



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABOU-BEKR BELKAID – TLEMCCEN

THÈSE

Présentée à :

FACULTE DES SCIENCES – DEPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

DOCTORAT

Spécialité: *Informatique*

Par :

BENOSMAN Amina

Sur le thème

Médiation sémantique de données basée sur les services web : Application dans le domaine médical

Soutenue publiquement le 21/11/2015 à Tlemcen devant le jury composé de :

Mme DIDI Fedoua,	Maître de Conférences A	Université de Tlemcen	Présidente
Mr BOUAMRANE Karim	Professeur	Université d'Oran	Examineur
Mr YAKOUBI Belabess,	Professeur	Université d'Oran	Examineur
Mr BENSLIMANE Sidi Mohammed	Professeur	Université d'Oran	Examineur
Mr CHIKH Mohammed Amine	Professeur	Université de Tlemcen	Directeur de thèse

Laboratoire Génie Biomédical(GBM)

Remerciements

Avant tout, je remercie Allah tout puissant de m'avoir donné la force et la volonté pour achever ce travail.

Il m'est particulièrement agréable d'exprimer ma gratitude envers les personnes qui m'ont apporté leur soutien et qui m'ont aidée de près ou de loin à effectuer ce travail, tout d'abord mes plus sincères remerciements sont adressés à mon directeur de thèse, Monsieur CHIKH Mohammed Amine qui m'a guidée tout au long de la réalisation de cette thèse. A monsieur Ladjel BELLATRECHE, professeur à l'ENSMA / LISI - Université de Poitiers pour ses conseils et ses encouragements.

Mes sincères remerciements aux membres du jury qui nous ont honorés d'accepter de juger ce travail. Je remercie :

Madame Didi Fedoua, professeur à l'université de Tlemcen, qui m'a fait l'honneur d'avoir accepté de présider ce travail.

Monsieur Bouamrane Karim, professeur à l'université d'Oran 1, monsieur Yagoubi Belabbas, professeur à l'université d'Oran et monsieur Benslimane Sidi Mohammed, professeur à l'université de Sidi Belabbas d'avoir accepté d'examiner ce travail.

Enfin, mes plus vifs remerciements à tous les membres du laboratoire de Génie Biomédical.

A mes parents

Mes beaux parents

Mon mari

Mes sœurs et mon frère

Toute ma famille

Et ma belle famille

Tous mes amis

Résumé

Actuellement les systèmes d'intégration sont de plus en plus basés sur la notion d'ontologie. Ainsi, plusieurs applications offrent un accès à leurs sources de données à travers les services web. Les services web sont la nouvelle vague des applications web. Ce sont des applications modulaires, auto-contenues et auto-descriptives qui peuvent être publiées, localisées et invoquées depuis le web.

Les ontologies sont utilisées dans plusieurs applications telles que l'annotation de documents, la gestion de catalogues de composants industriels, l'intégration de données, etc... ceci a donné naissance à la notion de données sémantiques.

Dans ce travail, une architecture pour l'intégration de données est proposée. Cette architecture est basée essentiellement sur les ontologies et les services web, où les services web jouent le rôle d'adaptateurs dans une architecture médiateur-adaptateur. Nous proposons un méta-modèle de service web qui sera connecté au méta-modèle de l'ontologie, ensuite nous proposons d'enregistrer les descriptions de services web dans une base de données sémantique. Ceci permet la découverte, la composition et l'intégration automatique des services. Nous montrons ainsi l'architecture de médiation proposée et enfin son implémentation dans le domaine médical.

Mots clés : web services, ontologie, intégration de données, médiation, adaptateur

Abstract

Currently integration systems are increasingly based on the notion of ontology. Thus, multiple applications provide access to their data sources through web services. Web services are the new wave of web applications. They are modular applications, self-contained and self-describing that can be published, located and invoked from the web.

Ontologies are used in several applications such as document annotation, management of industrial components catalogs, data integration, etc ... this gave birth to the notion of semantic data.

In this work, an approach for data integration is proposed. This approach is based on services and ontologies, where web services play the role of wrappers in mediator-wrapper architecture. First, the web service is modeled and connected to ontology meta-model, and then a registration of services descriptions in a semantic database is provided. This allows automatic discovery, composition and integration of web services. Finally, we implement our proposed mediation architecture in the medical domain.

Key words: web services, ontology, data integration, mediation, wrapper

ملخص

حاليا تعتمد أنظمة الإدماج بشكل متزايد على مفهوم الأنطولوجيا. وبالتالي، توفر تطبيقات متعددة الوصول إلى مصادر البيانات الخاصة بهم من خلال خدمات الويب. هذه الخدمات هي موجة جديدة من تطبيقات الويب. وهي عبارة عن تطبيقات مستقلة، قائمة بذاتها، تستطيع وصف ذاتها، والتي يمكن نشرها على شبكة الانترنت.

تستخدم الأنطولوجيا في العديد من التطبيقات مثل الشرح، إدارة المكونات الصناعية، الدلائل، وتكامل البيانات، الخ ... وهذا أدى إلى ولادة مفهوم البيانات الدلالي.

في هذا العمل، نقترح بنية لتكامل البيانات التي تستند أساسا على مفهوم الأنطولوجيا وخدمات الويب، حيث تلعب خدمات الويب دور محولات في هذه البنية. ونحن نقدم نموذج عاما لخدمة الويب والتي نصلها بنموذج عاما للأنطولوجيا، ثم نقترح تسجيل خدمات الويب في قاعدة بيانات الدلالات. وهذا يسمح للاكتشاف والتكامل التلقائي للخدمات نقدم بنية تكامل البيانات المقترحة وأخيرا تنفيذها في المجال الطبي.

الكلمات الرئيسية: خدمات الويب، الأنطولوجيا، إدماج المعلومات، بنية، محول.

Table des matières

<i>Remerciements</i>	ii
<i>Résumé</i>	v
<i>Abstract</i>	vi
Table des matières	viii
Table des figures	xii
Liste des tableaux	xiii
Chapitre 1 : Introduction générale	1
1.1. Contexte.....	2
1.2. Problématique.....	2
1.3. Objectifs et contributions	4
1.4. Organisation de la thèse.....	5
1.5. Production scientifique	6
Partie I :.....	9
Etat de l’art	9
Chapitre 2 : Ontologies	10
2.1. Introduction	11
2.2. Définition d’ontologie.....	12
2.3. Composants d’une ontologie.....	12
2.3.1. Les concepts	12
2.3.2. Les relations	13
2.3.3. Les fonctions	13
2.3.4. Les axiomes	13
2.3.5. Les entités ou individus	13
2.4. Classification des ontologies	13
2.4.1. Classification de Guarino	13
2.4.1.1. Les ontologies supérieures	13
2.4.1.2. Les ontologies de domaine	14
2.4.1.3. Ontologies de tâches.....	14
2.4.1.4. Ontologies d’application	14
2.4.2. Classification de Pierra.....	15

2.4.2.1.	Ontologies linguistiques	15
2.4.2.2.	Ontologies conceptuelles.....	15
a.	Ontologies conceptuelles canoniques(OCC).....	15
b.	Ontologies conceptuelles non-canoniques(OCNC).....	16
2.5.	Langages de représentation des ontologies	17
2.5.1.	RDF	17
2.5.2.	RDF Schéma.....	17
2.5.3.	OIL	18
2.5.4.	DAML+OIL	18
2.5.5.	OWL.....	18
2.6.	Représentation formelle des ontologies	20
2.7.	Exemples d'ontologies dans le domaine (bio) médical	21
2.7.1.	L'ontologie OpenGalen.....	21
2.7.2.	Gene ontology	22
2.7.3.	UMLS.....	23
2.7.4.	FMA	24
2.7.5.	OntoMénélas	25
2.7.6.	SNOMED.....	25
2.8.	Conclusion.....	26
Chapitre 3 : Intégration de données		28
3.1.	Introduction	29
3.2.	Hétérogénéité des données.....	30
3.1	Approches d'intégration	31
3.3.1.	Approche entrepôt de données.....	33
3.3.2.	Approche médiation.....	35
3.3.2.1.	Types de mappings	36
-	Global-As-View (GAV),.....	36
-	Local-As-View (LAV).....	37
-	Global-Local-as-View (GLAV).....	37
-	Both as view (BAV)	37
3.3.2.2.	Exemples de projets d'intégration de données par médiation	37
-	Le projet TSIMMIS	37
-	Le système MOMIS (Mediator Environment for Multiple Information Sources)....	38
-	Le système HERMES	39
-	Information manifold.....	39

- Infomaster	39
3.3.3. Intégration de données à base d'ontologies.....	40
3.3.3.1. Architectures d'intégration à base d'ontologies	40
3.3.3.2. Bases de données à base ontologique	42
3.3.3.2.1. Représentation des données à base ontologique	42
➤ <i>Représentation binaire</i>	42
➤ <i>Représentation verticale</i>	43
➤ <i>Représentation horizontale</i>	43
3.3.3.2.2. Architectures des bases de données à base ontologique.....	43
➤ <i>Architecture de type I (deux quarts)</i>	43
➤ <i>Architecture de type II (trois quarts)</i>	44
➤ <i>Architecture de type III (quatre quarts)</i>	44
3.3.3.2.3. Le modèle OntoDB	45
3.3.3.2.4. Le langage OntoQL.....	45
3.4. Conclusion.....	46
Chapitre 4 : Services Web Sémantiques	48
4.1. Introduction.....	49
4.2. Architecture Orientée Services(AOS).....	49
4.3. Services web.....	50
6.2.1. Principaux standards des services web.....	51
i. Langage XML (eXtensible Markup Language) [64].....	51
ii. WSDL	52
iii. SOAP (Simple Object Access Protocol).....	53
iv. UDDI	55
6.2.2. Services web sémantiques.....	57
4.3.2.1. OWL-S.....	57
4.3.2.2. WSMO	58
4.3.2.3. WSDL-S.....	59
6.2.3. Services web pour l'intégration de données.....	59
4.4. Conclusion.....	60
Partie II :	63
Contributions.....	63
Chapitre 5 : Architecture de médiation	64
5.1. Introduction.....	65
5.2. Méta-modèle de service web.....	66

5.3.	Méta-modèle de l'ontologie.....	69
5.4.	Connexion du méta-modèle de service web avec le méta-modèle de l'ontologie	69
5.5.	Extension du méta-modèle en considérant les opérateurs flous.....	71
5.6.	UDDI sémantique	73
5.7.	Architecture globale de médiation	78
5.8.	Conclusion.....	80
Chapitre 6 : Implémentation de l'architecture proposée		84
6.1.	Introduction	85
6.2.	Implémentation.....	85
-	Exemple de motivation.....	86
6.3.	Langages et outils utilisés.....	91
6.4.	Conclusion.....	92
Conclusion générale		94
Annexe : Exemple d'un document WSDL		97
Références Bibliographiques		101
Glossaire.....		111

Table des figures

Figure 2.1 : Classification des ontologies selon Guarino	15
Figure 2.2 : Le modèle en oignon	16
Figure 2.3 : les sous langages d'OWL.....	20
Figure 2.4 : Partie de l'ontologie GO décrivant le métabolisme DNA	23
Figure 3.1 : Architecture générale d'un système d'intégration	32
Figure 3.2 : Architecture d'un entrepôt de données	34
Figure 3.3 : Architecture globale d'un médiateur	36
Figure 3.4: Architecture générale de TSIMMIS	38
Figure 3.5 : Architecture avec une ontologie unique	40
Figure 3.6 : Architecture utilisant plusieurs ontologies.....	41
Figure 3.7 : Architecture hybride	42
Figure 3.8 : Architecture de base de données sémantique de type I (deux quarts).....	43
Figure 3.9. Architecture de base de données sémantique de type II (trois quarts).....	44
Figure 3.10 : Architecture de base de données sémantique de type III (quatre quarts).....	44
Figure 4.1: Modèle général d'une architecture orientée services	50
Figure 4.2 : Méta-modèle de services web selon la description WSDL.....	52
Figure 4.3 : Structure d'un message SOAP.....	54
Figure 4.4 : Structure d'un annuaire UDDI.....	56
Figure 4.5 : Ontologie OWL-S.....	58
Figure 4.6 : Architecture proposée par par Zhu F. et al.....	60
Figure 5.1 : Structure d'un document WSDL	67
Figure 5.2 : Méta-modèle de service web proposé.....	68
Figure 5.3 : Méta-modèle d'ontologie.....	69
Figure 5.4: Méta-modèle de service web connecté au méta-modèle d'ontologie	70
Figure 5.5 : Méta-modèle de services web en prenant en considérations les opérations floues.....	71
Figure 5.6 : Registre sémantique pour services web	74
Figure 5.7 : Sous-ontologie d'OWL-S représentant le profile	76
Figure 5.8 : Sous-ontologie d'OWL-S représentant le Process	77
Figure 5.9 : Architecture globale du médiateur.....	80
Figure 6.1 : Exemple de motivation.....	87
Figure 6.2: Partie de l'ontologie construite.....	91
Figure 6.3 : Interface graphique du médiateur.....	92

Liste des tableaux

Tableau 6.1 : Exemples de services web	87
Tableau 6.2 : Exemples d'extensions d'opérations	89
Tableau 6.3 : Exemples d'extensions de services web.....	89

Chapitre 1 : Introduction générale

Sommaire

1.1.	Contexte.....	2
1.2.	Problématique.....	2
1.3.	Objectifs et contributions	4
1.4.	Organisation de la thèse.....	5
1.5.	Production scientifique	6

1.1. Contexte

La diversité des sources d'information et leur hétérogénéité posent un problème crucial pour les entreprises qui ont besoin d'intégrer leurs données afin de permettre un accès unifié à leurs sources. Beaucoup de solutions d'intégration ont été proposées afin d'assurer l'interopérabilité des systèmes et de résoudre le problème de l'hétérogénéité des sources d'information.

Un système d'intégration de données permet d'offrir à l'utilisateur une vue uniforme et transparente des informations issues de sources hétérogènes et distribuées sans avoir à connaître leur source ou la façon dont elles sont interrogées.

Deux approches majeures existent pour l'intégration de données, l'approche entrepôt de données (approche matérialisée) et l'approche médiation de données (approche virtuelle). Dans l'approche matérialisée, les sources sont migrées vers un entrepôt, cet entrepôt contient donc toutes les données des sources à intégrer, par contre, dans l'approche virtuelle les données restent au niveau des sources, et le médiateur se charge de diviser la requête de l'utilisateur en sous requêtes et les envoyer à des adaptateurs qui se chargent de traduire la requête et renvoyer les résultats au médiateur.

Généralement, une architecture de médiation est composée de trois couches principales, une couche contenant les sources de données, une couche d'adaptateurs, et la couche du médiateur.

Le rôle de l'adaptateur(ou wrapper) est de connecter le médiateur à une source de données, il doit être capable d'accepter une variété de questions posées par le médiateur et les traduire en termes de la source.

1.2. Problématique

Le problème principal des systèmes de médiation est le problème des conflits qui existent entre les sources hétérogènes. La plupart des solutions actuelles reposent sur l'utilisation des ontologies afin d'automatiser le processus d'intégration. Une ontologie joue le rôle de schéma global dans un système d'intégration, l'utilisation croissante des ontologies a amené la communauté des chercheurs à développer un nouveau type de base de données afin de réduire les conflits, il s'agit de bases de données à bases ontologiques(OBDB), où les ontologies et les instances sont stockées au niveau de la même base, elles sont gérées de la

même façon, et peuvent subir les mêmes traitements comme l'insertion, la suppression, la sélection...

D'un autre côté, plusieurs organisations offrent un accès à leurs sources à travers les services web permettant aux clients l'opportunité de créer et manipuler leurs données d'une manière interactive.

Les services web sont la nouvelle vague des applications web. Ce sont des applications modulaires, auto-contenues et auto-descriptives qui peuvent être publiées, localisées et invoquées depuis le web. Ils effectuent des actions allant de simples requêtes à des processus métiers complexes. Une fois qu'un service web est déployé, d'autres applications peuvent le découvrir et l'invoquer. Cette technologie a été largement adoptée dans les systèmes distribués. Plusieurs standards existent pour la description sémantique des services web, ces standards reposent en général sur l'utilisation des ontologies. Le problème posé est le choix du standard afin de décrire les services, comment retrouver les services qui satisfont la requête de l'utilisateur, comment résoudre les conflits qui peuvent exister entre différentes sources d'informations.

Plusieurs approches de médiation considère un service web comme un adaptateur c'est-à-dire il permet de traduire les requêtes et de retourner les résultats, or un service web peut effectuer des traitements sur les données, ces traitements peuvent être nécessaires afin de répondre à la requête de l'utilisateur.

Dans le cadre des travaux de cette thèse nous ciblons le domaine médical où les pratiques médicales sont en évolution rapide ces dernières années s'exerçant d'une manière distribuée et impliquant plusieurs experts de différentes spécialités et services pour la prise d'une décision fiable et robuste, ces experts ont besoin d'accéder à des informations sur les patients d'une manière pertinente faisant appel à différentes sources d'information.

Le problème de l'accès à l'information médicale pertinente pour le diagnostic est lié d'un côté à :

- 1) la dispersion des données médicales sur différentes structures dont les systèmes d'information sont autonomes et hétérogènes, ainsi plusieurs systèmes existants ne sont pas communicants,

D'un autre côté à :

- 2) l'explosion du nombre des sources accessibles via le web et qui évoluent rapidement,

Ceci pose des problèmes d'interopérabilité et nécessite de développer divers outils d'information et d'aide à la décision médicale, destinés aux professionnels de santé afin de faciliter l'accès à l'information médicale, ainsi des solutions d'intégration ont été développées pour résoudre le problème de distribution et de l'hétérogénéité des données.

1.3. Objectifs et contributions

Notre thèse s'inscrit principalement dans l'intégration sémantique de données par médiation qui repose principalement sur la notion d'ontologie, pour cela nous proposons une architecture de médiation où chaque source locale est associée à sa propre ontologie qui étend une ontologie globale (celle du schéma global). Ceci permet de remédier au problème d'hétérogénéité structurelle et sémantique des données.

Des services web jouent le rôle d'adaptateurs entre le médiateur et les sources de données. En plus du rôle d'adaptateur, les services web peuvent effectuer des traitements sur les données.

Les principales contributions de cette thèse sont :

- Un état de l'art détaillé sur les systèmes d'intégration en particulier les systèmes de médiation sémantique et les bases de données à base ontologique
- Proposition d'un méta-modèle générique de service web, où ce méta-modèle est connecté au méta-modèle de l'ontologie
- Extension du méta-modèle en prenant en considération les paramètres flous
- Proposition d'une approche pour un UDDI sémantique utilisant les bases de données à base ontologique
- Proposition d'une architecture de médiation sémantique
- Implémentation de l'architecture de médiation dans le domaine médical

1.4. Organisation de la thèse

Notre thèse est structurée en deux parties : une partie « état de l'art », comportant trois chapitres, dans laquelle un état de l'art est effectué sur les différentes notions autour de notre problématique. La seconde partie « contributions », comporte deux chapitres, elle permet de détailler notre approche pour l'intégration de données et son implémentation.

Partie I : état de l'art

Cette partie présente un état de l'art sur les différentes notions autour de notre problématique permettant une bonne compréhension de nos propositions, et de mieux positionner notre travail par rapport aux travaux dans la littérature. Elle comporte trois chapitres :

- *Chapitre 2 « Ontologies »* : dans ce chapitre nous détaillons les notions de base liées aux ontologies, nous décrivons les différents composants qui constituent une ontologie, les types des ontologies, les langages de description des ontologies, et nous faisons un tour sur quelques ontologies du domaine médical.
- *Chapitre 3 : « Intégration de données »* : présente un état de l'art sur les systèmes d'intégration, en décrivant les différentes classifications des hétérogénéités qui peuvent apparaître lors de l'intégration, les bases de données à base ontologique et les systèmes de médiation, nous citons ainsi quelques travaux de médiation de données.
- *Chapitre 4 : « Services web et services web sémantiques »* : Dans ce chapitre nous détaillons la notion de services web sémantiques en présentant les approches les plus représentatives de ce domaine. Nous commençons par la définition de l'architecture orientée service sur laquelle se basent les services web, nous étudions par la suite les principaux langages des services web sémantiques, et enfin nous présentons quelques travaux utilisant les services web pour l'intégration de données.

Partie II : contributions

La deuxième partie concerne nos contributions majeures, elle comporte deux chapitres :

- *Chapitre 5 « Contributions »* : détaille nos contributions, il est divisé en quatre parties, dans la première partie nous décrivons le méta-modèle de service web proposé, dans la deuxième partie nous présentons l'extension de ce méta-modèle en considérant les opérateurs flous, la troisième partie permet de présenter notre proposition pour un UDDI sémantique qui repose sur les bases de données à base ontologique et dans la quatrième partie nous détaillons l'architecture globale de notre système de médiation.
- *Chapitre 6 « Implémentation de l'architecture proposée »* : détaille l'implémentation de notre architecture de médiation dans le domaine médical. Ce domaine est caractérisé par des changements fréquents, dans lequel les activités s'exercent d'une manière distribuée et impliquant plusieurs professionnels de différentes spécialités et services pour la prise de décision efficace, ces professionnels ont besoin d'accéder à des informations sur les patients de manière pertinente en faisant appel à différentes sources d'information.

1.5. Production scientifique

Journal International

- Benosman, A., Chikh, M. A., & Bellatreche, L. (2014). Semantic Mediation of Medical Data Using Web Services. *Journal of Medical Imaging and Health Informatics*, 4(6), 858-862.

Conférences nationales et internationales

- Benosman, A., Bouchikhi S. & Chikh, M.A.(2015). A UDDI for Semantic Web services registration. 5ème Journée Doctorale de Génie Biomédical.
- Benosman, A. & Chikh, M.A.(2013). Ontologies pour la construction de bases de données médicales. 3^{ème} Journée Doctorale de Génie Biomédical.

Chapitre 1 : Introduction générale

- Benosman, A., Chikh, M.A.(2012) & Boublenza, A.(2012). Semantic Web Services in medical domain. Biomedical Engineering International Conference BIOMEIC'12.
- Benosman, A. & Chikh, M.A.(2012) & Boublenza, A.(2012). Médiation des données pour la composition des services web médicaux. 2ème Journée Doctorale de Génie Biomédical.

Chapitre 1 : Introduction générale

Partie I :

Etat de l'art

Chapitre 2 : Ontologies

Sommaire

2.1.	Introduction	11
2.2.	Définition d'ontologie.....	12
2.3.	Composants d'une ontologie.....	12
2.3.1.	Les concepts	12
2.3.2.	Les relations	13
2.3.3.	Les fonctions	13
2.3.4.	Les axiomes.....	13
2.3.5.	Les entités ou individus.....	13
2.4.	Classification des ontologies	13
2.4.1.	Classification de Guarino	13
2.4.1.1.	Les ontologies supérieures.....	13
2.4.1.2.	Les ontologies de domaine	14
2.4.1.3.	Ontologies de tâches	14
2.4.1.4.	Ontologies d'application.....	14
2.4.2.	Classification de Pierra.....	15
2.4.2.1.	Ontologies linguistiques	15
2.4.2.2.	Ontologies conceptuelles	15
a.	Ontologies conceptuelles canoniques(OCC)	15
b.	Ontologies conceptuelles non-canoniques(OCNC)	16
2.5.	Langages de représentation des ontologies	17
2.5.1.	RDF.....	17
2.5.2.	RDF Schéma	17
2.5.3.	OIL	18
2.5.4.	DAML+OIL	18

2.5.5.	OWL.....	18
2.6.	Représentation formelle des ontologies	20
2.7.	Exemples d'ontologies dans le domaine (bio) médical	21
2.7.1.	L'ontologie OpenGalen.....	21
2.7.2.	Gene ontology	22
2.7.3.	UMLS.....	23
2.7.4.	FMA	24
2.7.5.	OntoMénélas	25
2.7.6.	SNOMED	25
2.8.	Conclusion	26

2.1. Introduction

La notion d'ontologie a été exploitée dans des domaines variés tels que le web sémantique, le traitement de la langue naturelle, le web sémantique, la recherche d'information, l'intégration des données, les web services...etc. Une ontologie permet de représenter explicitement la sémantique des données en réduisant les hétérogénéités qui peuvent apparaître lors de l'intégration. Ainsi, Elles offrent un moyen pour décrire de manière formelle les connaissances d'un domaine particulier, en définissant des concepts et des relations entre ces concepts.

L'ontologie a été définie tout d'abord dans le domaine de la philosophie où on s'intéresse à l'existence, c'est-à-dire les éléments qui peuvent exister, de la nature des objets, des propriétés, des évènements, des processus et des relations dans chaque domaine de réalité. En grec, le mot « ontologie » veut dire « la science de l'être », (onto : l'être ou l'existant et logia: science ou univers).

La plupart des approches d'intégration actuelles reposent sur l'utilisation des ontologies offrant une description structurelle et sémantique des données, ces approches varient dans la manière d'utiliser les ontologies. Plusieurs formes et plusieurs langages existent pour représenter les ontologies ainsi de multiples ontologies ont été développées et se développent encore dans des domaines variés

Dans ce chapitre, nous définissons la notion d'ontologie, les différents composants d'une ontologie, les types d'ontologies, les langages de description qui permettent de construire les ontologies, la formalisation des ontologies et enfin nous donnons quelques exemples d'ontologies dans le domaine médical.

2.2. Définition d'ontologie

La notion d'ontologie existe depuis longtemps, elle a été définie tout d'abord dans le domaine de la philosophie où on s'intéresse à l'existence, c'est-à-dire les éléments qui peuvent exister, de la nature des objets, des propriétés, des événements, des processus et des relations dans chaque domaine de réalité. En grec, le mot « ontologie » veut dire « la science de l'être », (onto : l'être ou l'existant et logia: science ou univers).[1]

Le terme « ontologie » est apparu pour la première fois selon le dictionnaire anglais de Bailey OED (Oxford English Dictionary), en 1721. [14]

Des chercheurs en informatique, plus spécifiquement en intelligence artificielle se sont inspirés de cette notion d'ontologie pour la représentation des connaissances, plusieurs définitions ont été données, la plus célèbre est celle de Gruber [1] qui énonce qu' "une ontologie est une spécification explicite d'une conceptualisation d'un domaine de connaissance ". où le terme "conceptualisation" réfère à un modèle abstrait d'un certain phénomène de la réalité et qui permet d'identifier de manière structurée les concepts pertinents de ce phénomène. L'expression "spécification explicite» signifie que les concepts utilisés et les contraintes sur leur usage sont définis d'une manière explicite.

L'ontologie est utilisée, dans le domaine de l'Ingénierie des Connaissances (IC) et l'Intelligence Artificielle (IA) pour formaliser les connaissances. Dans ce contexte, tout ce qui « existe » peut être « formalisé », par l'utilisation de vocabulaire.

2.3. Composants d'une ontologie

Une ontologie est constituée d'un ensemble de composants de base qui sont : les concepts, les relations, les fonctions, les axiomes et les instances. [7] [11]

2.3.1. Les concepts

Appelés aussi termes ou classes, ils constituent les éléments de base au sein d'une ontologie et correspondent à l'ensemble des individus à représenter. Il représente des objets réels du monde réel, une idée...

2.3.2. Les relations

Les relations représentent les interactions qui peuvent exister entre les concepts présents dans l'ontologie considérée. De manière formelle, ces relations sont définies comme un sous ensemble d'un produit cartésien de n ensembles, c'est à dire $R : C_1 \times C_2 \times \dots \times C_n$. Ces relations permettent de capturer, la structuration ainsi que l'interaction entre les concepts, permettant de représenter une grande partie de la sémantique de l'ontologie.

Ces relations incluent les relations de spécialisation-généralisation (sous classes), relations d'agrégation ou de composition (partie de), relations d'association et d'instanciation.

2.3.3. Les fonctions

Les fonctions sont des cas particuliers des relations dans lesquelles le n ème élément de la relation est défini en fonction des $n-1$ éléments précédents.

De manière formelle, les fonctions sont définies comme suit :

$$F : C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n.$$

2.3.4. Les axiomes

Ce sont des expressions qui sont toujours vrais, permettant de combiner des concepts, des relations et des fonctions afin de définir des règles d'inférences

2.3.5. Les entités ou individus

Ce sont des extensions des concepts de l'ontologie. Ils représentent les éléments qui véhiculent les connaissances du domaine considéré.

2.4. Classification des ontologies

Plusieurs classifications ont été données aux ontologies, parmi lesquelles celle de GUARINO [8][87][88] qui se base sur le niveau de granularité, elles sont classifiées en quatre classes principales.

2.4.1. Classification de guarino

2.4.1.1. Les ontologies supérieures

Appelées aussi ontologies génériques, Upper ou encore Top-level ontologies [9]. Ce sont des ontologies génériques applicables dans des domaines variés. Leur but est l'étude

des catégories des choses qui existent dans le monde comme les concepts de plus haut niveau d'abstraction et de généralité, par exemple : les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés, etc. et qui sont décrits indépendamment d'un domaine particulier. Parmi les d'ontologies de haut niveau : l'ontologie DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [16], SOWA CYC [89].

2.4.1.2. Les ontologies de domaine

Ce sont des ontologies construites pour représenter les connaissances d'un domaine bien particulier. C'est-à-dire que le vocabulaire décrit par une ontologie de domaine est lié à un domaine comme la médecine, le commerce... Les ontologies de domaine constituent des méta-descriptions de la représentation de connaissances du domaine.

La différence qui existe entre les ontologies de domaine et les ontologies de haut niveau est que les concepts dans les ontologies de domaine sont généralement définis comme des spécialisations des concepts qui sont déjà définis dans les ontologies de haut niveau (Figure 2.1) [13] [10].

Plusieurs ontologies de domaine existent, parmi lesquelles l'ontologie MENELAS [91] dans le domaine médical, ENGMATH pour les mathématiques, PhysSys pour le domaine de la Physique [92], etc.

2.4.1.3. Ontologies de tâches

“Task ontologies can be seen as attributing context-specific semantics to domain knowledge elements” [8], ces ontologies sont utilisées pour gérer des tâches spécifiques à un domaine afin de résoudre les problèmes du système, le vocabulaire qu'elles décrivent est relié à une tâche générique telle que les tâches de classification, etc et cela à travers la spécialisation des termes dans les ontologies de haut niveau. Les ontologies de tâche permettent de fournir un vocabulaire systématique des termes utilisés pour résoudre les problèmes liés aux tâches [13]

2.4.1.4. Ontologies d'application

Elles contiennent toutes les définitions nécessaires pour modéliser la connaissance requise pour une application particulière [10]. Ces ontologies sont un mélange des concepts qui sont définis au niveau des ontologies de domaine et des ontologies génériques.

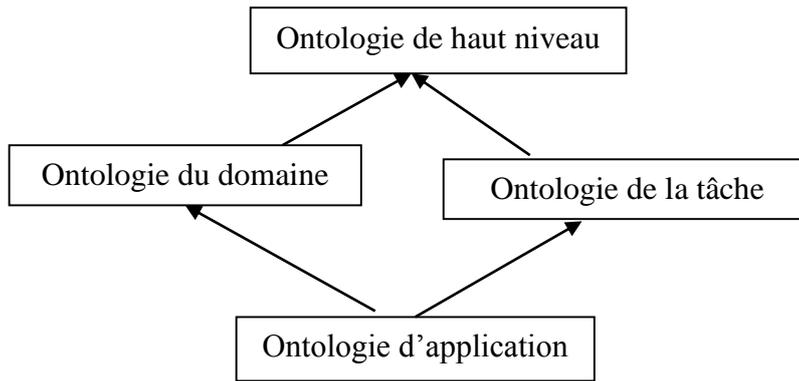


Figure 2. 1 : Classification des ontologies selon Guarino

2.4.2. Classification de Pierra

Une autre classification des ontologies est celle de Pierra [51] qui classe les ontologies en ontologies linguistiques et ontologies conceptuelles, le modèle en oignon (Figure 2.2) permet de représenter cette classification.

2.4.2.1. Ontologies linguistiques

Les ontologies linguistiques ont pour but la représentation de la signification des termes d'un domaine particulier dans un langage précis, permettant de définir le sens des mots et les relations linguistiques (comme les relations de la synonymie, hyponymie...) entre ces mots.

2.4.2.2. Ontologies conceptuelles

Elles permettent de définir de manière formelle les concepts d'un domaine et les relations entre les concepts, elles-mêmes sont catégorisées en deux types : Ontologies conceptuelles canoniques, Ontologies conceptuelles non-canoniques. [97]

a. Ontologies conceptuelles canoniques(OCC)

Elles contiennent des concepts qui sont définis de façon unique, et ne contiennent pas de redondances dans leur définition en incluant seulement les conditions nécessaires. Ces ontologies ne contiennent que des concepts qui sont appelés primitifs. Elles incluent des descriptions complexes et précises des concepts. Par contre, elles n'incluent pas d'opérateurs permettant de définir des équivalences entre les concepts.

Les ontologies de type OCC sont exploitées en général dans les domaines de l'échange de données et des bases de données. Un exemple d'ontologie OCC est l'ontologie IEC [101] portant sur le domaine du commerce électronique.

b. Ontologies conceptuelles non-canoniques(OCNC)

En plus des concepts primitifs, ces ontologies contiennent aussi des concepts définis. Des constructeurs sont utilisés pour exprimer l'équivalence entre les concepts. Ces constructeurs sont très utiles pour définir des mappings entre différentes ontologies. Ce type d'ontologies est utilisé généralement comme schéma global de requêtes. La redondance qu'elles introduisent permet d'augmenter le nombre de concepts en termes desquels il est possible d'exprimer une requête donnée.

OWL est un exemple de modèle d'ontologie qui définit des ontologies conceptuelles non-canoniques.

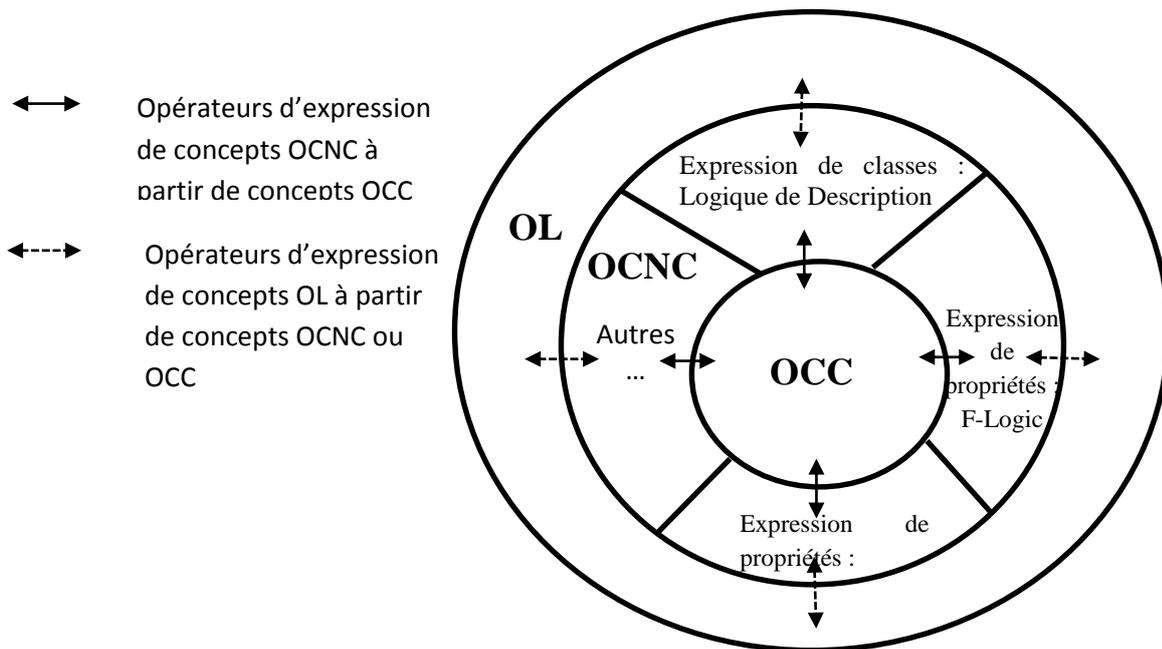


Figure 2.2 : Le modèle en oignon [98]

2.5. Langages de représentation des ontologies

Plusieurs langages ont été développés pour la représentation des ontologies, les premiers datent des années 90, ils se basent sur la logique du premier ordre comme le langage LOOM et KIF [90]. Ensuite la logique du premier ordre a été combinée avec la notion des frames par exemple Ontolingua et OCML, puis d'autres langages ont été créés comme le langage SHOE par la suite les langages qui se basent sur le langage XML sont apparus comme XOL, RDF, RDFS, OIL, DAML-OWL et OWL, dans ce qui suit nous présentons une brève description des langages les plus connus.

2.5.1. RDF

Resource Description Framework est un langage qui représente un modèle conceptuel, abstrait et formel pour la représentation des ressources et les relations entre elles. Il est fondé sur un modèle de graphe, permettant de décrire les éléments de manière simple et sans ambiguïté selon un mécanisme basé sur des déclarations RDF [17] [19]. L'objectif initial de RDF est une bonne représentation et une meilleure exploitation des métadonnées.

Les déclarations RDF sont des triplets (ressource, attribut, valeur) qui peuvent être traitées par la machine pour permettre à celle-ci de le faire tout en comprenant la signification de ces triplets. Chaque élément du triplet est identifié par un URI (Uniform Resource identifier). Cette identification se fait de manière unique à l'aide d'un nom sans avoir à localiser la ressource. Les triplets sont interprétables comme sujet-prédicat-objet où le prédicat exprime la propriété,

RDF est un langage qui permet de définir une ontologie de manière très simple, son inconvénient est qu'il ne supporte pas la vérification de la cohérence des données (vérification que le champ « date de naissance » est vraiment une date par exemple).

2.5.2. RDF Schéma

RDFS (RDF Schema) est une évolution de RDF. Ce langage est simple, il ajoute à RDF la possibilité de définir des hiérarchies de classes et des propriétés et permet l'implémentation du modèle RDF pour la définition des ontologies avec une approche orientée objet.

Trois notions principales permettant la définition des primitives : la ressource (rdfs:Resource), la classe (rdfs:Class) et la propriété. L'avantage de RDFS est pouvoir créer une hiérarchie de classes et de propriétés, grâce aux notions de subClassOf et subPropertyOf. Et donc on peut par la suite instancier des classes en utilisant le rdf:type.

Le langage RDFS offre en plus du contrôle la terminologie et la structure des descriptions RDF, la possibilité de raisonner sur liens de types « est-un » (is a) qui existent entre les concepts et les propriétés. [17][20]

Ces langages sont les langages de base du Web Sémantique. Et c'est dans ce contexte que trois autres langages ont été développés comme des extensions de RDF(S) qui sont :

2.5.3. OIL

Ontology Interchange Language and Ontology Inference Layer, développé au début des années 2000 dans le cadre du projet européen On-To-Knowledge. Il étend les langages basés sur des "frames". Sa sémantique formelle est basée sur les logiques de description. [18]

2.5.4. DAML+OIL

DARPA Agent Markup Language, Il a été créé après le langage OIL entre 2000 et 2001, par la DARPA (Defense Advanced Research Projects Agency) dans le contexte du projet DAML sur la spécification précédente de DALM-ONT, qui a été construit en fin 2000, et sur OIL. DAML+OIL est basés sur les logiques de description [18] [12]. DAML est une combinaison de l'XML et du RDF.

La dernière version de DAML a été combinée avec OIL pour construire un nouveau langage (DAML+OIL) et qui permet d'exprimer les axiomes comme l'équivalence.

2.5.5. OWL

OWL (Ontology Web Language) est un langage qui a été créé par le W3C en 2004, il hérite du langage DAML+OIL. C'est le langage le plus expressif des autres langages. Il est dédié à définir les classes et les types de propriétés. Il repose sur la syntaxe des triplets RDF et réutilise certaines des constructions RDFS. [2] [44]

C'est un langage qui permet de fournir des mécanismes pour créer tous les composants de l'ontologie c'est-à-dire les classes, les instances, les propriétés et les

axiomes. Comme dans le langage RDFS, les classes peuvent avoir des sous-classes, fournissant ainsi un mécanisme pour le raisonnement et l'héritage des propriétés. Néanmoins, en OWL, on distingue :

- *les propriétés des objets (object property)*, ce sont les relations, qui relient des instances de classes à d'autres instances de classes. C'est l'équivalent des triplets RDF dont l'objet est une ressource.
- *les propriétés des types de données (datatype property)*, ce sont les attributs, qui relient des instances de classes à des valeurs de types de données (nombres, chaînes de caractères,...). C'est l'équivalent des triplets RDF dont l'objet est une valeur littérale.
- *Les axiomes* permettent de fournir de l'information au sujet des concepts et des relations, en spécifiant l'équivalence entre deux classes par exemple.
OWL permet d'annoter les données et de faire certains raisonnements sur les données et d'assurer que ces raisonnements qui peuvent être réalisés sur les ontologies soient décidables.

OWL se compose de trois sous-langages qui diffèrent selon le niveau d'expressivité, qui sont : OWL Lite, OWL DL et OWL Full. [96], OWL DL est défini comme une extension d'OWL Lite, et OWL Full comme une extension d'OWL DL.

- Le langage *OWL Lite* peut être vu comme une extension du langage RDFS, mais avec moins de fonctionnalités. Il contient un ensemble réduit de constructeurs en fournissant l'essentiel pour la construction d'une hiérarchie de classes. C'est le sous langage d'OWL le plus simple. Son principe est de permettre une modélisation d'ontologies moins compliquées, afin de faciliter l'implémentation des raisonneurs corrects et complets.
- Le langage OWL DL contient des constructeurs supplémentaires, mais avec des contraintes bien particulières, il ne peut être utilisé qu'avec certaines restrictions. Il est plus complexe que OWL Lite et est fondé sur la logique de description d'où son nom, « OWL Description Logics ». Malgré cette complexité par rapport à OWL Lite, il garantit la complétude des raisonnements et leur décidabilité.

- Le langage OWL Full est le plus complexe des sous-langages d'OWL, il dispose de tous les constructeurs du langage OWL permettant ainsi une interprétation plus large. Il permet de traiter les classes comme des individus. OWL Full est utile typiquement pour les gens qui veulent combiner l'expressivité du langage OWL avec la flexibilité et méta-modélisation des caractéristiques de RDF afin d'avoir un haut niveau de capacité de description, néanmoins, son utilisation ne garantit pas toujours la complétude du raisonnement.

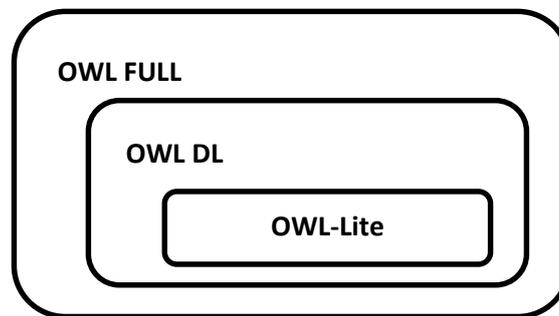


Figure 2.3 : Les sous langages d'OWL

2.6. Représentation formelle des ontologies

En plus des langages de représentation des ontologies, une ontologie peut être représentée de manière formelle, plusieurs formalisations ont été proposées dans la littérature, nous considérons celle proposée par Pierra[6], où une ontologie est représentée comme un quadruplet :

$$O : \langle C, P, \text{Sub}, \text{Applic} \rangle$$

Où:

- C : est l'ensemble des classes utilisées pour décrire les concepts de l'ontologie, un identifiant universel unique est associée pour chaque classe.
- P : est l'ensemble des propriétés utilisées afin de décrire les instances de l'ensemble des classes C. Chaque propriété est associée à un identifiant universel globalement unique.

– Sub : $C \rightarrow 2^C$ est une fonction de subsomption qui permet d'associer à chaque classe c de l'ontologie, ses classes subsumées directes.

– Applic : $C \rightarrow 2^P$ est une fonction permettant d'associer à chaque classe de l'ontologie les propriétés qui sont applicables pour chaque instance de cette classe. Ces propriétés sont héritées à travers la relation is-a. [3] [5]

2.7. Exemples d'ontologies dans le domaine (bio) médical

Dans le domaine médical plusieurs ontologies ont été créées, nous citons dans ce qui suit quelques une de ces ontologies :

2.7.1. L'ontologie OpenGalen

Le projet GALEN [85] (Generalised Architecture for Languages, Encyclopaedias and Nomenclatures in Medicine) est un projet fondé par une commission européenne en 1991[100] pour le développement de systèmes multilingues de terminologies médicales permettant de décrire les ontologies dans tous les domaines médicaux. .

OpenGalen est construit autour d'un modèle sémantique puissant de terminologie clinique qui est le CORE (COncept REference model), et utilise un langage de représentation appelé GRAIL (GALEN Représentation And Integration Language) dont le noyau est basé sur les logiques de description. Ce modèle consiste en une hiérarchie de subsomption d'entités élémentaires et un ensemble de déclarations liant ces entités.

La terminologie de Galen intègre trois modules :

- Concept Module : qui implémente le formalism GRAIL et permet de gérer la représentation interne des concepts,
- Multilingual Module : il permet de gérer les mapping des concepts au langage naturel
- Code Conversion Module : il gère les mapping des concepts avec le codage existant et les schémas de classification.

En 1999 l'ontologie OpenGalen a été créée afin de permettre l'accès à ses ressources sous forme d'ontologies accessibles en open source.

Une ontologie Galen est composée d'une hiérarchie de catégories élémentaires, une hiérarchie de liens sémantiques appelés attributs, rôles ou relations permettant d'établir des relations entre les catégories.

Les concepts peuvent être simples ou composés. Les concepts composés sont une combinaison de concepts simples. [104] [105]

2.7.2. Gene ontology

Le but de cette ontologie est de produire un vocabulaire structuré, précis, commun et contrôlé pour décrire les rôles des produits génétiques dans un organisme. Elle a été construite à partir de trois bases de données sur des organismes modèles : Flybase1 (Drosophila genome database); SGD2 (Saccharomyces Genome Database), et MGD/GXD3 (Mouse Genome Informatics databases) [102]

GO contient trois sous-ontologies qui décrivent la fonction moléculaire, les composants cellulaires et le processus biologique des produits génétiques d'autres bases de données biologiques. Sa structure est simple : les termes sont reliés par deux types de relations IsA ou PartOf, créant ainsi une structure d'arbre. [106]

¹ FlyBase (<http://flybase.org/>)

² Saccharomyces Genome Database (<http://genome-www.stanford.edu>)

³ Mouse Genome Database and Gene Expression Database (<http://www.informatics.jax.org>)

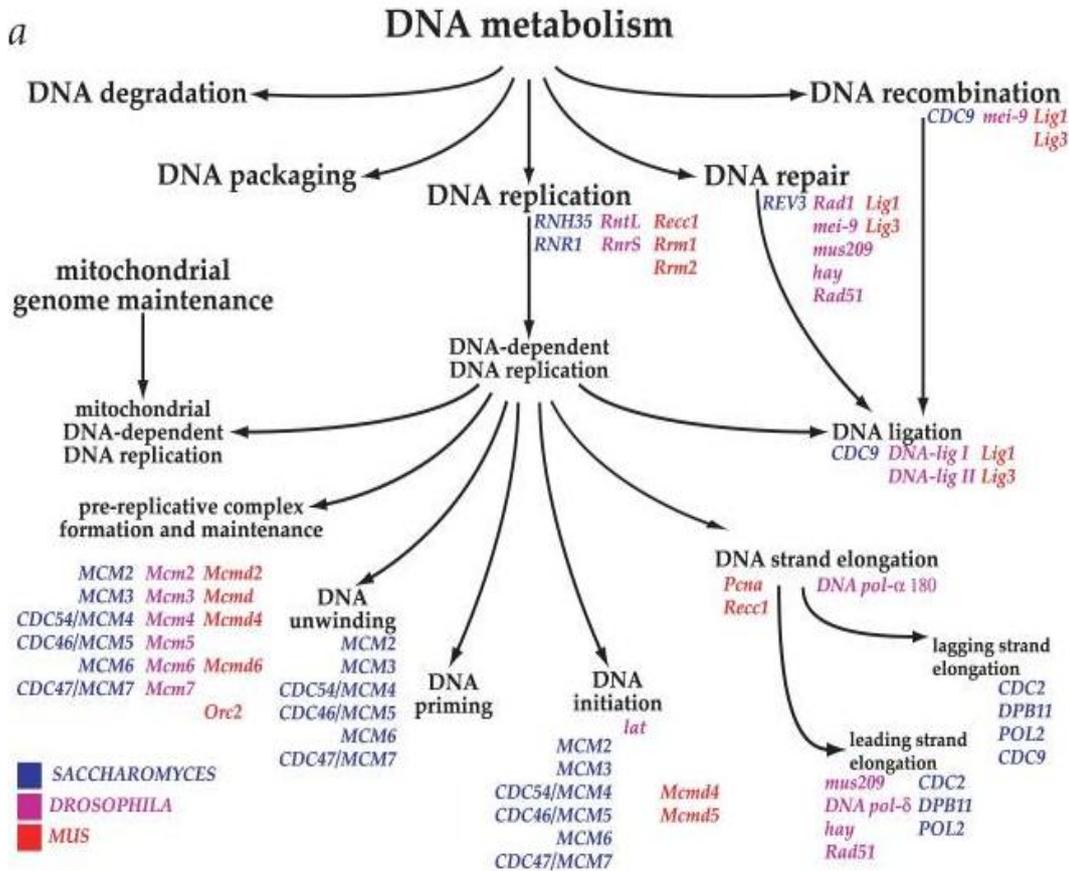


Figure 2.4 : Partie de l'ontologie GO décrivant le métabolisme DNA [107]

2.7.3. UMLS

UMLS [86] est désigné pas la NLM (*National Library of Medicine*), c'est l'un des grands projets qui combinent un grand nombre de différentes terminologies, il intègre environ 2 millions de noms pour environ 900000 concepts de plus de 60 familles de vocabulaires biomédicaux, en plus de 12 millions de relations entre les concepts. [103]

Le vocabulaire d'UMLS inclut la taxonomy de NCBI, la Gene Ontology, MeSH (Medical Subject Headings), OMIM et la base "Digital Anatomist Symbolic Knowledge Base".

Chaque concept dans UMLS a un identifiant unique qui est le CUI (Concept Unique Identier), ainsi pour chaque concept un ensemble de termes est associé dans des lexiques différents. D'autres identifiants uniques existent comme le LUI (Lexical Unique Identifier) qui permet d'identifier les variations lexicales pour un terme donné.

L'objectif principal d'UMLS est de fournir une plate-forme commune permettant de regrouper tous les thésaurus, nomenclatures, ontologies et les autres classifications qui existent dans le domaine médical et les relier dans un seul system.

2.7.4. FMA

L'ontologie FMA (Foundational Model of Anatomy) est une ontologie de référence sur l'anatomie humaine, elle a été créée initialement afin d'améliorer le contenu anatomique décrit par le vocabulaire d'UMLS, c'est une ontologie de domaine modélisant les concepts et les relations concernant l'organisation structurelle du corps humain. Elle englobe les objets matériels du niveau moléculaire au niveau macroscopique qui constituent le corps humain et associe avec eux les entités non-matérielles comme les espaces, les surfaces..., qui permettent de définir les relations structurelles. [108]

L'approche adoptée pour le développement de la FMA repose sur un ensemble de principes, des schémas de haut niveau et d'un environnement basé sur les frames et elle est stockée dans une base de données relationnelle.

De manière formelle l'ontologie FMA est représentée à travers un schéma de haut niveau (High level schema) comme un quadruplet : $FMA = (AT, ASA, ATA, Mk)$

Où :

- **AT** : « *Anatomy Taxonomy* », spécifiant les relations taxonomiques des entités anatomiques en les assignant à des classes, selon des attributs bien définis ;
- **ASA** : « *Anatomical Structural Abstraction* » représentant les relations spatiales des concepts qui sont représentés dans la taxonomy ;
- **ATA** : « *Anatomical Transformation Abstraction* » décrit en fonction du temps les transformations morphologiques des concepts représentés par la taxonomie pendant le cycle de vie de l'humain, comme le développement prénatal, vieillissement...
- **MK** : « *Metaknowledge* », elle comprend les règles et les principes correspondant à la représentation des relations dans les trois autres composants.

En 2003, la FMA a atteint 70,000 concepts différents de l'anatomie, associés à plus de 110,000 termes et reliés par plus de 1.5 million d'instanciation à travers 170 types de relations.

2.7.5. OntoMénélas

Le but initial de cette ontologie est la conception et l'implémentation d'un système pilote qui permet d'accéder à des rapports médicaux rédigés en langage naturel permettant l'analyse du contenu de rapports médicaux et l'archivage dans une base de données sous la forme d'un ensemble de structures conceptuelles, ces structures, qui constituent la représentation de chaque compte rendu d'hospitalisation(CRH), peuvent être consultées pour accéder à des informations spécifiques contenues dans le CRH.

Une partie des informations d'OntoMénélas est encodée à l'aide de nomenclatures internationales comme Cim-10, ce qui permet leur échange à partir de CRH écrits en différentes langues. Cette ontologie est appliquée au domaine de la chirurgie cardiaque. La compréhension d'un texte repose sur l'utilisation de connaissances médicales et de connaissances de sens commun qui permettent d'inférer de nombreuses informations implicites. Ces informations correspondent à des inférences effectuées naturellement par un spécialiste du domaine. [99]

2.7.6. SNOMED

SNOMED (Systematized Nomenclature of Human and Veterinary Medicine), est une nomenclature permettant de classifier et de normaliser les termes dans le domaine médical.

Elle contient environ 150 000 termes dans 12 axes différents organisés de manière hiérarchique, incluant les concepts de l'anatomie, la morphologie, les fonctions normales et anormales, les symptômes et les signes de maladies, les protéines, les relations spatiales, les organismes vivants, les diagnostics, les procédures ... [109][110]

Le processus de développement de la SNOMED est divisé en six étapes différentes :

- Initiation et mise en route (Start-up and Initiation), dans cette étape l'initiation de la SNOMED a été effectuée par les États-Unis et le Royaume-Uni
- La conception de la terminologie (Terminology Design) : cette étape a vu le développement de multiples documents de consultation à partir d'un groupe de travail technique.

- La production, c'est l'étape de la fusion du contenu de la SNOMED avec une autre terminologie qui est la CTV3 (Clinical Terms Version 3), en incluant les définitions sémantiques.
- Test Alpha, c'est la première phase de test de la terminologie, elle est limitée à 6 domaines cliniques.
- Test Béta, dans cette étape les données qui n'ont pas été raffinées au niveau de l'étape de production seront mis à la disposition des développeurs afin d'effectuer des tests.
- Mise en place (Release Process) elle consiste à effectuer des révisions et des corrections afin d'assurer l'intégrité des données, après cette étapes la terminologie peut être mise en disponibilité pour téléchargement.

2.8. Conclusion

Face à l'émergence des sources d'informations qui sont de plus en plus complexes et distribuées, il est nécessaire d'offrir une description de ces informations non seulement au niveau syntaxique mais aussi au niveau sémantique.

Dans les systèmes d'intégration, une ontologie permet de représenter explicitement la sémantique des données en réduisant les hétérogénéités qui peuvent apparaître lors de l'intégration.

Nous avons présenté dans ce chapitre les notions de base liées aux ontologies, nous avons décrit les différents composants qui constituent une ontologie, les types des ontologies, les langages de description des ontologies, et nous avons fait un tour sur quelques ontologies du domaine médical.

Dans notre contribution l'ontologie joue un rôle primordial où nous supposons qu'une ontologie partagée existe et qui couvre le domaine de la médecine, elle permet de décrire chaque élément de l'architecture proposée.

Le chapitre suivant permet de détailler les systèmes d'intégration de données et le rôle que jouent les ontologies pour ces systèmes.

Chapitre 3 : Intégration de données

Sommaire

3.1.	Introduction	29
3.2.	Hétérogénéité des données	30
3.1	Approches d'intégration	31
3.3.1.	Approche entrepôt de données	33
3.3.2.	Approche médiation	35
3.3.2.1.	Types de mappings	36
-	Global-As-View (GAV),.....	36
-	Local-As-View (LAV).....	37
-	Global-Local-as-View (GLAV).....	37
-	Both as view (BAV).....	37
3.3.2.2.	Exemples de projets d'intégration de données par médiation	37
<input type="checkbox"/>	Le projet TSIMMIS	37
<input type="checkbox"/>	Le système MOMIS (Mediator Environment for Multiple Information Sources)	38
<input type="checkbox"/>	Le système HERMES	39
<input type="checkbox"/>	Information manifold	39
<input type="checkbox"/>	Infomaster	39
3.3.3.	Intégration de données à base d'ontologies.....	40
3.3.3.1.	Architectures d'intégration à base d'ontologies	40
3.3.3.2.	Bases de données à base ontologique	42
3.3.3.2.1.	Représentation des données à base ontologique	42
<input type="checkbox"/>	Représentation binaire.....	42
<input type="checkbox"/>	Représentation verticale	43
<input type="checkbox"/>	Représentation horizontale	43
3.3.3.2.2.	Architectures des bases de données à base ontologique	43
<input type="checkbox"/>	Architecture de type I (deux quarts).....	43

□ Architecture de type II (trois quarts)	44
□ Architecture de type III (quatre quarts)	44
3.3.3.2.3. Le modèle OntoDB	45
3.3.3.2.4. Le langage OntoQL	45
3.4. Conclusion	46

3.1. Introduction

La diversité des sources d'information distribuées et leur hétérogénéité posent un problème important pour les entreprises. Beaucoup de solutions d'intégration ont été proposées afin d'assurer l'interopérabilité des systèmes et de résoudre le problème de l'hétérogénéité des sources d'information.

Dans ce contexte, des recherches ont été investies dans le domaine des bases de données, où ont été développées les bases de données relationnelles, objets, objets-relationnelles..., un autre type de bases de données a été créé ces dernières années, il s'agit des bases de données sémantiques ou bases de données à base ontologique [97] afin de remédier au problème de l'hétérogénéité entre différentes sources d'informations. Cette notion de base de données sémantique est la fusion des deux technologies : gestion de connaissances où la notion d'ontologie a été établie et le domaine des bases de données. L'ontologie permet de définir de manière consensuelle l'ensemble des connaissances d'un domaine bien particulier [21]. Une base de données sémantique permet de représenter de manière explicite une ontologie au sein d'une base de données.

Un système d'intégration de données permet d'offrir à l'utilisateur une vue uniforme et transparente des informations issues de sources hétérogènes et distribuées sans qu'il soit amené à savoir leur source ou la façon dont elles sont interrogées.

Deux approches majeures existent pour l'intégration de données, l'approche entrepôt de données (approche matérielle) et l'approche médiation de données (approche virtuelle). Notre étude s'inscrit dans l'approche de médiation.

Dans la suite de ce chapitre nous commençons par étudier les différents types d'hétérogénéités qui peuvent exister entre les différents systèmes et les solutions proposées

dans la littérature, ensuite nous étudions les approches entrepôt et médiation de données en mettant le point sur différents projets de médiation de données dans la littérature.

3.2. Hétérogénéité des données

Le problème majeur des systèmes d'intégration est l'hétérogénéité des données à intégrer puisque chaque système est conçu de manière différente, par des personnes différentes et utilisant des vocabulaires différents. Avant de définir divers types d'intégration de données, il est important d'étudier les différentes hétérogénéités qui peuvent exister lors de l'intégration, deux types majeurs d'hétérogénéité existent :

- Hétérogénéité structurelle : on parle d'hétérogénéité structurelle lorsqu'il existe différentes représentations des mêmes concepts c'est-à-dire utiliser des modèles différents pour décrire la même données ou inversement des données différentes sont décrites dans le même modèle, par exemple utiliser différents unités pour exprimer la même mesure. On l'appelle aussi hétérogénéité des schémas.
- Hétérogénéité Sémantique : cette hétérogénéité se trouve lorsque l'on exprime le même concept mais avec des significations différentes, Assia Soukane[22] définit deux types de conflits sémantiques :
 - *les conflits sémantiques liés au schéma* où on décrit le même concept utilisant des terminologies différentes,
 - *et les conflits sémantiques liés aux données*. Dans le cas où les données proviennent de différentes origines, saisies à des moments différents par des personnes différentes qui n'ont pas la même perception, et utilisent des conventions différentes.

Dung Xuan Nguyen[23] définit quatre types de conflits :

- conflits de représentation, qui apparaissent lorsqu'on utilise des propriétés différentes ou des schémas différents pour décrire un même concept.
- conflits de noms (termes), dans le cas où on utilise des noms différents pour décrire le même concept, ou bien utiliser des noms identiques pour décrire des concepts différents
- conflits de contextes : ils se produisent dans le cas où on définit un concept mais dans des contextes différents

- et les conflits de mesure de valeur : on les trouve dans le cas où on utilise des unités différentes pour mesurer la valeur d'une propriété.

Bakhtouchi [113] distingue trois types de conflits :

1) les conflits technologiques qui sont liés à la technologie utilisée pour représenter les données, par exemple utiliser des SGBD différents,

2) les conflits de schéma qui peuvent être soit des conflits sémantiques, de description, d'hétérogénéité ou des conflits structurels et

3) les conflits d'instances sont causés par les erreurs de qualité, telles que l'exactitude, la complétude, la fraîcheur, et les erreurs de cohérence ; ces conflits sont divisés en deux classes : conflits de référence qui surgissent lorsque des instances provenant de relations différentes réfèrent au même objet mais contiennent des références différentes et conflits de valeurs d'attributs qui apparaissent lorsque des instances, qui correspondent aux mêmes objets du monde réel et partagent une même référence, diffèrent dans d'autres attributs

Lors de la conception d'un système d'intégration, il est nécessaire de considérer les différents types de conflits, un système d'intégration doit être capable de résoudre le problème de l'hétérogénéité entre les sources, de nombreux travaux ont proposé la résolution des conflits, la plupart des approches actuelles utilisent les ontologies pour résoudre le problème de l'hétérogénéité sémantique entre les différentes sources, ces ontologies permettent de représenter de manière formelle et explicite les informations contenues dans les sources.

3.1 Approches d'intégration

L'intégration de données est un processus permettant de combiner plusieurs sources afin qu'elles puissent être interrogées à travers une interface commune, en éliminant les conflits entre ces données, et les présenter de manière cohérente. Permettant ainsi un accès uniforme à de multiples sources d'information à travers un schéma qui fournit une vue globale d'un ensemble de sources de données.

Un système d'intégration est composé en général en trois couches principales :

- Une couche de données : contenant l'ensemble des sources de données à intégrer,
- Une couche des adaptateurs ou des chargeurs : qui permettent d'extraire les données et de les représenter dans le schéma global, c'est le moyen avec lequel on peut accéder à une source de données, et avec lequel la source peut interagir avec les autres composants de l'architecture.
- Une couche qui contient le composant qui permet aux utilisateurs d'interroger les différentes sources à travers un schéma global, ce composant peut être un entrepôt de données où toutes les données des sources sont dupliquées (approche matérialisée), ou bien un médiateur (approche virtuelle), dans cette approche les données restent au niveau des sources et ne sont pas migrées vers un schéma global, la requête de l'utilisateur est divisée en sous requêtes.

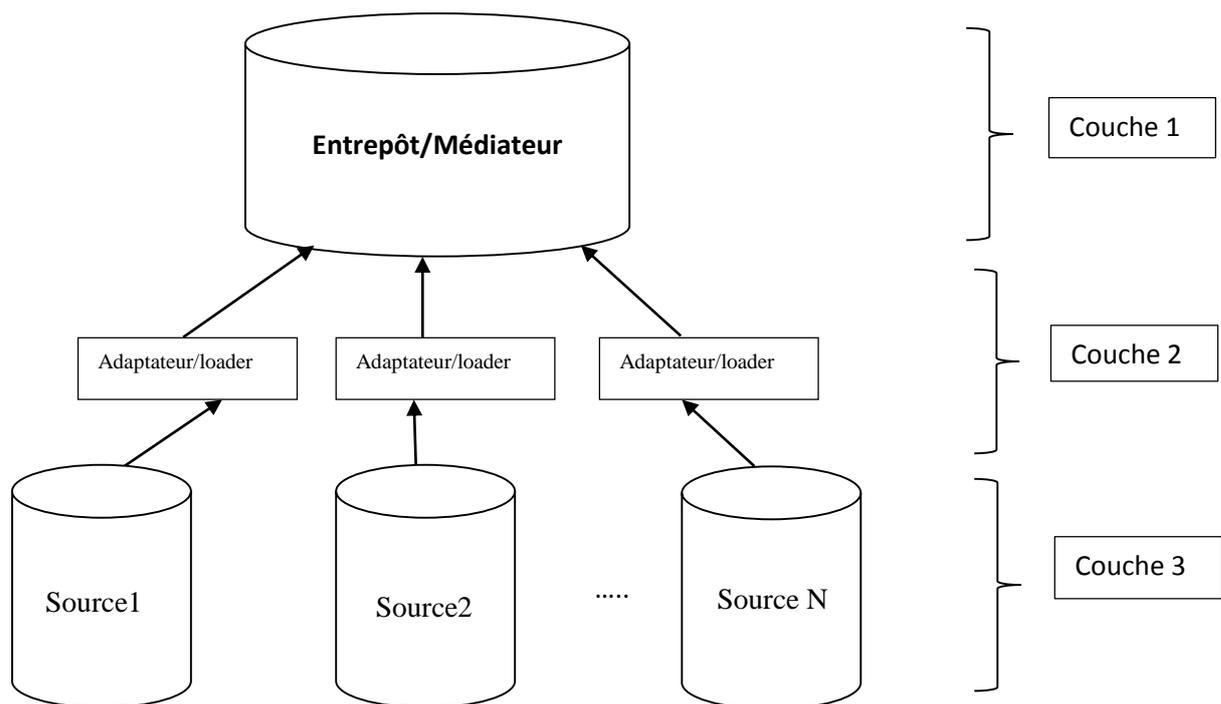


Figure 3.1 : Architecture générale d'un système d'intégration

De manière formelle un système d'intégration peut être défini comme un triplé (G,S,M) [33] [37]

Où :

- G : représente le schéma global
- S : est l'ensemble des schémas des sources de données
- et M : représente les mappings entre le schéma global et les schémas des sources

3.3.1. Approche entrepôt de données

Cette approche permet de centraliser l'accès aux données en les combinant dans un seul endroit appelé entrepôt de données.

La notion d'entrepôt de données a été définie par Inmon[25], où il définit un entrepôt de données comme une méthode de stockage de données intégrées pour être utilisées dans les systèmes d'aide à la décision, en offrant des méthodes d'analyse comme OLAP(On-Line Analytical Processing), Un cube OLAP contient des données servant à faire des analyses de données provenant de différentes sources hétérogènes et distribuées. Cette analyse est effectuée en organisant les données de manière multidimensionnelle.

La construction d'un entrepôt de données, passe les étapes suivantes :

- l'Extraction des données à partir des sources,
- la Transformation des données,
- l'intégration des données, et le stockage des données intégrées au niveau de l'entrepôt (Load)

Ces étapes correspondent au processus ETL (Extract, Transform and Load), qui permet d'extraire les données à partir des sources, les transformer, et les charger au niveau de l'entrepôt de données. [47][48]

- Extraction : elle consiste à récupérer les données à partir des sources de données.
- Transformation : cette étape est constituée d'un ensemble d'opérations permettant de formater et nettoyer les données afin de les rendre homogènes pour pouvoir les extraire au niveau de l'entrepôt.
- Chargement : c'est l'étape qui permet de charger les données qui sont préparées au niveau de l'entrepôt de données, cette étape peut être longue. Ces données peuvent être utilisées par des outils décisionnels tels qu'OLAP.

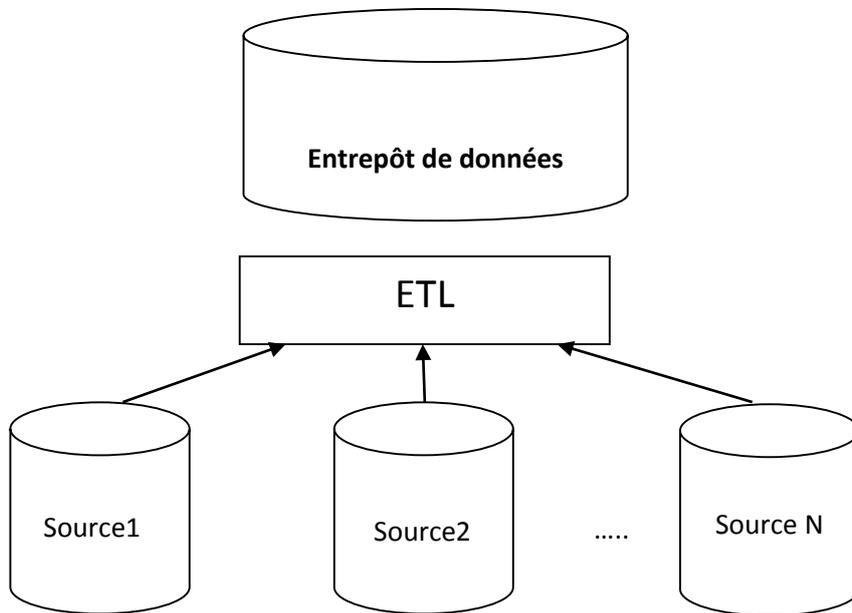


Figure 3.2 : Architecture d'un entrepôt de données

Ces étapes sont réalisées à l'aide d'un composant logiciel appelé adaptateur, cet adaptateur offre une interface d'accès à une source de données, extraire les données et les stocker dans l'entrepôt de données.

L'avantage de cette approche est que les données sont accessibles à partir d'un seul endroit, par contre elle est très coûteuse en terme de stockage, où une mise à jour de l'entrepôt de données est nécessaire pour chaque mise à jour au niveau des sources de données et surtout dans un environnement où les sources sont nombreuses et changent de manière fréquente.

[26][28][29][30] sont des exemples de travaux sur les entrepôts de données dans le domaine médical.

3.3.2. Approche médiation

La médiation est une approche d'intégration de données permettant de répondre à des requêtes complexes de manière transparente, elle a été introduite par wiederhold en 1992[39] une architecture de médiation est constituée généralement des composants suivants (Figure 3.3) :

- Les sources de données : contiennent les données à intégrer
- Les wrappers (adaptateurs) : le wrapper joue le rôle de traducteur entre une source de données et le médiateur, où il traduit la requête en un langage compréhensible par la source et inversement, traduit les données retournées dans le format du médiateur. il permet de dialoguer avec une source de données à partir d'une interface commune (médiateur).
- Le médiateur : c'est l'élément principal dans une architecture de médiation, il permet de formuler la requête de l'utilisateur en sous requêtes destinées au wrappers pour interroger une source de données, à travers un schéma global qui contient des vues sur les sources de données, puis il recompose les résultats obtenus à partir des sources de données en une seule réponse à l'utilisateur.

Les correspondances (mappings) entre le schéma global et les schémas locaux se font de manière manuelle, cependant des travaux ont été réalisés afin d'automatiser ces correspondances. Quatre types de méthodes existent pour définir les correspondances entre le schéma global (schéma du médiateur) et les schémas locaux (schémas des sources). [33] [34][35]. Dans ce qui suit nous détaillons les différents types de mappings.

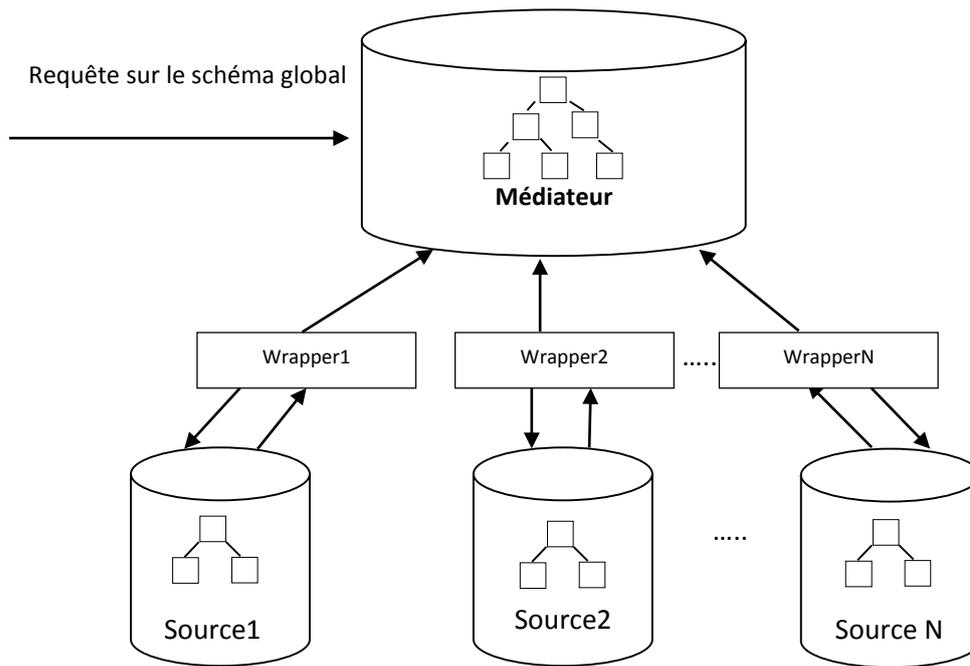


Figure 3.3 : Architecture globale d'un médiateur

3.3.2.1. Types de mappings

Quatre types de mapping existent permettant de définir les correspondances entre le schéma du médiateur et les schémas des sources :

- **Global-As-View (GAV),**

Dans cette approche les entités du schéma global sont définies comme des vues sur les schémas sources. Pour répondre à une requête on fait référence au schéma global. La requête est réécrite en utilisant les termes des schémas locaux, il suffit juste de remplacer les termes par leurs définition, l'inconvénient de cette approche est que la mise à jour du schéma global est coûteuse, où mise à jour au niveau du schéma global nécessite la mise à jour de tous les schémas locaux (c'est-à-dire mettre à jour les vues des sources sur le schéma global), ainsi ajouter une nouvelle source nécessite de changer la définition es concepts au niveau du schéma [37]

- ***Local-As-View (LAV)***

Contrairement à l'approche GAV, dans l'approche LAV, les entités des schémas locaux sont définies comme des vues sur le schéma global. L'avantage de cette approche est qu'il n'y a pas une forte dépendance entre le schéma global et les schémas locaux. L'ajout ou la suppression de nouvelles sources nécessite seulement la définition des correspondances nécessaires entre le schéma local et le schéma global. Cependant, dans cette approche, la réponse à une requête nécessite une reformulation qui est une tâche un peu difficile. Une requête est réécrite sous forme de requête conjonctive selon les vues sur les schémas des sources de données [41], la réponse à la requête est l'union des résultats obtenues pour les sous requêtes

- ***Global-Local-as-View (GLAV)***

Cette approche permet de combiner les deux approches précédentes permettant de créer des vues sur les sources en générant des vues sur le schéma global, afin de combler les lacunes des deux approches précédentes [36]. C'est-à-dire que dans cette approche, on dispose des vues au niveau global et local.

- ***Both as view (BAV)***

C'est une autre méthode pour définir les mappings entre le schéma global et les schémas locaux, dans cette approche la transformation est faite dans les deux directions c'est-à-dire du schéma global vers les schémas locaux, et inversement des schémas locaux vers le schéma global. AutoMed est un exemple de projet utilisant ce mapping. [38]

Dans ce qui suit nous décrivons brièvement quelques travaux sur la médiation des données dans la littérature, ces travaux diffèrent selon le sens de mapping entre le schéma global et les schémas locaux.

3.3.2.2.Exemples de projets d'intégration de données par médiation

- ***Le projet TSIMMIS***

C'est l'un des premiers projets qui utilisent l'approche médiateur-wrapper pour l'intégration de données, il est basé sur l'approche GAV offrant un moyen pour intégrer des sources d'information multiples et hétérogènes [31]. Dans ce projet, un traducteur (wrapper) est associé à chaque source d'information, qui est chargé de réécrire une requête en sous requêtes. Il utilise un modèle de données

orienté objet appelé Modèle d'Echange d'Objet (Object Exchange Model : OEM) et un langage de règles (MSL, Mediator Specification Language) pour l'interrogation. Plusieurs médiateurs peuvent exister dans cette architecture, ces médiateurs reçoivent des informations des wrappers afin de les traiter. Le médiateur central permet de générer un plan à la requête. (Figure 3.4)

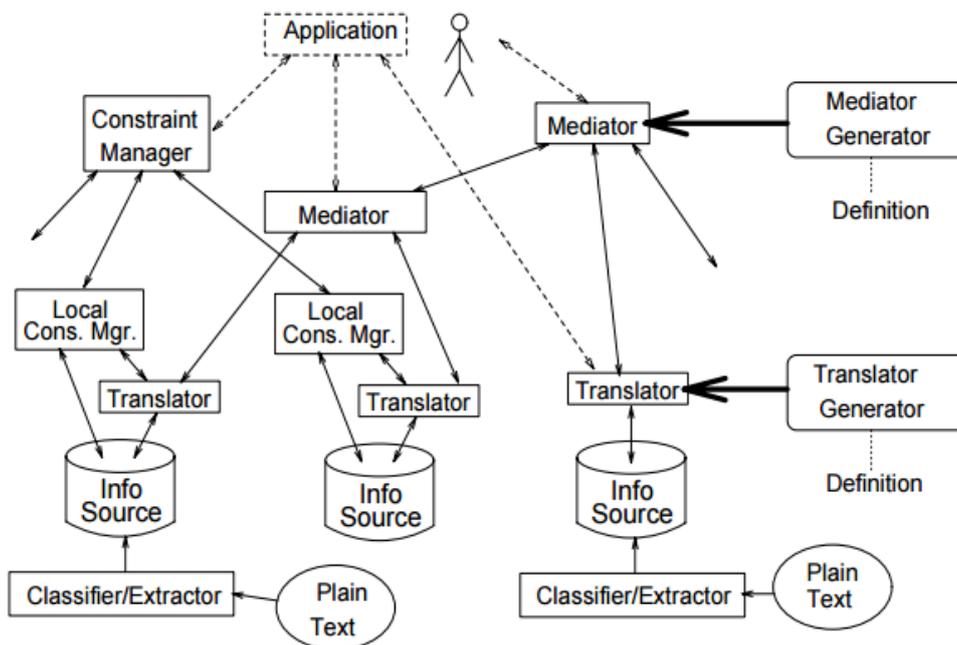


Figure 3.4 : Architecture générale de TSIMMIS [31]

- ***Le système MOMIS (Mediator Environment for Multiple Information Sources)***

Ce système est basé sur l'utilisation de la logique de description de type (ODL-I3) pour décrire les schémas des sources de données à intégrer. Il repose sur un thesaurus dérivé de la base de données lexicale WordNet, il vise à intégrer les données structurées et semi-structurées de manière semi-automatique, dans ce système un adaptateur est associé à chaque source de données, cet adaptateur vise à traduire les descriptions des métadonnées vers une représentation commune basée sur le modèle (ODL-I3).[40]

- ***Le système HERMES***

C'est un système de médiation suivant l'approche GAV. Il repose sur un langage de règles pour exprimer les requêtes [42]. La construction des médiateurs dans ce système est basée sur deux tâches principales [43] : intégration de domaine qui permet de lier physiquement les sources d'information, et l'intégration sémantique qui permet d'extraire et de combiner les informations des sources.

- ***Information manifold***

Ce système décrit de manière déclarative le contenu des sources d'information. Le contenu des sources est décrit par des requêtes sur un ensemble de relations et de classes. Des relations et des classes virtuelles jouent le rôle de schéma global sur lequel l'utilisateur pose ses requêtes.

Ce système suit l'approche LAV où les relations sont décrites au niveau des sources sous forme de requêtes à travers les relations du schéma global.

Le plan d'exécution d'une requête est généré en utilisant un algorithme composé en deux phases : dans la première un plan sémantique est généré sous forme de requête conjonctive qui utilise les relations des sources et qui est contenue dans la requête de l'utilisateur.

- ***Infomaster***

Permet l'accès à diverses sources hétérogènes. Il utilise le format d'échange de connaissances (KIF : knowledge interchange format) comme langage pour décrire le contenu. Il considère trois types de relations : 1-relations d'interface, utilisées pour formuler les requêtes de l'utilisateur. 2-relations des sources qui décrivent les données stockées au niveau des sources. 3-relations globales, qui représentent un schéma de référence.

3.3.3. Intégration de données à base d'ontologies

L'intégration sémantique de données consiste à offrir une représentation conceptuelle des données afin d'éliminer les conflits entre ces données. Actuellement, les ontologies sont utilisées dans divers domaines pour représenter explicitement les connaissances offrant ainsi une description sémantique des informations d'un domaine particulier. Dans les systèmes d'intégration utilisant les ontologies, une ontologie peut représenter le schéma global des sources à intégrer afin de pouvoir résoudre les hétérogénéités entre les sources de données, dans ce cas l'ontologie représente un modèle pour interroger les sources de données.

3.3.3.1. Architectures d'intégration à base d'ontologies

Trois types d'architectures existent pour l'intégration de données en fonction de la façon dont les ontologies sont utilisées dans un système d'intégration [111] :

a. Architecture avec une ontologie unique

Cette approche utilise une seule ontologie partagée, chaque source à intégrer est liée à une seule et même ontologie globale.

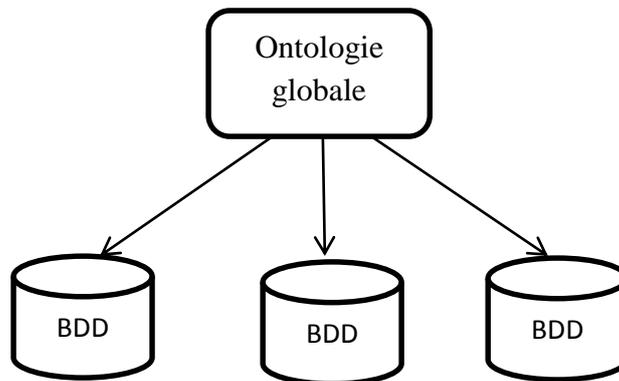


Figure 3.5 : Architecture avec une ontologie unique

b. Architecture utilisant plusieurs ontologies

Dans cette approche, chaque source d'information est décrite par sa propre ontologie, indépendamment des autres sources. Des mappings entre les ontologies des sources sont définies permettant d'identifier les correspondances sémantiques entre les termes de ces ontologies.

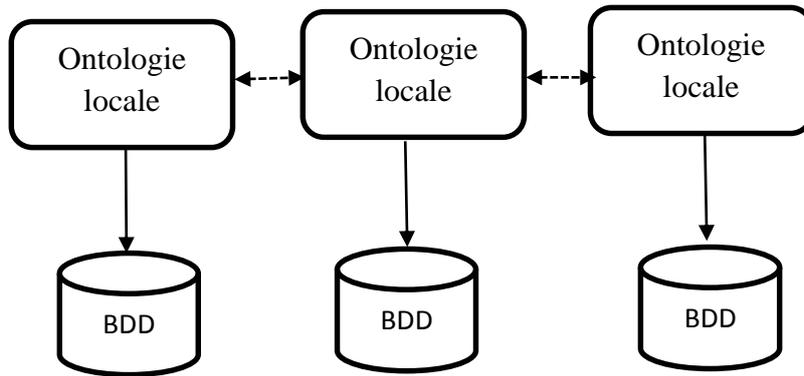


Figure 3.6 : Architecture utilisant plusieurs ontologies

c. Architecture hybride

Cette architecture combine les deux autres types d'architectures. Dans cette approche, chaque source est décrite par sa propre ontologie, les différentes ontologies sont connectées entre elles par un vocabulaire commun partagé.

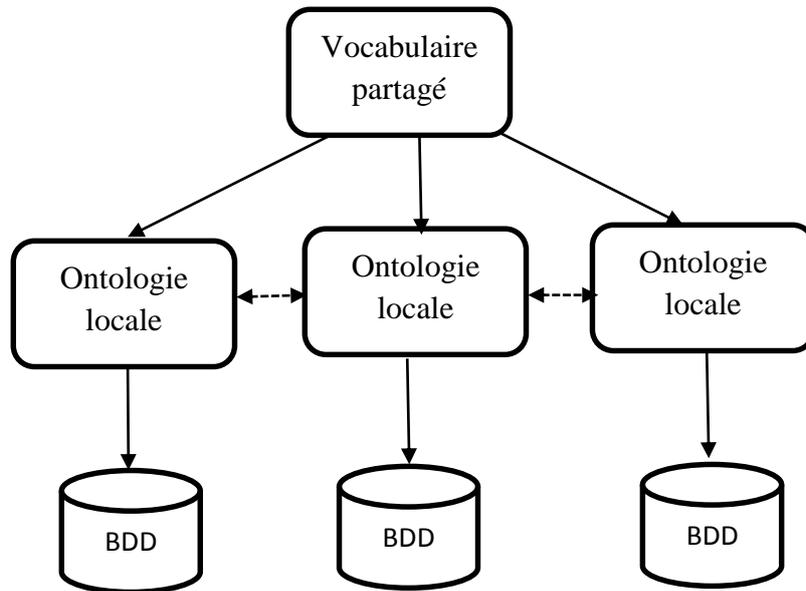


Figure 3.7 : Architecture hybride

3.3.3.2. Bases de données à base ontologique

C'est une extension des bases de données traditionnelles permettant de gérer les données ontologiques offrant la possibilité de stocker ces données et leur définition ontologique. Le but est d'associer à chaque donnée un concept ontologique qui définit son sens au niveau de la même base de données.

Les bases de données à base ontologiques diffèrent selon la représentation des données c'est-à-dire le mode de stockage de données au niveau de la base et selon l'architecture de la base de données.

3.3.3.2.1. Représentation des données à base ontologique

Il existe trois modes de stockage de données ontologiques au niveau de la base de données :

➤ **Représentation binaire**

Au niveau de cette représentation une table est créée pour chaque classe de l'ontologie, et une table pour représenter les propriétés des instances.

➤ **Représentation verticale**

Ce mode permet de représenter l'ontologie par une table à trois colonnes. La première colonne contient l'identifiant d'une entité qui peut être une classe, une propriété ou une instance de l'ontologie, la deuxième colonne contient son nom et la troisième contient sa valeur.

➤ **Représentation horizontale**

Elle permet d'associer à chaque classe de l'ontologie une table avec une colonne pour chaque propriété de cette classe.

3.3.3.2. Architectures des bases de données à base ontologique

Trois types d'architecture de BDBO existent [46]:

➤ **Architecture de type I (deux quarts)**

Comme dans les bases de données traditionnelles une base de données de cette architecture est constituée de deux parties : une partie contenant les données et une partie contenant le schéma des données. Dans cette architecture le schéma de l'ontologie et les instances sont stockés au même niveau et ne peuvent pas être séparés. Oracle [50], et Jena [49] sont des exemples de cette architecture.

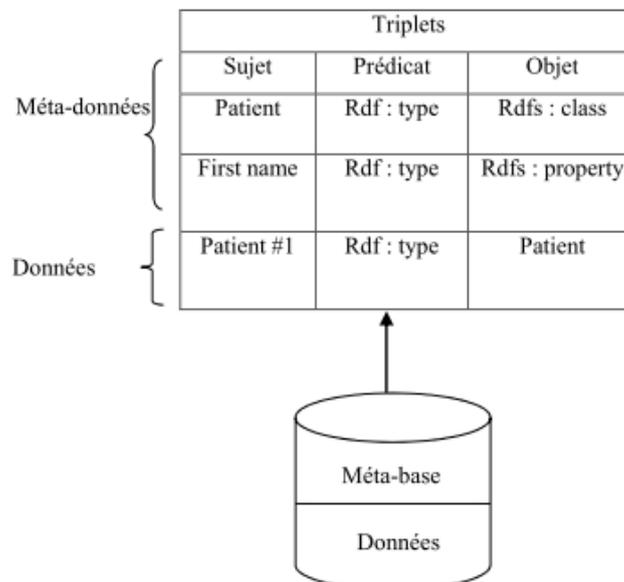


Figure 3.8 : Architecture de base de données sémantique de type I (deux quarts).

➤ **Architecture de type II (trois quarts)**

Cette architecture a été proposée pour pallier l'inconvénient de l'architecture deux quarts où les instances et le modèle de l'ontologie ont été séparés, contenant ainsi trois parties.

Au niveau de cette architecture il est difficile d'ajouter de nouveaux concepts de l'ontologie car un schéma est imposé.

RDFSuite [53], Sesame [54] et SOR [55] sont des exemples de SGBD de type II

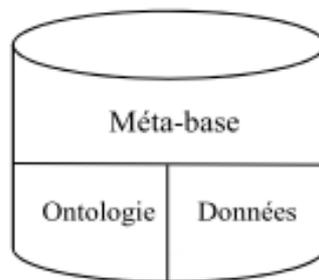


Figure 3.9. Architecture de base de données sémantique de type II (trois quarts).

➤ **Architecture de type III (quatre quarts)**

L'architecture quatre quarts étend l'architecture trois quarts, où une quatrième partie a été ajoutée, il s'agit du méta-schéma qui représente le schéma de l'ontologie.

Cette architecture a été proposée dans le cadre du projet OntoDB [51] et son extension OntoDB2 [52]

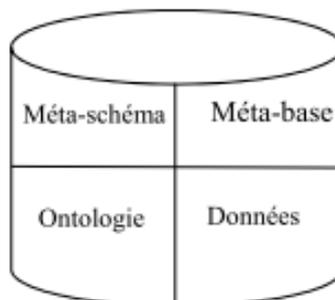


Figure 3.10 : Architecture de base de données sémantique de type III (quatre quarts).

3.3.3.2.3. Le modèle OntoDB

OntoDB [51] est une architecture de base de données à base ontologique utilisant l'approche horizontale c'est-à-dire que pour chaque classe de l'ontologie, une table est créée avec une colonne pour chaque propriété. Il utilise l'architecture de type III. Il est construit en utilisant le SGBD Postgres au sein du laboratoire LIAS⁴, il est constitué de quatre parties:

1. La partie ontologie qui permet de représenter l'ontologie qui définit la sémantique des données de la base,
2. La partie méta-schéma permet de représenter le modèle de l'ontologie utilisé et le méta-schéma.
3. La partie méta-base permet de représenter les schémas logiques des autres parties. Elle est appelée aussi catalogue système.
4. La partie « données » représente les instances c'est-à-dire les objets du domaine qui sont décrits en termes de classe d'appartenance et d'un ensemble de valeurs de propriétés applicables à cette classe.

Les ontologies gérées sous OntoDB, sont conformes au modèle PLIB⁵. Cependant, il est conçu afin de pouvoir supporter n'importe quel type d'ontologie, car elle dispose d'un méta-schéma susceptible de prendre en compte tout schéma d'ontologie.

3.3.3.2.4. Le langage OntoQL

Le modèle OntoDB est doté d'un langage de requêtes afin d'exploiter les données, il s'agit du langage OntoQL [56], ce langage permet d'effectuer les différents traitements sur des données ainsi que sur les ontologies d'une base de données OntoDB comme l'interrogation, l'insertion, la modification...

Le langage OntoQL est une extension du langage SQL offrant la possibilité d'exploiter la partie ontologique, il est associé au formalisme PLIB et basé sur un noyau commun aux différents modèles ontologiques, qui peut être facilement étendu par des

⁴ Laboratoire d'Informatique et d'Automatique pour les Systèmes

⁵ PLIB est la norme européenne ISO 13584 permettant la modélisation, l'échange et le référencement de catalogues informatisés des composants ou des objets techniques préexistants

instructions dédiées. Ainsi, OntoQL est indépendant de la représentation de l'ontologie (formalisme) et de la représentation des données (logique). [97]

3.4. Conclusion

Nous avons décrit dans ce chapitre les différents types d'hétérogénéités qui peuvent exister entre les systèmes hétérogènes, et les solutions proposées dans la littérature. Nous avons vu que la plupart des solutions se basent sur l'utilisation des ontologies, ensuite nous avons étudié les approches entrepôt et médiation de données et les différentes facettes liées à l'approche médiation et les différents travaux de médiation de données réalisés dans la littérature, nous avons fait aussi un état de l'art sur les solutions d'intégration de données à base de données plus précisément celle basées sur les bases de données à base ontologique.

Chapitre 4 : Services Web Sémantiques

Sommaire

4.1.	Introduction	49
4.2.	Architecture Orientée Services	49
4.3.	Services web	50
4.3.1.	Principaux standards des services web.....	51
	i. Langage XML	51
	ii. WSDL	52
	iii. SOAP	53
	iv. UDDI.....	55
4.3.2.	Services web sémantiques	57
	4.3.2.1. OWL-S	57
	4.3.2.2. WSMO	58
	4.3.2.3. WSDL-S.....	59
4.3.3.	Services web pour l'intégration de données.....	59
4.4.	Conclusion.....	60

4.1. Introduction

Le web joue un rôle de plus en plus important pour les entreprises où elles se basent essentiellement sur la collaboration et le partage d'informations qui sont généralement hétérogènes et distribuées, celles-ci nécessitent une bonne gestion afin d'être accessibles de manière pertinente. Dans le domaine médical par exemple, une grande masse de données est disponibles aux praticiens, ces données proviennent des analyses, des rapports de consultations...etc, et peuvent contribuer à l'amélioration des soins médicaux si elles sont exploitées et présentées de manière appropriée. Des techniques se développent pour gérer ces données de manière pertinente. Parmi ces techniques l'utilisation des services web, qui permettent l'interopérabilité des applications distribuées. Un service web est une application accessible et invocable par d'autres applications utilisant les protocoles d'internet tel que le HTTP et le XML permettant l'échange d'informations provenant de machines différentes et écrits avec des langages différents.

Dans ce chapitre nous détaillons la notion de services web sémantiques en présentant les approches les plus représentatives de ce domaine. Nous commençons par la définition de l'architecture orientée service sur laquelle se basent les services web, nous étudions par la suite les principaux langages des services web sémantiques, et enfin on présente quelques travaux utilisant les services web pour l'intégration de données.

4.2. Architecture Orientée Services(AOS)

Après les approches procédurales, les approches objets sont apparues, celles-ci reposent sur la notion de classe, d'héritage, d'instanciation...etc ; ensuite, les approches composants telles que CORBA et DCOM, permettant la réutilisation de programmes dans des environnements distribués, l'inconvénient de ces approches est qu'elles permettent un couplage fort entre applications. Enfin, l'architecture orientée services est apparue en 1996 par le groupe américain de recherche en technologie Gartner Group, afin de répondre aux enjeux des approches à base de composants notamment le problème du couplage en réduisant les dépendances entre les systèmes. Elle permet ainsi d'offrir des services indépendants des plates-formes, de la localisation, des systèmes ou des réseaux [71].

Cette architecture présente un modèle d'intégration moderne, permettant de réduire les coûts d'intégration, beaucoup de systèmes d'intégration actuels se basent sur cette architecture [57][58][59][60][61], elle permet de structurer les fonctionnalités d'un système d'information en un ensemble de composants logiciels réutilisables (appelés services) qui

communiquent par des messages. Ces services sont définis indépendamment de la plateforme et des autres services et permettent de créer des applications composites selon un processus métier de l'entreprise. Une architecture orientée services est constituée principalement de trois composants essentiels : le fournisseur de services, le client et le registre. Le fournisseur publie la description du service dans le registre, le client cherche et localise un service au niveau du registre et enfin il invoque le service trouvé. (Figure 4.1). Avec cette architecture se fut la naissance des services web et par la suite des services web sémantiques.

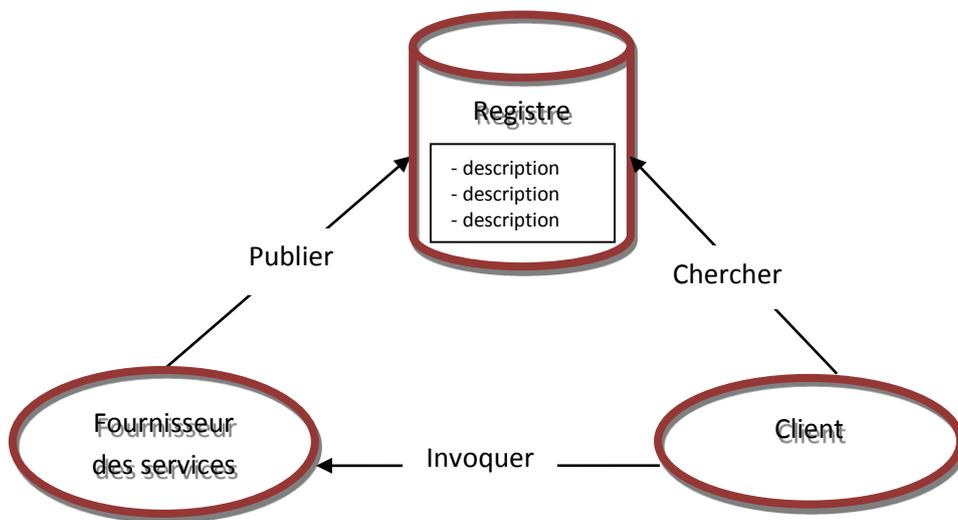


Figure 4.1: Modèle général d'une architecture orientée services

4.3. Services web

Les services web reposent sur l'architecture SOA, un service web est une application accessible et invocable par d'autres applications utilisant les protocoles d'internet tel que le HTTP et le XML permettant l'échange d'informations provenant de machines différentes et écrits avec des langages différents. Trois standards importants spécifiés permettant la description, le transport et la publication des services web, ces standards se basent sur XML. L'interface de service web est décrite par un document WSDL qui inclut toutes les informations qui peuvent être échangées par le service, le transport d'informations entre services web se fait à travers le protocole SOAP (Simple Object Access Protocol), et l'UDDI

(Universal Description, Discovery and Integration) qui représente l'annuaire des services web où ils sont stockés et où ils peuvent être localisés par des clients. [62][68][69][70]

Un service web peut être découvert par d'autres services et peut interagir avec d'autres services, D'autres standards ont été développés pour la composition de services web comme le BPEL, XLANG et le WSFL.

Dans ce qui suit nous nous limitons à présenter brièvement les trois standards principaux des services web, WSDL, SOAP et l'UDDI.

6.2.1. Principaux standards des services web

i. *Langage XML (eXtensible Markup Language) [64]*

XML est un langage de description des données développé au sein du W3C¹. Sa première spécification est apparue en février 1998, afin de combler les lacunes du langage HTML. XML permet de séparer le contenu de la présentation de documents. Il est utilisé dans différents domaines et diverses applications telles que le commerce électronique, la gestion des documents, le partage et l'échange d'information, la description de l'interface et l'encodage des messages des web services...

Un document XML est lisible par l'humain mais il est destiné aussi à être lu par la machine, sa syntaxe unique lui permet d'être adopté par différents systèmes d'exploitation, il permet de définir des balises et de leur associer une interprétation, il a une structure arborescente qui facilite sa lisibilité. C'est aussi un langage extensible où d'autres langages ont été développés en se basant sur ce langage, nous citons quelques exemples :

- MathML (Mathematical Markup Language)[65] c'est un langage qui utilise le langage XML pour définir les formules mathématiques.
- SVG (Scalable Vector Graphics)[66] : c'est une spécification pour décrire les graphiques, il définit un format pour le dessin vectoriel utilisant le langage XML.
- SMIL (Synchronized Multimedia Integration Language)[67] cette spécification permet l'intégration de contenu multimédia d'un document en définissant le déroulement temporel et spatial des éléments à afficher.

Hospitexte [78] est un exemple de projet permettant de présenter le dossier médical patient sous le format XML.

ii. **WSDL**

Le WSDL (Web Service Description Language) est le langage de description de services web spécifié par W3C, basé sur le langage XML, il permet de décrire les fonctionnalités du service c'est-à-dire les opérations qu'il peut effectuer, sa localisation et comment il peut être invoqué.

La Figure 4.2 montre un méta-modèle de services web selon sa description WSDL, un service web est modélisé comme un ensemble d'opérations représentant les échanges des messages.

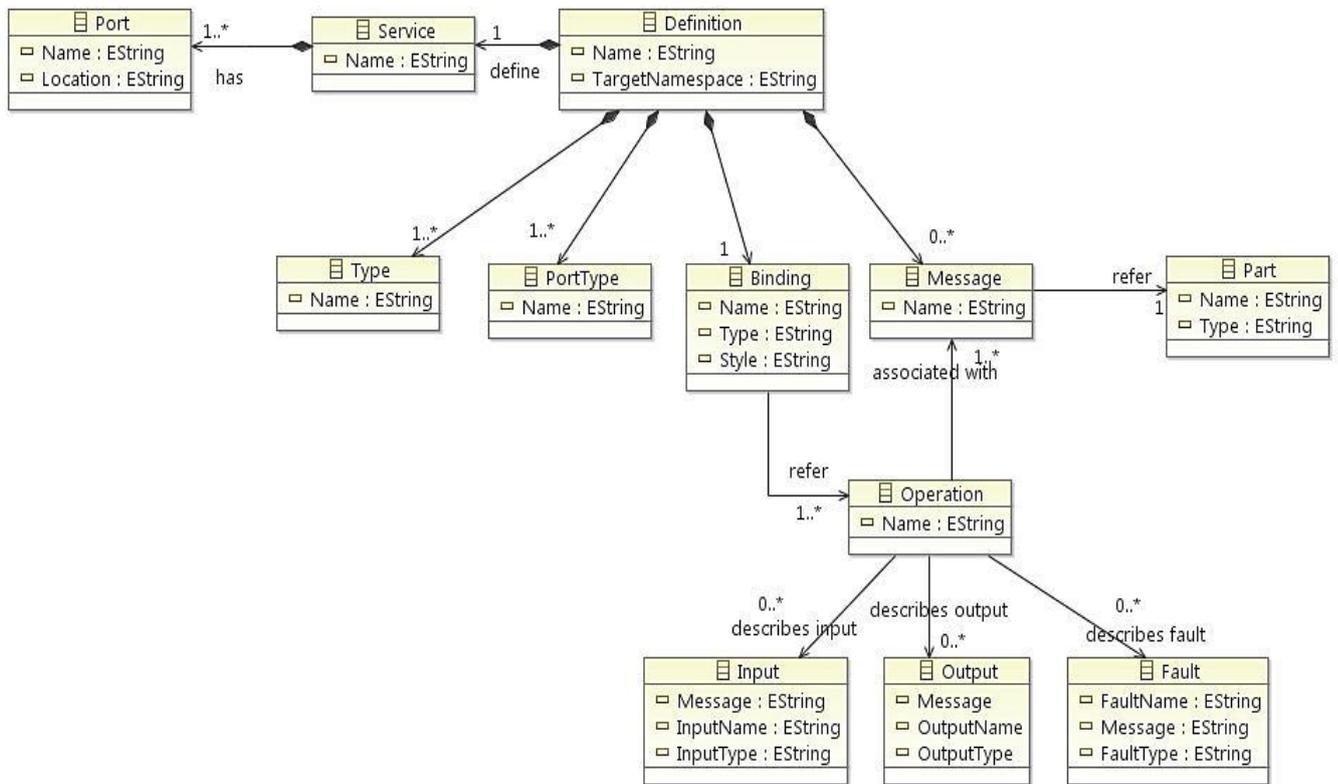


Figure 4.2 : méta-modèle de services web selon la description WSDL

- L'élément "**définition**" permet la définition du nom de services ainsi que la déclaration des types de données suivant les types XML Schéma, des schémas XML peuvent être importés. Cet élément est l'élément racine de toute description WSDL
- Les éléments **message** sont des descriptions abstraites de ce qu'il peut être communiqué, c'est-à-dire les messages entrants et sortants, et leurs types.
- Les types sont définis dans l'élément **part** et peuvent référencés les types déjà définis
- Un "**portType**" peut contenir un ou plusieurs opérations ; chaque opération spécifie une action effectuée par le service web et les messages qui peuvent être envoyés à partir et au service par le biais des éléments « **input** » et « **output** ». L'élément « **Fault** » est un élément optionnel et permet d'identifier les erreurs.
- L'élément "**binding**" décrit le format des messages, les détails des opérations et les messages définis par le « **portType** » c'est-à-dire le mapping wsdl-to-soap du service.

Or, cette description de services web ne supporte pas la description sémantique des services. Pour combler cette lacune, des techniques de représentation de connaissances développées pour le web sémantique sont utilisées. (voir section2)

iii. SOAP (Simple Object Access Protocol)

C'est le protocole de transmission de messages entre services web, standardisé par le W3C, il permet de définir les messages échangés entre un client et un fournisseur de services web à travers un protocole tel que HTML, SMTP, FTP... en se basant sur le langage XML.

Deux modes de communication du protocole SOAP existent :

- Le mode RPC (Remote Procedure Call) qui permet un appel synchrone de procédures
- le mode Messagerie permet de fonctionner au choix, en mode synchrone ou synchrone.

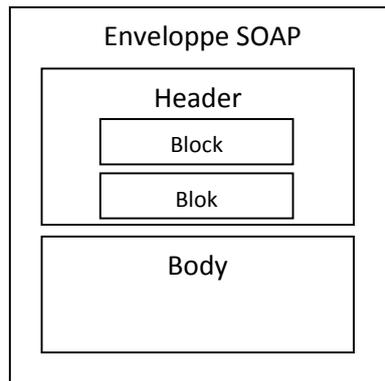


Figure 4.3 : Structure d'un message SOAP

Un message SOAP est structuré en quatre blocs :

- **l'enveloppe** SOAP, c'est l'élément racine qui définit le contexte d'un message, son destinataire, son contenu et d'autres options; Il peut contenir un attribut `encodingStyle` dont la valeur est une URL vers un fichier XML qui décrit les types appliqués dans le message SOAP.
- **l'en-tête est un élément facultatif**, permettant d'ajouter à un message SOAP des attributs spécifiques à la transaction ;
- **le corps** : c'est lui qui contient le contenu du message à transporter, incluant les détails d'une demande SOAP, et permettant généralement de décrire le nom de la méthode et les valeurs des paramètres d'appel dans le cas d'une demande de service ou les valeurs des paramètres de retour dans le cas d'une réponse. C'est un élément obligatoire et doit apparaître une seule fois.
- **Fault**: cet élément permet de reporter des erreurs lors du traitement du message, ou lors de son transport. Il ne peut apparaître qu'une seule fois par message. Sa présence n'est pas obligatoire.

```

POST /StockQuote HTTP/1.1 Host: www.stockquoteserver.com Content-Type:
text/xml; charset="utf-8" Content-Length: nnnn SOAPAction: "Some-URI"
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"> <soapenv:Body>
<m:GetLastTradePrice xmlns:m="Some-URI">
<m:tickerSymbol>DIS</m:tickerSymbol> </m:GetLastTradePrice> </soapenv:Body>
</soapenv:Envelope>
  
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:q0="http://DefaultNamespace"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<q0:getInfosPatient>
<q0:SSN>0</q0:SSN>
<q0:output>Last name</q0:output>
</q0:getInfosPatient>
</soapenv:Body>
</soapenv:Envelope>
```

Exemple de requête SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <soapenv:Body>
- <getInfosPatientResponse xmlns="http://DefaultNamespace">
<getInfosPatientReturn>Jean</getInfosPatientReturn>
</getInfosPatientResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Exemple de réponse SOAP

iv. UDDI

Universal Description, Discovery and Integration est une spécification pour la publication de services web, créée par IBM, Microsoft et Ariba, il permet aux utilisateurs de trouver un service à travers un registre de services centralisé, l'objectif est de construire un référentiel commun pour décrire et localiser des entreprises ou des services d'entreprises sous le format XML. L'interface du registre elle-même est implémentée sous

forme de service web SOAP et La communication avec l'UDDI est effectuée à travers des messages XML encapsulés dans un message SOAP. [63][64]

Les informations enregistrées au sein d'un UDDI sont organisées dans types de structures de données qui sont(Figure 4.4) :

- *businessEntity*, permet de fournir les informations concernant le fournisseur de services incluant le nom, informations de contact et sa localisation. Il peut contenir d'autres éléments businessServices;
 - *businessService*, permet de définir les informations sur une famille de services techniques, il peut comporter des éléments bindingTemplates
 - *bindingTemplate*, contient des informations sur des points d'entrée c'est-à-dire les informations nécessaires pour invoquer le service, des références à un ou plusieurs tModels, ces références désignent des spécifications d'interface pour le service ;
 - *tModels(technical Models)* contient des descriptions de spécifications
- Un autre élément peut exister qui est le *PublisherAssertion* permettant de définir les liens entre entreprises.

L'ensemble de ces éléments forment :

- *les pages blanches* : fournissant les données à propos du fournisseur
- *les pages jaunes* : définissent les types de services offerts
- et *les pages vertes* : donnent les informations techniques sur les services

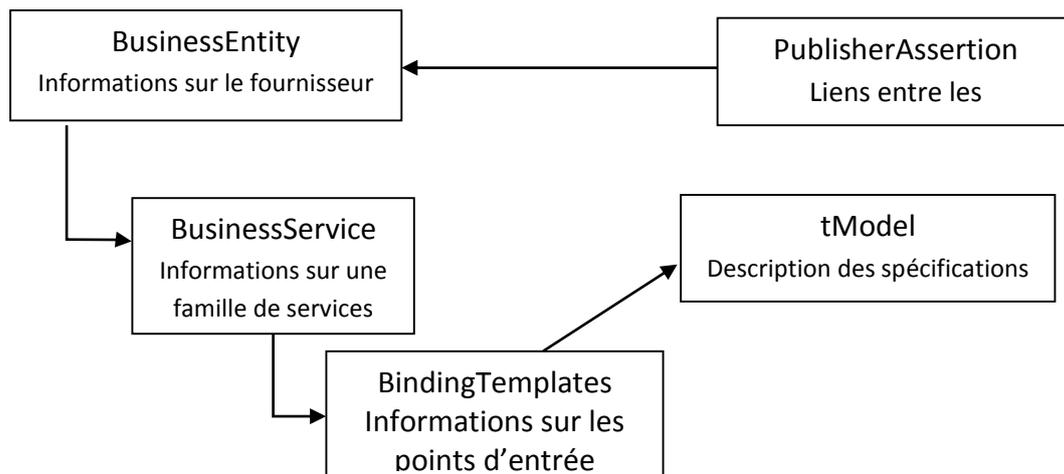


Figure 4.4 : Structure d'un annuaire UDDI

6.2.2. Services web sémantiques

La description WSDL permet de décrire les services web au niveau syntaxique, et ne permet pas d'exprimer la sémantique de ces services, dans un autre côté le domaine du web sémantique gagne de plus en plus d'importance dans l'intégration des données grâce à sa capacité de représenter les connaissances pour cela les techniques du web sémantique ont été utilisées.

Plusieurs standards ont été développés afin de décrire les services web sémantiques. On distingue deux types d'approches pour la description sémantique de services web, celles qui basent sur les annotations comme le WSDL-S et le SAWSDL et celles qui se basent sur les langages sémantiques tel que le WSMO et le OWL-S.

Toutes ces approches utilisent les ontologies pour décrire sémantiquement les web services, il existe d'autres langages pour la composition de services web comme le langage BPEL(Business Process Execution Language). Dans ce qui suit nous détaillons quelques langages pour la description sémantique des services web.

4.3.2.1. OWL-S

C'est une ontologie de web service reposant sur le langage OWL, son objectif est d'ajouter des descriptions sémantiques aux services afin d'automatiser les tâches effectuées par le service, incluant la découverte, l'exécution et la composition de services, où un service est considéré comme une ontologie OWL contenant trois parties essentielles qui sont, elles même des sous ontologies :

- Le *Profile* permet de définir ce que le service peut faire en fournissant une description abstraite du service et les informations sur le fournisseur.
- Le *Process* contient des informations descriptives sur la fonctionnalité du service et de sa composition avec d'autres services.
- Le *Grounding* permet de définir comment accéder au service, en définissant les détails concrets sur le service. [72]

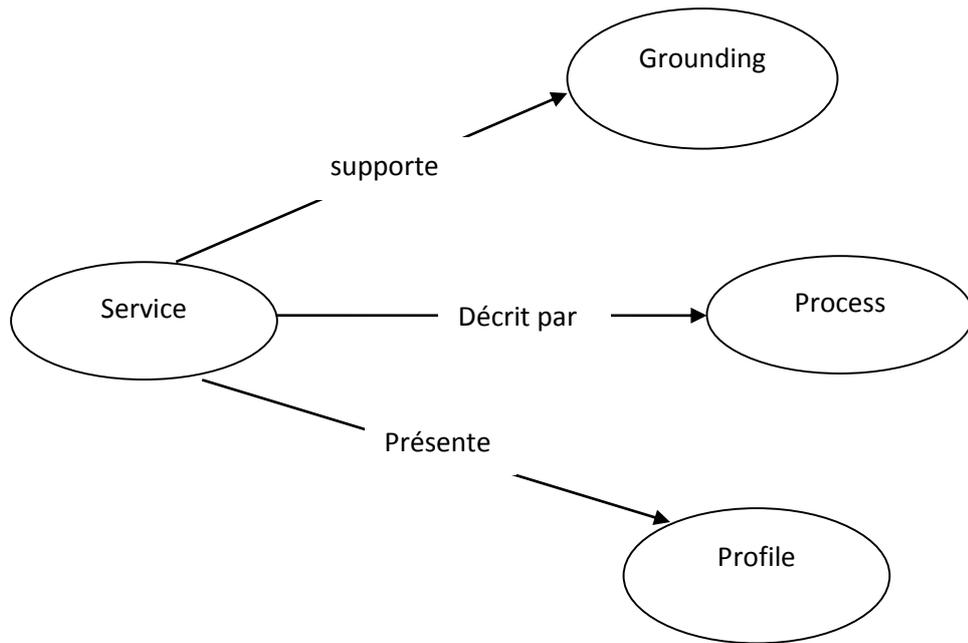


Figure 4.5 : Ontologie OWL-S

Les premières versions de cette ontologie étaient connues sous le nom DAML-S [77] et construites avec le langage DAML+OIL [78] qui est le prédécesseur du langage OWL.

4.3.2.2. WSMO

Web Service Modeling Ontology, est une extension du WSMF (Web Service Modeling Framework) [74] [75]. C'est une ontologie permettant de définir les différents éléments pour décrire les web services sémantiques. Ces éléments incluent les ontologies, les buts, les services web et les médiateurs.

- *Les ontologies* permettent de définir la terminologie utilisée par d'autres éléments WSMO pour décrire les aspects pertinents ;
- *Les buts (goals)* indiquent ce que l'utilisateur désire.
- Les éléments *services web* définissent les fonctionnalités offertes par le service, ils décrivent les entités permettant d'effectuer les traitements c'est-à-dire le fonctionnement interne du service.
- les *Médiateurs* décrivent les éléments qui permettent de surmonter les problèmes d'interopérabilité entre les différents éléments de WSMO.

Le langage WSML (Web Service Modeling Language) [76] est utilisé pour décrire les éléments de WSMO d'une manière formelle. WSMX (Web Service Modelling eXecution environment) [82] est un environnement d'exécution de services web permettant la découverte, la sélection, la médiation et l'invocation des services sémantiques. Il est basé sur le modèle conceptuel offert par le WSMO.

4.3.2.3. WSDL-S

Le WSDL-S est une extension du WSDL, il permet d'ajouter des annotations à la description WSDL [73], ces annotations peuvent être des références à des concepts d'ontologies, les éléments ajoutés à la description sont décrits dans ce qui suit :

- L'élément *modelReference* permettant de gérer les mappings des éléments du schéma avec des concepts d'un modèle sémantique.
- L'attribut *schemaMapping* : cet attribut est ajouté aux types complexes et aux éléments afin d'associer les éléments
- Deux nouveaux éléments **precondition** et **effect** qui sont des éléments fils de l'élément **operation**, ils sont utilisés pour la découverte de services web.
- L'attribut **category**, permet de définir la fonctionnalité du service

6.2.3. Services web pour l'intégration de données

Plusieurs systèmes offrent un accès à leurs sources de données à travers les services web, ces données ont besoin d'être intégrées à travers une interface commune. Dans cette partie nous citons quelques travaux dans la littérature qui utilisent les services web pour l'intégration des données.

Dans [93] une approche pour l'intégration de données basée sur les services web est proposée dans le domaine des télécommunications. Dans ce travail on définit la notion d'agrégation comme un cas particulier de l'intégration, pour cela, un agrégateur est mis en place. Cet agrégateur représente une entité qui permet de collecter et d'analyser de manière transparente les informations à partir de sources de données différentes, permettant de résoudre le problème d'hétérogénéité contextuelle et sémantique.

Des approches comme proposée par [95] et [4] définissent la notion de DAS (ou DAAS pour Data As A Service) pour les services web qui permettent d'accéder à des sources de données, où le fournisseur du service publie ses données à travers les services web pour avoir un accès uniforme à ces données. Ces approches adressent en général le

problème de composition de services. La figure 4.6 montre l'architecture proposée par Zhu F. et al. [95], cette architecture est caractérisée par la mise en place d'un broker, qui permet de réconcilier les données provenant de différentes sources.

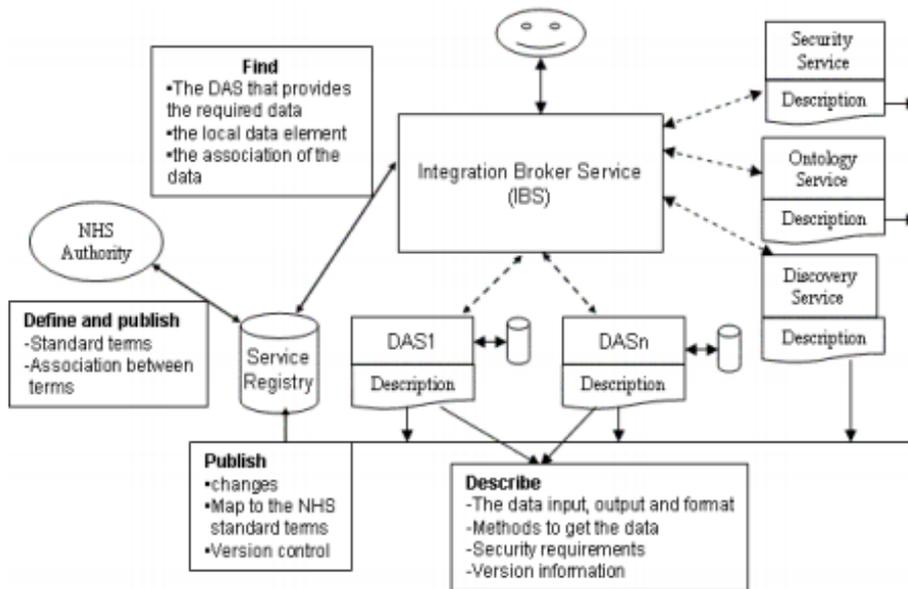


Figure 4.6 : Architecture proposée par Zhu F. et al. [95]

Fujun Zhu et al. [80] proposent un système d'intégration à base de services web dans un environnement fédéré. Abiteboul et al. [81] Développent un nouveau langage basé sur XML appelé Active XML où on considère des éléments XML comme des appels aux services web afin de permettre une vue uniforme de sources de données différentes.

4.4. Conclusion

Des techniques se développent de plus en plus pour gérer les données distribuées de manière pertinente. Parmi ces techniques l'utilisation des services web, qui permettent l'interopérabilité des applications distribuées. Un service web est une application accessible et invocable par d'autres applications utilisant les protocoles d'internet tel que le HTTP et le XML permettant l'échange d'informations provenant de machines différentes et écrits avec des langages différents.

Dans ce chapitre nous avons étudié les services web, particulièrement les services web sémantiques, les différents standards liés, les langages des services web sémantiques, ainsi que quelques travaux qui utilisent les services web pour effectuer l'intégration de données.

A travers les travaux cités si dessus, nous concluons que les services web sont utilisés dans un système d'intégration pour la traduction de requêtes entre le médiateur et les sources, et jouent le rôle de wrapper dans une architecture médiateur-wrapper. Dans notre contribution, en plus du rôle de wrapper, le service web peut effectuer les traitements sur les données.

Partie II :

Contributions

Chapitre 5 : Architecture de médiation

Sommaire

5.1.	Introduction	65
5.2.	Méta-modèle de service web	66
5.3.	Méta-modèle de l'ontologie.....	69
5.4.	Extension du méta-modèle en considérant les opérateurs flous.....	71
5.5.	UDDI sémantique	73
5.6.	Architecture globale de médiation	78
5.7.	Conclusion.....	80

5.1. Introduction

Les chapitres précédents ont permis de faire un état de l'art sur les différentes approches d'intégration à base d'ontologies et de services web, ceci a permis de positionner nos travaux par rapport à l'état de l'art. Ce chapitre permet de présenter les contributions de notre travail de thèse dans lequel nous nous intéressons à l'intégration de données par médiation sémantique.

Nous proposons dans un premier lieu un méta-modèle de services web, après l'étude de sa description WSDL (version 1.0). Dans ce méta-modèle nous essayons de modéliser les différentes opérations qui peuvent être effectuées par le service ensuite nous le connectons au méta-modèle d'ontologie. Nous étendons ce méta-modèle afin de prendre en considération les opérations floues qui peuvent être effectuées par un service web, ceci nous amène à proposer une approche de stockage de services web au niveau d'un UDDI sémantique en se basant sur les bases de données à base ontologique, ensuite nous détaillons l'architecture de médiation proposée.

Soulignons que dans notre système, la construction de l'ontologie n'est pas un but en soi, pour cela nous faisons l'hypothèse pour la réalisation de notre approche, qu'une ontologie partagée couvrant le domaine étudié existe et qui joue le rôle du schéma global, pour se faire nous avons réutilisé d'autres ontologies existantes (ou des parties de ces ontologies) pour la construction de notre propre ontologie comme UMLS, OpenGalen...

Ce chapitre est divisé en quatre parties, dans la première partie nous décrivons le méta-modèle de service web proposé, dans la deuxième partie nous présentons l'extension de ce méta-modèle en considérant les opérateurs flous, la troisième partie présentera notre proposition pour un UDDI sémantique qui repose sur les bases de données à base ontologique, nous montrons comment un service web peut être décrit par une ontologie et est stocké au niveau d'une base de données sémantique. Dans la quatrième partie nous détaillons l'architecture globale de notre système de médiation.

5.2. Méta-modèle de service web

Les systèmes d'intégration reposent de plus en plus sur l'architecture orientée services, en particulier l'utilisation des services web. Un service web est une application accessible et invocable par d'autres applications utilisant les protocoles d'internet tel que le HTTP et le XML permettant l'échange d'informations provenant de machines différentes et écrites avec des langages de programmation différents permettant l'interopérabilité des systèmes distribués.

Nous avons vu à travers les travaux étudiés dans le chapitre précédent que les services web sont généralement utilisés dans les systèmes d'intégration pour la traduction de requêtes entre le médiateur et les sources, et jouent le rôle de wrappers (adaptateurs) dans une architecture médiateur-wrapper, or un service web peut effectuer des traitements sur les données, ces traitements peuvent varier de simples opérations à des opérations plus complexes. Pour cela notre première contribution consiste à proposer un méta-modèle de service web dans lequel nous modélisons les différents types d'opérations pouvant être effectuées par un service.

Les services web sont caractérisés par l'utilisation d'une pile de standards pour la description de leur interface, l'échange de données entre services et leur publication. Le WSDL est le langage qui permet la description de services web, basé sur le langage XML, il permet de décrire les fonctionnalités du service c'est-à-dire les opérations qu'il peut effectuer, sa localisation et comment il peut être invoqué.

Une description WSDL est constituée d'un ensemble d'éléments (voir Figure 4.2, Chapitre 4, section 4.1.3) :

- L'élément "**définition**" permet la définition du nom du service ainsi que la déclaration des types de données suivant les types XML Schéma, des schémas XML peuvent être importés. Cet élément est l'élément racine de toute description WSDL.
- Les éléments "**message**" sont des descriptions abstraites de ce qu'il peut être communiqué, c'est-à-dire les messages entrants et sortants, et leurs types.
- Les types sont définis dans l'élément "**part**" et peuvent référencer les types déjà définis.

- Un “**portType**” peut contenir un ou plusieurs opérations ; chaque opération spécifie une action effectuée par le service web et les messages qui peuvent être envoyés ou reçus par le service à travers les éléments “**input**” et “**output**”. **un** autre élément peut exister qui est l’élément “**Fault**” c’est un élément optionnel permettant d’identifier les erreurs.
- L’élément “**binding**” décrit le format des messages, les détails des opérations et les messages définis par le “**portType**” c'est-à-dire le mapping wsdl-to-soap du service.
- L’élément “**operation**” représente une définition abstraite d’une opération effectuée par le service.

```
<definitions>  
  <types>  
  </types>  
  <message>  
  </message>  
  <portType>  
  </portType>  
  <binding>  
  </binding>  
</definitions>
```

Figure 5.1 : Structure d’un document WSDL

La Figure 5.1 montre en XML la structure d’un document WSDL, nous proposons un méta-modèle (Figure 5.2) où les différents types d’opérations qui peuvent être effectuées par un service web sont considérés.

Une **opération** d'un service web est une **expression** qui peut être une **fonction**, une expression **conditionnelle**, **itérative** ou bien une expression **floue** (nous la détaillons dans la section 5.4). Cette expression contient des variables représentant les **inputs**, les **outputs** des services et les opérateurs ; un opérateur peut être un opérateur **logique**, **arithmétique** ou bien une **relation** entre variables. Une expression peut **contenir** (représentée par la relation « **containment** » au niveau du méta-modèle) une autre expression, peut être un **raffinement** (relation « **refinement** ») d'une autre expression ou bien **requiert** (relation « **requirement** ») une autre expression. [15]

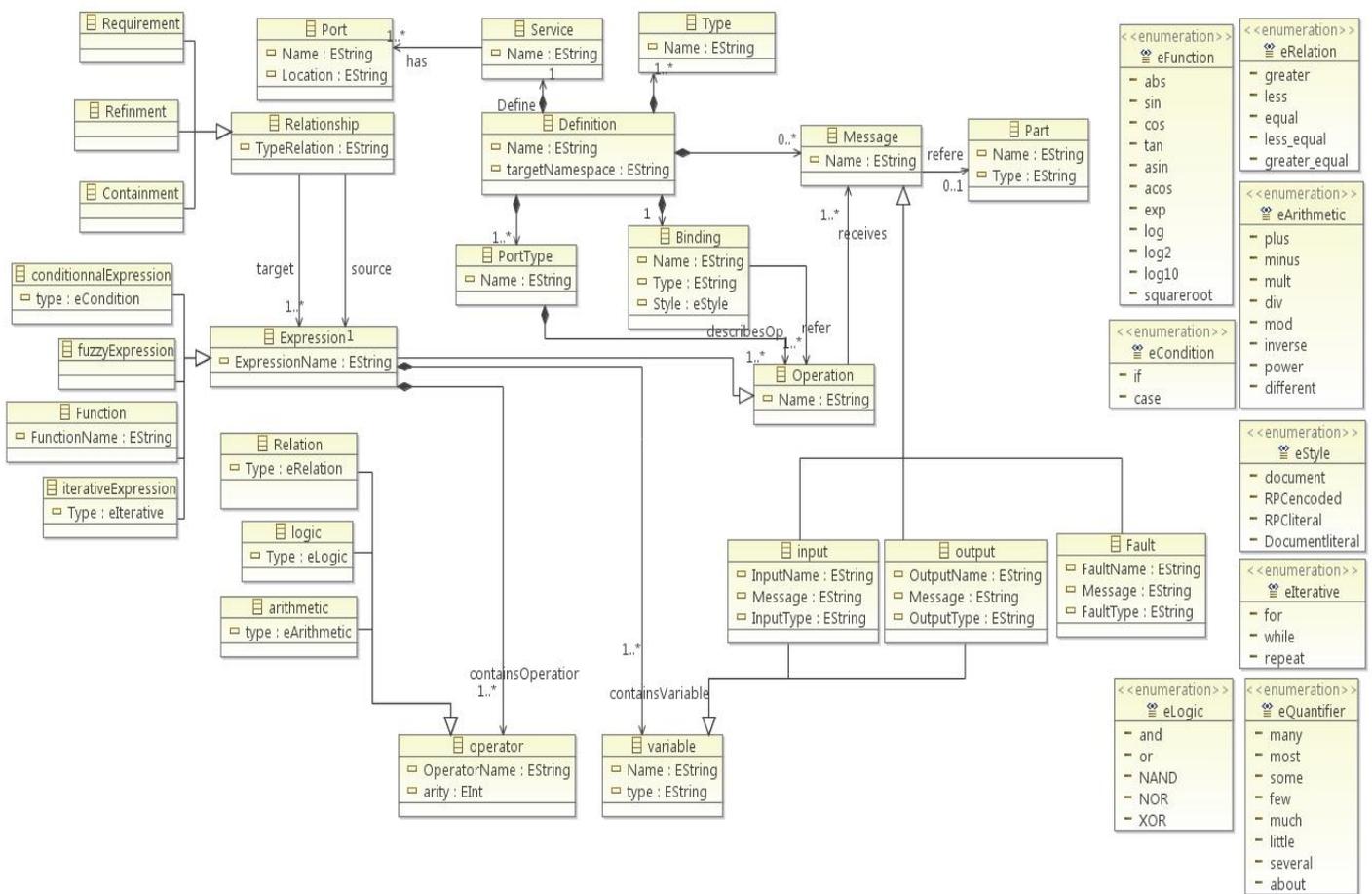


Figure 5.2 : Méta-modèle de service web proposé

L’instanciation du méta-modèle donne lieu à un modèle de services qui pourra être utilisé par un ensemble de personnes ou d’applications. Ce méta-modèle a été connecté au méta-modèle de l’ontologie.

5.3. Méta-modèle de l’ontologie

Une ontologie est constituée d’un ensemble de classes regroupées dans le méta-modèle dans la classe « class », d’un ensemble de propriétés (classe Property), ces propriétés peuvent être de types objet (Object Property) ou bien des propriétés de données (Data Property). La figure 5.3 représente le méta-modèle d’une ontologie.

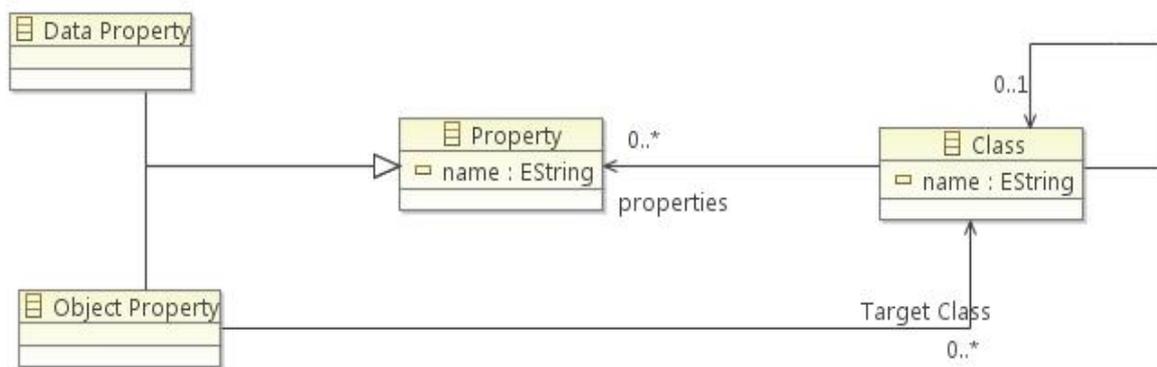


Figure 5.3 : Méta-modèle d’ontologie

5.4. Connexion du méta-modèle de service web avec le méta-modèle de l’ontologie

Le méta-modèle de service web a été connecté au méta-modèle de l’ontologie (Figure 5.4) où les variables et les opérateurs sont représentés comme des propriétés des éléments d’une ontologie.

Cette connexion du méta-modèle avec l’ontologie permet de considérer les classes d’un service web comme des classes ontologiques qui pourraient être exploitées au niveau d’une base de données sémantique.

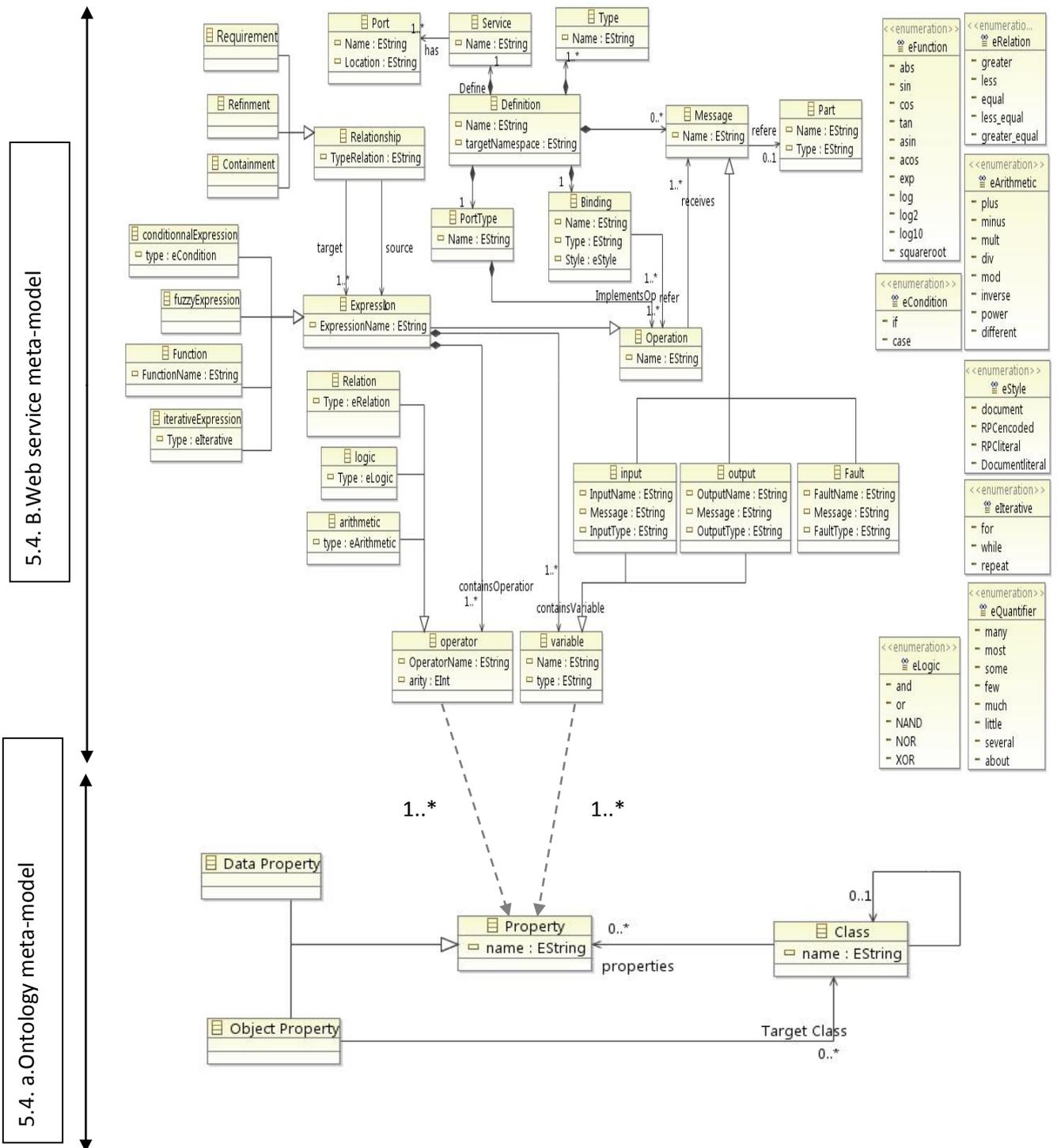


Figure 5.4 : Méta-modèle de service web connecté au méta-modèle d'ontologie

5.5. Extension du méta-modèle en considérant les opérateurs flous

Comme nous l'avons déjà vu dans la section précédente, un service web permet aussi d'effectuer des opérations de types flous pour cela nous avons étendu le méta-modèle proposé dans la première partie afin de prendre en considération les opérateurs flous.

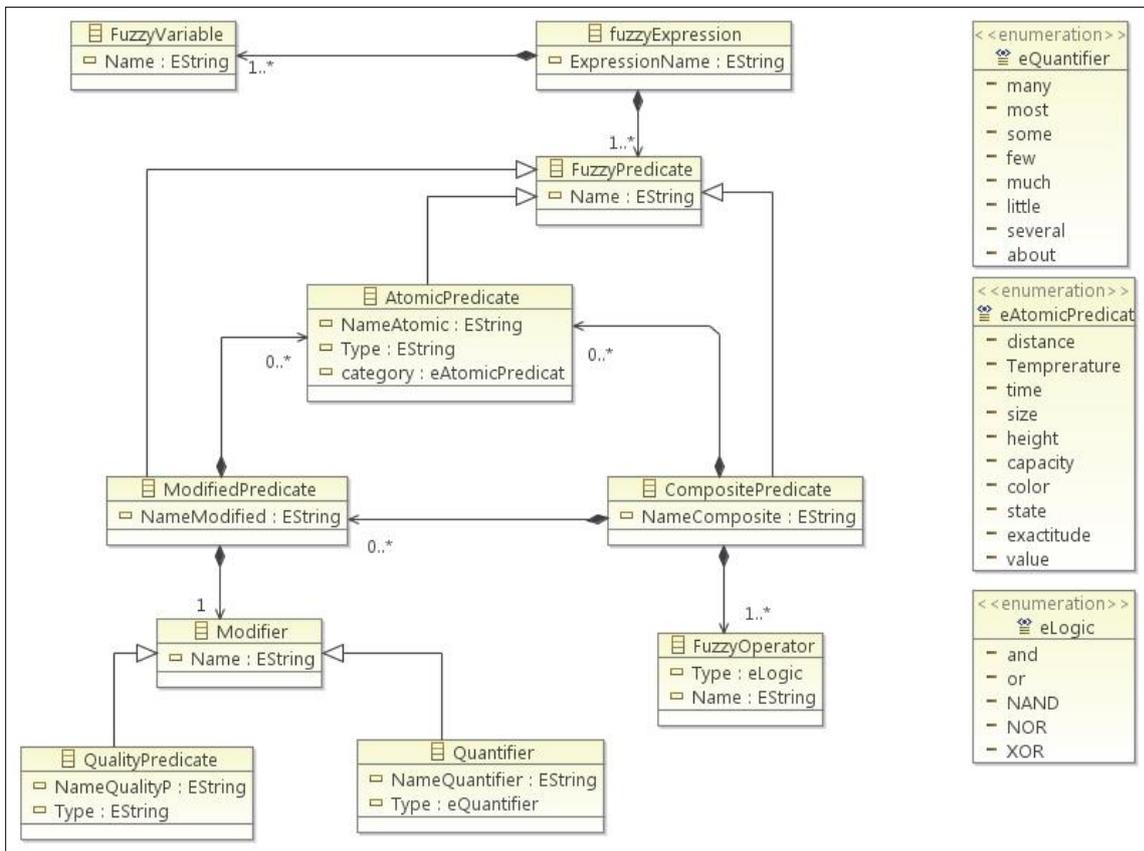


Figure 5.5 : Méta-modèle de services web en prenant en considérations les opérations floues

Une expression floue contient des **variables** et des **prédicats** flous, selon [84] ces prédicats peuvent être **atomiques**, **modifiés** ou **composés** :

- Les prédicats atomiques : correspondent généralement à des adjectifs.

Un prédicat atomique peut exprimer :

- ✓ la distance : comme proche, loin, court...
- ✓ la température : chaud, froid, tiède

- ✓ le temps : récent, ancien...
 - ✓ taille : grand, petit, large...
 - ✓ altitude : haut, bas...
 - ✓ capacité : puissant, fort, élevé, faible....
 - ✓ couleur : foncé, clair
 - ✓ un état : grave, simple, jeune, sec....
 - ✓ valeur : cher, coûteux, important
 - ✓ exactitude : précis, exact, certain, sûr, concis...
- Les prédicats modifiés : ils correspondent à des adjectifs dont le sens est modifié au moyen d'un adverbe tel que très, plus, on appelle cet adverbe un modificateur « modifier » où il peut exprimer une certaine propriété/dimension, nous pouvons citer : les **quantificateur** (ex :plus, moins...), un prédicats exprimant une qualité qu'on note « **quantityPredicate** »(ex : bon, doux, raisonnable)

Exemple 1 : la température est **plus élevée**

- Les prédicats composés : les prédicats peuvent être combinés grâce à des opérateurs logiques, ils peuvent être une combinaison de prédicats atomiques et/ou modifiés d'où les cardinalités 0..*.

Exemple 2 : Le poids du patient est **plus élevé et** sa taille est **petite et** il est en **bonne santé**.

Cette expression est une expression floue, représentant une instance du méta-modèle représenté dans la Figure 5.5 ; où la donnée « poids » est une instance de la classe « FuzzyVariable », « plus élevé » est une instance de la classe CompositePredicate, composée d'un quantificateur « plus » et d'un prédicat atomique « élevé ». Le « et » est une instance de la classe FuzzyOperator qui est de type eLogic, « petite » est une instance de la classe AtomicPredicate, « bonne santé » est une instance de la classe CompositePredicate où « bonne » est un prédicat de type « quantityPredicate ».

5.6. UDDI sémantique

Comme mentionné dans le chapitre 4, un UDDI permet d'enregistrer les services web afin qu'ils puissent être découverts par des personnes ou par des applications ayant l'objectif de construire un référentiel commun pour décrire et localiser les services.

La description de services web joue un rôle très important et elle varie entre les standards adoptés par les entreprises, la plupart de ces standards utilisent les ontologies, cette description est publiée dans un registre UDDI, généralement sous forme de documents XML.

Les UDDI classiques n'offrent pas la possibilité de représenter les métadonnées sur les services web. Ces métadonnées permettent de capturer la sémantique des services et sont nécessaires pour leur découverte. Des travaux sur l'enrichissement sémantique de services ont été réalisés en se basant essentiellement sur les langages de description sémantique des services web [114][115][116], parmi ces langages ceux qui proposent de décrire un service web par une ontologie par exemple OWL-S, WSMO... (Voir chapitre 4, section 4.3.2).

Puisqu'un service web peut être décrit par une ontologie, il pourra être donc stocké au niveau d'une base de données à base ontologique, pour cela nous proposons un nouveau mode de stockage de services web en se basant sur les BDBO [112]. La Figure 5.6 montre comment les descriptions de services web sont stockées au niveau de la base de données sémantique.

- La partie 'meta-schema' représente le schéma de l'ontologie des services
- La partie 'Data-schema' est la partie traditionnelle des bases de données relationnelles.
- La partie 'Service Ontology Instances' contient les données qui sont des instances de l'ontologie du service

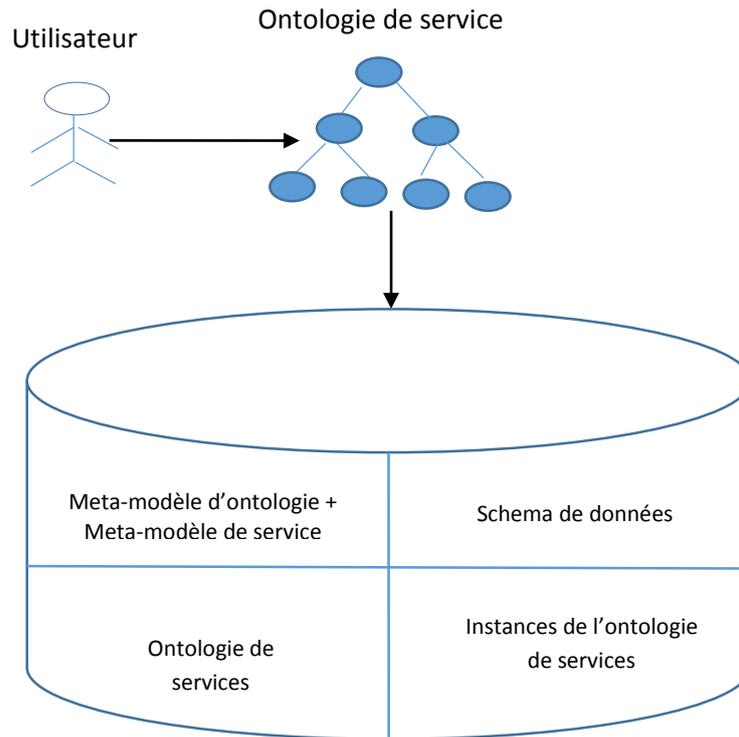


Figure 5.6 : Registre sémantique pour services web

- Etude de cas : le langage OWL-S

Dans cette section, nous prenons l'exemple de l'ontologie OWL-S et comment peut-on représenter ses concepts et ses propriétés au niveau d'une BDBO.

OWL-S est une ontologie de web service reposant sur le langage OWL, son objectif est d'ajouter des descriptions sémantiques aux services afin d'automatiser les tâches effectuées par le service, incluant la découverte, l'exécution et la composition de services, où un service est considéré comme une ontologie OWL (voir Figure 4.5, Chapitre 4 section 4.3.2.1) contenant trois parties essentielles qui sont, elles même des sous ontologies :

- Le concept « *Profile* » : permet de définir ce que le service peut faire en fournissant une description abstraite du service et les informations sur le fournisseur.
- Le concept « *Process* » : contient des informations descriptives sur la fonctionnalité du service et de sa composition avec d'autres services.

- Le concept « Grounding » permet de définir comment accéder au service, en définissant les détails concrets sur le service.

Les concepts de l'ontologie OWL-S peuvent être représentés au niveau de la BDBO pour cela trois classes sont créées : classe « Profile », classe « Process » et la classe « Grounding », ensuite deux sous-ontologies existent pour représenter les concepts « profile » (Figure 5.7) et « process » (Figure 5.8)

La requête OntoQL suivante est utilisée pour créer le concept « profile » :

```
CREATE #CLASS Profile (DESCRIPTOR(#name[fr]=Profile))
```

Ceci permet de créer une classe Profile avec le descripteur nom[fr] (nom en français de la classe) qui a la valeur Profile, d'autres descripteurs peuvent être créés.

Nous prenons l'exemple de la sous ontologie « Profile » où la classe « Profile » a comme propriétés :

- “serviceName” Qui réfère au nom du service, cette propriété peut être utilisée comme identifiant du service
- “textDescription” C'est une brève description du service résumant ce que le service requiert pour son fonctionnement
- “contactInformation” permet d'offrir des informations référant à des personnes responsables du service

La requête OntoQL suivante est utilisée pour créer la propriété « serviceName » de la classe « Profile » :

```
ALTER #CLASS Profile ADD serviceName String DESCRIPTOR (#name[fr] =  
    'nomService', #definition[fr] = 'Nom du service')
```

Ceci permet de créer la propriété « serviceName » avec le descripteur « name[fr] » : 'nomService', et le descripteur definition[fr] ayant la valeur 'Nom du service' permettant de donner une définition en français de la propriété.

De la même manière nous créons les classes et les propriétés au sein de chaque sous-ontologie de l'ontologie globale OWL-S.

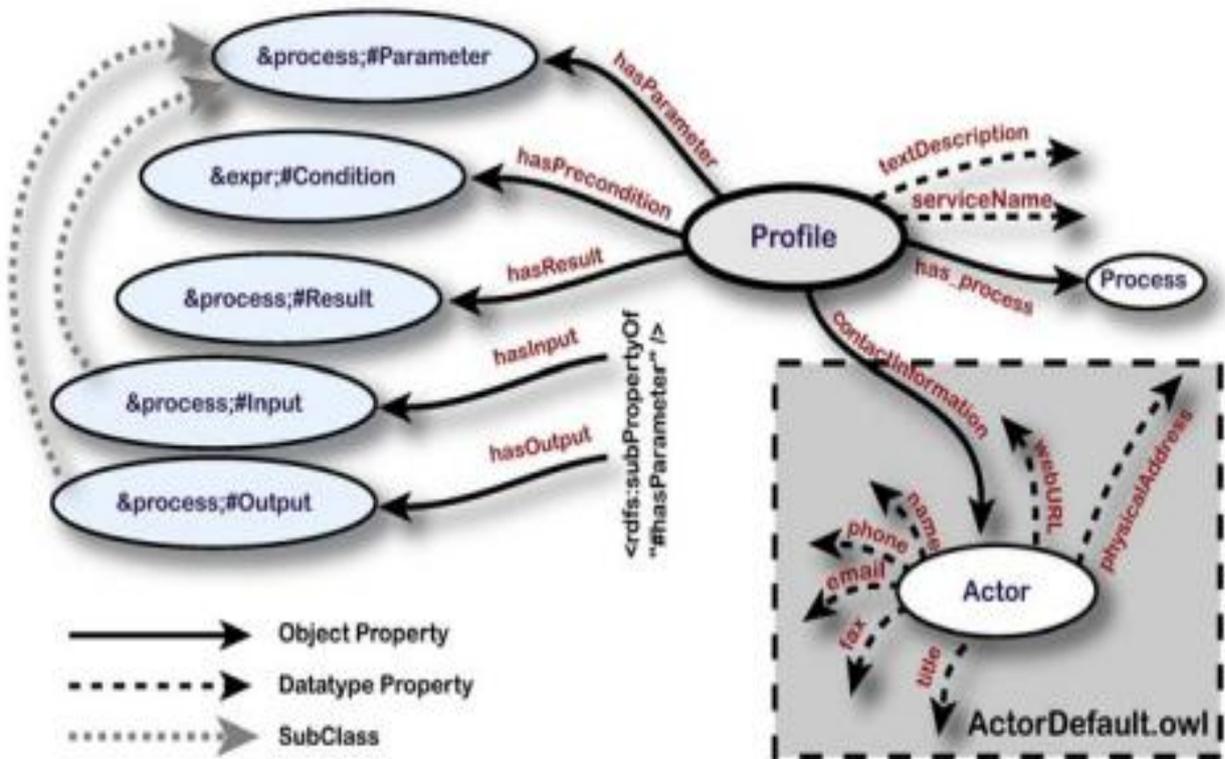


Figure 5.7 : Sous-ontologie d'OWL-S représentant le profile [72]

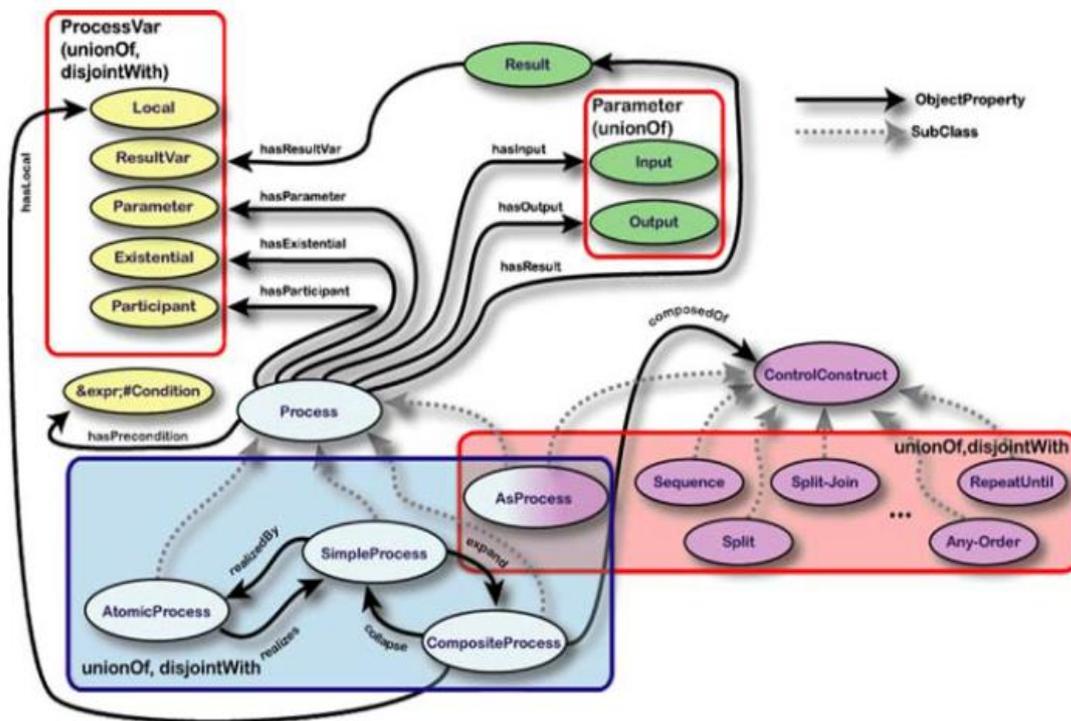


Figure 5.8 : Sous-ontologie d'OWL-S représentant le Process [72]

Afin de pouvoir manipuler les données au niveau de la base de données OntoDB des extensions de classes doivent être créées. Par exemple la requête suivante permet de créer une extension de la classe service :

```
CREATE EXTENT OF Service ("serviceName", "TextDescription",
                          "ContactInformation")
```

Ceci permet de créer une table service contenant les colonnes « serviceName », « TextDescription » et « ContactInformation »

Enfin La requête suivante permet d'insérer les données au niveau de la table service

```
INSERT INTO Service (serviceName, TextDescription, ContactInformation)
VALUES ('getInfosPatient', 'Service qui permet de récupérer les
      informations sur le patient', 'phone, 040587145');
```

5.7. Architecture globale de médiation

Comme nous l'avons déjà mentionné au début de ce chapitre, nous considérons dans cette section que chaque élément de l'architecture (le médiateur, les services et les sources de données) est décrit par la même ontologie qui joue le rôle du schéma global.

Les descriptions des services web sont stockées au niveau de la base de données globale ; chaque description inclut la localisation du service, les entrées, les sorties ...de chaque opération effectuée par le service. Cette base de données est considérée comme un registre de services web où chaque service web doit être enregistré pour qu'il puisse être trouvé par le médiateur ainsi que par d'autres applications.

Les descriptions des services web sont stockées de la même manière que le stockage des données au niveau de la base de données sémantique, par la définition des concepts et des propriétés du service, puis les extensions du modèle de service sont créées où chaque service peut être enregistré.

L'extension de service web inclut l'identifiant (oid) du service, son nom, son localisation, et les noms des opérations effectuées par ce service. Les extensions des opérations incluent leurs noms, l'identifiant du service, ses entrées et ses sorties. Pour chaque service, cette information doit être fournie dans la base de données globale.

En premier lieu, le médiateur traite la requête de l'utilisateur, localise puis sélectionne les services web liés, ensuite il fournit un plan d'invocation de services web qui doivent être invoqués. Finalement il rassemble les résultats et les retourne à l'utilisateur.

La requête définie sur l'ontologie du médiateur est traitée selon les opérations stockées dans la base de données. Il sélectionne les opérations qui ont les entrées et les sorties désignées dans la requête. Il sélectionne les services contenant ces opérations, il invoque les services concernés, il recompose les résultats et enfin il les retourne à l'utilisateur.

Le service reçoit les paramètres associés à l'information (sorties) nécessaires pour certaines entrées.

Le médiateur sélectionne l'opération qui a les entrées et les sorties liées à la requête ainsi que la localisation du service qui effectue cette opération à travers un ensemble de modules :

- Le module 'Query rewriter' : permet de réécrire la requête de l'utilisateur exprimée en termes de l'ontologie globale selon les descriptions de services web stockées dans la base de données. Les descriptions de services web représentent des vues sur le schéma global. Cette réécriture est basée sur l'approche de la réécriture de requête proposée par halevy[41] où la requête de l'utilisateur est une conjonction de requêtes.
- Le module 'services selector' permet de sélectionner les services correspondant aux opérations selon les entrées et les sorties contenues dans les sous-requêtes. Ces entrées et ces sorties sont des propriétés de classes des descriptions de services web qui sont stockées au niveau d'une base de données à base ontologique
- Le module 'planner' permet d'invoquer les services selon l'ordre des classes de la requête de l'utilisateur.
- Le module 'Results reconciler' permet de réconcilier les résultats obtenus à partir des services web invoqués et de les retourner à l'utilisateur.

- **Formalisation**

De manière formelle le système d'intégration est représenté en un triplé $\langle G, S, M \rangle$ où :

G : représente le schéma global qui est l'ontologie.

S : représente l'ensemble des schémas des descriptions de services web qui représentent des vues sur le schéma global.

M : représente les correspondances entre l'ontologie globale et les schémas des sources.

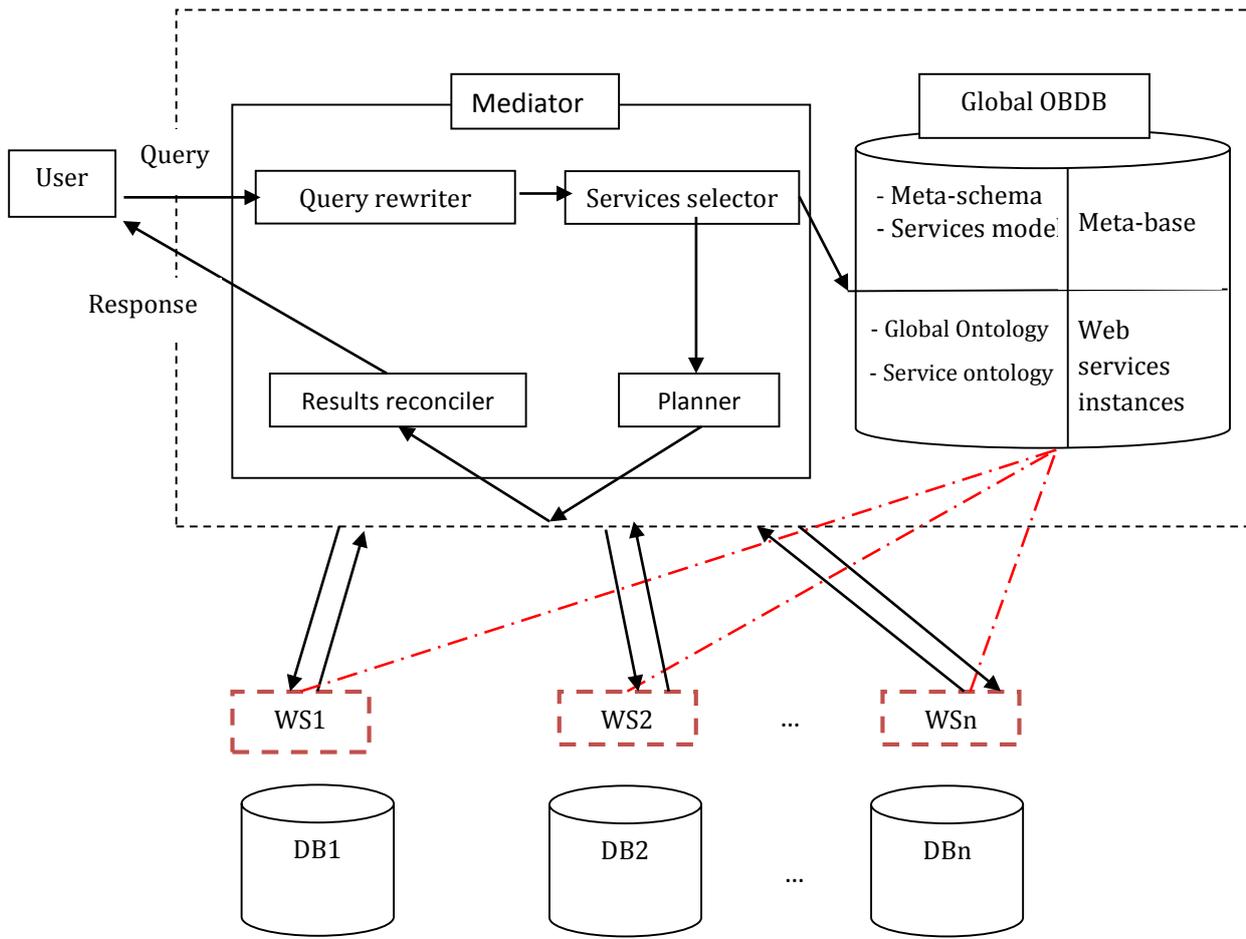


Figure 5.9 : Architecture globale du médiateur

5.8. Conclusion

Dans ce chapitre nous avons présenté et expliqué nos contributions pour la construction d'un système d'intégration de données qui repose essentiellement sur les ontologies et l'utilisation des services web au niveau de la deuxième couche, celle-ci existe dans tout système d'intégration qui est la couche des adaptateurs (wrappers). Ce service peut effectuer plusieurs traitements qui ont été représentés dans un méta-modèle, une extension de ce méta-modèle a été proposée qui prend en charge des opérateurs flous. Nous avons proposé de stocker ces services au niveau d'une base de données à base ontologique, ceci permet d'utiliser une ontologie globale qui décrit les services web contenus dans un registre sémantique. Enfin Nous avons présenté notre architecture de médiation proposée.

A travers notre proposition pour un registre sémantique, les métadonnées de services web représentant leurs sémantiques peuvent être enregistrées, facilitant leur découverte. Les différentes sources de données sont accessibles à travers une interface unique et commune qui est l'interface du médiateur.

Dans le chapitre suivant nous détaillons l'implémentation de cette architecture dans le domaine médical et les outils utilisés pour implémenter cette architecture.

Chapitre 6 : Implémentation de l'architecture proposée

6.1.	Introduction	85
6.2.	Implémentation.....	85
	-Exemple de motivation	86
6.3.	Langages et outils utilisés.....	91
6.4.	Conclusion.....	92

6.1. Introduction

Dans ce chapitre nous présentons l'implémentation de notre architecture dans le domaine médical. Ce domaine est caractérisé par des changements fréquents, dans lequel les activités s'exercent de manière distribuée impliquant plusieurs experts de différentes spécialités et services pour la prise de décision efficace, ces derniers ont besoin d'accéder à des informations concernant les patients de manière pertinente faisant appel à différentes sources d'information.

Le problème de l'accès à l'information médicale pour les soins est lié d'un côté à la dispersion des données médicales sur différentes structures dont les systèmes d'information sont autonomes et hétérogènes, ainsi plusieurs systèmes existants ne sont pas communicants. D'un autre côté, à l'explosion du nombre des sources accessibles via le web et qui évoluent rapidement. Pour cela, intégrer les données médicales en utilisant l'approche entrepôt de données est une tâche très coûteuse. L'utilisation des ontologies permet d'avoir un schéma unifié des données des sources. Pour cette raison nous optons pour la solution de médiation de données en s'appuyant essentiellement sur l'utilisation des ontologies.

Nous illustrons dans ce chapitre la faisabilité de notre architecture avec des exemples. Le système est évalué en termes du temps de réponse d'un ensemble de requêtes sur un ensemble de sources de données.

6.2. Implémentation

Pour valider nos contributions, nous avons choisi d'implémenter l'architecture proposée dans le domaine médical. Ce domaine est caractérisé par des changements fréquents, dans lequel les activités s'exercent d'une manière distribuée impliquant plusieurs experts de différentes spécialités et services pour la prise de décision efficace.

- **Exemple de motivation**

Nous prenons l'exemple des urgences, où un médecin a besoin d'accéder à plusieurs examens et bilans médicaux qui sont effectués à l'arrivée d'un patient. De manière très simplifiée, ces examens se répartissent en plusieurs types :

- a) les examens cliniques, qui concernent les examens effectués directement sur le patient, tels que la prise de la tension, du pouls ou de la température ;
- b) les examens biologiques, qui concernent les études faites à partir d'un prélèvement, comme une prise de sang du patient, et
- c) les examens d'imagerie, qui correspondent à l'échographie ou à la radiographie

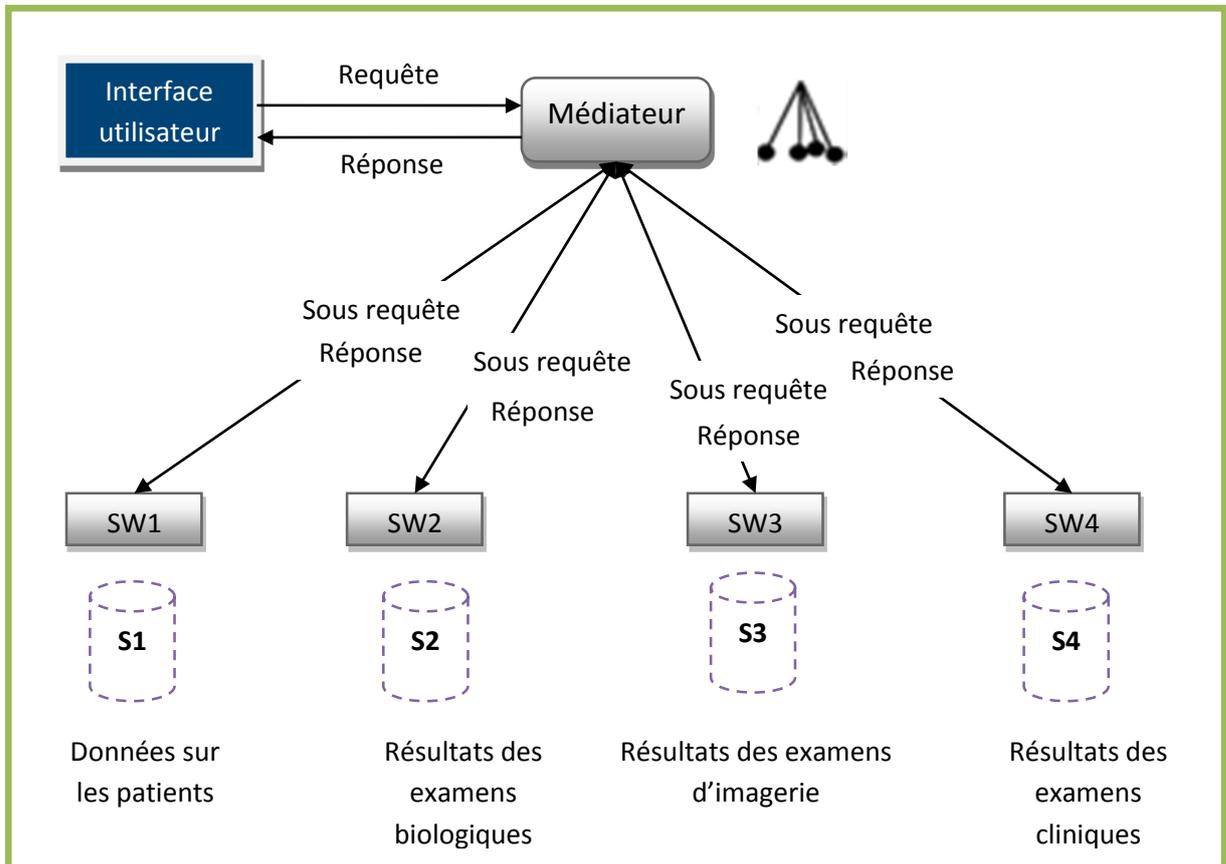


Figure 6.1 : Exemple de motivation

Dans le tableau 6.1, nous donnons des exemples de services web avec leurs entrées et leurs sorties.

Services	Entrées	Sorties
SW1	IdPatient	Informations sur le patient (nom, prénom, sexe, adresse...)
SW2	IdPatient	Résultats examens biologiques
SW3	IdPatient	Résultats examens imagerie
SW4	IdPatient	Résultats examens cliniques

Tableau 6.1 : Exemples de services web

Premièrement la construction de l'ontologie a été basée sur d'autres ontologies du domaine médical comme l'ontologie OpenGalen et UMLS. Ensuite des bases de données sémantiques ont été construites utilisant la même ontologie au niveau du framework OntoDB ; incluant les informations sur le patient comme les tests de laboratoire, informations des urgences, tests cliniques...etc. pour chaque base de données, un service web est associé, qui est responsable de l'interrogation de cette base, le traitement des données et de retourner les résultats au médiateur. Ces web services peuvent être non seulement interrogés par le médiateur mais aussi par d'autres applications qui ont besoin des données que le service est capable de récupérer. L'utilisateur pose ses requêtes au niveau de l'ontologie utilisant le langage de requêtes OntoQL. Le médiateur et les services web ont été implémentés utilisant le langage de programmation Java, les services ont été hébergés dans un serveur apache tomcat.

Ces expérimentations ont été réalisées dans un CPU intel Core i5, avec 3G de RAM sur un système d'exploitation Windows7. Nous donnons dans ce qui suit quelques exemples de services et de requêtes.

SW1 : contient une opération qui consiste à retourner les informations personnelles d'un patient (nom, prénom, sexe, adresse) étant donné son SSN (Social Security Number).

SW2 : contient deux opérations :

- Op1 : permet de retourner les informations sur un patient sachant son SSN ;
- Op2 : permet de retourner le LDL Cholesterol qui est calculé par la formule Friedewald Formula [1] comme suit :

$$\text{LDL} = \text{TC} - \text{HDL} - \text{TG}/5.0 \text{ (mg/dL)}$$

Où:

- LDL représente le mauvais cholesterol,
- HDL est le bon cholesterol,
- TC est le cholesterol total et
- TG représente le taux des Triglycerides.

Les services retournent les données de manière dynamique, et le médiateur sélectionne les entrées et les sorties nécessaires pour répondre à la requête et les envoie comme des arguments aux opérations des services.

Étant donné la requête suivante :

Requête1:

```
SELECT P."First name", P."Last name", P.Gender, C." LDL Cholesterol" FROM
      Patient P, Cholesterol C WHERE P.SSN = '123';
```

Où 'first name' est une propriété de la classe 'person', 'patient' est une sous classe de la classe 'person', et 'LDL Cholesterol' est une propriété de la classe 'Cholesterol' de l'ontologie globale.

À partir de cette requête (requête de l'utilisateur) nous obtenons les entrées et les sorties nécessaires à la requête 2 qui permet de sélectionner les services avec les opérations en mesure de retourner les données demandées par l'utilisateur.

Requête 2 :

```
SELECT ServiceID, Name FROM Operation WHERE input = "SSN" AND 'Last
name' = ANY(Outputs) AND 'First name' = ANY(Outputs) AND 'Gender' = ANY(Outputs)
AND 'LDL Cholesterol' = ANY(Outputs)
```

Le résultat de cette requête nous présente deux opérations de deux services différents, donc la requête initiale et réécrite en deux sous requêtes, le sélecteur de service (service selector) sélectionne les services correspondant à ces opérations à partir des extensions des services. Par exemple, la requête 3 est envoyée afin d’avoir la localisation du service de la première opération dans la requête 2.

Requête 3:

```
SELECT Location FROM Service where OID=6
```

Où OID correspond à l’identifiant du service. (qui est le résultat serviceID de la requête 2).

Le tableau 1 présente des exemples d’extensions de services web, tableau 2 présente quelques extensions sur des opérations.

oid	Name	ServiceID	Inputs	Outputs
5	getInfosPatient	6	{SSN}	{'First name','Last name','Gender','Address'}
9	getCholesterolRate	8	{SSN}	{'LDL Cholesterol'}

Tableau 6.2 : Exemples d’extensions d’opérations

oid	Name	Location	Operations
3	InfosPatient	http://localhost:8080/MedicalWS/wsdl/InfosPatient.w...	{getInfosPatient}
6	getPatientInfosService	http://localhost:8080/testSW1/services/getPatientInfos	{getInfosPatient}
8	getCholesterol	http://localhost:8080/testSW1/services/getCholesterol	{getCholesterolRate}

Tableau 6.3 : Exemples d’extensions de services web

Nous avons construit une ontologie contenant environ 72 concepts et 200 propriétés (Figure 6.1 représente une partie de cette ontologie), environ 100 bases de données couvrant les différents domaines de la médecine incluant les informations administratives, les analyses, les tests cliniques ..., chaque base de données est associée à un service web qui permet d'effectuer les différentes opérations nécessaires.

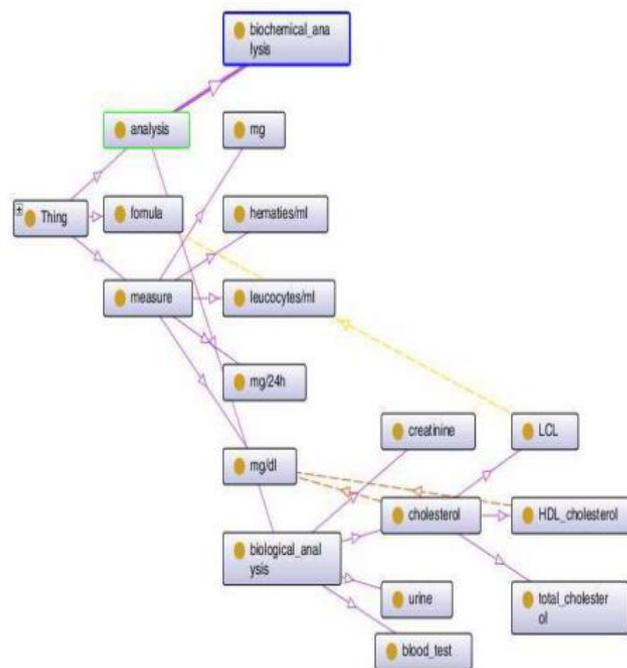


Figure 6.2: Partie de l'ontologie construite

Nous avons implémenté nos bases de données, ainsi que les descriptions des services web au niveau de la base de données à base ontologiques OntoDB, celle-ci offre un stockage flexible des données à base ontologique. La Figure 6.3 montre l'interface de l'utilisateur qui lui permet de saisir sa requête et de visualiser les résultats ainsi que le temps de réponse de la requête.

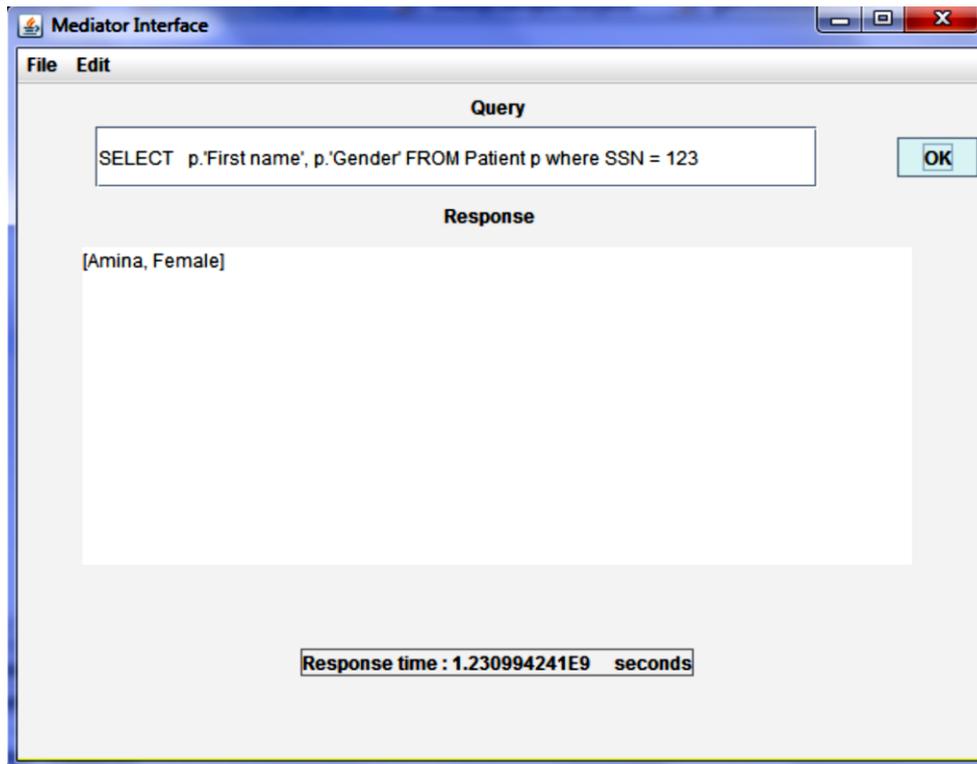


Figure 6.3 : Interface graphique du médiateur

6.3. Langages et outils utilisés

Pour l'implémentation de notre architecture nous avons utilisé les outils et les langages suivants :

- *Protégé 4.1.0*, c'est un éditeur open source pour la construction des ontologies, développé à l'école de médecine du centre Stanford pour la recherche en informatique biomédicale, nous avons utilisé cet éditeur pour la construction de notre ontologie.
- *Le langage OWL*, l'ontologie a été construite sous le langage OWL
- *Le langage java*, pour le développement des services web, du médiateur ainsi que pour la construction des bases de données sous l'environnement de développement *Eclipse (Eclipse IDE for Java EE Developers)*

- *OntoDB* est une architecture de base de données à base ontologique utilisant l'approche horizontale c'est-à-dire que pour chaque classe de l'ontologie, une table est créée avec une colonne pour chaque propriété. Il utilise l'architecture de type III. Il est construit en utilisant le SGBD Postgres au sein du laboratoire LIAS
- *OntoQL*, Le modèle OntoDB est doté d'un langage de requêtes afin d'exploiter les données, il s'agit du langage OntoQL [56], ce langage permet d'effectuer les différents traitements sur des données ainsi que sur les ontologies d'une base de données OntoDB comme l'interrogation, l'insertion, la modification...
- Postgres, c'est un système de gestion de base de données relationnelle et objet, les bases de données ont été construites sur ce SGBD.
- Le plugin "*ecore*" pour la modélisation, et "Visual Editor" pour la programmation de l'interface graphique.

6.4. Conclusion

Nous avons décrit dans ce chapitre l'implémentation de notre architecture de médiation utilisant les ontologies et les services web à travers un exemple de motivation. L'ontologie permet de décrire les sources de données et joue le rôle de schéma global de l'architecture de médiation. Nous avons construit une ontologie contenant environ 72 concepts et 200 propriétés.

Un service web joue le rôle d'adaptateur entre une source et le médiateur. Ce service peut effectuer plusieurs traitements qui ont été représentés dans un méta-modèle, nous avons construit environ 100 bases de données couvrant les différents domaines de la médecine incluant les informations administratives, les analyses, les tests cliniques ..., chaque base de données est associée à un service web qui permet d'effectuer les différentes opérations nécessaires.

L'évaluation de notre système est effectuée en mesurant le temps d'exécution d'un ensemble de requêtes sur un ensemble de sources de données. Ce temps varie de 1 à 4 secondes selon le nombre de services web interrogés pour répondre à la requête.

Conclusion générale

L'intégration sémantique de données consiste à offrir une représentation conceptuelle des données afin d'éliminer les conflits entre ces données. Actuellement, les ontologies sont utilisées dans divers domaines pour représenter explicitement les connaissances d'un domaine particulier. Dans les systèmes d'intégration une ontologie joue le rôle de schéma global sur lequel l'utilisateur pose ses requêtes, nos propositions sont basées essentiellement sur l'utilisation des ontologies afin de pouvoir résoudre les hétérogénéités entre les sources de données.

Nous avons décrit dans la première partie de cette thèse les différents types d'hétérogénéités qui peuvent exister entre les systèmes et les solutions proposées dans la littérature pour résoudre ces hétérogénéités après l'étude de la notion d'ontologie, ensuite nous avons étudié les approches d'intégration de données qui sont l'approche entrepôt et l'approche médiation de données et les différentes facettes liées à la médiation ainsi que les travaux réalisés dans la littérature. Nous avons aussi étudié les services web et les services web sémantiques et leur rôle dans les systèmes d'intégration. Dans la deuxième partie, nous avons présenté nos travaux qui concernent l'intégration sémantique de données de type médiation, cette approche permet d'accéder à des données distribuées sans avoir à les migrer vers un schéma global, pour cela choisir cette solution est préférable dans des systèmes où l'information évoluent rapidement, l'approche entrepôt de données est très coûteuse puisque pour chaque changement au niveau d'une source de données, l'entrepôt de données doit être mis à jours, à ce propos, nous avons proposé une approche de médiation en se basant essentiellement sur l'utilisation des ontologies.

Nous utilisons les services web en tant qu'adaptateurs dans cette architecture qui, en plus du rôle de traducteurs, ils peuvent effectuer des traitements sur les données pour cela nous avons proposé un méta-modèle de services web dans lequel nous considérons les différents types d'opération qui peuvent être effectuées, nous avons étendu par la suite ce méta-modèle en considérant les opérateurs flous. Nous avons proposé ainsi une approche pour l'enregistrement de services web sur des bases de données à base ontologiques ce qui permet de créer un registre sémantique [112]. Nous avons détaillé notre architecture de médiation et enfin, nous l'avons implémenté dans le domaine médical [15].

Comme perspectives nous envisageons d'enrichir cette architecture par d'autres sources couvrant tout le domaine de la médecine, nous souhaitons aussi proposer un benchmark sur les approches d'intégration dans le domaine médical permettant à ces systèmes de comparer et d'évaluer leurs approches parmi d'autres.

Annexe : Exemple d'un document WSDL

Dans cette annexe nous donnons un exemple d'un document WSDL qui concerne le service getCholesterolRate :

```
<?xml version="1.0" encoding="UTF-8"?>

<wsdl:definitions targetNamespace="http://DefaultNamespace"
xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="http://DefaultNamespace"
xmlns:intf="http://DefaultNamespace" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
xmlns:wSDLsoap="http://schemas.xmlsoap.org/wSDL/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <wsdl:types>

    <schema elementFormDefault="qualified" targetNamespace="http://DefaultNamespace"
xmlns="http://www.w3.org/2001/XMLSchema">

      <element name="getCholesterolRate">

        <complexType>

          <sequence>

            <element name="SSN" type="xsd:int"/>

            <element maxOccurs="unbounded" name="output" type="xsd:string"/>

          </sequence>

        </complexType>

      </element>

      <element name="getCholesterolRateResponse">

        <complexType>

          <sequence>

            <element maxOccurs="unbounded" name="getCholesterolRateReturn"
type="xsd:string"/>

          </sequence>

        </complexType>

      </element>

    </schema>

  </wsdl:types>

</wsdl:definitions>
```

```
</wsdl:types>

<wsdl:message name="getCholesterolRateRequest">
  <wsdl:part element="impl:getCholesterolRate" name="parameters">

    </wsdl:part>
  </wsdl:message>

<wsdl:message name="getCholesterolRateResponse">
  <wsdl:part element="impl:getCholesterolRateResponse" name="parameters">
    </wsdl:part>
  </wsdl:message>

<wsdl:portType name="getCholesterol">
  <wsdl:operation name="getCholesterolRate">
    <wsdl:input message="impl:getCholesterolRateRequest"
name="getCholesterolRateRequest">
    </wsdl:input>
    <wsdl:output message="impl:getCholesterolRateResponse"
name="getCholesterolRateResponse">
    </wsdl:output>
  </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="getCholesterolSoapBinding" type="impl:getCholesterol">
  <wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="getCholesterolRate">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getCholesterolRateRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="getCholesterolRateResponse">
```

```
<wsdlsoap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="getCholesterolService">
  <wsdl:port binding="impl:getCholesterolSoapBinding" name="getCholesterol">

    <wsdlsoap:address location="http://localhost:8080/testSW1/services/getCholesterol"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```


Références Bibliographiques

- [1] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2), 199-220.
- [2] Baget, J. F., Canaud, É., Euzenat, J., & Saïd-Hacid, M. (2004). Les langages du web sémantique. *Information-Interaction-Intelligence*.
- [3] Selma, K., Ilyès, B., Ladjel, B., Eric, S., Stéphane, J., & Michael, B. (2012). Ontology-based structured web data warehouses for sustainable interoperability: requirement modeling, design methodology and tool. *Computers in Industry*, 63(8), 799-812.
- [4] Amarouche, I. A., Benslimane, D., Barhamgi, M., Mrissa, M., & Alimazighi, Z. (2011). Electronic health record data-as-a-services composition based on query rewriting. In *Transactions on large-scale data-and knowledge-centered systems IV* (pp. 95-123). Springer Berlin Heidelberg.
- [5] Khouri, S. (2013). Cycle de vie sémantique de conception de systèmes de stockage et manipulation de données (Doctoral dissertation, ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d'Aérotechnique-Poitiers).
- [6] Pierra, G. (2003, July). Context-explication in conceptual ontologies: the PLIB approach. In *ISPE CE* (pp. 243-253).
- [7] Gómez-Pérez, A. (1999). Ontological engineering: A state of the art. *Expert Update: Knowledge Based Systems and Applied Artificial Intelligence*, 2(3), 33-43.
- [8] Guarino, N. (1997). Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, 46(2), 293-310.
- [9] Guarino N., "Some Ontological Principles for Designing Upper Level Lexical Resources". *Proceedings of First International Conference on Language Resources and Evaluation ELRA - European Language Resources Association, Granada, Spain, pp. 527-534, 1999.*
- [10] Van Heijst, G., Schreiber, A. T., & Wielinga, B. J. (1997). Using explicit ontologies in KBS development. *International journal of human-computer studies*, 46(2), 183-292.
- [11] Bachimont, B. (2000). Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances. *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, 305-323.

- [12] Keita, A. K. (2007). Conception coopérative d'ontologies pré-consensuelles: Application au domaine de l'urbanisme. Institut National des Sciences Appliquées. Lyon, Ecole Doctorale Informatique et Information pour la Société.
- [13] Mizoguchi, R., Vanwelkenhuysen, J., & Ikeda, M. (1995). Task ontology for reuse of problem solving knowledge. *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing*, 46-59.
- [14] Smith, Barry (2003). Ontology. In Luciano Floridi (ed.), *Blackwell Guide to the Philosophy of Computing and Information*. Blackwell. 155-166.
- [15] Benosman, A., Chikh, M. A., & Bellatreche, L. (2014). Semantic Mediation of Medical Data Using Web Services. *Journal of Medical Imaging and Health Informatics*, 4(6), 858-862.
- [16] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., & Schneider, L. (2003). Dolce: a descriptive ontology for linguistic and cognitive engineering. WonderWeb Project, Deliverable D17 v2, 1.
- [17] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, 284(5), 28-37.
- [18] <http://www.w3.org/TR/daml+oil-reference>
- [19] <http://www.w3.org/RDF/>
- [20] <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>
- [21] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowl. Creat. Diffus. Util.*, vol. 5, no. April, pp. 199–220, 1993.
- [22] Soukane, A. (2005). Génération automatique des requêtes de médiation dans un environnement hétérogène (Doctoral dissertation, Université Versailles St-Quentin en Yvelines).
- [23] D. Xuan and N. Int, "Intégration de bases de données hétérogènes par articulation à priori d'ontologies: application aux catalogues de composants industriels."
- [24] D. C. FAYE, "Médiation de données sémantique dans SenPeer , un système pair-à-pair de gestion de données," *Gestion*, 2007.
- [25] Inmon, W. H., & Hackathorn, R. D. (1994). *Using the data warehouse*. Wiley-QED Publishing.
- [26] Azzaz, A., Malki, M., Bellatreche, L., & Benmimoun, Y. (2009). Une Approche Ontologique d'Intégration de Sources de Données dans un Environnement de Pair à Pair. In *CIIA*.

- [27] Wisniewski, M. F., Kieszkowski, P., Zagorski, B. M., Trick, W. E., Sommers, M., Weinstein, R. A., & Chicago Antimicrobial Resistance Project. (2003). Development of a clinical data warehouse for hospital infection control. *Journal of the American Medical Informatics Association*, 10(5), 454-462.
- [28] Banek, M., Tjoa, A. M., & Stolba, N. (2006). Integrating different grain levels in a medical data warehouse federation. In *Data Warehousing and Knowledge Discovery* (pp. 185-194). Springer Berlin Heidelberg.
- [29] Kerkri, E. M., Quantin, C., Allaert, F. A., Cottin, Y., Charve, P., Jouanot, F., & Yétongnon, K. (2001). An approach for integrating heterogeneous information sources in a medical data warehouse. *Journal of Medical Systems*, 25(3), 167-176.
- [30] Prather, J. C., Lobach, D. F., Goodwin, L. K., Hales, J. W., Hage, M. L., & Hammond, W. E. (1997). Medical data mining: knowledge discovery in a clinical data warehouse. In *Proceedings of the AMIA annual fall symposium* (p. 101). American Medical Informatics Association.
- [31] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. The tsimmi project: Integration of heterogeneous information sources. 1994.
- [32] Cruz, I. F., & Xiao, H. (2005). The role of ontologies in data integration. *Engineering intelligent systems for electrical engineering and communications*, 13(4), 245.
- [33] Lenzerini, M. (2002, June). Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 233-246). ACM.
- [34] Ullman, J. D. (1997). Information integration using logical views. In *Database Theory—ICDT'97* (pp. 19-40). Springer Berlin Heidelberg.
- [35] Halevy, A., Rajaraman, A., & Ordille, J. (2006, September). Data integration: the teenage years. In *Proceedings of the 32nd international conference on Very large data bases* (pp. 9-16). VLDB Endowment.
- [36] Friedman, M., Levy, A. Y., & Millstein, T. D. (1999). Navigational plans for data integration. *AAAI/IAAI*, 1999, 67-73.
- [37] Calvanese, D., Lembo, D., & Lenzerini, M. (2001). Survey on methods for query rewriting and query answering using views. *Integrazione, Warehousing e Mining di sorgenti eterogenee*, 25.

- [38] McBrien, P., & Poulouvasilis, A. (2003, March). Data integration by bi-directional schema transformation rules. In *Data Engineering, 2003. Proceedings. 19th International Conference on* (pp. 227-238). IEEE.
- [39] G. Wiederhold, Mediators in the architecture of future information systems. *IEEE Computer* 38 (1992).
- [40] Beneventano, D., Bergamaschi, S., Castano, S., Corni, A., Guidetti, R., Malvezzi, G., & Vincini, M. (2000, September). Information integration: the MOMIS project demonstration. In *VLDB* (pp. 611-614).
- [41] Halevy, A. Y. (2001). Answering queries using views: A survey. *The VLDB Journal*, 10(4), 270-294.
- [42] Subrahmanian, V. D., Adali, S., Brink, A., Emery, R., Lu, J. J., Rajput, A. & Ward, C. (1995). HERMES: A heterogeneous reasoning and mediator system.
- [43] Ouzzani, M., & Bouguettaya, A. (2004). Query processing and optimization on the web. *Distributed and Parallel Databases*, 15(3), 187-218.
- [44] <http://www.w3.org/TR/owl-ref/>
- [45] Sujansky, W. (2001). Heterogeneous database integration in biomedicine. *Journal of biomedical informatics*, 34(4), 285-298.
- [46] Fankam C. (2009). Ontodb2 : un système flexible et efficient de base de données à base ontologique pour le web sémantique et les données techniques. Ph.D. Thesis. Poitiers University. 1.3, 1.4, 1.4.3
- [47] Vassiliadis, P., Simitsis, A., & Skiadopoulou, S. (2002, November). Conceptual modeling for ETL processes. In *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP* (pp. 14-21). ACM.
- [48] Karakasidis, A., Vassiliadis, P., & Pitoura, E. (2005, June). ETL queues for active data warehousing. In *Proceedings of the 2nd international workshop on Information quality in information systems* (pp. 28-39). ACM.
- [49] Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., & Wilkinson, K. (2004, May). Jena: implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters* (pp. 74-83). ACM.
- [50] Wu, Z., Eadon, G., Das, S., Chong, E. I., Kolovski, V., Annamalai, M., & Srinivasan, J. (2008, April). Implementing an inference engine for RDFS/OWL

- constructs and user-defined rules in Oracle. In Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on (pp. 1239-1248). IEEE.
- [51] Dehainsala H., Pierra G., Bellatreche L., « OntoDB : An Ontology-Based Database for Data Intensive Applications », DASFAA'07, p. 497-508, 2007.
- [52] Fankam C., OntoDB2 : un système flexible et efficient de Base de Données à Base Ontologique pour le Web sémantique et les données techniques, Ph.d. thesis, Poitiers University, 2009
- [53] Alexaki, S., Christophides, V., Karvounarakis, G., Plexousakis, D., & Tolle, K. (2001, May). On Storing Voluminous RDF Descriptions: The Case of Web Portal Catalogs. In WebDB (pp. 43-48).
- [54] Broekstra, J., Kampman, A., & Van Harmelen, F. (2002). Sesame: A generic architecture for storing and querying rdf and rdf schema. In The Semantic Web—ISWC 2002 (pp. 54-68). Springer Berlin Heidelberg.
- [55] Lu, J., Ma, L., Zhang, L., Brunner, J. S., Wang, C., Pan, Y., & Yu, Y. (2007, September). SOR: a practical system for ontology storage, reasoning and search. In Proceedings of the 33rd international conference on Very large data bases (pp. 1402-1405). VLDB Endowment.
- [56] Jean, S., Aït-Ameur, Y., & Pierra, G. (2006). Querying ontology based database using OntoQL (an ontology query language). In On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE (pp. 704-721). Springer Berlin Heidelberg.
- [57] Wang, J., Yu, A., Zhang, X., & Qu, L. (2009, August). A dynamic data integration model based on SOA. In Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on (Vol. 2, pp. 196-199). IEEE.
- [58] Sha, Z., & Xie, Y. (2010). Design of service-oriented architecture for spatial data integration and its application in building web-based GIS systems. *Geo-spatial Information Science*, 13(1), 8-15.
- [59] Wang, J. (2009). The Design and Implementation of Marine Data Integration System Based on SOA. In 2009 International Conference on Communication Software and Networks (pp. 469-472).
- [60] R. Malarvizhi and S. Kalyani, “SOA based Open Data Model for Information Integration in Smart Grid,” pp. 143–148, 2013.

- [61] S. Zhang, J. Li, F. Zhang, and X. Li, "Design of Data Center and Implementation of Data Integration Based on SOA Framework," 2011 IEEE 3rd Int. Conf. Commun. Softw. Networks, pp. 267–271, 2011.
- [62] Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., & Weerawarana, S. (2002). Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet computing*, 6(2), 86-93.
- [63] Clement, L., Hatley, A., Von Riegen, C., Rogers, UDDI Version 3.0.2, 2004. URL http://www.uddi.org/pubs/uddi_v3.htm
- [64] Lecomte, S., & Boulanger, T. (2008). XML par la pratique : bases indispensables, concepts et cas pratiques. Editions ENI.
- [65] Buswell, S., Devitt, S., Diaz, A., Ion, P., Miner, R., Poppelier, N., & Watt, S. Mathematical Markup Language (MathML) 1.01 Specification, 1999.] soit [World-Wide Web Consortium. (1999). Mathematical Markup Language (MathML [tm]) 1.01 Specification.]
- [66] World Wide Web Consortium. (2000). Scalable Vector Graphics (SVG) 1.1 Specification. W3C Candidate Recommendation, 2.
- [67] W3C World-Wide Web Consortium. (2008). Synchronized Multimedia Integration Language–SMIL 3.0 Specification, W3C Recommendation. DOI: <http://www.w3.org/TR/SMIL3>.
- [68] Gottschalk, K., Graham, S., Kreger, H., & Snell, J. (2002). Introduction to web services architecture. *IBM systems Journal*, 41(2), 170-177.
- [69] Erl, T. (2004). Service-oriented architecture: a field guide to integrating XML and web services. Prentice Hall PTR.
- [70] Alonso, G., Casati, F., Kuno, H., & Machiraju, V. (2004). Web services (pp. 123-149). Springer Berlin Heidelberg.
- [71] Huhns, M. N., & Singh, M. P. (2005). Service-oriented computing: Key concepts and principles. *Internet Computing, IEEE*, 9(1), 75-81.
- [72] D. Martin et al., "Bringing Semantics to Web Services with OWL-S," *World Wide Web*, vol. 10, no. 3, pp. 243-277, Jul. 2007.
- [73] Akkiraju, R., Farrell, J., Miller, J. A., Nagarajan, M., Sheth, A. P., & Verma, K. (2005). Web service semantics-wsdl-s.
- [74] De Bruijn, J., Bussler, C., Domingue, J., Fensel, D., Hepp, M., Kifer, M., ... & Stollberg, M. (2005). Web service modeling ontology (wsmo). *Interface*, 5, 1.

- [75] Fensel, D., & Bussler, C. (2002). The web service-modeling framework WSMF. *Electronic Commerce Research and Applications*, 1(2), 113-137.
- [76] De Bruijn, J., Lausen, H., Polleres, A., & Fensel, D. (2006). The web service modeling language WSML: an overview (pp. 590-604). Springer Berlin Heidelberg.
- [77] Ankolekar, A., Burstein, M., Hobbs, J. R., Lassila, O., Martin, D., McDermott, D. & Sycara, K. (2002). DAML-S: Web service description for the semantic web. In *The Semantic Web—ISWC 2002* (pp. 348-363). Springer Berlin Heidelberg.
- [78] Horrocks, I. (2002). DAML+OIL: A Description Logic for the Semantic Web. *IEEE Data Eng. Bull.*, 25(1), 4-9.
- [79] Charlet, J., Bachimont, B., Brunie, V., El Kassar, S., Zweigenbaum, P., & Boisvieux, J. F. (1998). Hospitexte: towards a document-based hypertextual electronic medical record. In *Proceedings of the AMIA Symposium* (p. 713). American Medical Informatics Association.
- [80] Zhu, F., Turner, M., Kotsiopoulos, I., Bennett, K., Russell, M., Budgena, D., & Xu, J. (2004, July). Dynamic data integration using web services. In *Web Services, 2004. Proceedings. IEEE International Conference on* (pp. 262-269). IEEE.
- [81] Abiteboul, S., Benjelloun, O., & Milo, T. (2002, December). Web services and data integration. In *Web Information Systems Engineering, International Conference on* (pp. 3-3). IEEE Computer Society.
- [82] Zaremba, M., Moran, M., Haselwanter, T., & Lee, H. K. (2005). Wsmx architecture. D13. 4v0, 2.
- [83] Gottschalk, K., Graham, S., Kreger, H., & Snell, J. (2002). Introduction to web services architecture. *IBM systems Journal*, 41(2), 170-177.
- [84] DANIEL ROCACHER, « Multi-ensembles flous et quantités graduelles- Application à l'interrogation flexible de bases de données », Habilitation à diriger les recherches, 2005
- [85] <http://www.opengalen.org/>
- [86] <http://www.nlm.nih.gov/research/umls/>
- [87] Guarino, N. (1998). Formal ontology in information systems: *Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy (Vol. 46)*. IOS press.
- [88] Roussey, C., Pinet, F., Kang, M. A., & Corcho, O. (2011). An Introduction to Ontologies and Ontology Engineering. In *Ontologies in Urban Development Projects* (pp. 9-38). Springer London.

- [89] Lenat, D. B., & Guha, R. V. (1993). Building large knowledge-based systems: Representation and inference in the CYC project. *Artificial Intelligence*, 61(1).
- [90] Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & knowledge engineering*, 46(1), 41-64.
- [91] Bodenreider, O., & Burgun, A. (2005). Biomedical ontologies. In *Medical Informatics* (pp. 211-236). Springer US.
- [92] Borst, P., Akkermans, H., & Top, J. (1997). Engineering ontologies. *International Journal of Human-Computer Studies*, 46(2), 365-406.
- [93] Hansen, M., Madnick, S., & Siegel, M. (2003). Data integration using web services (pp. 165-182). Springer Berlin Heidelberg.
- [94] Yé tongnon, K., Suwanmanee, S., Benslimane, D., & Champin, P. A. (2006). A web-centric semantic mediation approach for spatial information systems. *Journal of Visual Languages & Computing*, 17(1), 1-24.
- [95] Zhu, F., Turner, M., Kotsiopoulos, I., Bennett, K., Russell, M., Budgena, D., & Xu, J. (2004, July). Dynamic data integration using web services. In *Web Services, 2004. Proceedings. IEEE International Conference on* (pp. 262-269). IEEE.
- [96] McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, 10(10), 2004.
- [97] Hondjack, D. (2007). Explication de la sémantique dans les bases de données : base de données à base ontologique et le modèle OntoDB (Doctoral dissertation, Thèse de Doctorat Université de Poitiers).
- [98] Jean, S. (2007). OntoQL, un langage d'exploitation des bases de données à base ontologique (Doctoral dissertation, Université de Poitiers).
- [99] Charlet, J., Bachimont, B., Mazuel, L., Dhombres, F., Jaulent, M. C., & Bouaud, J. (2012). OntoMénélas. Motivations et retours d'expérience sur l'élaboration d'une ontologie noyau de la médecine. *Technique et Science Informatiques*, 31(1), 125-147.
- [100] Rector, A. L., Rogers, J. E., Zanstra, P. E., & Van Der Haring, E. (2003). OpenGALEN: open source medical terminology and tools. In *AMIA Annual Symposium Proceedings* (Vol. 2003, p. 982). American Medical Informatics Association.
- [101] IEC 61360 Standard data element types with associated classification scheme for electric components", <http://webstore.iec.ch>

- [102] Botstein, D., Cherry, J. Ms., Ashburner, M., Ball, C. A., Blake, J. A., Butler, H., & Eppig, J. T. (2000). Gene Ontology: tool for the unification of biology. *Nat Genet*, 25(1), 25-29.
- [103] Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1), D267-D270.
- [104] Rector, A. L., Solomon, W. D., Nowlan, W. A., Rush, T. W., Zanstra, P. E., & Claassen, W. M. (1995). A Terminology Server for medical language and medical information systems. *Methods of information in medicine*, 34(1-2), 147-157.
- [105] Rodrigues, J. M., Trombert-Paviot, B., Baud, R., Wagner, J., & Meusnier-Carriot, F. (1998). GALEN-In-Use: Using artificial intelligence terminology tools to improve the linguistic coherence of a national coding system for surgical procedures. *Studies in health technology and informatics*, (1), 623-627.
- [106] Aranguren, M. E. (2005). Ontology design patterns for the formalization of biological ontologies. University of Manchester, Faculty of Engineering and Biological Sciences.
- [107] <http://www.ncbi.nlm.nih.gov/pubmed/10802651>
- [108] Rosse, C., & Mejino, J. L. (2003). A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *Journal of biomedical informatics*, 36(6), 478-500.
- [109] Spackman, K. A., Campbell, K. E., & Côté, R. A. (1997). SNOMED RT: a reference terminology for health care. In *Proceedings of the AMIA annual fall symposium* (p. 640). American Medical Informatics Association.
- [110] Stearns, M. Q., Price, C., Spackman, K. A., & Wang, A. Y. (2001). SNOMED clinical terms: overview of the development process and project status. In *Proceedings of the AMIA Symposium* (p. 662). American Medical Informatics Association.
- [111] Wache, H., Voegelé, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., & Hübner, S. (2001, August). Ontology-based integration of information-a survey of existing approaches. In *IJCAI-01 workshop: ontologies and information sharing* (Vol. 2001, pp. 108-117).
- [112] Benosman, A., Bouchikhi S. & Chikh, M.A.(2015). A UDDI for Semantic Web services registration. 5ème Journée Doctorale de Génie Biomédical. Laboratoire Génie Biomédical, Université de Tlemcen

- [113] Abderlghani, B. (2013, 03 20). Une approche ontologique pour l'intégration et la réconciliation des sources de données hétérogènes dans une architecture de médiation.
- [114] Paolucci, M., Kawamura, T., Payne, T. R., & Sycara, K. (2002). Importing the semantic web in UDDI. In *Web Services, E-Business, and the Semantic Web*(pp. 225-236). Springer Berlin Heidelberg.
- [115] Akkiraju, R., Goodwin, R., Doshi, P., & Roeder, S. (2003, August). A Method for Semantically Enhancing the Service Discovery Capabilities of UDDI. *InIIWeb* (pp. 87-92).
- [116] Sivashanmugam, K., Verma, K., Sheth, A. P., & Miller, J. (2003). Adding semantics to web services standards.

Glossaire

BAV	Both As View
BDBO	Base de Données à Base Ontologique
BDD	Base De Données
BPEL	Business Process Execution Language
CTV3	Clinical Terms Version 3
DAAS	Data As A Service
FMA	Foundational Model of Anatomy
GAV	Global As View
GLAV	Global Local as View
GO	Gene Ontology
LAV	Local as View
LIAS	Laboratoire d'Informatique et d'Automatique pour les Systèmes
MOMIS	MediatOr environment for Multiple Information Sources
MSL	Mediator Specification Language
NLM	<i>National Library of Medicine</i>
OEM	Object Exchange Model
OWL	Ontology Web Language
OWL-S	Ontology Web Language for Services
RDF	Resource Description Framework
RDF-S	RDF Schema
SNOMED	Systematized Nomenclature of Human and Veterinary Medicine
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
UDDI	Universal Description, Discovery and Integration
UMLS	Unified Medical Language System
W3C	World Wide Web Consortium
WSDL	Web Service Description Language
WSDL-S	Web Service Description Language Semantics
WSMO	Web Service Modeling Ontology
XML	eXtensible Markup Language

