia a considérablement amélioré les performances des systèmes de recherche de données multimédias.

Mots clés : Services web sémantiques, Recherche d'information, Données multimédias, Description de services, Découverte de services.

ملخص

مصادر بيانات الوسائط المتعددة (multimédia) الآتية من مختلف المجالات (الطب، السياحة، التجارة، الفن والثقافة، إلخ) أصبحت أساسية و ضرورية على شبكة الإنترنت. الوصول إلى هذه المصادر الإعلامية في النظم الموزعة يطرح مشاكل جديدة بسبب العديد من المعايير: حجم وتنوع الواجهات، شكل التمثيل، الموقع، إلخ. وبالإضافة إلى ذلك، شرط مزيد من المستخدمين والتطبيقات لدمج معاني الكلمات في البحث عن المعلومات يطرح مشاكل جديدة يجب حلها. لأخذ بالإعتبار هذه الصعوبات الجديدة، نهتم في عملنا هذا إلى حلول تكامل البيانات المستندة على خدمات ويب.

نقترح في هاته الأطروحة، نهج قائم على خدمات الواب للبحث الدلالي على بيانات الوسائط المتعددة. نسمي هذا النهج SeSaM (البحث الدلالي للوسائط المتعددة) . SeSaM يستند على تعريف نوع جديد من خدمات الواب للوصول إلى محتوى الوسائط المتعددة، وهي خدمات MaaS (الوسائط المتعددة كخدمات). SeSaM ترتكز على عملية من مرحلتين: وصف واكتشاف خدمات MaaS. فيما يتعلق بوصف خدمات MaaS، قمنا بتعريف لغة SA4MaaS (تعليق دلالي لخدمات MaaS)، التي تعد امتداداً لSAWSDL (توصية W3C). الفكرة الرئيسية في هاته اللغة هو بالإضافة إلى دلالات تسلسل العمل (sémantique métier)، نضيف دلالات المعلومات المتعددة الوسائط (sémantique multimédia) في وصف خدمات MaaS.

فيما يتعلق باكتشاف خدمات MaaS، اقترحنا طريقة مطابقة اسمها MaaS-MX (مطابقة خدمات MaaS) تنطبق على نموذج وصف خدمات MaaS. MaaS-MX تتكون من خطوتين أساسيتين: مطابقة تسلسل العمل (métier Appariement) ومطابقة الوسائط المتعددة (multimédia Appariement). مطابقة تسلسل العمل تعتمد على مقارنة وصف تسلسل العمل لخدمات MaaS مع وصف تسلسل العمل لطلب المستخدم (terequê)، بينما مطابقة الوسائط المتعددة تعتمد على مقارنة وصف الوسائط المتعددة لخدمات MaaS مع وصف الوسائط المتعددة لطلب المستخدم (terequê).

قمنا بتنفيذ تطبيق على الكمبيوترللنهج المقترح و تم تقييمه في مجالين مختلفين: الطبي والسياحي. تبين النتائج أن استخدام مطابقة تسلسل العمل مع مطابقة الوسائط المتعددة تحسن أداء نظم البحث على بيانات الوسائط المتعددة.

الكلمات الرئيسية: خدمات الويب الدلالي، البحث عن المعلومات، بيانات الوسائط المتعددة، وصف لخدمات الويب ، خدمة الاكتشاف والمطابقة.