

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

BADJI MOKHTAR-ANNABA UNIVERSITY  
UNIVERSITE BADJI MOKHTAR-ANNABA



جامعة باجي مختار - عنابة

Faculté des Sciences de L'Ingénieur

Département d'Informatique

## MEMOIRE

Présenté en vue de l'obtention  
du diplôme de Magister en Informatique  
Option : Texte, Parole et Images

## THEME

Une architecture orientée services pour la  
mise en œuvre de mécanismes d'interopérabilité  
des systèmes d'information d'entreprises

Présenté par :  
**MECHERI Karima**

Soutenu le 02/12/2009 devant le jury:

Président	M <sup>r</sup> FARAH Nadir	MC	Université d'Annaba
Rapporteur	M <sup>r</sup> BOUFAIDA Mahmoud	PR	Université Mentouri de Constantine
Examineur	M <sup>me</sup> SERIDI Hassina	MC	Université d'Annaba
Examineur	M <sup>r</sup> BOUHADADA Tahar	MC	Université d'Annaba

# Remerciements

Ce travail a été réalisé sous la direction de monsieur BOUFAIDA Mahmoud, professeur à l'université Mentouri de Constantine. Je tiens à le remercier vivement pour ses conseils, son aide, sa disponibilité et son soutien tout au long de ce travail de recherche. Ses compétences et son expérience m'ont profondément marquées. Qu'il trouve ici, ma sincère reconnaissance et ma profonde gratitude.

Je souhaite adresser mes sincères remerciements à monsieur FARAH Nadir, maître de conférences à l'université d'Annaba, pour m'avoir fait l'honneur de présider le jury de ma soutenance.

Je tiens aussi à remercier particulièrement les personnes qui ont accepté d'examiner et de juger mon travail et qui sont : monsieur BOUHADADA Tahar et madame SERIDI Hassina, maîtres de conférences à l'université d'Annaba.

J'adresse mes sincères remerciements à M<sup>me</sup> GHOUALMI Nassira, maître de conférences à l'université d'Annaba, surtout pour sa compréhension. Un grand merci à M<sup>me</sup> SOUCI Labiba, maître de conférences à l'université d'Annaba, pour son aide précieuse et son encouragement. Je remercie également tous les enseignants et responsables du département d'informatique de l'université d'Annaba.

Je présente mes remerciements à M<sup>elle</sup> TEBIB Assia de l'université Mentouri de Constantine, pour son aide.

Je tiens à remercier mon amie M<sup>me</sup> KLAI Sihem et tous mes collègues, amies et responsables de l'INSFP Didouche Mourad Annaba.

Enfin, mes remerciements vont à tous les membres de ma famille qui m'ont tant soutenue, il s'agit en particulier de mon mari Azzeddine, mes fils Abderrahmane, Nacereddine et monder, ma petite sœur Sabah, son mari Sofiane et mon adorable neveu Iheb Hocine. Sans oublier mes chers neveux Ali et Kamélia qui me manquent beaucoup.

A la mémoire de mes chers parents

# ملخص

في الإطار الاقتصادي والتقني الحالي، تعاون أنظمة معلومات المؤسسات ضرورة محتومة، يَسْمَحُ لمُواجهَةِ القيود التي ترضخُ لها. إلا أن، أنظمة المعلومات طُوِّرتُ بشكل مستقل البعض عن الآخر. لذا، المشكلة الرئيسية التي تواجهها الأنظمة المسماة بالتعاونية هي عدم التجانس على عدة مستويات. ما عدا ذلك، قاد ظهور الأنظمة والتقنيات الجديدة المؤسسات لتكثيف أنظمة معلوماتها. هذا، يَتطلَّبُ طرق جديدة للتطوير مع إدماج أدوات تصويرية وتقنية جديدة.

يَظهرُ عملنا في هذه الإشكالية. حيث يَتضمنُ اقتراح هيكله خدماتية للعمليات المتبادلة لأنظمة معلومات المؤسسات.

بعد عرض الإشكالية والأهداف المسطرة، قَدَمْنَا خصائص أنظمة المعلومات، ودرَسْنَا بَعْضَ أدوات تفعيل آليات العمليات المتبادلة على المستوى التقني، ما سَمَحَ لنا باختيار الهياكل الخدمائية، معايير خدمات الويب وهندسة النماذج. إنَّ العيب الرئيسي لهذه التقنيات، هو نقصهم فيما يتعلق بالسمة السيمانطيقية (أي المعنى)، التي تُشكِّلُ الحصة الرئيسية للتعاون.

انطلاقاً من هذا المنظور، درَسْنَا الأدوات والمفاهيم لتثبيت السيمانطيقية (المعنى)، و هي، مفهوم الأنطولوجية، نظريات وصف معاني خدمات الويب، ووساطتهم.

اعتماداً على دراسة نقدية للأعمال الموجودة، اقترحنا نظرية قادتنا إلى هيكله مكونة من طبقات. لبناء الطبقة الأولى، تَبَيَّنَّا منهج يعتمد على هندسة النماذج يَسْمَحُ لأنظمة المعلومات بالتحول نحو الهياكل الخدمائية. الخدمات المتحصل عليها سَتعرضُ للمنظمات الخارجية بفضل معايير خدمات الويب عن طريق حافلة خدمات المؤسسات. الطبقة الثانية تَتعلقُ بالهيكله التي تَسْمَحُ بتمثيل سيمانطيقية (معاني) الخدمات المتحصل عليها من الطبقة الأولى. و هي تعتمد على فكرة الأنطولوجية والسياقات، وتُشكِّلُ نموذجاً مرجعياً لتدليل أوصاف خدمات الويب بمساعدة Sawsdl. إن الطبقة الأخيرة، تتمثل في هيكله للعمليات المتبادلة السيمانطيقية التي تَسْتغلُّ الطبقتان الأوليتان. إنَّ عملية التعاون تتم من قِبَلِ خدمة ويب السمسارة التي تنظِّمُ الخدمات الأخرى للعمليات المتبادلة، هم خدمات الويب: للوصف، للنشر، للاكتشاف، للوساطة وللتنفيذ. للمُصادقة على نظريتنا وهيكلتنا عرضنا دراسة حالة.

**الكلمات المفتاح:** نظم المعلومات التعاونية، العمليات المتبادلة، الهياكل الخدمائية، خدمات الويب، هندسة النماذج، الأنطولوجيات، السياق، تدويل سيمانطيقية، وساطة.

# Abstract

In the present economic and technological setting, the cooperation of the systems of information of enterprises is an inescapable necessity, permitting to face the constraints that weigh on the enterprises. Only, the systems of information have been developed independently some of the other. Therefore, the main problem that the cooperative so-called systems face is the one of the multilevel heterogeneities. Otherwise, the advent of new paradigms and technologies drove the enterprises to adapt their systems of information. This, require new methods of development as there integrating new conceptual and technological tools.

Our work appears in this problematic. It consists in the proposition of an architecture oriented services for the interoperability of the systems of information of enterprises.

After having exposed the problematic and the objectives aimed, we presented the features of the systems of information, and studied some tools of bets of the mechanisms of interoperability to the technical level, what allowed us to keep the architectures oriented services, the standards of the Web services and the engineering of the models. The main limit of these paradigms, is their lack with regard to the semantic aspect, that constitutes the major stake for cooperation.

Leaving from this report, we studied the tools and concepts to raise the semantic interoperability, to know, the notion of ontologies, the approaches of description of the semantics of the Web services, and their mediation.

To basis of a critical study of works existing, we proposed an approach that drove to an architecture in layers. To construct the first layer, we adopted a methodology based on the engineering of the models permitting the migration of the systems of information toward the architectures oriented services. The gotten services will be exposed to the tierce organizations with the help of the standards of the Web services via the enterprise bus of services (ESB). The second layer concerns the architecture permitting to capture the semantics of the services gotten of the first layer. It is based on the notion of ontologies and contexts, and constitutes the model of reference to annotate the descriptions of the Web services with the help of the SAWSDL. The last layer, is an architecture for the semantic interoperability exploiting the two first layers. The process of cooperation is taken in charge by the service Web broker that orchestrates the other services of interoperability, they are the Web services: of description, publication, discovery, mediation and execution.

In order to validate our approach and architecture a study of case is exposed.

***Key words:*** *Cooperative systems of information, Interoperability, Architectures oriented services, Service Web, Engineering of the models, Ontologies, Context, semantic Annotation, Mediation.*

# Résumé

Dans le cadre économique et technologique actuel, la coopération des systèmes d'information d'entreprises est une nécessité incontournable, permettant de faire face aux contraintes qui pèsent sur les entreprises. Or, les systèmes d'information ont été développés indépendamment les uns des autres. Par conséquent, le principal problème qu'affrontent les systèmes dits coopératifs est celui des hétérogénéités qui apparaissent à plusieurs niveaux. Par ailleurs, l'avènement de nouveaux paradigmes et technologies a conduit les entreprises à adapter leurs systèmes d'information. Ceci, nécessite de nouvelles méthodes de développement en y intégrant de nouveaux outils conceptuels et technologiques.

Notre travail s'inscrit dans cette problématique. Il consiste en la proposition d'une architecture orientée services pour la mise en œuvre de mécanismes d'interopérabilité des systèmes d'information d'entreprises.

Après avoir exposé la problématique et les objectifs visés, nous avons présenté les caractéristiques des systèmes d'information, et étudié quelques outils de mise en œuvre des mécanismes d'interopérabilité au niveau technique, ce qui nous a permis de retenir les architectures orientées services, les standards des services Web et l'ingénierie des modèles. La principale limite de ces paradigmes, est leur manque en ce qui concerne l'aspect sémantique, qui constitue l'enjeu majeur pour la coopération.

Partant de ce constat, nous avons étudié les outils et concepts pour dresser l'interopérabilité sémantique, à savoir, la notion d'ontologies, les approches de description de la sémantique des services Web, et leur médiation.

A base d'une étude critique des travaux existant, nous avons proposé une approche qui a conduit à une architecture en couches. Pour construire la première couche, nous avons adopté une méthodologie basée sur l'ingénierie des modèles permettant la migration des systèmes d'information vers les architectures orientées services. Les services obtenus seront exposés aux organisations tierces à l'aide des standards des services Web via le bus de services d'entreprise. La deuxième couche concerne l'architecture permettant de capturer la sémantique des services obtenus de la première couche. Elle est basée sur la notion d'ontologies et de contextes, et constitue le modèle de référence pour annoter les descriptions des services Web à l'aide du SAWSDL. La dernière couche, est une architecture pour l'interopérabilité sémantique exploitant les deux premières couches. Le processus de coopération est pris en charge par le service Web broker, qui orchestre les autres services d'interopérabilité et qui sont les services Web: de description, de publication, de découverte, de médiation et d'exécution.

Afin de valider notre approche et architecture une étude de cas est exposée.

***Mots clés :*** *Systèmes d'information coopératifs, Interopérabilité, Architectures orientées services, Services Web, Ingénierie des modèles, Ontologies, Contexte, Annotation sémantique, Médiation.*

# Sommaire

<b>Remerciements</b>	i
<b>ملخص</b>	iii
<b>Abstract</b>	iv
<b>Résumé</b>	v
<b>Sommaire</b>	vi
<b>Tables des illustrations</b>	xi
<b>Introduction générale</b>	1
a. Problématique et motivations	1
b. Contribution et objectifs	2
c. Organisation du document	2
<b>Chapitre 1 : Caractéristiques des systèmes d'information coopératifs et interopérables</b>	
1.1. Introduction	4
1.2. Notion de système d'information d'entreprise	4
1.2.1. Définitions	4
1.2.1.1. Aspect organisationnel	4
1.2.1.2. Aspect systémique	5
1.2.2. Notion d'application d'entreprise	6
1.2.2.1. Définition	6
1.2.2.2. Classification des applications d'entreprise	7
1.2.2.3. Dimensions des applications d'entreprise	7
1.3. Coopération et interopérabilité des systèmes d'information	9
1.3.1. Définition des systèmes d'information coopératifs	9
1.3.2. Définitions de l'interopérabilité	10
1.3.3. Niveaux d'interopérabilité	10
1.3.3.1. Interopérabilité technique	10
1.3.3.2. Interopérabilité sémantique	10
1.3.3.3. Interopérabilité organisationnelle	11
1.4. Interopérabilité et intégration	11
1.5. Outils permettant l'interopérabilité technique	11
1.5.1. Intergiciels	11

1.5.1.1. Définition	12
1.5.1.2. Avantages des intergiciels	12
1.5.1.3. Typologie des intergiciels	13
1.5.2. Quelques notions sur les architectures orientées services	13
1.5.2.1. Définition	13
1.5.2.2. Définition du service	13
1.5.2.3. Notion de service métier	14
1.5.2.4. Différents types de service	15
1.5.2.5. Avantages des architectures orientées services	16
1.5.3. Description des services Web	16
1.5.3.1. Définition	16
1.5.3.2. Définition des rôles	16
1.5.3.3. Fonctions et normes de base des services Web	17
1.5.3.4. Composition des services Web	20
1.5.3.5. Bus de services d'entreprise	20
1.5.4. Architecture guidée par les modèles	21
1.5.4.1. Définition	21
1.5.4.2. Modèles et niveaux méta	21
1.5.4.3. Avantages des architectures guidées par les modèles	23
1.6. Synthèse de quelques travaux sur l'interopérabilité technique	23
1.7. Discussion	24
1.8. Conclusion	24

## **Chapitre 2 : Outils conceptuels et technologiques pour la mise en œuvre des mécanismes d'interopérabilité sémantique**

2.1. Introduction	25
2.2. Notion de sémantique	25
2.2.1. Définition	25
2.2.2. Typologie des hétérogénéités sémantiques	25
2.2.3. Classification de la sémantique	26
2.2.4. Représentation de la sémantique	26
2.3. Apport des ontologies	27
2.3.1. Définitions	27

2.3.2.	Rôle des ontologies	27
2.3.3.	Structuration des ontologies	28
2.3.4.	Classification d'ontologies	28
2.3.4.1.	Classification selon le niveau de granularité	28
2.3.4.2.	Classification selon le niveau de formalité	29
2.3.5.	Méthodologies de construction d'ontologies	29
2.3.6.	Langages de représentation et outils de développement des ontologies	31
2.3.6.1.	Langages de représentation des ontologies	31
2.3.6.2.	Outils de développement des ontologies	34
2.3.7.	Approches d'utilisation des ontologies	35
2.3.7.1.	Approche mono-ontologie	35
2.3.7.2.	Approche multi-ontologie	36
2.3.7.3.	Approche hybride	36
2.3.7.4.	Evaluation des approches ontologiques	37
2.3.7.5.	Approche de mapping d'ontologies	38
2.4.	Classification des hétérogénéités des services web	38
2.4.1.	Hétérogénéités des propriétés non fonctionnelles	39
2.4.2.	Hétérogénéités des propriétés fonctionnelles	39
2.4.3.	Hétérogénéités des données	39
2.5.	Description sémantique de services web	39
2.5.1.	Langages de description sémantique	40
2.5.2.	Approches d'annotation des langages existant	41
2.6.	Médiation des services Web	44
2.6.1.	Définition	44
2.6.2.	Classification de la médiation	44
2.7.	Interopérabilité et intégration des services Web	45
2.7.1.	Classification des approches d'intégration des services Web	45
2.7.2.	Etude des travaux portant sur l'interopérabilité sémantique	46
2.7.3.	Discussion	48
2.8.	Conclusion	49

### **Chapitre 3 Une Architecture orientée services sémantiques pour**

## **l'interopérabilité des systèmes d'information**

3.1. Introduction	50
3.2. Approche globale	51
3.3. Passage vers une architecture orientée services	51
3.3.1. Approche de construction de la SOA métier	53
3.3.2. Phase de construction des services métiers	54
3.3.3. Phase de construction des services applicatifs	54
3.3.4. Phase de publication	56
3.4. Construction de l'architecture sémantique	56
3.4.1. Construction du sous modèle sémantique des données (SMSD)	57
3.4.1.1. Notion de contexte et d'objet sémantique	57
3.4.1.2. Description du modèle sémantique des données	58
3.4.1.3. Approche de conception des ontologies	59
3.5. Mécanismes d'annotation sémantique	60
3.6. Architecture pour l'interopérabilité sémantique des services Web	60
3.6.1. Rôle du service Web broker	61
3.6.2. Publication sémantique	62
3.6.3. Découverte sémantique	63
3.6.4. Médiation sémantique	63
3.6.5. Exécution des services Web	63
3.7. Description de la médiation des services Web	63
3.7.1. Principe de médiation sémantique des données	64
3.7.2. Description de l'architecture de médiation	65
3.7.3. Fonctions de conversion et de transformation utilisées	66
3.8. Conclusion	67

## **Chapitre 4      E t u d e d e c a s**

4.1. Introduction	68
4.2. Enoncé de l'étude de cas	68
4.3. Construction de l'architecture orientée services de l'entreprise SETA	70
4.3.1. Construction de la SOA applicative	70

4.3.2. Construction de la SOA métier	71
4.3.2.1. Définition des services fonctionnels	71
4.3.2.2. Définition des services métiers	72
4.3.3. Publication des services Web	73
4.3.4. Hétérogénéités sémantiques	73
4.4. Construction du modèle sémantique	74
4.4.1. Construction du sous modèle sémantique de données (SMSD)	74
4.4.2. Publication du modèle sémantique de données	76
4.5. Annotation des descriptions des services par le SMSD	77
4.6. Description de la médiation sémantique de données	78
4.7. Conclusion	79
<b>Conclusion générale</b>	<b>80</b>
<b>Références bibliographiques</b>	<b>82</b>

# Tables des illustrations

## Table des figures

<i>Figure I.1.</i>	<i>Structure d'un système d'information</i>	5
<i>Figure I.2.</i>	<i>Vue systémique d'un système d'information</i>	6
<i>Figure I.3.</i>	<i>Dimensions des applications d'entreprise</i>	8
<i>Figure I.4.</i>	<i>Schéma général d'un SIC</i>	10
<i>Figure I.5.</i>	<i>Organisation de l'intergiciel</i>	12
<i>Figure I.6.</i>	<i>Organisation des services en couplage faible</i>	14
<i>Figure I.7.</i>	<i>Différents types de services</i>	15
<i>Figure I.8.</i>	<i>Acteurs des services Web</i>	17
<i>Figure I.9.</i>	<i>Méta-modèle pour WSDL2.0</i>	18
<i>Figure I.10.</i>	<i>Différents éléments du message SOAP</i>	19
<i>Figure I.11.</i>	<i>Architecture d'un ESB</i>	20
<i>Figure I.12.</i>	<i>Approche MDA et les quatre niveaux de modèles</i>	22
<i>Figure I.13.</i>	<i>Transformation de méta-modèles</i>	22
<i>Figure II.1.</i>	<i>Différents types d'ontologies selon le niveau de granularité</i>	29
<i>Figure II.2.</i>	<i>Méthode d'Uschold et King</i>	30
<i>Figure II.3.</i>	<i>Composantes de Methontologie</i>	31
<i>Figure II.4.</i>	<i>Pile des langages d'annotation des ontologies</i>	32
<i>Figure II.5.</i>	<i>Extrait du méta modèle OWL</i>	34
<i>Figure II.6.</i>	<i>Editeur d'ontologies Protégé 2000</i>	35
<i>Figure II.7.</i>	<i>Approche mono-ontologie</i>	36
<i>Figure II.8.</i>	<i>Approche multi-ontologies</i>	36
<i>Figure II.9.</i>	<i>Approche hybride</i>	37
<i>Figure II.10.</i>	<i>Processus de mapping d'ontologies</i>	38
<i>Figure II.11.</i>	<i>Document WSDL2.0 avec les annotations SAWSDL</i>	43
<i>Figure II.12.</i>	<i>Parties annotées par le SAWSDL</i>	43
<i>Figure II.13.</i>	<i>Classification des niveaux de médiation entre services Web</i>	45
<i>Figure III.1.</i>	<i>Différentes couches pour l'interopérabilité sémantique des systèmes d'information</i>	52
<i>Figure III.2.</i>	<i>Cycles de construction d'une SOA</i>	53
<i>Figure III.3.</i>	<i>Approche de construction d'une SOA métier</i>	55
<i>Figure III.4.</i>	<i>Sous modèles sémantiques</i>	56
<i>Figure III.5.</i>	<i>Exemple d'un objet sémantique et son contexte</i>	58
<i>Figure III.6.</i>	<i>Processus général d'interopérabilité</i>	61
<i>Figure III.7.</i>	<i>Architecture des services Web broker</i>	62
<i>Figure III.8.</i>	<i>Principe de publication sémantique</i>	62
<i>Figure III.9.</i>	<i>Processus de médiation de données</i>	65
<i>Figure III.10.</i>	<i>Architecture conceptuelle de la médiation proposée</i>	66

<i>Figure IV.1.</i>	Articulation des systèmes d'information	69
<i>Figure IV.2.</i>	Outil Protégé 2000	75
<i>Figure IV.3.</i>	Extrait de l'ontologie locale de données de la SETA	75
<i>Figure IV.4.</i>	Extrait de l'ontologie locale de données du laboratoire	76
<i>Figure IV.5.</i>	Ontologie contextuelle de données associée au concept 'price'	76
<i>Figure IV.6.</i>	Annotation des descriptions relatives aux services Web de l'exemple.	77
<i>Figure IV.7.</i>	Ontologies contextuelles de données temporaires et conversions.	78
<i>Figure IV.8.</i>	Description du service Web médiateur	79

### **Table des tableaux et listings**

<i>Tableau I.1.</i>	Différents types d'utilisation d'un UDDI.	19
<i>Tableau I.2.</i>	Tableau comparatif des travaux existants	48
<i>Tableau II.1.</i>	Typologie des conflits sémantiques.	26
<i>Tableau II.2.</i>	Evaluation des approches ontologiques.	37
<i>Tableau IV.1.</i>	Services applicatifs de la SETA.	70
<i>Tableau IV.2.</i>	Services fonctionnels de la SETA.	71
<i>Tableau IV.3.</i>	Services métiers de la SETA et ses partenaires.	72
<i>Tableau IV.4.</i>	Hétérogénéités sémantiques des données des deux services Web exemples.	74
<i>Listing II.1.</i>	Extrait de description OWL-S	40

# Introduction Générale

Avec l'évolution des technologies de l'information, la prolifération d'Internet et la mondialisation de l'économie, les problèmes auxquels doit faire face l'entreprise sont entre autres : le changement sans cesse de son environnement, et les hétérogénéités des systèmes d'information de plus en plus complexes. Pour surmonter ces problèmes les entreprises sont contraintes à mieux communiquer avec leurs clients, leurs partenaires, leurs fournisseurs, et même avec leurs concurrents. Autrement, les entreprises ont besoin de coopérer.

En réalité, le besoin de travailler en coopération a été souligné il y a plus de deux siècles par Adam Smith<sup>1</sup> dans son texte sur une " fabrique d'épingles" : « dix ouvriers travaillant chacun de son côté ne parviennent pas à produire plus de 20 épingles par jour et par ouvrier. S'ils travaillent ensemble et se spécialisent chacun dans une étape de fabrication, les cadences montent à 4 800 épingles par ouvrier et par jour ! ».

De la même manière, si on remplace les ouvriers par des applications d'entreprises, on obtient le principe du paradigme de Systèmes d'Information Coopératifs (SIC), qui fait appel en plus du travail coopératif à d'autres technologies telles que les systèmes de base de données, l'intelligence artificielle, les langages de programmation, le génie logiciel, le génie cognitif, l'informatique répartie, ...etc.

Les avantages des SIC sont multiples, tels que, gain de temps et de productivité, et une meilleure image de l'entreprise, aussi ils permettent d'offrir de meilleures conditions de travail pour les employés, et par conséquent, une production d'informations fiables et de haute qualité nécessaires à la prise de décisions pertinentes.

A l'origine la quasi-totalité des travaux de recherche sur les SIC dits Intelligents (SICI), exploitent la notion d'Agents d'Information Intelligents (AII).

Aujourd'hui, avec l'évolution des Technologies de l'Information et de la Communication (TIC), le langage XML, les architectures orientées services, les standards des services Web, et bien d'autres outils techniques et conceptuels, nous allons voir dans le cadre de cette thèse, quels sont les avantages et les bénéfices tirés par les SIC grâce à ces technologies.

## a. Problématique et motivations

Dans le contexte économique et technologique actuel, la coopération des systèmes d'information des entreprises est une nécessité incontournable, permettant de faire face aux contraintes qui pèsent sur les entreprises.

Parallèlement, l'avènement des paradigmes d'architecture orientée services<sup>2</sup>, et des services Web qui se sont imposés rapidement, a profondément changé l'orientation des entreprises en raison des avantages qu'ils procurent, tels que : couplage faible, accès simple et rapide aux fonctionnalités offertes, et interopérabilité des plates-formes. Ceci a incité les entreprises à passer vers les architectures orientées services, un passage qui n'est pas aussi évident à cause d'une part de la complexité des systèmes d'information existants, et d'autre part au manque d'une méthodologie de migration vers les architectures orientées services.

---

1 : Economiste écossais partisan du libre échange.

2 : SOA pour Service Oriented Architecture. Nous retenons tout au long de ce document le terme SOA.

Le SOA et les services Web offrent des solutions d'interopérabilité des applications des entreprises aux niveaux technique et syntaxique, mais ne traitent pas la problématique liée à l'interopérabilité sémantique qui, constitue l'enjeu majeur pour la coopération des systèmes d'information. En effet, les hétérogénéités des données et fonctionnalités échangées entre services Web sont complexes et rendent leur interopérabilité difficile. De ce fait, il est impératif de capturer la sémantique liée aux informations échangées, et de résoudre les conflits causés par les hétérogénéités sémantiques.

### **b. Contribution et objectifs**

Notre travail de recherche s'inscrit dans le cadre des systèmes d'information coopératifs, et des architectures orientées services. IL a pour objectif principal la proposition d'une architecture orientée services pour la mise en œuvre de l'interopérabilité des systèmes d'information. Cette architecture répond aux problématiques que nous avons citées par :

- L'adoption d'une méthodologie basée sur l'ingénierie des modèles (MDA pour Model Driven Architecture) qui permet d'une part une migration progressive et adéquate des systèmes d'information existants vers des architectures orientées services, et d'autre part la construction de nouveaux systèmes d'information orientés services. L'architecture obtenue est basée sur les technologies des services Web qui sont normalisées et standardisées sous le contrôle des organisations W3C<sup>3</sup>, OASIS<sup>4</sup>, et WS-I<sup>5</sup>.
- La proposition d'une architecture sémantique basée sur la notion d'ontologies pour capturer la sémantique des propriétés fonctionnelles et non fonctionnelles des services Web. Nous présentons en particulier les ontologies décrivant les données échangées.
- L'annotation des descriptions des services Web par le modèle sémantique en utilisant le 'Semantic Annotation of Web Service Description Language', un standard récent qui fait partie des recommandations du consortium W3C. Il s'agit en particulier de l'annotation des données échangées entre services Web.
- Et enfin, la proposition d'une architecture basée services pour l'interopérabilité des systèmes d'information, et la définition des mécanismes d'interopérabilité sémantique des systèmes d'information basés sur les concepts précédents. Notamment la médiation sémantique des données.

### **c. Organisation du document**

Le document est structuré en quatre chapitres comme suit :

- Le premier chapitre est consacré aux notions de systèmes d'information, des applications d'entreprises et leurs caractéristiques, des systèmes d'information coopératifs, de l'interopérabilité et ses niveaux. Il présente quelques outils permettant l'interopérabilité aux niveaux technique et syntaxique. Il se termine par une synthèse des travaux relatifs à l'interopérabilité technique.

---


3 : World Wide Web Consortium.

4 : Organization for the Advancement of Structured Information Standards

5 : Services Web Interoperability Organization

- Le deuxième chapitre étudie les approches d'interopérabilité sémantique, notamment les ontologies, et la médiation. Il se termine par une étude des travaux présentant des solutions sémantiques.
- Le troisième chapitre est consacré à la description de notre architecture. Dans un premier temps, nous présentons l'approche globale de notre architecture en couches. Puis, nous détaillons chaque point de cette approche, en présentons d'abord une méthodologie de construction de la couche de base qui est une architecture orientée services. Ensuite, nous décrivons le modèle sémantique et en particulier le sous modèle de données grâce auquel nous enrichissons sémantiquement les descriptions des services Web, ce qui constitue la couche sémantique. Tous ces concepts et mécanismes seront exploités par la dernière couche en l'occurrence l'architecture d'interopérabilité basée services.
- Le dernier chapitre fera l'objet d'une étude de cas qui met en évidence les concepts et mécanismes de notre architecture.

Enfin, cette thèse est clôturée par une conclusion dans laquelle nous dressons un bilan et une synthèse du travail effectué, et nous discutons un ensemble de perspectives que nous jugeons nécessaires à l'achèvement de notre travail.



# Chapitre I

## Caractéristiques des systèmes d'information coopératifs et interopérables

# Chapitre I

## Caractéristiques des systèmes d'information coopératifs et interopérables

### I.1. Introduction

Depuis longtemps, le besoin en information sans cesse a conduit au développement d'un nombre considérable de systèmes d'information chargés d'acquérir de traiter de stocker, d'interroger, et de restituer des données. Chaque système d'information interagit avec son environnement en amont et en aval. De nos jours, avec l'avènement d'Internet et l'évolution des Technologies de l'Information et des Communications (TIC), les systèmes d'information sont affrontés à de multiples problèmes à savoir : hétérogénéités, complexité coûteuse et peu économique, et changement rapide de leur environnement concurrentiel. Pour surmonter ces problèmes, les entreprises sont contraintes à mieux communiquer avec les autres acteurs économiques, et son système d'information doit s'adapter de façon souple et rapide aux changements de son environnement et coopérer avec les autres systèmes.

L'objet de ce chapitre est de présenter les notions de systèmes d'informations et de leur coopération et de discuter quelques outils permettant leur interopérabilité au niveau technique.

### I.2. Notion de système d'information d'entreprise

#### I.2.1. Définitions

Nous distinguons deux définitions, la première selon un aspect organisationnel, et la deuxième selon un aspect systémique :

##### I.2.1.1. Aspect organisationnel

R. Reix définit le système d'information comme un ensemble organisé de ressources (figure I.1.) [3] :

- matériels (machines informatiques, support, ...)
- personnel (utilisateurs, informaticiens...)
- données (connaissances, modèles...)
- logiciels et procédures (programmes informatiques, des méthodes de travail...)

permettant :

- d'acquérir
- de traiter
- de stocker
- et de communiquer

des informations sous des formes variées au sein d'une entreprise.

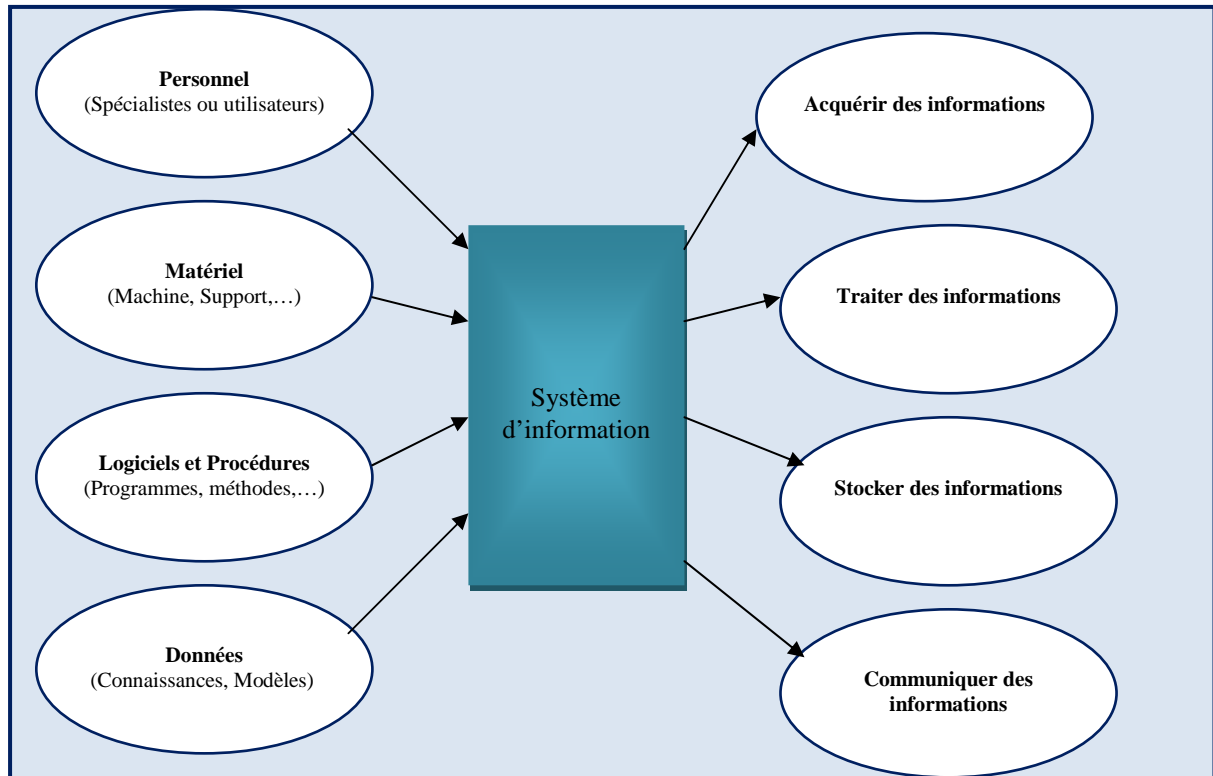
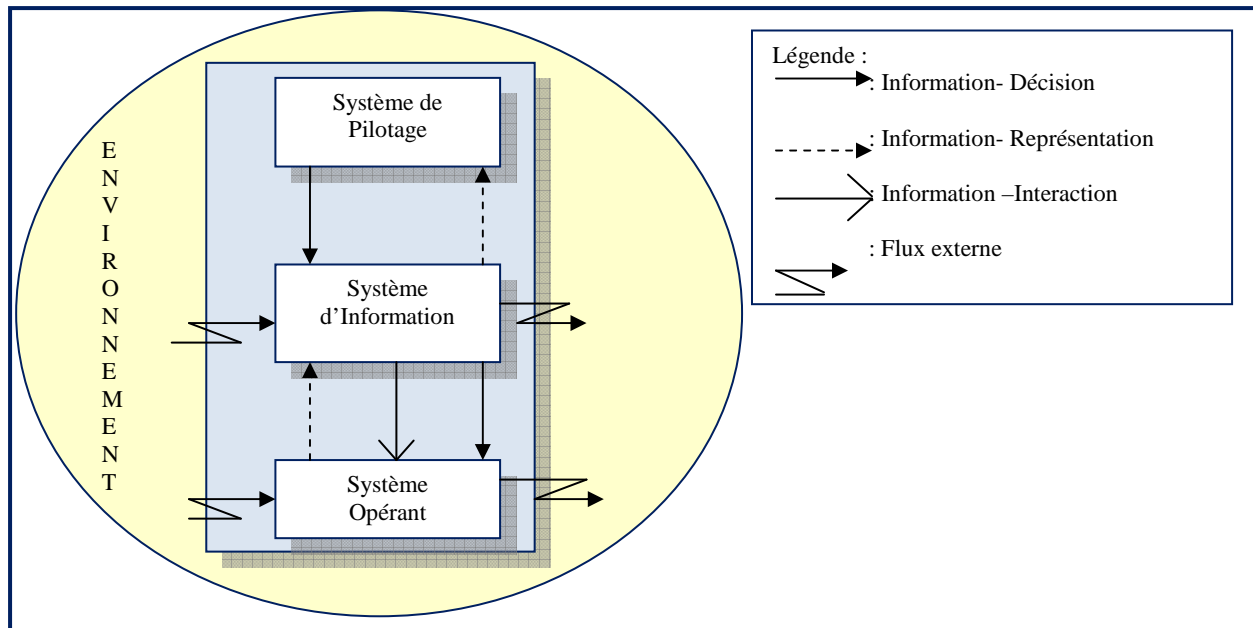


Figure I.1. Structure d'un système d'information [3].

### I.2.1.2. Aspect systémique

D'après J.L. Lemoigne [6] une organisation est composée de trois types de systèmes qui sont (figure I.2.) :

- ✓ *Le système opérant*, qui réagit aux événements quotidiens, qui viennent de l'environnement, selon les règles définies. Il est chargé de transformer des ressources ou flux primaires (flux financiers, flux de personnel, flux de matière, flux d'information).
- ✓ *Le système de pilotage*, qui permet d'engager le processus de décision tout en définissant au préalable les objectifs, les critères d'évaluation et les règles de gestion. Il dirige l'entreprise et maintient le cap sur les objectifs choisis.
- ✓ *Le système d'information*, qui relie les deux systèmes précédents tout en jouant un rôle de coupleur. Il correspond à la partie chargée de la collecte, du traitement, du stockage et de la diffusion des informations. Il peut être perçu comme une représentation de l'activité du système opérant et/ou du système de pilotage.



Selon leur finalité principale C. Tessier distingue la typologie des systèmes d'information suivante [8]: des systèmes d'information supports d'opérations (traitement de transaction, contrôle de processus industriels, etc.) et des systèmes d'information supports de gestion (aide à la production de rapports, aide à la décision, etc.).

Un système d'information peut être totalement manuel comme il peut comporter une partie automatisée appelée système informatique ou système automatisé d'information ou TIC.

## 1.2.2. Notion d'application d'entreprise

Un système automatisé d'information peut être composé de plusieurs applications (appelées également applications d'entreprise ou applications informatiques).

### 1.2.2.1. Définition

D'après S. Izza [1] une application (ou application d'entreprise) est un ensemble de programmes (logiciels) articulés entre eux, utilisés pour automatiser ou assister des tâches de traitement de l'information dans un domaine particulier au sein de l'entreprise.

D'après [2] et [3] une application est principalement caractérisée par :

- Les composants applicatifs qu'elle comporte et qui représentent des sous-ensembles cohérents (modules logiciels) de l'application ;
- Le ou les champs d'application (contexte d'utilisation) définis soit structurellement (un poste de travail, un département, etc.) soit fonctionnellement (une fonction de gestion telle que la gestion de la maintenance, la gestion des commandes, etc.) ;
- Des fonctionnalités qui correspondent à un ensemble de tâches qui peuvent être prises en charge par l'application ;
- Les données manipulées qui correspondent aux différentes données utilisées et produites par l'application.

- Et les différentes ressources humaines, logicielles et matérielles utilisées par l'application.

### I.2.2.2. Classification des applications d'entreprise

Dans la littérature des systèmes d'information on trouve une panoplie de classifications selon des critères différents :

- Classification des applications selon trois tiers [1] :
  - Front-end, ensemble d'applications qui interagissent directement avec l'utilisateur.
  - Back-end, ensembles d'applications liées aux systèmes de production de l'entreprise dans lesquels se concentre la valeur ajoutée de l'entreprise.
  - Middle-end, comprend l'ensemble des applications et des technologies nécessaires pour interconnecter le front-end et le back-end.
- Classification selon les couches logiques d'une application [1] :
  - Architecture 1-tiers, correspond aux applications monolithiques construites sur le principe d'un niveau physique qui englobe les trois couches logiques (couche présentation, couche applicative et couche données).
  - Architecture 2-tiers, correspond aux applications construites selon le mode client/serveur.
  - Architecture 3-tiers, implique une séparation physique des trois couches logiques au sein des applications.
- Classification selon le mode de traitement des événements [4];
  - Les applications batch (AB) : qui sont les applications les plus anciennes et qui sont conçues avec une philosophie d'application monolithique lourde et qui permettent de traiter en différé un ensemble d'évènements groupés dans des fichiers ou éventuellement dans des bases de données.
  - Les applications transactionnelles (AT) : qui sont des applications qui permettent de traiter les événements les uns après les autres et dont la plus part peuvent assurer des interfaces avec les utilisateurs.
  - Les applications client/serveur (AC) : qui sont une forme évoluée des applications transactionnelles et qui reposent sur le modèle client-serveur où une application cliente peut contacter le serveur et lui envoyer une requête que le serveur traite pour retourner ses réponses à l'application appelante.
  - Les applications Web (AW) : qui sont une forme particulière d'applications transactionnelles qui tournent sur une technologie Web.
  - Les applications au fil de l'eau (AF) : qui sont des applications relativement récentes et essentiellement asynchrones qui utilisent des technologies de messagerie inter-applications (MOM- Message Oriented Middleware) pour communiquer avec d'autres applications. Elles fonctionnent selon des logiques qui peuvent relever à la fois des modes batch et transactionnel.

### I.2.2.3. Dimensions des applications d'entreprise

Les applications peuvent être analysées selon trois principales caractéristiques, qui sont [9] [10] : l'autonomie, la distribution et l'hétérogénéité. La figure I.3 montre ces dimensions avec quelques approches générales permettant de les gérer (flèches en pointillées).

Une quatrième dimension est rajoutée par certains auteurs comme M. Singh et M. Huhns [11], il s'agit du dynamisme, en effet les applications peuvent évoluer avec le temps, ceci est dû aux évolutions et aux changements qui s'opèrent dans leur environnement.

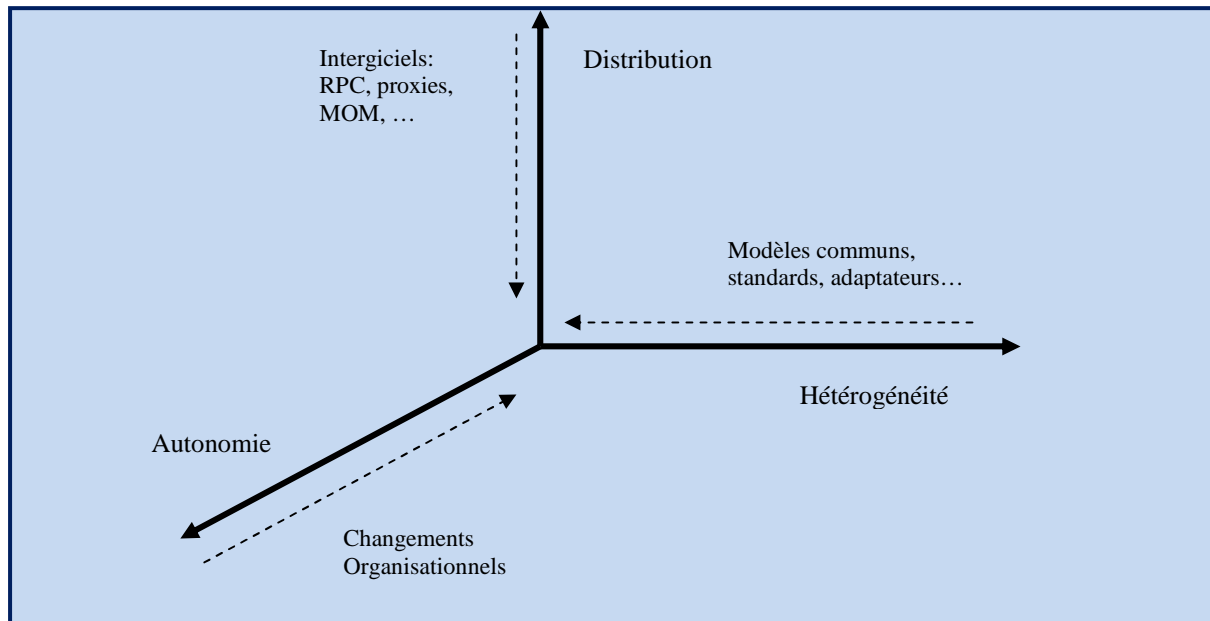


Figure I.3. Dimensions des applications d'entreprise [1].

#### **Autonomie :**

L'autonomie des applications désigne le fait qu'elles peuvent être conçues et exécutées indépendamment les unes des autres. D'après M.T. Ozsu et P. Valduriez [12] l'autonomie possède plusieurs aspects :

- Autonomie de conception, qui signifie qu'une application est indépendante d'autres applications dans sa conception intrinsèque (son modèle de données, son modèle de traitement, etc.).
- Autonomie de communication, qui signifie qu'une application peut localement choisir avec quelles applications elle peut communiquer.
- Autonomie d'exécution, qui signifie l'indépendance de l'application pour gérer les interactions avec son environnement extérieur.

#### **Distribution :**

La distribution désigne le fait que les applications sont très souvent réparties physiquement sur le réseau d'entreprise. Cette distribution des applications est réalisée grâce à des technologies telles que CORBA, Java/RMI, COM/DCOM, etc. (voir section I.5.1.3).

#### **Hétérogénéité :**

Dans un système interopérable les hétérogénéités peuvent survenir à différents niveaux. G. Wiederhold [13] distingue : (1) *l'hétérogénéité technique* qui correspond aux différences présentes dans les matériels et les logiciels (ordinateurs, réseaux, plates formes). (2) *l'hétérogénéité syntaxique* qui correspond aux différences présentes dans les formats de données et des interfaces. (3) *l'hétérogénéité sémantique* qui correspond aux différences liées à l'interprétation et au sens associé aux données et aux fonctions d'une application.

Dans le domaine des données D. Benslimane [14] distingue (1) *l'hétérogénéité syntaxique* qui traduit la diversité des choix au niveau des modèles logiques des données. (2) *l'hétérogénéité structurelle* qui concerne la structure des objets réels et leur classification qui varient d'un système à un autre. Elle inclut la différence de types de données et leur agrégation. (3) *l'hétérogénéité sémantique* qui concerne la différence de perception des objets réels d'un système à un autre. Elle concerne les noms des éléments de schémas (attributs, classes), les valeurs (différence d'échelles ou d'unités) etc.

Dans le cadre de nos travaux nous nous focalisons sur la dimension d'hétérogénéité des applications des systèmes d'information, et plus précisément sur l'hétérogénéité sémantique, qui présente plus de difficultés et freine l'interopérabilité des systèmes d'informations. Nous allons décrire la notion de sémantique et une classification de l'hétérogénéité liée à cette notion dans le deuxième chapitre (§ II.2)

#### ***Dynamisme :***

Les systèmes d'information sont ouverts sur un environnement en perpétuels changements et qui constituent le défi majeur qu'affrontent les entreprises. Pour faire face à ce problème les applications doivent s'adapter de manière souple et rapide et évoluer de façon dynamique.

### **I.3. Coopération et interopérabilité des systèmes d'information**

Comme nous l'avons déjà mentionné, le travail de recherche que nous présentons se focalise sur le paradigme de systèmes d'information coopératifs. De ce fait, nous tenons à le définir, et à présenter la notion d'interopérabilité qui le caractérise.

#### **I.3.1. Définition des systèmes d'information coopératifs (SIC)**

D'après Benslimane et al. [14], un SIC est un système ayant la capacité de supporter le partage de données et le traitement coopératif inter-applications. Comme le montre la figure I.4, une application X, appelée client, peut inter-opérer avec un ensemble de systèmes (sources de données)  $Y_i$   $i=1,2,\dots,n$ , appelés fournisseurs si et seulement si : (1) la demande de service de l'application X peut être correctement interprétée par chacun des systèmes  $Y_i$ , et (2) les réponses des systèmes  $Y_i$  peuvent être correctement interprétées par l'application X. Cette définition des SIC peut être vue comme une généralisation de l'approche client/serveur à deux notion [14] :

*La notion de compréhension mutuelle entre le client et ses fournisseurs :* le client ne s'adapte pas à la sémantique du fournisseur. Il exprime sa requête dans son propre contexte. De la même manière, le fournisseur répond à la requête dans son propre contexte. La compréhension mutuelle entre l'application X et les différents fournisseurs  $Y_i$  est réalisée au travers de l'utilisation d'un contexte de référence qui conceptualise un domaine d'application.

*La notion de découverte de fournisseur :* le client exprime ses besoins en matière de données sans avoir au préalable une connaissance sur les fournisseurs existants. L'ensemble des  $Y_i$  qui peuvent coopérer au traitement d'une demande de X est déterminé dynamiquement.

Nous soulignons deux caractéristiques importantes des SIC, à savoir : la transparence pour l'utilisateur, et l'autonomie des systèmes d'information composants.

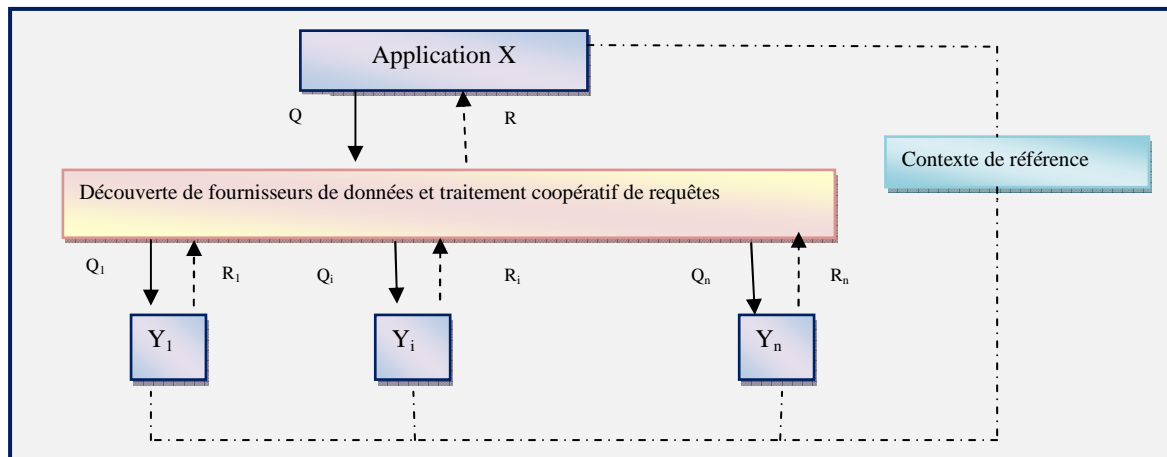


Figure I.4. Schéma général d'un SIC [14].

### I.3.2. Définitions de l'interopérabilité

Dans la littérature, de nombreuses définitions concernant l'interopérabilité des systèmes d'information ont été fournies, nous citons quelques unes :

- « L'interopérabilité est la capacité des systèmes informatiques et des processus qu'ils supportent d'échanger des données et de permettre le partage d'information et de connaissance », d'après EIF [15].
- « L'interopérabilité est la capacité que possèdent deux ou plusieurs systèmes ou composants à échanger des informations puis à exploiter les informations venant d'être échangées », d'après IEEE 1990 [16].
- « L'interopérabilité est la capacité, approuvée mais non garantie par la conformité à un ensemble de standards, d'équipements hétérogènes, généralement développés par différents fabricants, de travailler ensemble dans un environnement en réseau », d'après IEEE 2000 [17].
- « L'interopérabilité des applications d'entreprise est définie comme la capacité pour un système d'échanger de l'information et des services dans un environnement technologique et organisationnel hétérogène » d'après H. Panetto et al. [18].

### I.3.3. Niveaux d'interopérabilité

L'EIF (European Interoperability Framework) a proposé trois niveaux d'interopérabilité [19] :

**I.3.3.1. Interopérabilité technique**, relative à la mise en œuvre des technologies de l'information et de la communication concernant les normes pour présenter, stocker, échanger, traiter et communiquer les données au moyen de matériel informatique.

**I.3.3.2. Interopérabilité sémantique**, de niveau plus conceptuel, qui doit assurer que les informations échangées sont compréhensibles du point de vue de leur signification et de leur interprétation par les applications qui les utilisent mais ayant été développées pour des objectifs différents. L'interopérabilité sémantique permet à des systèmes de combiner

l'information reçue de la part d'autres sources d'information et de la traiter tout en conservant le sens.

**I.3.3.3. Interopérabilité organisationnelle**, définissant les responsabilités, les autorisations, les confiances, les aspects légaux, les propriétés intellectuelles et les structures organisationnelles nécessaires à l'acceptation des échanges d'information entre applications par les différents acteurs. Ce niveau d'interopérabilité est particulièrement mis en avant dans le cadre de l'interopérabilité de l'administration électronique et des gouvernements [18] [25].

## **I.4. Interopérabilité et intégration**

« L'intégration des systèmes consiste à assembler les différentes parties d'un système et à assurer leur compatibilité ainsi que le bon fonctionnement du système complet », d'après LAROUSSE.

Il s'agit de faire tomber les barrières fonctionnelles et organisationnelles au sein des entreprises afin que l'ensemble soit vu comme un tout cohérent. [5].

L'intégration n'est pas un but en soi, c'est un moyen d'assurer la cohérence fonctionnelle et informationnelle de l'entreprise. L'interopérabilité des systèmes, n'est elle aussi qu'un moyen, parmi d'autres pour faciliter l'intégration. On utilise généralement dans la littérature le vocable d'interopérabilité de systèmes quand les systèmes concernés sont aptes à s'échanger des informations et à agir ensemble, et on réserve le vocable d'intégration de systèmes pour le cas où les systèmes coopèrent au sein d'un système unique et homogène [20].

Ces deux termes peuvent devenir synonymes seulement dans le cas où l'on s'intéresse à l'intégration faiblement couplée, c'est-à-dire que l'on cherche à préserver l'hétérogénéité et l'autonomie des systèmes constituants [1].

L'objectif de notre travail de recherche est d'assurer l'interopérabilité technique et sémantique des systèmes d'information en proposant une architecture adéquate pour cela. Aussi, nous décrivons dans ce qui suit les outils permettant l'interopérabilité technique et syntaxique. Le deuxième chapitre sera consacré à l'étude des outils techniques et conceptuels permettant la mise en œuvre de l'interopérabilité sémantique.

## **I.5. Outils permettant l'interopérabilité technique**

Tout au long de ces dernières années une panoplie de techniques ont été mises en œuvre pour permettre l'interopérabilité des systèmes d'information. Nous décrivons dans ce qui suit les outils techniques et conceptuels suivants: les intergiciels, les architectures orientées services, les services Web et l'ingénierie des modèles.

### **I.5.1. Les intergiciels**

Le domaine de l'intergiciel est apparu dans les années 90, pour répondre aux problèmes d'interopérabilité des applications.

### I.5.1.1. Définition

Traditionnellement, le terme intergiciels (en anglais middleware) désigne l'ensemble des technologies permettant de faire communiquer des composants informatiques en fournissant des services de transport et de routage [22] [21].

La figure I.5 montre l'intergiciel en tant que couche intermédiaire installée au-dessus des systèmes d'exploitation et des protocoles de communication.

Les intergiciels réalisent les fonctions suivantes [23] :

- cacher la répartition, c'est à dire le fait qu'une application est constituée de parties interconnectées s'exécutant à des emplacements géographiquement répartis;
- cacher l'hétérogénéité des composants matériels, des systèmes d'exploitation et des protocoles de communication utilisés par les différentes parties d'une application;
- fournir des interfaces uniformes, normalisées, et de haut niveau aux équipes de développement et d'intégration, pour faciliter la construction, la réutilisation, le portage et l'interopérabilité des applications;
- fournir un ensemble de services communs réalisant des fonctions d'intérêt général, pour éviter la duplication des efforts et faciliter la coopération entre applications.

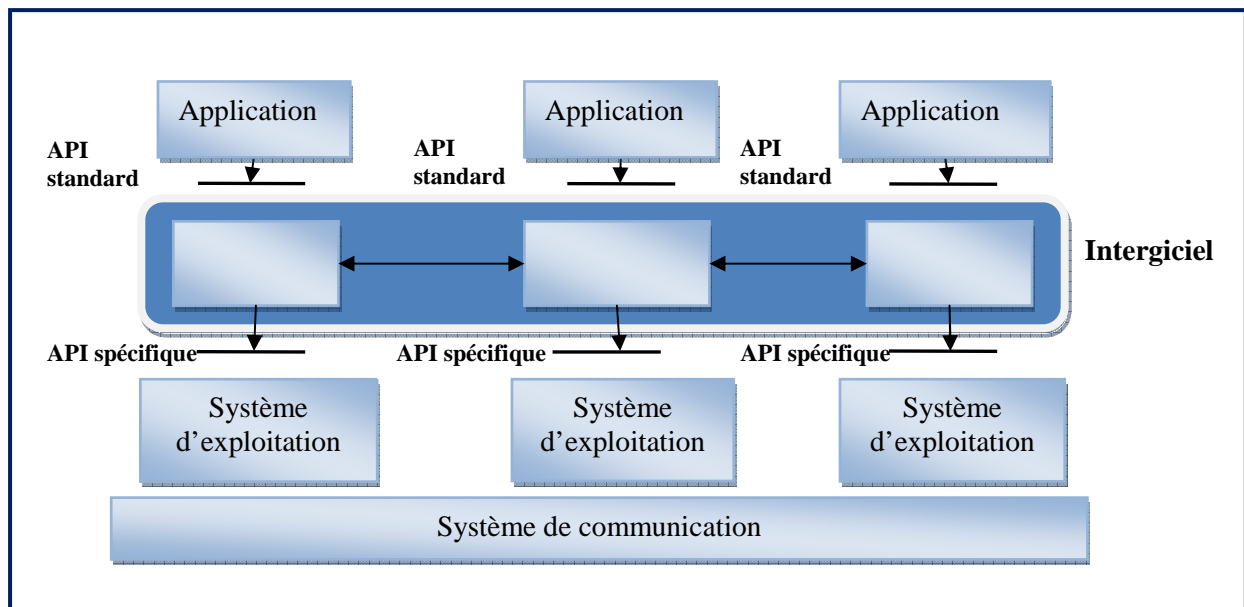


Figure I.5. Organisation de l'intergiciel [23].

### I.5.1.2. Avantages des intergiciels

Grâce aux fonctions réalisées par les intergiciels, l'utilisation de ces derniers présente plusieurs avantages qui sont:

- ✓ cacher les détails des mécanismes de bas niveau,
- ✓ assurer l'indépendance vis-à-vis des langages et des plates-formes,
- ✓ permettre de réutiliser l'expérience et parfois le code,
- ✓ faciliter l'évolution des applications,
- ✓ réduire les coûts et la durée de développement des applications et améliorer leur portabilité et leur interopérabilité.

### I.5.1.3. Typologie des intergiciels

Les intergiciels sont multiples et en évolution permanente, dans cette section, nous allons présenter en bref les intergiciels de base, lesquels beaucoup d'auteurs ont regroupés en sept grandes familles [1]:

- ✚ **Les intergiciels d'accès aux bases de données** : ODBC (Open Data Base Connectivity) et JDBC (Java DBC).
- ✚ **Les intergiciels d'appel de procédures à distance RPC (pour, Remote Procedure Call)** constituent un des moyens les plus élémentaires pour exécuter des programmes distants à travers le réseau.
- ✚ **Intergiciels orientés composants**, tels que :
  - CORBA (Common Object Request Broker Architecture).
  - COM (Component Object Model).
  - DCOM (Distributed COM).
- ✚ **Intergiciels orientés messages**: permettant l'échange de messages en utilisant une communication asynchrone, appelés MOM (pour, Message Oriented Middleware).
- ✚ **Intergiciels orientés transactions**: permettant principalement de gérer les accès aux bases de données. L'intégrité des transactions est assurée en définissant des transactions de type ACID (Atomicité, Consistance, Isolation, Durabilité).
- ✚ **Serveurs d'applications**: sont des environnements de développement et d'exécution. Il existe deux catégories de serveurs qui sont les serveurs J2EE (pour, Java 2 Entreprise Edition) et les serveurs propriétaires.  
Exemples de serveurs J2EE: BEA Weblogic, IBM Websphere, SUN/ netscape iplanet, Jonass, Jboss.  
Exemples de serveurs propriétaires: microsoft.Net, Python Zope, openACS.

## I.5.2. Quelques notions sur les architectures orientées services

### I.5.2.1. Définition

D'après P. Bonnet, les architectures orientées services ou SOA (pour, Service Oriented Architecture), est une démarche méthodologique de construction des systèmes d'information et d'implémentation du logiciel sous la forme de service [24].

Il s'agit de bâtir des architectures applicatives adaptées aux serveurs d'applications, capables d'exposer des services interopérables (legacy, progiciels) et de collaboration entre les entreprises (services métiers) [24].

### I.5.2.2. Définition du service

Un service expose une interface qui définit le traitement offert sous la forme d'un message d'entrée et d'un autre de réponse. Pour son implémentation, le service peut solliciter une ou plusieurs entités physiques de logiciel (des composants).

Le service exprime donc un niveau « logique » d'accès aux traitements et pas un niveau « physique » d'implémentation [24].

Le service respecte cinq propriétés qui sont: couplage faible, activable à distance et interopérable, asynchrone; expose un contrat d'utilisation (interface), et respecte le pattern d'architecture SOA. Nous détaillons dans ce qui suit les propriétés obligatoires qui sont le couplage faible, l'exposition d'un contrat et le respect du pattern d'architecture SOA [24] :

- a- couplage faible:** un service ne peut pas appeler un autre service. Il délègue cette fonction à un traitement spécialisé dans l'enchaînement appelé fonction d'orchestration (voir figure I.6.).

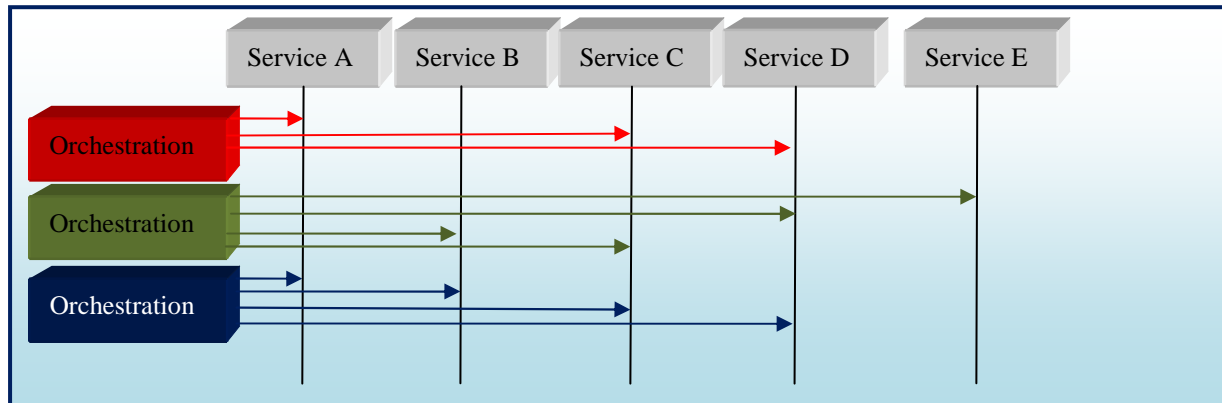


Figure I.6 Organisation des services en couplage faible [24].

Aussi, un service est activable indépendamment de sa technologie. Pour ce faire l'activation se réalise par l'envoi (et la réception) d'un message XML [40].

Un service peut être activé suivant un mode asynchrone. Dans ce cas, le service s'abonne à un événement auprès d'une fonction d'orchestration.

- b- expose un contrat d'utilisation (interface de service):** ce contrat est décrit en deux parties: une abstraite qui déclare les messages d'entrée et de réponse du traitement offert. Une partie concrète qui décrit les standards et protocoles techniques utilisés pour l'activation du service (XML- RPC, SOAP- HTTP, SOAP-JMS...).
- c- respecte le pattern d'architecture SOA:** ce pattern consiste à créer une architecture applicative qui décompose les traitements sous la forme de services rattachés à des paquets de classes. Ces paquets forment des catégories (objets métiers, sujets métiers), chacune dotée d'une façade d'accès qui contient l'ensemble des services qu'elle expose un service a le droit d'interagir uniquement avec les classes de sa catégorie.

#### I.5.2.4. Notion de service métier

Le service métier est un service qui n'est pas découvert à l'occasion de la mise en place du pattern d'architecture SOA. Il est découvert directement au niveau de la modélisation des processus (diagramme d'activités).

Il correspond à un périmètre fonctionnel que l'on souhaite exposer à des consommateurs indépendamment des choix d'architecture applicative. Le service métier est implémenté à l'aide de services (applicatifs) [24].

En pratique, il y a une correspondance directe entre le service Web et le service métier, de ce fait la meilleure implémentation de ce dernier est d'utiliser les standards des services Web.

### I.5.2.5. Les différents types de service

Le concept de service intervient à plusieurs niveaux dans l'architecture du système d'information. P. Bonnet [24] présente trois niveaux selon la granularité du service (voir figure I.7):

- **service métier:** c'est le « plus haut » niveau (Coarse grained, pour forte granularité). Il s'agit du service qui fait sens pour le métier, donc pour la maîtrise d'ouvrage<sup>6</sup> (MOA) du fournisseur mais aussi pour la maîtrise d'ouvrage du consommateur.

- **service exposé par une catégorie:** il s'agit du niveau de l'architecture applicative du logiciel (medium grained, pour moyenne granularité). Ces services peuvent être visibles de la maîtrise d'ouvrage du fournisseur car ils sont au carrefour de la préoccupation métier et d'architecture du logiciel. Ces services forment le grain privilégié d'administration et de réutilisation des services.

- **service interne à une catégorie:** il s'agit des services qui implémentent ceux exposés par les catégories (fine grained, pour fine granularité). Ces services sont proches de l'implémentation logicielle notamment du concept de composants. Ils ne sont pas visibles de la maîtrise d'ouvrage.

Bien entendu tous ces services sont visibles de la maîtrise d'œuvre du fournisseur<sup>7</sup> (MOE).

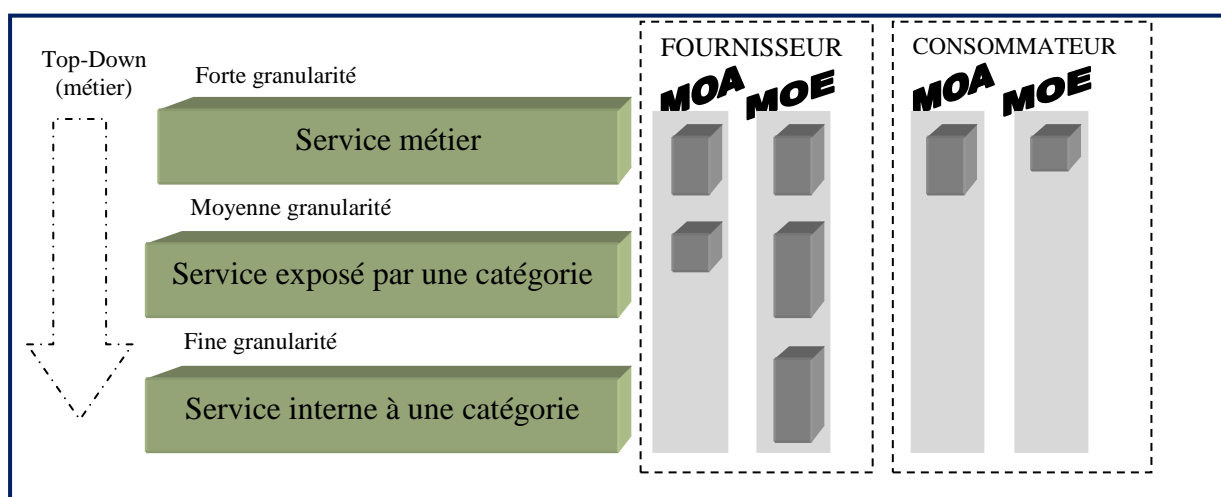


Figure I.7 Différents types de services.

Une autre classification est donnée selon le type d'usage comme suit [24]:

- **usage métier et fonctionnel:** Il s'agit de l'exposition d'une fonctionnalité à des organisations clientes tierces qui consomment le service. Si ces organisations sont internes à l'organisation du fournisseur, on parle aussi d'usage fonctionnel.

- **usage technique:** Il s'agit de l'exposition d'une fonctionnalité qui permet la communication entre des systèmes informatiques hétérogènes (par exemple, contexte de type EAI ou ESB).

6 : les gens métiers.

7 : les gens informatiques (développeurs, ingénieurs).

### **I.5.2.6. Avantages des architectures orientées services**

Les architectures orientées services :

- ✓ favorisent la réutilisation,
- ✓ permettent l'interopérabilité,
- ✓ encapsulent la complexité,
- ✓ et conservent l'existant grâce à leur couplage faible.

### **I.5.3. Description des services Web**

#### **I.5.3.1. Définition**

Nous devons préciser que les services Web sont eux aussi des intergiciels, mais de haut niveau, car ils sont implantés sur des intergiciels de base tels que les serveurs d'applications.

Vu l'importance des services Web, plusieurs définitions de ces derniers ont été mises en avant, nous retenons les suivantes:

- « Un Service Web est un composant implémenté dans n'importe quel langage, déployé sur n'importe quelle plate-forme et enveloppé dans une couche de standards dérivés du XML. Il doit pouvoir être découvert et invoqué dynamiquement par d'autres services » [121].
- «Un service Web est un logiciel identifié par une URI (Uniforme Ressource Identifier), dont l'interface est publique et les liaisons sont définies et décrites en utilisant XML. L'environnement du service offre le moyen à d'autres logiciels de découvrir celui-ci. Ce service interagit avec les autres services Web en respectant cette définition, donc en utilisant des messages basés sur XML acheminés par des protocoles Internet.» [26].

#### **I.5.3.2. Définition des rôles**

Les services Web sont une technologie permettant à des applications de dialoguer via Internet, par l'échange de messages fondés sur des standards, et ceci indépendamment des plates-formes et des langages sur lesquelles elles reposent [27].

Il ya trois acteurs dans l'utilisation des services Web qui interagissent ensemble (figure I.8):

- le fournisseur de service, qui publie le service.
- le demandeur de service qui demande l'utilisation d'un service.
- l'annuaire de services, permettant de répertorier l'ensemble des fournisseurs de services.

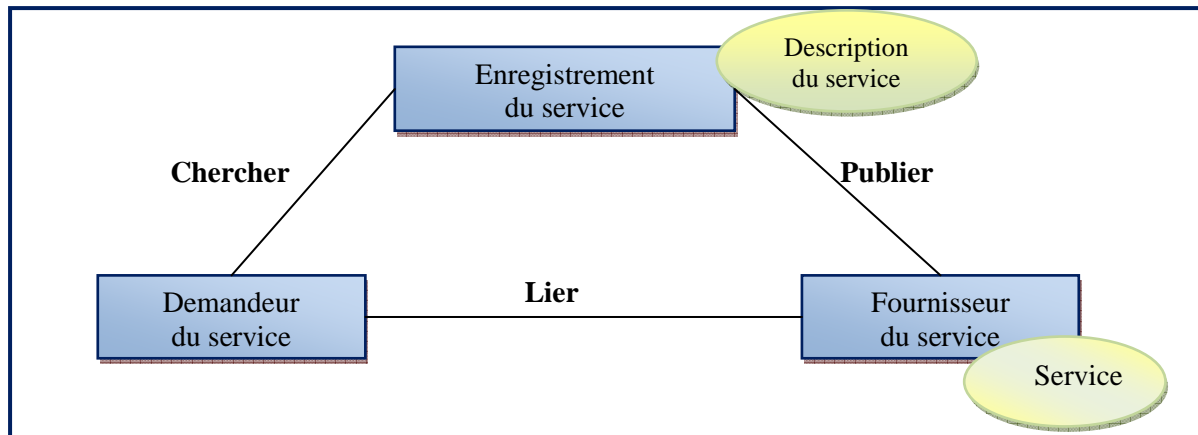


Figure I.8. Les trois acteurs des services Web [27].

### I.5.3.3. Les fonctions et les normes de base des services Web

Les services Web s'appuient sur un ensemble de fonctions et de protocoles standards. Dans cette section nous allons présenter les fonctions de base, et les normes qui les supportent.

Les fonctions permettent d'assurer la [27]:

- **découverte**: comment identifier et localiser les services Web.
- **description**: comment exposer les fonctions des services Web.
- **échange**: comment échanger les messages entre les services Web.

Ces fonctions sont supportées par les normes suivantes [27]:

#### WSDL:

Cette norme est le résultat de la consolidation de NASL (Ariba), SCL (Microsoft) et SDL (IBM), est devenue un standard du W3C. WSDL (Web Service Description Language) est fondée sur XML [40]. Elle permet de spécifier le format des messages, les protocoles qui doivent être utilisés et la localisation des différentes machines qui mettent en œuvre un service Web. Elle définit un format de document qui permet de décrire l'interface d'un service Web. La récente version de WSDL est WSDL2.0 (recommandation de W3C en 2005) [41], dont nous présentons son méta-modèle dans la figure I.9 ci-après.

#### SOAP:

Le principal objectif du protocole SOAP est de permettre la normalisation des échanges de données. SOAP (Simple Object Access Protocol) est un standard du consortium W3C définissant un protocole indépendant du transport. Il assure l'interopérabilité entre composants en restant indépendant des plates formes et des langages de programmation. Il repose sur deux standards XML et HTTP respectivement pour la structure des messages et pour le transport.

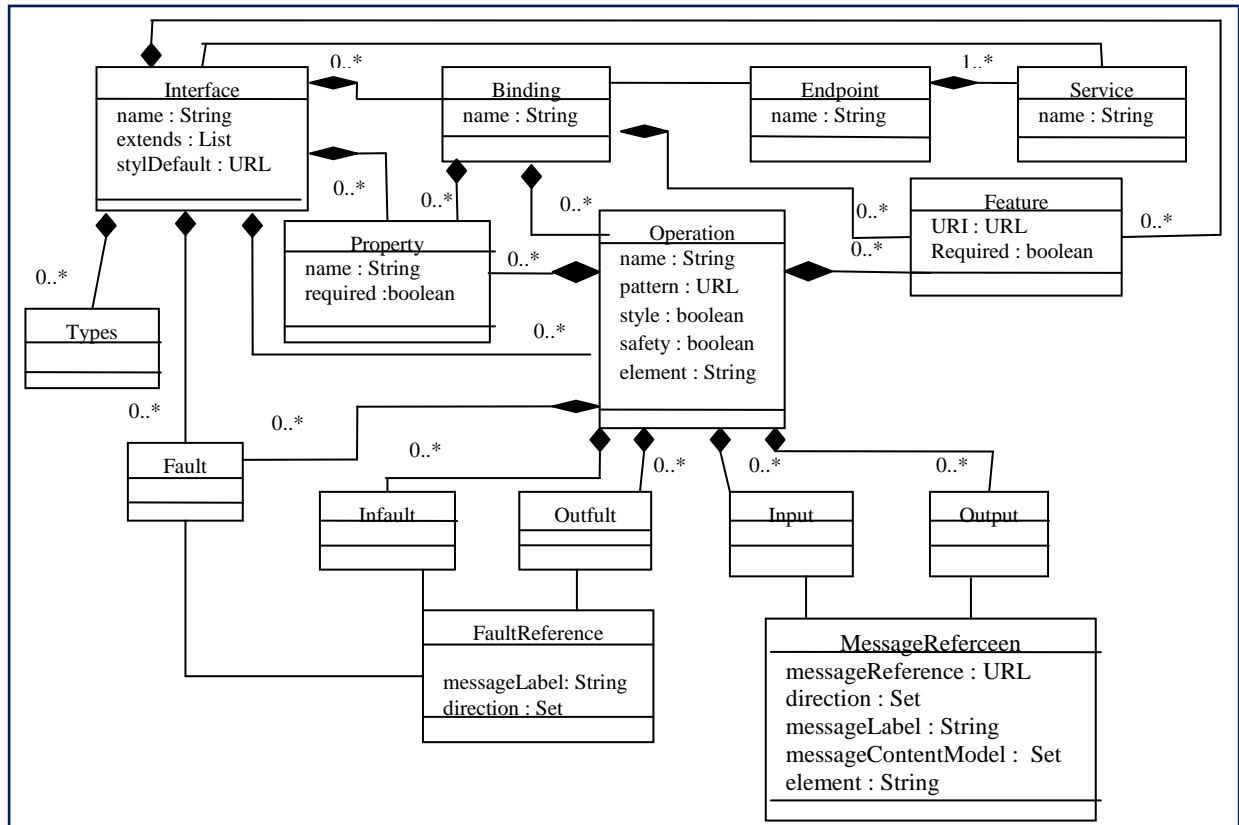


Figure I.9. Méta-modèle pour WSDL 2.0 [42].

Un message SOAP est une enveloppe qui est constituée de deux parties (figure I.10.) :

- Un en-tête qui contient des éléments enfants, appelés entrées, qui permettent d'ajouter des extensions à un message. Ces extensions permettront, par exemple, de prendre en charge les transactions et la sécurité. SOAP se voulant un protocole léger et simple n'a, en effet, volontairement pas pris en compte la sécurité.
- Un corps qui contient la charge utile du message, c'est à dire les informations spécifiques à l'application. Par exemple, la spécification décrit clairement la manière dont les appels de procédure à distance (RPC Remote Procedure Call) devraient être représentés dans les messages SOAP pour invoquer des fonctionnalités de services Web. Les messages SOAP peuvent aussi servir à transmettre des documents XML arbitraires tel qu'un ordre d'achat ou une feuille de style XML.

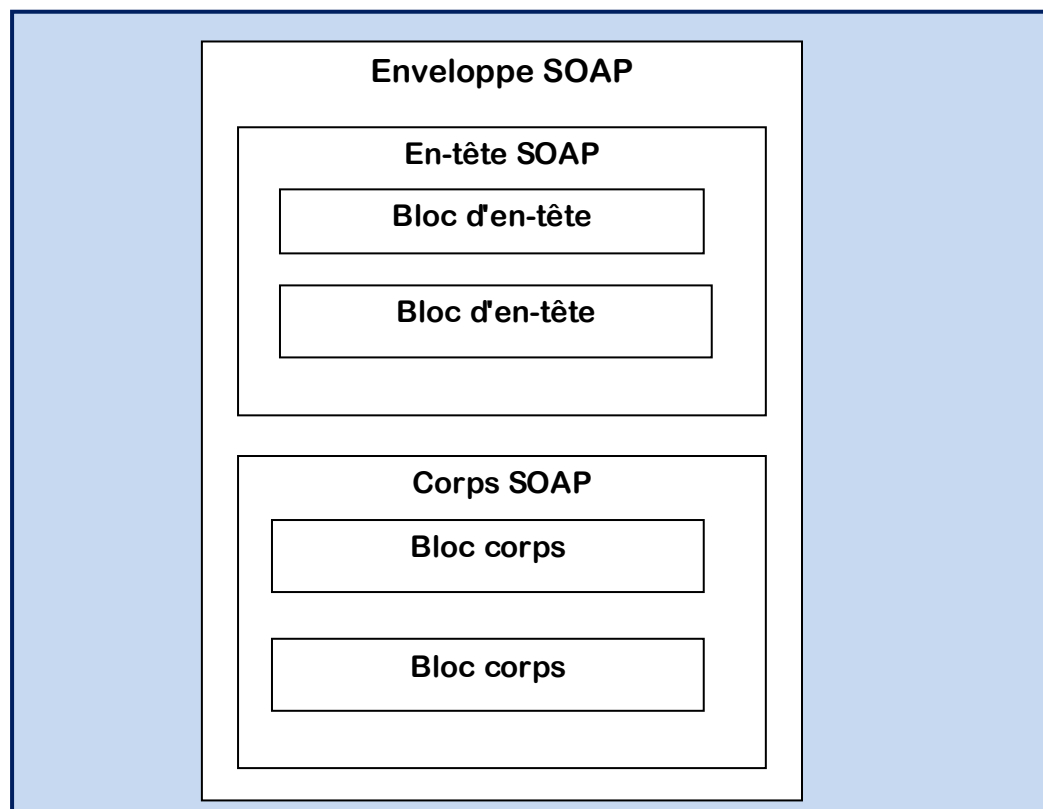


Figure I.10. Différents éléments du message SOAP

### UDDI:

UDDI (Universal Description, Discovery and Integration) est une norme basée sur XML qui définit des formats permettant de créer des catalogues (page blanches, pages jaunes et pages vertes) pour publier et rechercher des services Web (voir tableau I.1) [27].

Cette norme est issue d'une initiative d'IBM et Microsoft rejoints par d'autres éditeurs tels que SAP, Oracle, Commerce One... [27].

Les versions publiées de cette norme sont UDDI 1.0 (septembre 2000), UDDI 2.0 (juin 2001), et récemment UDDI 3.0. [108].

Information	Utilité
<b>Pages blanches :</b> Informations publiques de type nom, adresse, téléphone et autres coordonnées.	Permet au fournisseur du service Web de se référencer lui-même.
<b>Pages jaunes :</b> Information sur la classification de l'entreprise au moyen de taxinomies standards.	Permet de chercher et trouver un service Web particulier.
<b>Pages vertes :</b> Description technique des différents services Web offerts par l'entreprise	Permet de comprendre comment une application peut se connecter et interagir avec le Web service trouvé.

Tableau I. 1. Les différents types d'utilisation d'un UDDI.

### I.5.3.4. Composition des services Web

La composition consiste à combiner les fonctionnalités de plusieurs services Web au sein d'un même processus métier (ou business process) dans le but de répondre à des demandes complexes qu'un seul service ne pourrait pas satisfaire [33] [34]. Ceci nous permet d'obtenir un ou des services Web dits 'complexes' modélisant un processus métier dans sa globalité plutôt qu'une de ses opérations [35]. Les langages de description de ces services Web complexes font apparaître plusieurs méthodes d'exécutions : orchestration (centralisée) ou chorégraphie.

**Orchestration** : un processus principal (service Web) prend le contrôle du déroulement de la composition et coordonne donc les différentes opérations des différents services Web [35]. Aucun service n'a connaissance de cette composition. L'orchestration est la méthode la plus utilisée en raison de sa simplicité.

**Chorégraphie** : Chaque service a une connaissance plus ou moins globale du processus métier dans lequel il se trouve. Chacun des services intervenant dans la composition sait exactement ce qu'il doit faire, quand il doit le faire et avec qui [35].

Plusieurs langages ont été développés pour répondre aux besoins de la composition : WSFL (Web Service Flow Language) [36], BPM (Business Process Modeling Language) [37], XLANG [38], et le récent langage est BPEL4WS (Business Process Enterprise Language for Web Services) [39] qui constitue une fusion de XLANG et WSFL et présente plusieurs avantages, il est basé sur XML, et supporte les protocoles standards des services Web.

### I.5.3.5. Bus de service d'entreprise (ESB).

L'ESB (Entreprise Service Bus) ou bus de services d'entreprise [28] est un concept qui a été défini en 2003 par le Gartner group. Il est considéré comme la convergence des outils EAI (Entreprise Application Integration) [105] et des services Web mais en évitant l'aspect monolithique des EAI. L'ESB est une solution d'intégration distribuée basés sur un Message-Oriented- Middleware (MOM) et fournissant des services de transformation des données, de routage et ce en s'appuyant sur l'utilisation systématique des standards des services Web (SOAP, WSDL, UDDI) et les normes WS-\*

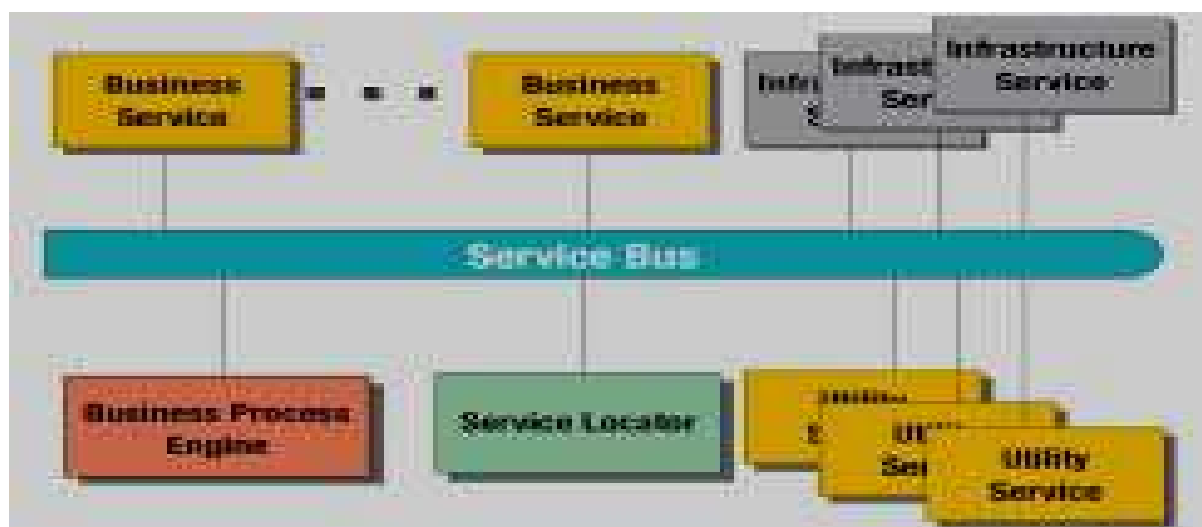


Figure I.11. Architecture d'un ESB [29].

Comme illustré dans la figure I.11, l'ESB est centré sur la notion de bus qui permet aux différents services métiers (encapsulant des applications d'entreprise) de communiquer et de coopérer. Le bus se base sur certains services de base comme [29] :

- le moteur d'orchestration (Business process engine) qui joue le rôle de service d'orchestration permet d'exécuter des processus basés sur BPEL4WS;
- le service de localisation (Service locator) qui permet de trouver de façon transparente les services;
- les services utilitaires (Utility services) qui sont des services techniques qui sont sollicités par les services métiers (ex: services permettant le routage, la transformation, le support des transactions, etc....);
- les services d'infrastructure (Infrastructure services) qui sont des services qui permettent de fournir un support système et d'infrastructure aux services métiers (ex: services liés à la sécurité et au monitoring, etc....).

### I.5.4. Architecture guidée par les modèles

Le Model Driven Architecture (MDA) proposé par l'OMG (Object Management Group) est une approche qui propose un cadre méthodologique et Architectural de développement et d'intégration de systèmes qui assure la pérennisation des architectures métier de l'entreprise en les découplant des préoccupations technologiques.

#### I.5.4.1 Définition de MDA

MDA est une approche fondée sur la réalisation de modèles (représentations) de différents niveaux d'abstraction et offrant des règles de passage d'un niveau à l'autre. Il s'agit de modéliser pour réutiliser, et composer des modèles entre eux, du niveau métier jusqu'au niveau technique, en misant sur L'interopérabilité des modèles, sur leur indépendance entre les différents niveaux d'abstraction, et enfin, en apportant des solutions implémentées [27].

#### I.5.4.2. Modèles et niveau méta

Le MDA permet de séparer les spécifications fonctionnelles d'un système des spécifications de son implémentation sur une plate-forme donnée. A cette fin, elle définit une architecture de spécifications structurée en modèles du domaine (CIM : Computation Independent Model), en modèle indépendant des plates-formes (PIM : Platform Independent Model) et en modèles spécifiques (PSM : Platform Specific Model) [5] :

**Modèle CIM :** Le modèle CIM est un modèle conceptuel du domaine, dont le vocabulaire utilisé est celui du praticien du domaine. Il fait le pont entre les experts du domaine et les experts de la conception et construction des artefacts qui satisfont les besoins du domaine. Ce modèle se définit suivant les exigences du client et représente l'application dans son environnement. Il ne contient pas d'informations sur la réalisation de l'application ni sur les traitements.

**Modèle PIM :** Le PIM de base représente uniquement les capacités fonctionnelles métiers et le comportement du système, sans "dégradations" par des considérations technologiques. La clarté de ce modèle doit permettre à des experts du domaine de comprendre bien mieux qu'un modèle d'implémentation.

**Modèle PSM :** Une fois le PIM suffisamment détaillé, il est projeté vers un modèle spécifique (PSM). Pour obtenir un modèle spécifique, il faut choisir la ou les plates-formes d'exécution.

Les caractéristiques d'exécution et les informations de configuration qui ont été définies de façon générique sont converties pour tenir compte des spécificités de la plate-forme.

L'approche MDA est basée sur quatre niveaux de modèles [30] (voir figure I.12.).

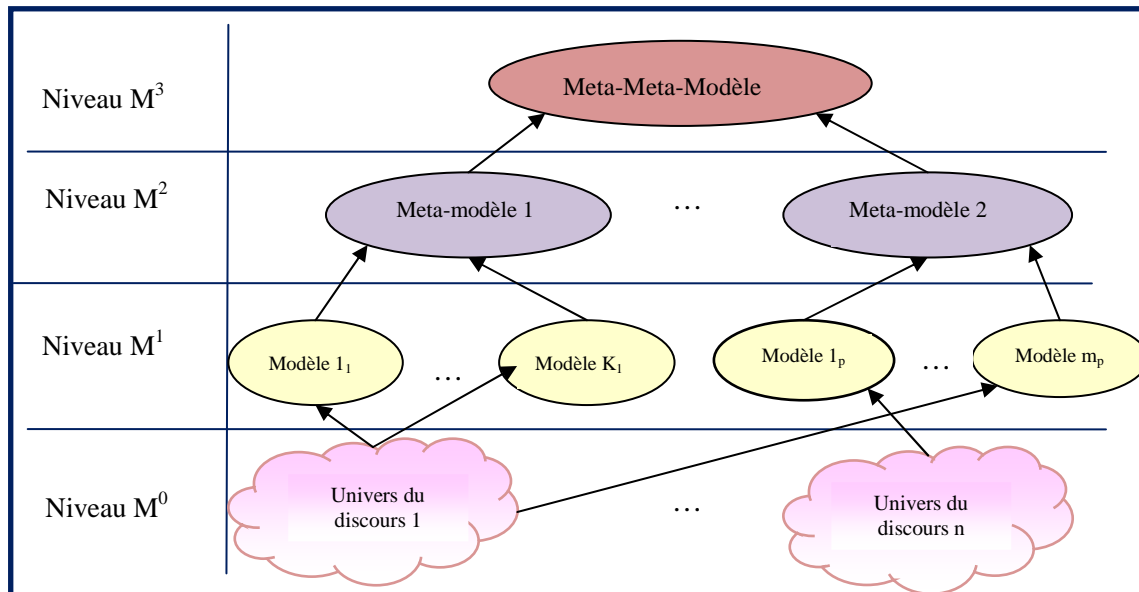


Figure I.12. Approche MDA et les quatre niveaux de modèles [30].

Le niveau le plus bas M0 représente les différents sujets de modélisation aussi appelé « univers du discours ». Le niveau M1 contient les différents modèles de chaque univers de discours. Les modèles du niveau M1 appartiennent à différents domaines d'intérêt relatifs aux univers du discours représentés par les modèles. Le niveau M2 représente les méta-modèles spécifiques à chaque domaine : un méta-modèle pour chaque domaine d'intérêt pertinent pour les modèles de niveau M1. Finalement, le niveau M3 présente le méta-méta-modèle et il doit être conçu pour permettre la définition de tous les concepts nécessaires pour la modélisation de méta-modèles et pour leur unification sous un cadre commun. Un méta-méta-modèle est indépendant du domaine, il contient des méta-caractéristiques pour des méta-modèles spécifiques.

Pour assurer l'interopérabilité des applications, des transformations entre leurs modèles sont définies. Lemesle [31] introduit la transformation de modèles en établissant des règles de transformation entre méta-modèles (figure I.13.).

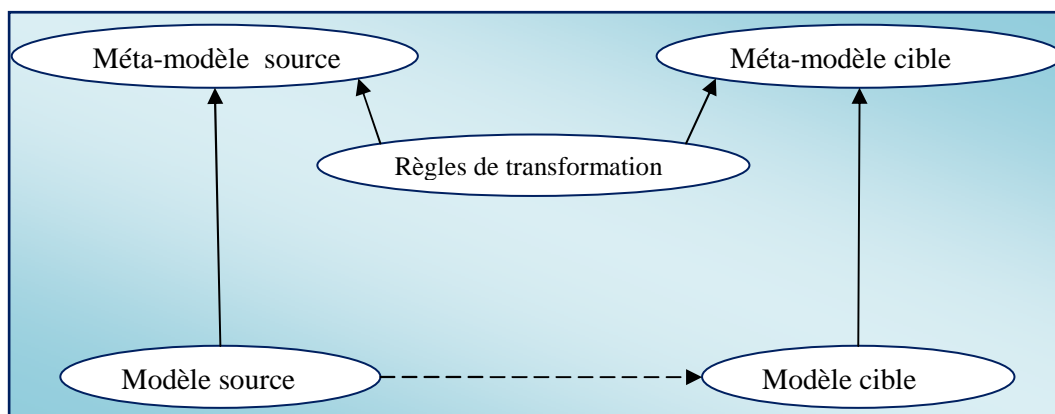


Figure I.13. Transformation de méta-modèles [31].

On appelle donc mapping, l'ensemble des règles de transformation de modèles permettant de traduire les instances d'un méta-modèle source en des instances d'un méta-modèle cible [32].

MDA s'appuie sur de nombreux standards comme [27]:

- Meta Object Facilities (MOF) : spécification de l'OMG (Object Management Group) permettant d'établir « un canevas générique pour représenter les méta-informations dans un contexte d'application ».
- XML Meta data Interchange (XMI) : définit comment les balises sont utilisées pour représenter les modèles compatibles MOF sérialisés en XML. XMI permet d'échanger des méta- données UML entre outils de modélisation et le repository de méta-données.
- UML: est un langage de modélisation [109].
- OCL: langage de description formelle des contraintes [55].

#### I.5.4.3. Avantages des architectures guidées par les modèles

Les architectures guidées par les modèles présentent plusieurs avantages comme [106] :

- ✓ La pérennité des savoir-faire, afin de permettre aux entreprises de capitaliser sur leur métier sans avoir à se soucier de la technique.
- ✓ Les gains de productivité, afin de permettre aux entreprises de réduire les coûts de mise en œuvre des applications informatiques nécessaires à leur métier.
- ✓ La prise en compte des plates-formes d'exécution, afin de permettre aux entreprises de bénéficier des avantages des plates-formes sans souffrir d'effets secondaires.

## I.6. Synthèse de quelques travaux relatifs à l'interopérabilité technique

Nous trouvons dans la littérature une multitude de travaux qui traite de l'interopérabilité technique. Dans cette section nous présentons quelques uns, qui se rapportent à notre travail.

Le protocole **PRESTO**<sup>8</sup>, a pour objectif de définir une couche générique d'échange de messages pour les échanges de l'administration électronique [124] [125]. Ce protocole possède la spécification d'un profil service Web (ensemble de spécifications de services Web sélectionnées par le WS-I). Ce protocole est implémenté dans un premier temps par des mandataires PRESTO. La distinction entre un mandataire émetteur PRESTO et un mandataire récepteur PRESTO est purement logique et a pour but de refléter leurs rôles respectifs dans échange de message. Le premier est un service qui génère et transmet un message via PRESTO au second. Par conséquent, un mandataire récepteur PRESTO est un service qui reçoit et consomme des messages par le protocole. Ainsi les mandataires implémentent et exposent des points d'accès aux partenaires de l'administration électronique et leur permet d'inter-opérer avec tous les services qui "*parlent*" PRESTO [124].

Les spécifications de ce protocole imposent des restrictions à l'utilisation des services Web afin d'en garantir l'interopérabilité.

La plate forme *Mélusine* [126] permet de concevoir et d'exécuter des systèmes de coordination de services hétérogènes via un workflow<sup>9</sup>. C'est une architecture composée de quatre couches principales : La couche services hétérogènes à interopérer.

La couche de médiateur permettant de faire le lien entre les activités du workflow et leurs implémentations en termes de services réalisant ces activités,

La couche de coordination définit l'enchaînement des services participants, elle se divise en une couche de workflow, et une couche d'adaptateurs chargés de faire le lien entre les activités du workflow et les services chargés de l'exécution.

Et enfin, la couche métier permettant de relever le niveau d'abstraction d'analyser les besoins et de modéliser le domaine métier de l'entreprise selon l'approche MDA.

Ces travaux présentent un point commun en termes d'interopérabilité des plates-formes, et l'utilisation des services, mais ils ne présentent aucune solution quant à l'interopérabilité sémantique.

## I.7. Discussion

Nous avons présenté des solutions et quelques travaux pour dresser l'interopérabilité technique et syntaxique des systèmes d'information. Vu les avantages offerts par chacune de ces solutions, nous sommes convenus à l'utilisation d'une combinaison des techniques de SOA, services Web et ingénierie des modèles (MDA). Ceci nous permet de profiter des points forts de chaque paradigme.

Ces techniques favorisent la flexibilité, la résolution du problème d'hétérogénéités des plates-formes tout en encapsulant la complexité des systèmes hétérogènes. Mais, elles présentent un manque en ce qui concerne la présentation de l'aspect sémantique. Ceci freine l'interopérabilité des systèmes d'information en raison de multiples conflits qui sont liés aux problèmes de compréhension.

## I.8. Conclusion



Ce chapitre a présenté de façon générale la notion de système d'information et d'applications d'entreprise, où nous nous sommes focalisés sur la dimension d'hétérogénéité qui constitue un problème pour l'interopérabilité des systèmes d'information en inter et intra-entreprise. Ce chapitre a aussi présenté la notion de coopération et d'interopérabilité qui est définie à plusieurs niveaux : technique (et syntaxique), sémantique et organisationnel. Nous nous intéressons aux deux premiers niveaux, pour cela nous avons décrit quelques outils permettant l'interopérabilité technique, et nous avons retenue une combinaison des paradigmes d'architecture orientée services, d'architecture guidée par les modèles et les services Web ce qui pallie les problèmes d'hétérogénéité au niveau technique, de manque de flexibilité et de complexité.

La principale limite de ces solutions est la non prise en charge de l'aspect sémantique, ce qui conduit à l'échec les échanges de données et de fonctions entre les services Web.

Pour remédier à ce manque, des solutions basées sur la sémantique ont été mises en avant, elles feront l'objet du second chapitre.

---

9 : description abstraite de haut niveau de la coordination, elle est souvent graphique.



# Chapitre II

## Outils conceptuels et techniques pour la mise en œuvre des mécanismes d'interopérabilité sémantique

# Chapitre II

## Outils conceptuels et techniques pour la mise en œuvre des mécanismes d'interopérabilité sémantique

### II.1. Introduction

Dans le chapitre précédent nous avons opté pour une combinaison des techniques des architectures orientées services, de l'ingénierie des modèles et des standards des services Web, pour dresser l'interopérabilité technique et syntaxique des systèmes d'information. Nous avons signalé leur principale limite en ce qui concerne l'aspect sémantique. Cet aspect fera l'objet de ce chapitre qui sera consacré à la notion de sémantique, sa typologie et sa classification, ainsi que les méthodes permettant sa représentation, notamment les ontologies que nous allons voir de près et, qui constituent le prolongement de l'ingénierie des modèles.

Nous présentons une classification des hétérogénéités sémantiques des services Web, les méthodes permettant leur description au niveau sémantique, et une classification des architectures de médiation nécessaire à la résolution des conflits entre services Web en interaction pour permettre l'interopérabilité sémantique des systèmes d'information.

Enfin, nous clôturons ce chapitre par une classification des approches d'interopérabilité et d'intégration des services Web et une étude des travaux existants.

### II.2. Notion de sémantique

#### II.2.1. Définition

La sémantique est l'étude du sens des mots (LAROUSSE), elle constitue une branche de la linguistique. En informatique, Woods [43] la définit « comme une forme formelle de représentation des connaissances humaines ».

Dans le domaine des systèmes d'information [1], la sémantique se réfère plus précisément au sens des différents éléments d'un système d'information qui peuvent être des données, des fonctions, voire des processus. La sémantique peut différer d'un système à un autre, en fonction du contexte d'utilisation. Ces hétérogénéités de type sémantique sont à l'origine des conflits entre applications ce qui contraint leur interopération.

#### II.2.2. Typologie des hétérogénéités sémantiques

Kavouras [44] propose une typologie d'hétérogénéités sémantiques qui prend en considération les termes (T1, T2) et leurs définitions (D1, D2) associés à des entités de la réalité. Le tableau II.1 illustre les notions suivantes :

*Equivalence* : deux entités sont équivalentes si elles ont le même terme et la même définition.

*Disjonction* : deux entités sont disjointes si elles ont des termes différents et des définitions différentes.

*Synonymie* : cas où deux entités sont représentées par deux termes différents et une même définition.

*Homonymie* : cas où deux entités sont représentées par le même terme et deux définitions différentes.

*Complétude* : cas où les deux entités sont représentées par le même terme, et la définition de l'une est incluse dans l'autre.

*Spécialisation* : cas où les deux entités sont représentées par deux termes différents, et la définition de l'une est incluse dans l'autre.

*Chevauchement* : Cas où les définitions des deux entités se recouvrent.

	$T_1 = T_2$	$T_1 \neq T_2$
$D_1 = D_2$	<b>Equivalence</b>	<b>Synonymie</b>
$D_1 > D_2$	<b>Complétude</b>	<b>Spécialisation</b>
$D_1 \sim D_2$	<b>Chevauchement</b>	<b>Chevauchement</b>
$D_1 \neq D_2$	<b>Homonymie</b>	<b>Disjonction</b>

Légende

	Pas de conflit
	Faible conflit
	Moyen conflit
	Grand conflit

Tableau II. 1. Typologie des conflits sémantiques [44].

Pour réussir l'interopérabilité des systèmes d'information, une représentation de la sémantique est impérative, avec l'appel à des mécanismes de médiation pour résoudre les conflits inhérents des différentes hétérogénéités sémantiques.

### II.2.3. Classification de la sémantique

Uschold et Gruninger [45] proposent une classification de la sémantique qui permet de distinguer les différents types de sémantique qui peuvent exister dans le contexte du Web comme suit :

- ✓ *La sémantique implicite* : qui existe seulement dans le mental des gens.
- ✓ *La sémantique semi-informelle (explicite et informelle)* : qui est une sémantique explicite mais qui est souvent représentée de façon informelle en utilisant en général des langages naturels tels que le français ou l'anglais.
- ✓ *La sémantique semi-formelle* : qui désigne une sémantique explicite et relativement formelle qui est destinée principalement aux humains en utilisant des formalismes, le plus souvent graphiques tels que les modèles sémantiques, ou les diagrammes UML.
- ✓ *La sémantique formelle* : qui est une sémantique qui se base sur des formalismes mathématiques rigoureux (tels que la logique de description, la logique de premier ordre, etc.) qui lui permettent d'être traitée de façon automatique.

### II.2.4. Représentation de la sémantique

Pour assurer l'interopérabilité des systèmes d'information au niveau sémantique, il est nécessaire de représenter cette dernière à l'aide des techniques de représentation de la connaissance. La représentation des connaissances étudie comment transformer l'expression

du sens en une représentation formelle manipulable par une machine [1]. Pour modéliser la sémantique nous pouvons utiliser l'une des deux approches à savoir :

- L'approche procédurale qui utilise des procédures ou des règles pour représenter la sémantique.
- L'approche déclarative qui se base sur la modélisation des faits.

Cette dernière est meilleure que la première du fait qu'elle présente des avantages tels que standardisation, capture, réutilisation, inférence et flexibilité [45].

L'un des outils les plus utilisés dans le cadre de la représentation sémantique est la notion d'ontologies que nous détaillons dans ce qui suit.

## II.3. Apport des ontologies

### II.3.1. Définitions

Le terme ontologie est issu du domaine de la philosophie et signifie " science de l'être" (Onto : être et logia : science). En informatique les ontologies font partie de l'ingénierie des connaissances (du domaine de l'IA). Il existe plusieurs définitions de la notion d'ontologie, nous retenons quelques unes :

- \* « Une ontologie est une spécification explicite d'une conceptualisation », Gruber [46].
- \* « Les ontologies se définissent comme une spécification formelle d'une conceptualisation partagée », Borst [47].
- \* « Une ontologie est une théorie logique qui permet une spécification explicite et partielle d'une conceptualisation », Guarino [48].
- \* « Une ontologie est un ensemble de termes structurés de façon hiérarchique, conçu afin de décrire un domaine et qui peut servir de charpente à une base de connaissances », Swartout [49].
- \* « Une ontologie fournit les moyens de décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances », Gomez-Perez [50].
- \* « Une ontologie est un modèle (au sens large, composé d'objets et de relations) réutilisable et partageable d'un domaine particulier spécifié pour créer un langage commun (éventuellement basé sur des standards basés sur XML) afin de faciliter l'échange d'informations et le partage de fonctionnalités entre les personnes et les systèmes d'information », Izza [1].

### II.3.2. Rôle des ontologies

Les ontologies peuvent jouer divers rôles qui sont [51]:

- ✓ acquisition et représentation des connaissances ;
- ✓ recherche et extraction des connaissances : inférer la connaissance qui est pertinente face à la requête de l'utilisateur ;

- ✓ partage et intégration des connaissances : intégration des différentes sources d'information ;
- ✓ gestion des connaissances ;
- ✓ simplification du dialogue homme-machine.

### II.3.3. Structuration des ontologies

Une ontologie est structurée de plusieurs composants qui sont [46] [50] :

*Les concepts* (termes ou classes) de l'ontologie, constituent les objets de base manipulés par les ontologies. Ils correspondent aux abstractions pertinentes du domaine du problème, retenues en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie ; par exemple, la description d'un ensemble d'objets, d'une tâche, d'une fonction, d'une stratégie, d'un processus de raisonnement, etc.

*Les relations* traduisent les interactions existant entre les concepts. Ces relations sont formellement définies comme tout sous-ensemble d'un produit cartésien de  $n$  ensembles, c'est-à-dire  $R : C_1 \times C_2 \times \dots \times C_n$  et incluent 1) la relation de spécialisation (subsumption), 2) la relation de composition (méronymie), 3) la relation d'instanciation, etc. Ces relations nous permettent de capturer, la structuration ainsi que l'interaction entre les concepts, ce qui permet de représenter une grande partie de la sémantique de l'ontologie.

*Les fonctions* sont des cas particuliers de relations dans les quelles le nième élément (extrant) de la relation est défini de manière unique à partir des  $n-1$  éléments précédents (intrants). Formellement, les fonctions sont définies ainsi :  $F : C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$ . Comme exemple de fonctions binaires, nous pouvons citer la fonction *mère de*.

*Les axiomes* permettent de modéliser des assertions toujours vraies, à propos des abstractions du domaine traduites par l'ontologie. Ils permettent de combiner des concepts, des relations et des fonctions pour définir des règles d'inférences et qui peuvent intervenir, par exemple, dans la déduction, la définition des concepts et des relations, ou alors pour restreindre les valeurs des propriétés ou les arguments d'une relation.

*Les instances ou individus* constituent la définition extensionnelle de l'ontologie. Ils représentent des éléments singuliers véhiculant les connaissances (statiques, factuelles) à propos du problème.

### II.3.4. Classification des ontologies

Il existe deux classifications d'ontologies, selon deux dimensions : le niveau de granularité et le niveau de formalité [45] [48].

#### II.3.4.1. Classification selon le niveau de granularité

Uschold [45] et Guarino [48] classent les ontologies selon leur granularité en quatre types qui sont (voir figure II.1.):

*Les ontologies supérieures (Upper or Top-level Ontologies)* [45] : ont pour objet l'étude des catégories de choses qui existent dans le monde, comme les concepts de haute abstraction tels que : les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés, etc. et qui sont indépendants d'un domaine particulier. Ces ontologies intègrent en sus les fondements philosophiques comme étant des principes à suivre pour concevoir l'ontologie de haut niveau, et dont le but ultime est de standardiser la conception des ontologies.

*Les ontologies de domaine (Domain ontologies) [52] [53] :* Ce sont des ontologies qui sont construites sur un domaine particulier de la connaissance. Elles fournissent le vocabulaire des concepts, les théories et les principes de base du domaine de connaissance. Ces ontologies sont des méta-descriptions d'une représentation de connaissances du domaine. Nous citons comme exemples d'ontologies de domaine ENGMATH pour les mathématiques, et TOVE dans le domaine de la gestion des entreprises [54].

*Les ontologies de tâches (Task ontologies) [52] :* Ces ontologies sont utilisées pour gérer des tâches spécifiques liées à la résolution de problèmes dans les systèmes, tels que les tâches de diagnostic, de planification, de configuration, etc.

*Les ontologies d'application [53] :* ce sont les ontologies les plus spécifiques. Elles permettent de décrire des concepts dépendants à la fois d'un domaine et d'une tâche. Ces ontologies décrivent la sémantique des informations et des services manipulés par une ou un ensemble d'applications sur un même domaine.

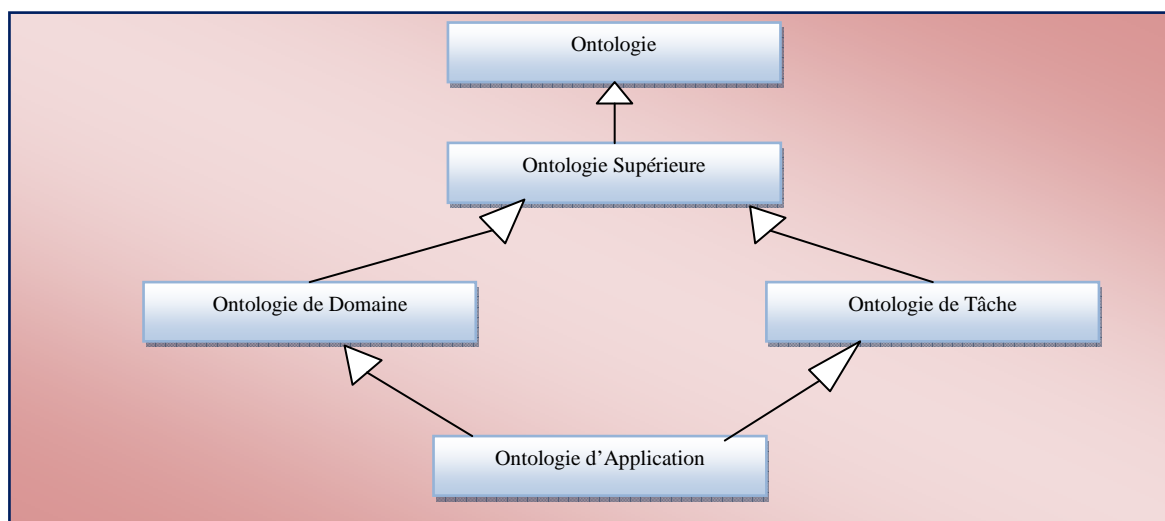


Figure II.1. Différents types d'ontologies selon le niveau de granularité.

#### II.3.4.2. Classification selon le niveau de formalité

Cette classification correspond aux types de sémantique vus dans la section II.2.3, et elle est en fonction du niveau de formalité du langage utilisé pour la représentation de l'ontologie. Cette classification permet de mettre en évidence quatre catégories principales qui vont des ontologies hautement informelles jusqu'aux ontologies rigoureusement formelles et qui sont [45] [48] :

*Ontologies informelles :* qui sont des ontologies opérationnalisées dans un langage naturel, ce qui correspond à ce qui est communément appelé la sémantique ouverte.

*Ontologies semi-informelles :* ce sont des ontologies qui sont décrites à l'aide d'un langage naturel structuré et limité.

*Ontologie semi-formelles :* ce sont des ontologies qui utilisent un langage artificiel défini formellement.

*Ontologies formelles :* ce sont des ontologies qui se basent sur l'utilisation d'un langage artificiel contenant une sémantique formelle telles que les logiques de description.

#### II.3.5. Méthodologies de construction d'ontologies

Nous trouvons dans la littérature plusieurs méthodologies de construction d'ontologies. Par exemple, [56] [57] [58] présentent une étude comparative d'une multitude de méthodologies,

tels que les méthodologies d'Uschold et King [45], de Gruninger et Fox, et Methontology [57]. Cette étude révèle que Methontology est la méthode la plus mûre, d'ailleurs elle représente une recommandation de FIPA (Foundation for Intelligent Physical Agent) en matière de construction d'ontologies. Nous présentons la méthodologie 'Methontology' et la méthodologie d'Uschold et King qui est inspirée de la construction d'ontologies pour la modélisation d'entreprises.

- Méthodologie d'Uschold et King [45] (figure II.2.): Les auteurs proposent quatre étapes qui sont 1) Identification du but et de la portée de l'ontologie, 2) Sa construction, 3) Son évaluation, 4) et sa documentation. Pour la construction de l'ontologie et l'identification des concepts, les auteurs proposent trois stratégies : une approche descendante, dans laquelle les concepts les plus abstraits sont définis et qui sont ensuite spécialisés ; une approche ascendante, dans laquelle les concepts spécifiques sont identifiés et qui sont ensuite généralisés pour définir d'autres concepts plus génériques ; et une approche mixte dans laquelle les concepts les plus importants sont identifiés et auxquels on applique les techniques de spécialisation et/ou de généralisation pour déterminer les autres concepts.

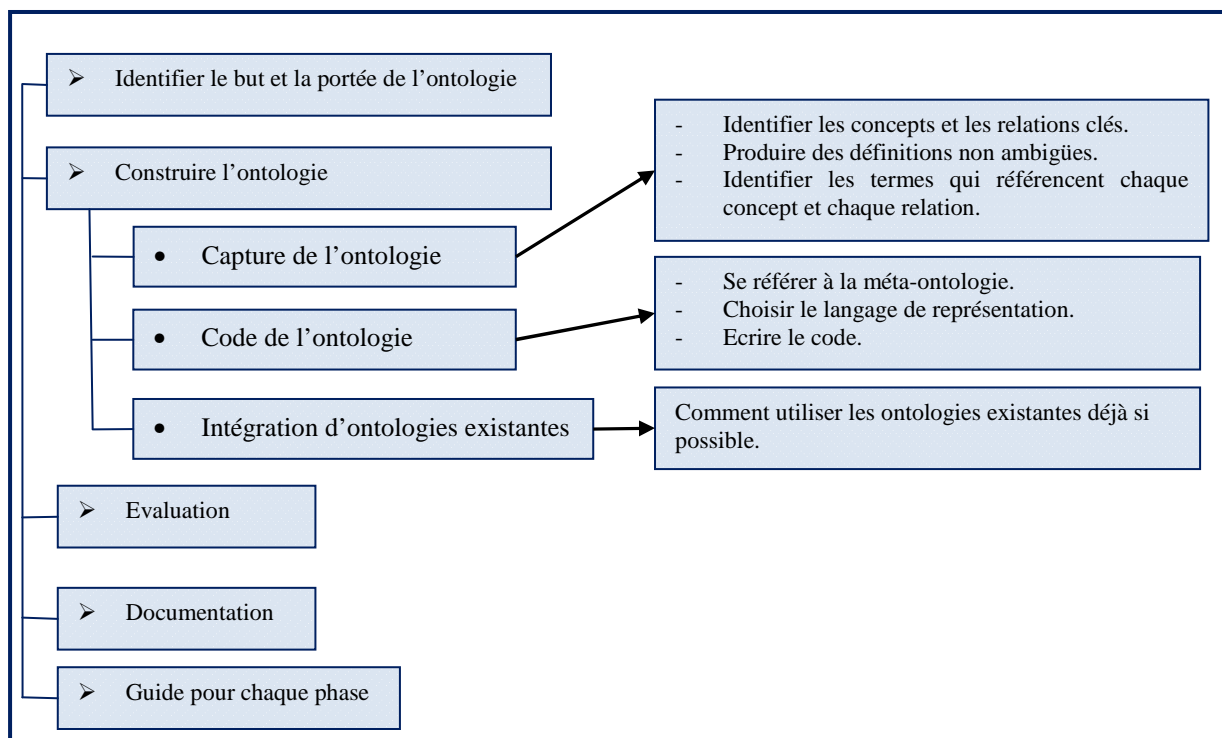


Figure II.2. Méthode d'Uschold et King [57].

- Méthodologie 'Methontology' : proposée en 1998 par l'équipe du LAI de l'université Polytechnique de Madrid (figure II.3.), permet de couvrir tout le cycle de vie d'une ontologie à savoir, spécification, conceptualisation, formalisation, implémentation, et maintenance. Elle s'intéresse pratiquement à toutes les activités liées aux ontologies (activités de développement, de gestion de projet et de support). Cette méthodologie est soutenue par l'outil WebODE et ODE [56].

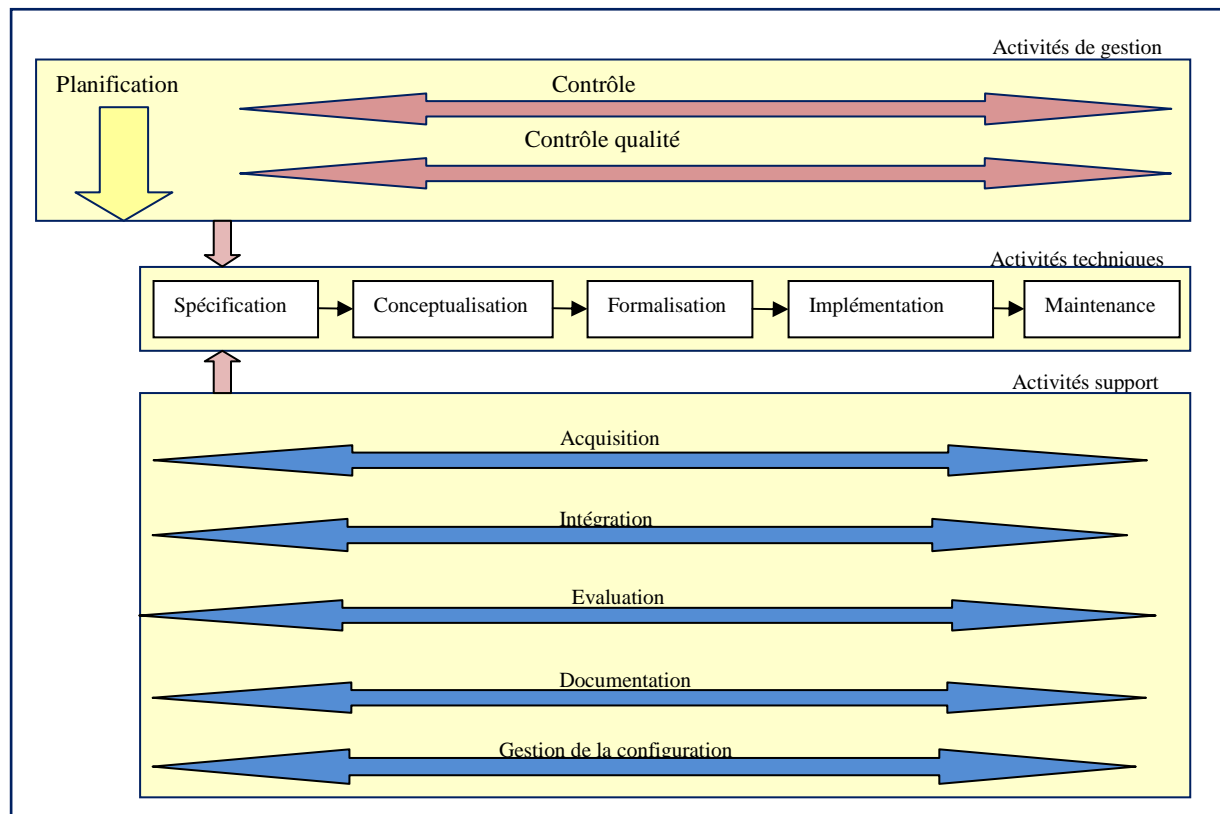


Figure II.3. Composantes de Methontology [57].

En ce qui concerne les ontologies de domaine une méthode connue consiste à transformer le modèle de données en une ontologie :

Un modèle de classe UML peut être transformé en une ontologie pour le Web sémantique si l'on se limite à un sous ensemble d'UML qui est traduisible [59].

Plusieurs auteurs ont expliqué comment développer des ontologies de domaine à partir du modèle conceptuel [51] [121].

Notons que toutes les méthodologies ne sont pas encore mûres [56] [57].

### II.3.6. Langages de représentation et outils de développement des ontologies

Vu l'importance des ontologies et leur prolifération dans de nombreux domaines, on assiste depuis les années 90 et jusqu'à nos jours à la naissance d'une panoplie de langages de descriptions d'ontologies et parallèlement de nombreux outils pour leur construction. Nous allons présenter dans ce qui suit quelques langages et outils. Pour une ample description nous trouvons dans la littérature abondante des listes exhaustives ainsi que des comparaisons de ces langages et outils [56] [50] [21] [57].

#### II.3.6.1. Langages de représentation des ontologies

Les langages de représentation des ontologies basés Web sont résumés par la figure II.4. [56]

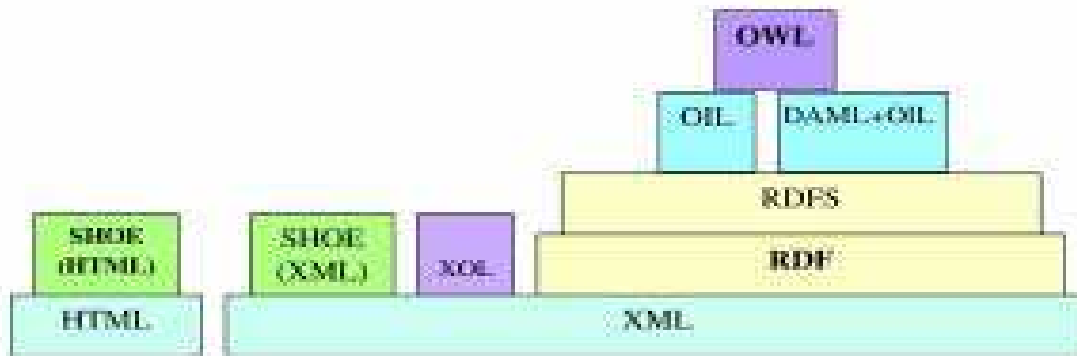


Figure II.4. Pile des langages de représentation des ontologies [56].

### **SHOE :**

Le langage SHOE (Simple HTML Ontology Extensions) a été créé en 1996 comme une extension de HTML à l'université de Maryland. Il permet d'insérer des ontologies dans des documents HTML grâce à de nouvelles balises. Il permet de représenter des concepts, des taxonomies, des relations et aussi des règles qui, permettent d'inférer de nouvelles connaissances. En 1998, et avec l'avènement du XML adopté comme un standard pour les échanges d'informations sur le Web par le W3C, SHOE a été modifié pour supporter des documents décrits en XML [1].

### **XOL :**

Le XOL (XML Ontology Language) a été développé par le centre AI de SRI International en 1999. Il permet de spécifier des concepts, des taxonomies et des relations binaires. Ce langage ne permet pas d'effectuer des raisonnements

### **RDF et RDFS :**

RDF (Ressource Description Framework) a été développé par W3C comme un langage basé sur les réseaux sémantiques pour décrire des ressources ou des relations entre ressources, en leur affectant des métadonnées.

Le modèle RDF définit trois types d'objets :

- des ressources : les ressources sont tous les objets décrits par RDF ;
- des propriétés : une propriété est un attribut, un aspect, une caractéristique qui s'applique à une ressource ;
- des valeurs : les valeurs en question sont celles particulières que prennent les propriétés.

Ces trois types d'objets peuvent être mis en relation par des assertions, c'est à dire des triplets (ressource, propriété, valeur), ou encore (sujet, prédicat, objet). Une description RDF est une suite d'assertions.

RDFS (RDF Schema) offre les moyens de définir un modèle (ou bien encore un schéma) de méta données qui permet de :

- donner du sens aux propriétés associées à une ressource ;
- formuler des contraintes sur les valeurs associées à une propriété afin de lui assurer une signification [61], [62].

### **DAML+OIL :**

Créer dans le cadre du projet DARPA le langage DAML-ONT (DARPA Agent Markup Language) intervient pour permettre aux agents de partager de la sémantique. DAML est associé à OIL (Ontology Inference Layer), qui est un autre langage de description d'ontologies. Le couple DAML+OIL repose sur RDF.

OIL est un langage de description et d'inférence sur les ontologies, basé sur RDF. Il prend appui sur les logiques de description. Il est composé de plusieurs couches :

- le coeur : il correspond presque exactement avec le langage RDF Schema, qui permet de décrire les vocabulaires RDF ;
- Standard OIL : il permet de définir la sémantique de façon plus précise, et donne ainsi la possibilité d'utiliser les mécanismes d'inférence ;
- Instance OIL : il introduit les fonctionnalités des bases de données.

Tout comme OIL ou RDF Schema, DAML est un langage de description d'ontologies. En ce sens, on peut définir des classes et des propriétés, et les mettre en relation.

Ainsi, DAML+OIL cherche à combiner toutes les caractéristiques de DAML, d'OIL, de RDF Schema, et de RDF. DAML+OIL permet de modéliser les aspects suivants :

- définition de classes de propriétés ;
- définition de classes de ressources ;
- relations logiques entre classes (disjonction, union, équivalence, etc.) ;
- relations d'héritage entre classes ;
- restriction de propriétés (cardinalité, etc.) et typage ;
- prise en charge des collections (listes) ;
- instanciation de classes de propriétés et de ressources [61].

### **OWL :**

OWL ( Ontology Web Language) est un vocabulaire XML basé sur RDF. Il constitue une recommandation du W3C. Il permet de spécifier ce qui peut être compris. Dans le cadre du Web, une ontologie est considérée comme une représentation des termes (information) d'un vocabulaire et ses relations aux autres termes, OWL est utile pour aider les machines à gérer ces informations plutôt que pour les rendre lisibles à l'œil humain. Un document OWL est composé de triplets RDF qui peuvent être écrits dans plusieurs syntaxes RDF. Il existe trois sortes d'OWL [63]:

- OWL Lite : c'est une version d'OWL aux fonctionnalités réduites, mais suffisant pour bien des usages, comme la constitution de taxonomies ou de thesaurus ;
- OWL DL : il correspond exactement aux logiques de description, et inclut OWL Lite ;
- OWL Full : il inclut OWL DL, et donne à l'utilisateur une expressivité maximale [61] [62].

OWL Lite (voir figure II.5.) réutilise les constructeurs du RDFS tels que Class, Property, subclassOf, subPropertyOf, domain, range, etc. Il définit d'autres constructeurs tels que : intersectionOf, equivalentClass, equivalentProperty, inverseOf, DataTypeProperty, ObjectProperty, allValuesFrom, someValuesFrom, Cardinality, MinCardinality, MaxCardinality, Ontology, Imports, etc. [63].

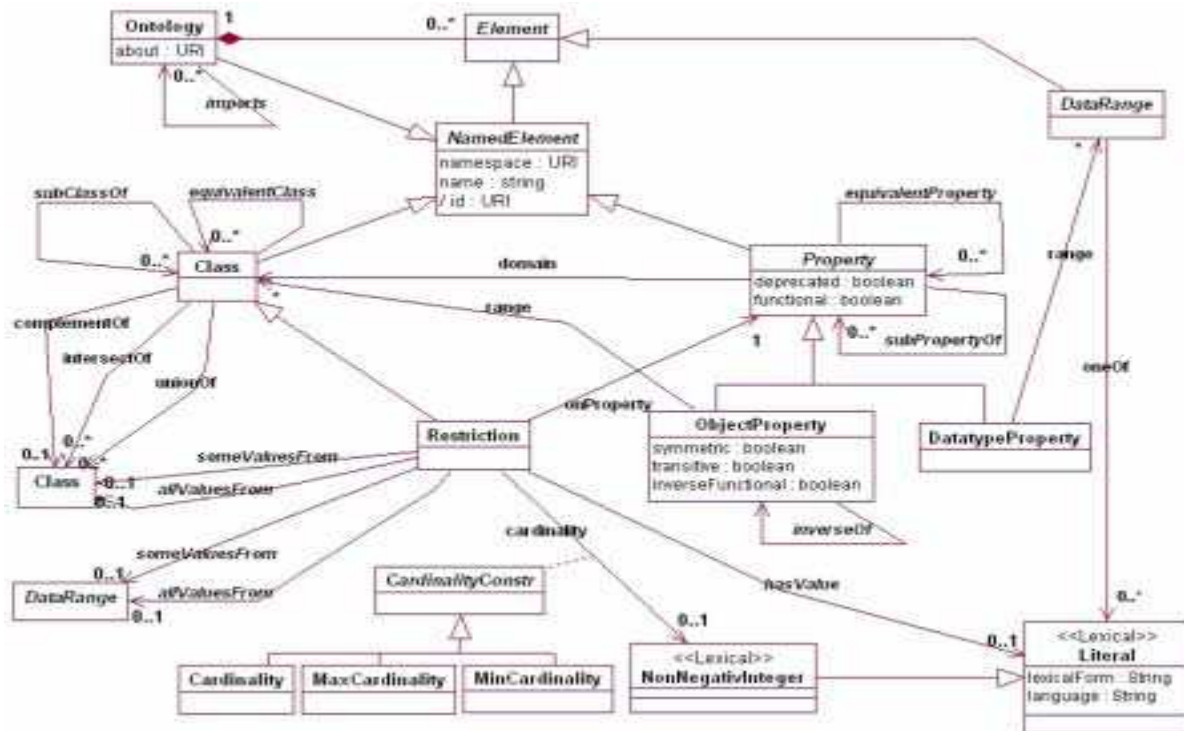


Figure II.5. Extrait du méta-modèle OWL [63].

### II.3.6.2. Outils de développement des ontologies

Comme nous l'avons déjà mentionné il existe une panoplie d'outils de construction d'ontologies. Mais dans le cadre de cette thèse nous présentons seulement quelques uns : Ontolingua, Ontosaurus, Tadzebao, WebOnto, et Protégé 2000 [50] [56] [57] [60].

#### **Ontolingua :**

Développé au début des années 90 par l'université Stanford, l'outil Ontolingua, un serveur d'édition d'ontologies, comprend un ensemble d'outils et de services qui supportent la construction en coopération d'ontologies, entre groupes séparés géographiquement.

#### **Ontosaurus :**

Développé au début des années 90 par l'Information Science Institute de l'université de South California. Comprend deux modules : un serveur d'ontologies et un navigateur d'ontologies.

#### **Tadzebao et WebOnto :**

Développés en 1997 par le Knowledge Media Institute d'Open University. Les deux outils sont complémentaires. Le premier permet aux ingénieurs de connaissances de tenir des discussions sur les ontologies, en mode synchrone et asynchrone, tandis que le second supporte la navigation collaborative, la création et l'édition d'ontologies sur le Web.

#### **Protégé 2000 :**

Développé au département d'Informatique Médicale de l'Université de Stanford, appartient aux outils dits de la seconde génération, Protégé-2000 est un éditeur (figure II.6.) qui permet de construire une ontologie pour un domaine donné, de définir des formulaires d'entrée de données, et d'acquérir des données à l'aide de ces formulaires sous forme d'instances de cette ontologie. Protégé est également une librairie Java qui peut être étendue pour créer de

véritables applications à bases de connaissances en utilisant un moteur d'inférence pour raisonner et déduire de nouveaux faits par application de règles d'inférence aux instances de l'ontologie et à l'ontologie elle-même (méta-raisonnement) [64]. Des « plugin » pour les langages RDF, DAML+OIL et OWL ont été développés pour Protégé. On peut intégrer à Protégé des outils exemple : un moteur d'inférence tel que JESS [65], ou des outils d'inférence spécifiques au Web sémantique basés sur des logiques de description [66] tel que RACER [67].

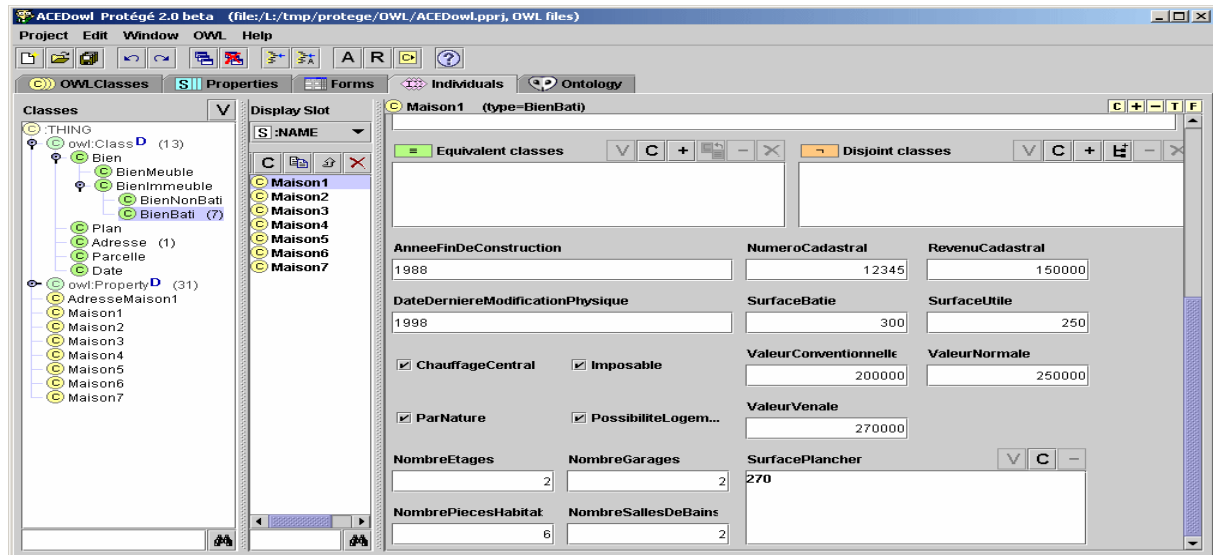


Figure II.6 . Editeur d'ontologies 2000 [64].

### II.3.7. Approches d'utilisation des ontologies

Il existe principalement trois approches d'utilisation des ontologies [68] qui sont : L'approche mono-ontologie, l'approche multi-ontologie et l'approche hybride.

#### II.3.7.1. L'approche mono-ontologie

L'approche mono-ontologie (Single ontology approach) utilise une ontologie globale fournissant un vocabulaire partagé pour la spécification de la sémantique. Toutes les sources d'informations sont reliées à une ontologie globale (figure II.7.). Le meilleur exemple d'implémentation de cette approche est le projet SIMS [107]. Les approches mono-ontologie sont les approches les plus simples à mettre en œuvre, mais dans le cas où les sources de données concernent des domaines hétérogènes, il devient difficile voire impossible à concrétiser un engagement ontologique. Aussi, la modification des sources peut affecter l'ontologie globale.

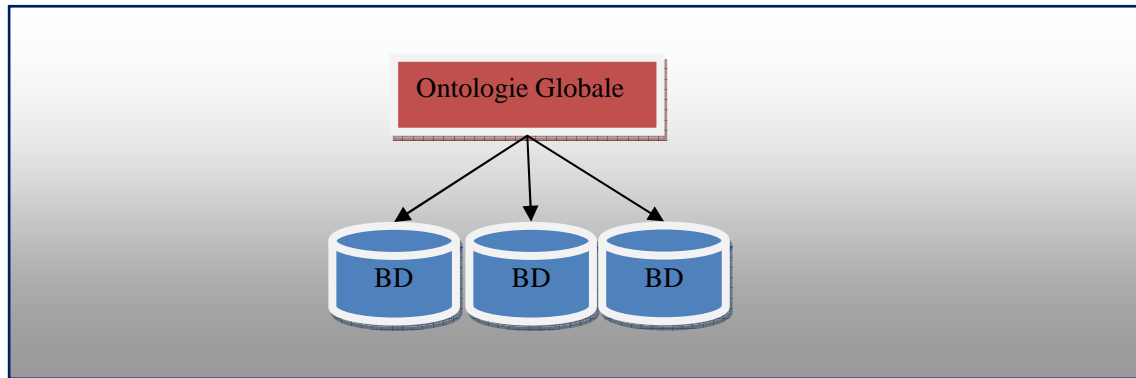


Figure II.7. Approche mono-ontologie [68].

### II.3.7.2. L'approche multi-ontologies :

L'approche multi-ontologies (multi-ontology approach) permet d'associer une ontologie locale à chaque Système d'Information (figure II.8.). Des mappings inter-ontologiques sont alors nécessaires afin d'établir une interprétation commune des données. Comme exemple d'implémentation, le projet OBSERVER [69] s'inscrit dans cette approche. Il associe à chaque source de données une ontologie locale. L'intérêt de cette approche est le fait que les sources peuvent être développées de façon indépendante, mais l'absence d'un vocabulaire commun conduit généralement à une difficulté extrême pour comparer différentes ontologies sources. Pour pallier cette difficulté, un formalisme de représentation additionnelle définissant le mapping inter-ontologie est utilisé. Ce dernier identifie sémantiquement les termes correspondants des différentes ontologies sources.

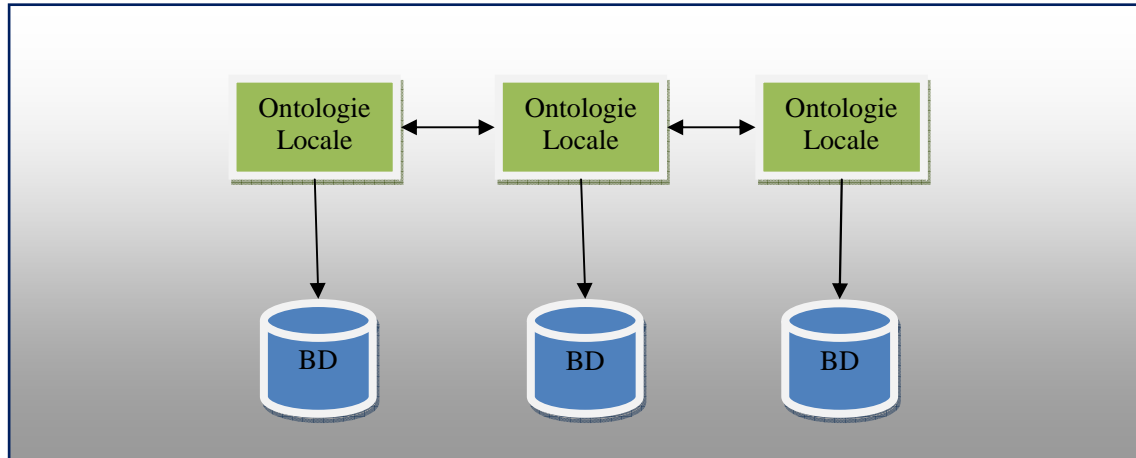


Figure II.8. Approche multi-ontologies [68].

### II.3.7.3. L'approche hybride :

Les approches hybrides combinent les deux approches précédentes en utilisant à la fois des ontologies locales et une ontologie partagée (parfois un vocabulaire commun) afin de simplifier les mappings ontologiques (figure II.9.). L'ontologie partagée fournit un vocabulaire commun et global ce qui rend les ontologies locales comparables. Cette ontologie partagée comprend les termes de base ou primitives d'un domaine. Afin de construire des termes complexes, ces termes de base sont combinés par des opérateurs, et les termes peuvent être comparés plus facilement que dans une approche multi-ontologies. Comme exemples de projets associés à cette approche, nous pouvons citer le projet COIN [70], le projet MECOTA

[71], et le projet BUSTER [72] [73] et qui utilisent respectivement la notion de contexte local, des mécanismes d'annotation, des raffinements d'une ontologie générale pour décrire localement les sources de données.

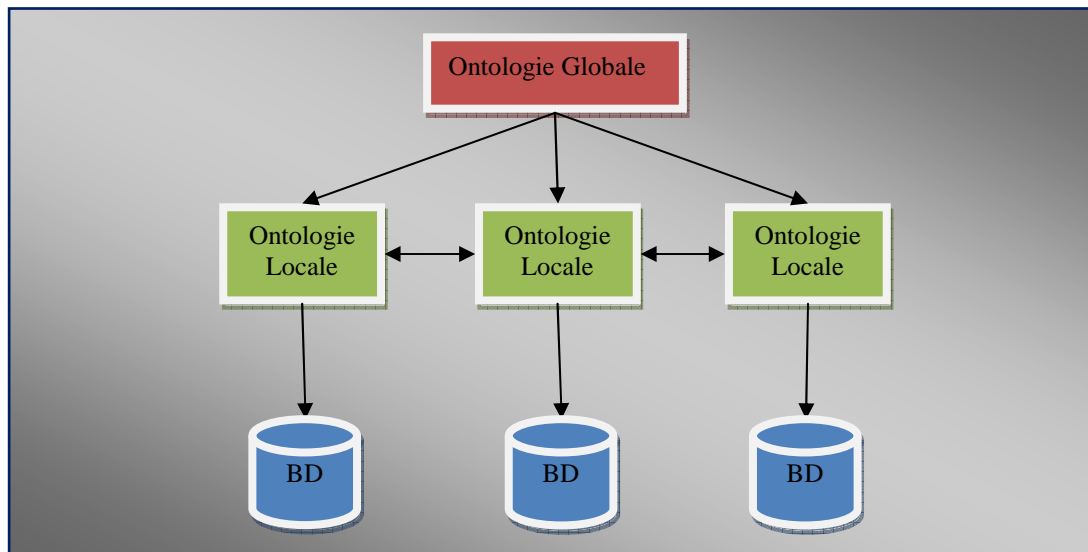


Figure II.9. Approche hybride [68].

#### II.3.7.4. Evaluation des approches ontologiques

	<i>Approche mono-ontologie</i>	<i>Approche multi-ontologie</i>	<i>Approche hybride</i>
<b>Effort d'implémentation</b>	Simple	Elevé	Raisonnable
<b>Hétérogénéité sémantique</b>	Sémantique unifiée d'un domaine	Hétérogénéité sémantique	Hétérogénéité sémantique
<b>Impact lors l'ajout/ mise à jour d'applications</b>	Besoin d'adaptation de l'ontologie globale	Ajouter une autre ontologie locale, mettre à jour l'ontologie locale associée à l'application mise à jour, liaison avec les autres ontologies locales	Ajouter une autre ontologie locale, mettre à jour l'ontologie locale associée à l'application mise à jour
<b>Comparaison d'ontologies</b>	Pas de comparaison d'ontologies (car il n'existe qu'une seule ontologie)	Difficile du fait qu'il n'existe pas d'ontologie partagé	Simple du fait qu'il existe une ontologie partagée

Tableau II.2. Evaluation des approches ontologiques [68].

Il est clair d'après ce tableau (Tableau II.2.), que l'approche hybride est plus avantageuse du fait que l'ajout de nouvelles sources est simple, et ne nécessite pas de modifications. Aussi, elle supporte l'acquisition et l'évolution des ontologies, et l'utilisation d'un vocabulaire partagé rend les ontologies sources comparables [68].

Les approches à ontologies multiples, et les approches hybrides peuvent être confrontées au problème d'hétérogénéité ontologique [74]. Dans ce cas, il devient nécessaire d'intégrer et de faire interopérer des ontologies [1]. Pour cela, il existe plusieurs approches telles que : le

mapping, l'alignement, la transformation et la fusion d'ontologies [75]. L'interopérabilité des ontologies revient à obtenir une médiation entre celles-ci, de ce fait nous choisissons la technique de mapping ontologique, qui est une solution faiblement couplée et apporte plus de flexibilité quant à l'interopérabilité des systèmes d'information.

### II.3.7.5. Approche de mapping d'ontologies

Le mapping d'ontologie peut être définie comme un ensemble de correspondances au sens mathématique entre un certain nombre d'ontologies [76]. Il repose sur la définition de relations de correspondance entre les entités de deux ontologies qui présentent une similitude. La principale caractéristique de cette approche est qu'elle ne modifie pas les ontologies impliquées et qu'elle produit en sortie un ensemble de correspondances. Le processus de mapping comprend généralement les étapes suivantes (figure II.10.) [77] :

- L'import des ontologies : qui consiste à charger les ontologies dans l'outil de mapping en effectuant éventuellement des translations de format ;
- La recherche de similarité : qui consiste à trouver et de façon semi-automatique les similarités qui peuvent exister entre les entités des deux ontologies ;
- La spécification des mappings : qui consiste à spécifier les mappings et qui peut aussi se faire de façon semi-automatique en utilisant un outil tel que PROMPT [81].

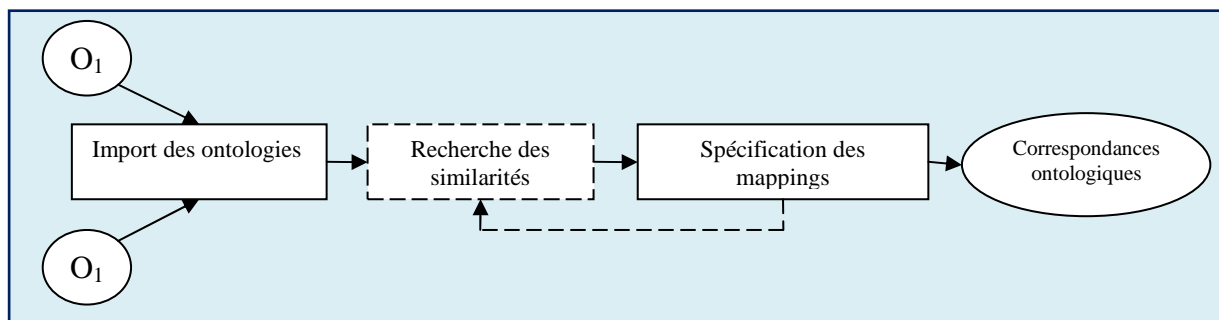


Figure II.10. Processus de mapping d'ontologies [78].

Plusieurs outils de mapping ont été développés, nous citons à titre d'exemple : GLUE [119], FCA-Merge [80], et PROMPT [81].

Par exemple, le système PROMPT [81], permet de stocker les mappings entre des ontologies sources créés par le système et par l'utilisateur. Ces mappings sont représentés sous forme déclarative (Les mappings sont représentés en tant qu'instances dans une ontologie de mappings). Ce qui leur permet d'être utilisés par un moteur d'inférence (Racer [67] ou Jena [60]).

## II.4. Classification des hétérogénéités des services Web

En raison de notre choix des architectures orientées services et des services Web, nous avons jugé nécessaire de présenter une classification des hétérogénéités de ces derniers qui est en relation directe avec nos travaux de recherche. Sachant que plusieurs classifications des hétérogénéités existent dans la littérature [13] [14] [82] [83] [84] (nous avons déjà évoqué quelques unes [13] [14] à la section I.2.2.3), nous avons choisi celle de Mrissa [34] du fait qu'elle tient compte des services Web :

### II.4.1. Hétérogénéités entre propriétés non fonctionnelles

Toutes les propriétés qui caractérisent un service Web et qui ne sont pas directement liées à la fonctionnalité délivrée par le service, sont considérées comme des propriétés non fonctionnelles (non décrites dans le WSDL). Les hétérogénéités des propriétés non fonctionnelles concernent :

- Les séquences d'échanges de message (ordre de messages différents, présence de messages supplémentaires, absence de messages requis etc).
- Les propriétés transactionnelles : ce sont les quatre propriétés essentielles d'un système de traitement de transactions (ACID).
- La qualité de service (QoS) : ce terme regroupe les propriétés non fonctionnelles relatives aux performances des services Web, telles que la disponibilité, la rapidité, le coût, la fiabilité, etc.

### II.4.2. Hétérogénéités entre propriétés fonctionnelles

Ce sont des hétérogénéités qui concernent les propriétés des services décrites dans un document WSDL. Les paramètres d'entrée/sortie, la fonctionnalité du service, le protocole d'échange utilisé, et l'encodage des données sont des propriétés fonctionnelles.

### II.4.3. Hétérogénéités des données échangées entre services Web

Les données échangées entre services Web peuvent présenter des hétérogénéités que Mrissa [34] classe en trois niveaux :

- Le niveau syntaxique : concerne l'encodage des données. les hétérogénéités de ce niveau sont résolues par la syntaxe XML.
- Le niveau structurel : est relatif aux différentes représentations des données au niveau schéma.
- Le niveau sémantique : englobe la signification véhiculée par les données.

Pour résoudre le problème d'hétérogénéités des services Web une description sémantique de ces derniers est nécessaire ainsi qu'une médiation. Pour cela, nous allons consacrer les deux sections suivantes aux approches de description sémantique, et de médiation des services Web.

## II.5. Description sémantique des services Web

Comme nous l'avons déjà mentionné la pile des standards des services Web ne prend pas en considération la sémantique de ces derniers. Pour atteindre l'interopérabilité sémantique les services Web doivent être capables [34]: (1) d'interpréter correctement la sémantique des données qu'ils envoient et reçoivent, (2) de décrire les fonctionnalités qu'ils fournissent en utilisant une sémantique explicite et compréhensible par les machines.

Pour cela, nous distinguons principalement deux approches permettant de décrire sémantiquement les services Web et qui sont [34] : Les approches exploitant les langages de description sémantique et les approches d'annotations de langages existants.

## II.5.1. Langages de description sémantique

Les langages qui décrivent sémantiquement les services Web sont issus du Web sémantique, et sont :

### II.5.1.1. Le langage OWL-S

Ontology Web Language for Web Services (OWL-S) est un sous ensemble du langage OWL (Ontologie Web Language) [85] dédié à la description sémantique de services Web. Une description OWL-S se compose de trois éléments [86] :

- *Le service profile* décrit les fonctionnalités des services Web, il est utile pour leur découverte et leur sélection.
- *Le process model* détaille la sémantique des données échangées, au niveau des messages échangés entre services Web.
- *Le grounding* spécifie l'encodage des données échangées, les protocoles de communication, ainsi que tous les détails concrets nécessaires à l'invocation du service.

OWL-S sépare les vues de haut et de bas niveau (listing II.1.). La vue abstraite (de haut niveau), relie le service Web à des descriptions conceptuelles en OWL, décrites dans des ontologies. La vue concrète (de bas niveau) décrit la représentation physique du service Web. Cette représentation permet différentes représentations physiques du même concept, et renforce le rôle des ontologies dans la représentation abstraite de la sémantique des données [34].

```
<!-- Description abstraite -->
  <process:Input rdf:ID="`InputLanguage">
    <process:parameterType rdf:datatype="`&xsd:anyURI">
      &this;#SupportedLanguage
    </process:parameterType>
    <rdf:label>Input Language</rdf:label>
  </process:Input>
<!-- Description concrète -->
  <grounding:WsdInputMessageMap>
    <grounding:owl:parameter rdf:resource="`#InputLanguage"/>
    <grounding:wsd:MessagePart rdf:datatype="`&xsd:anyURI">
      &groundingWSDL;#inputLanguage
    </grounding:wsd:MessagePart>
  </grounding:WsdInputMessageMap>
```

Listing II.1. Extrait de description OWL-S.

### II.5.1.2. Web Service Modelling Ontology (WSMO)

Web Services Modelling Ontology (WSMO), proposée par DERI [87], est une architecture conceptuelle dont le but est de décrire la sémantique des services Web. Le méta-modèle du WSMO définit quatre éléments principaux :

**Les ontologies** : fournissent la fondation pour une description sémantique des données dans le but de réaliser l'interopérabilité sémantique. Elles sont utilisées par les autres éléments du WSMO,

**Les objectifs** : servent à décrire les souhaits des utilisateurs en termes de fonctionnalités requises.

**Les services Web** : une description est associée à chaque service, dans le but de décrire sa fonctionnalité, son interface, et ses détails.

**Les médiateurs** : sont utilisés pour résoudre les incompatibilités inhérentes des hétérogénéités des systèmes pour assurer leur interopérabilité. Ce sont principalement les incompatibilités de données et de processus.

Une caractéristique principale du WSMO est que les médiateurs constituent des composants centraux qui lient les autres éléments (ontologies, objectifs et services Web). Il existe quatre sortes de médiateurs qui sont :

**OO-médiateurs** : permettent de résoudre les conflits entre ontologies.

**GG-médiateurs** : permettent d'établir des correspondances entre objectifs.

**WW-médiateurs** : permettent de résoudre les conflits des données, des protocoles et des processus.

**WG-médiateurs** : permettent de résoudre les conflits entre les fonctionnalités offertes par les services Web et les requêtes des utilisateurs.

### II.5.1.3. DIANE Elements (DE) et DIANE Service Description (DSD)

Les langages DE et DSD sont des langages orientés objets construits à partir d'une analyse des conditions requises pour la description des services Web sémantiques, et des difficultés de OWL-S et WSMO à remplir ces conditions [88]. Ces langages utilisent les notions d'ensembles configurables et de logique floue pour améliorer la découverte sémantique de services. DE est un langage général d'ontologie visant à améliorer la description de services Web sémantiques. Ce langage hérite de la F-logique pour décrire les concepts d'attributs, de types de données primitifs (empruntés à XML), et d'éléments restreints (seulement huit types primitifs). Le langage DSD, utilise les constructions fournies par DE afin de décrire les services Web. Il est construit autour des notions de descriptions de requête et d'offre de service [34] [88].

Les langages précédents font partie du Web sémantique, et ils tentent de remplacer les langages existants des services Web pour élever ces derniers au rang des services Web sémantiques.

## II.5.2. Approches d'annotation des langages existants

« L'annotation est une des formes les plus communes de méta données dans le contexte du Web, c'est aussi une information graphique ou textuelle attachée à un document et le plus souvent placée dans ce document. L'annotation sémantique est un cas particulier d'annotation parce qu'elle fait référence à une ontologie. » D'après N. Zouggar et al. [51].

Les travaux de cette approche consistent à annoter les langages existants des services Web en exploitant leurs éléments d'extensibilités ou en modifiant les fonctionnalités initiales des normes. Ces annotations peuvent concerner :

- Les processus métier comme le langage SESMA (SEmantic Service MArkup) [89], qui fournit un support pour la description de services composites, en annotant les processus métiers WS-BPEL ;
- Les registres UDDI et ebXML, en utilisant des langages comme DAML-S [90] [91] [92] et SAWSDL [93] (voir ci-dessous) ce qui permet d'améliorer la publication et la découverte des services Web
- Le langage de description WSDL soit en exploitant les éléments d'extensibilité de ce dernier avec le WSDL-S [94] [95] [96], soit en utilisant le SAWSDL qui constitue une recommandation du W3C. Vue son importance nous allons le décrire ci-dessous.

### Présentation du SAWSDL

Le Semantic Annotations for Web Service Description Language un des récents standards du World Wide Web Consortium W3C [97] définit comment ajouter des annotations sémantiques aux différents composants d'un document WSDL2.0 [41] (et supporte aussi le WSDL1.1 [98]) tels que les messages d'entrée et de sortie, les interfaces et les opérations. Le SAWSDL définit aussi des mécanismes d'annotation pour spécifier les mappings syntaxiques des types de schémas XML [99] [100] de et vers un modèle sémantique. Ces mappings sont utilisés pendant l'invocation et en particulier quand la médiation est nécessaire. Pour accomplir l'annotation sémantique, SAWSDL définit des attributs d'extension qui peuvent être appliqués aux éléments WSDL ainsi qu'aux éléments des schémas XML qui sont (figure II.11.):

**ModelReference** : SAWSDL définit cet attribut d'extension pour annoter les différentes parties du document de description, wsdl: interface, wsdl: operation, wsdl: fault, xs: element, xs: complexType, xs: simpleType et xs: attribute. Les annotations des quatre derniers composants du WSDL définissent les sémantiques des données en entrée et en sortie des opérations.

La valeur de l'attribut Model Reference est un ensemble de zéro ou plusieurs URI séparés par des blancs. Chaque URI est un pointeur vers un concept du modèle sémantique.

**LiftingSchemaMapping et LoweringSchemaMapping** : sont des attributs ajoutés aux définitions de types pour spécifier les correspondances entre les éléments du schéma des données et l'information sémantique du modèle de référence.

La valeur de l'attribut LiftingSchemaMapping ou LowringSchemaMapping est un ensemble de zéro ou plusieurs URI qui référencent les définitions des mappings.

Les mappings référencés par le premier attribut définissent comment une instance de donnée XML relative à un élément ou un type défini dans un schéma XML est transformée vers une ou plusieurs données conceptuelles du modèle sémantique.

Les mappings référencés par le second attribut définissent comment une données dans le modèle sémantique est transformée vers une instance de donnée XML.

En ce qui concerne les schémas XML tous les attributs peuvent être appliqués de la même manière pour les descriptions WSDL1.1 et WSDL2.0.

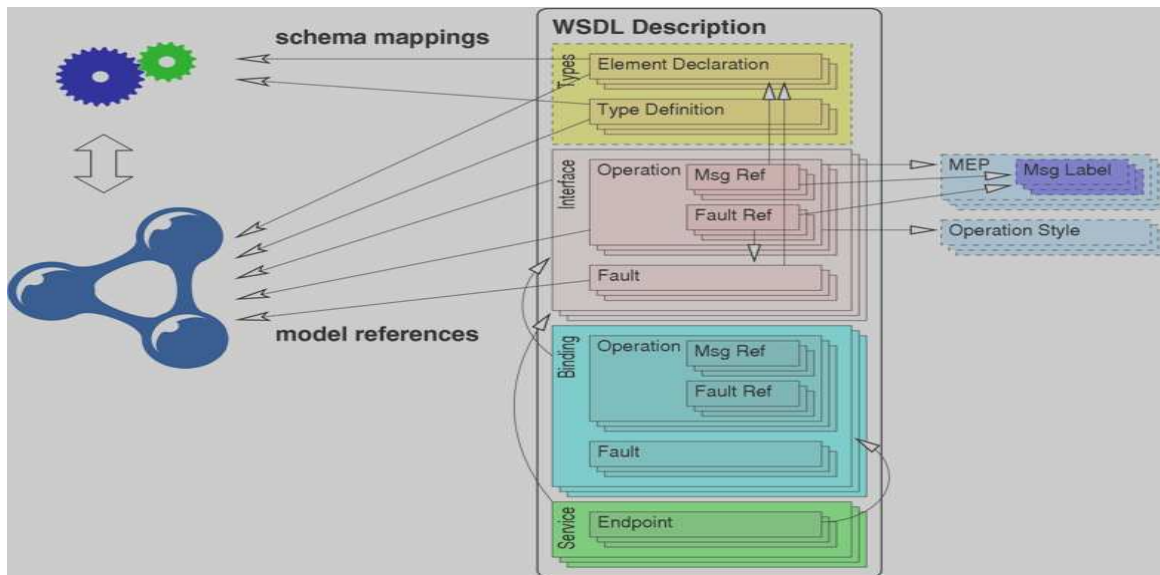


Figure II.11. Le document WSDL 2.0 avec les annotations SAWSDL [101].

Avec l'attribut **ModelReference** de ce standard on peut annoter tous les éléments de la partie abstraite de la description du service Web. Et avec les attributs **LiftingSchemaMapping** et **LoweringSchemaMapping** on peut annoter les définitions les éléments du XML Schema et les messages d'entrée/soties (figure II.12.).

Ces annotations sont utilisées pour permettre la publication, la découverte, la sélection et la médiation sémantiques.

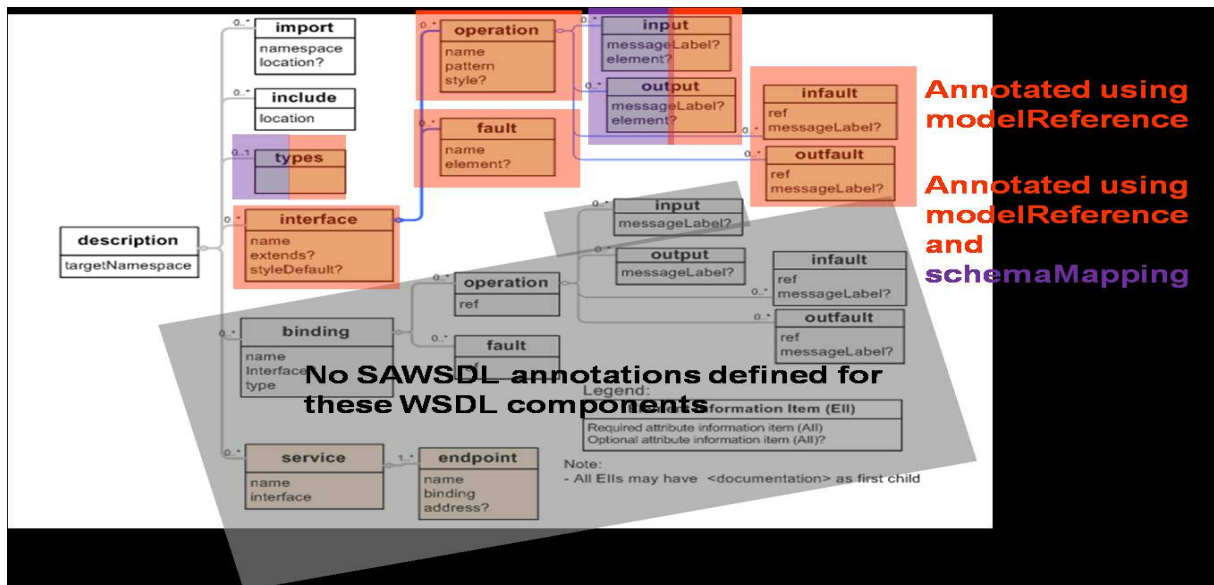


Figure II.12. Les parties annotées par le SAWSDL [101].

Les langages recommandés pour le lifingSchemaMapping sont : XSLT (XML Style Sheet Transformation) [110], et Xquery [111].

Les langages recommandés pour le loweringSchemaMapping sont : SPARQL [112], suivi de XSLT ou Xquery.

Les outils permettant l'annotation des documents WSDL sont : SAWSD4J [113], WSMO Studio [114], et Radiant [115].

## II.6. Médiation des services Web

La résolution des conflits causés par les hétérogénéités qui peuvent apparaître entre les services Web est prise en compte par une médiation nécessaire pour assurer l'interopérabilité des systèmes d'information. Cette médiation constitue le noyau des architectures orientées services. Nous allons la présenter dans ce qui suit :

### II.6.1. Définition

« D'une manière générale, la médiation consiste à résoudre les conflits entre deux acteurs. Cette tâche est effectuée par un élément spécifique appelé médiateur. » [34]. La première architecture de médiation a été conçue en 1992 par G. Weiderhold [13] qui introduit la notion de médiateur : « Un médiateur est un module logiciel qui exploite la connaissance codée relative à des ensembles ou sous-ensembles de données afin de créer des informations pour une couche supérieure d'application. ». La notion de médiation a été initialement utilisée dans le domaine des bases de données, puis adaptée aux services Web.

### II.6.2. Classification de la médiation

Il est possible de classifier les tâches de médiation selon diverses perspectives conformément à la classification des hétérogénéités [82] [83] [84], nous retenons celles de Mrissa [34] du fait qu'elle tient compte des propriétés fonctionnelles et non fonctionnelles des services Web. Mrissa propose trois niveaux de médiation comme suit :

- **Le niveau intégration des services Web** : a pour but de résoudre toutes les hétérogénéités entre les services Web à savoir, les hétérogénéités des propriétés fonctionnelles et non fonctionnelles et les hétérogénéités des données (section II. 4.)
- **Le niveau adaptation d'interfaces** : concerne la résolution des hétérogénéités des propriétés fonctionnelles.
- **Le niveau médiation des données** : Concerne la résolution des hétérogénéités des données.

Comme il apparaît sur la figure II.13., ces niveaux de médiation sont imbriqués l'un dans l'autre.

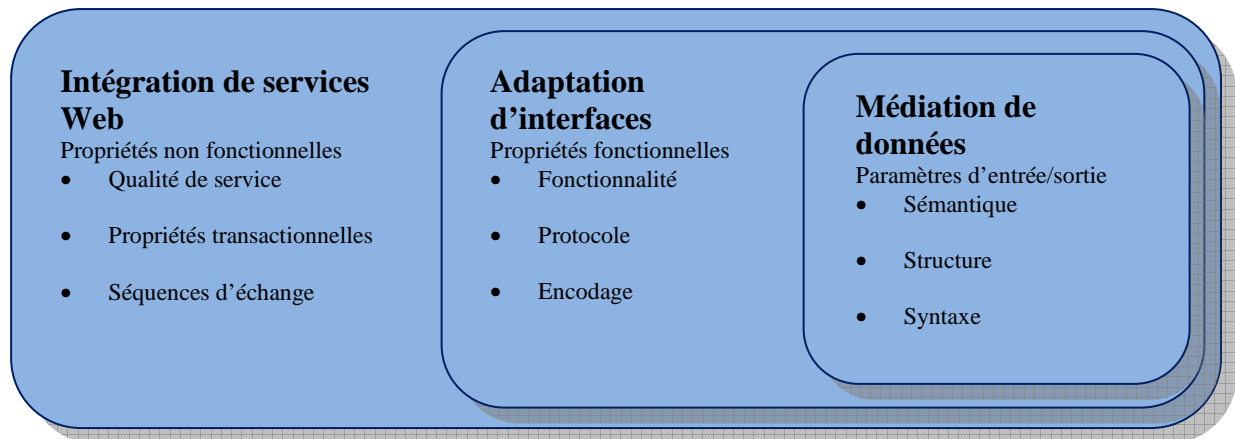


Figure II.13. Classification des niveaux de médiation entre services Web [34].

## II.7. Interopérabilité et intégration des services Web

La nouvelle orientation des entreprises est l'utilisation des services Web pour développer des systèmes d'information interopérables. Cependant, les éventuelles hétérogénéités sémantiques freinent cette interopérabilité. Pour surmonter ce problème, il est plus que nécessaire d'intégrer et faire inter-opérer des services Web par des mécanismes permettant leur publication, sélection et médiation au niveau sémantique. Dans cette section nous décrivons les approches d'intégration des services Web, et nous discutons quelques travaux qui intéressent de près notre recherche.

### II.7.1. Classification des approches d'intégration des services Web

Mrissa [34] distingue les types d'approches suivants :

- **Approches à base d'adaptateurs** : Ces approches reposent sur l'utilisation d'une couche d'adaptateurs qui encapsulent les services Web. Cette couche est intégrée à une architecture homogène qui supporte leur découverte, leur composition et leur administration. Les hétérogénéités sont résolues manuellement lors de l'intégration des services dans l'architecture par les fournisseurs, en établissant les correspondances entre leurs services et les adaptateurs [117].
- **Approches à base de communautés** : Ces approches regroupent plusieurs services fournissant une fonctionnalité équivalente derrière une interface unique. Ces services forment alors une communauté. Les correspondances entre l'interface de la communauté et les réalisations concrètes de la fonctionnalité sont établies par plusieurs médiateurs [118].
- **Approches à base de descriptions étendues** : Ces approches ajoutent de l'information supplémentaire dans les descriptions des services Web. Ces informations concernent les propriétés fonctionnelles et non fonctionnelles des services Web. Les solutions de

médiations utilisent ces descriptions améliorées pour réconcilier les services Web [116].

- **Approches à base de services Web** : Ces approches reposent sur les services Web pour la publication, l'invocation la médiation et l'exécution. Elles sont plus prometteuses par rapport aux premières, car elles bénéficient des avantages des architectures orientées services.

Dans notre recherche nous nous intéressons aux deux dernières approches et par conséquent, nous allons discuter quelques uns de leurs travaux dans la section suivante.

## II.7.2. Etude des travaux portant sur l'interopérabilité sémantique

Nous exposons quelques travaux qui touchent de près notre recherche.

### *L'architecture WSMX :*

L'architecture orientée services sémantiques WSMX de Haller et al. [102], est un environnement d'exécution des services Web basée sur le Web Services Modelling Ontology (WSMO). Cette architecture permet la découverte dynamique, l'invocation et la composition des services Web sémantiques, elle est utilisée par les fournisseurs pour enregistrer et offrir leurs services, et par les clients pour découvrir dynamiquement et invoquer les services sélectionnés. WSMX assure l'interopérabilité des systèmes B2B (Business to Business), elle-même se présente en tant que service Web.

La médiation entre les propriétés non fonctionnelles est réalisée grâce à un travail manuel effectué durant la phase conception du processus métier, par la définition de règles permettant d'identifier les équivalences entre les différentes chorégraphies. Ces règles étant stockées puis utilisées par le moteur de chorégraphie afin de réconcilier les gestions des séquences de messages.

La médiation d'interfaces est effectuée en suivant le format de description de services du modèle conceptuel WSMO. Les hétérogénéités entre les concepts sont résolues par des mécanismes de raisonnement, qui établissent des correspondances entre les concepts des différentes ontologies utilisées par les fournisseurs.

Le médiateur des données est constitué d'un premier composant identifiant les similarités entre les concepts des ontologies et génère des correspondances entre ces derniers. Le deuxième composant effectue la transformation des données durant l'exécution de la composition en utilisant les correspondances stockées du premier composant.

### *L'architecture IRS-III*

L'architecture IRS-III [84] est un framework permettant la publication, la configuration et l'exécution des services Web hétérogènes. Elle est basée aussi sur le Web Services Modelling Ontology (WSMO) et comprend les composants suivants : Server, Publisher et client qui communiquent à travers le protocole SOAP. IRS-III implémente la médiation des données, la médiation des objectifs et la médiation des processus.

Cabral et al. [84] utilisent un médiateur de processus pour résoudre les conflits de communication en exploitant un ensemble de règles logiques. Ce médiateur utilise un

interpréteur de chorégraphie, qui permet d'exécuter un service Web en créant les messages requis par ce dernier sur la base de sa description. Il mémorise les appels exécutés par un composant appelé invocateur.

La médiation d'interfaces est prise en charge par un service Web médiateur qui utilise un composant appelé interpréteur d'orchestration. Et fait appel à des médiateurs de fonctionnalité, dont le rôle est de faire correspondre la demande de fonctionnalité de l'application cliente avec une description sémantique de service fournie par l'architecture.

Le médiateur de données génère des correspondances entre les termes des ontologies, et les stocke dans une ontologie temporaire qui est une fusion des ontologies de départ. Grâce à laquelle il est possible de convertir les données d'une ontologie à l'autre en passant par une représentation intermédiaire.

Tous ces médiateurs appartiennent à un composant appelé 'broker' ou courtier.

L'architecture IRS-III est utilisée par [103] comme une infrastructure de base pour l'interopérabilité sémantique au sein d'une architecture d'un portail e-gouvernemental. Cette architecture est appelée "*Architecture basée services Web sémantiques pour l'interopérabilité des services E-government*", et inclut trois couches distinctes qui sont : Interaction utilisateur, Middleware et services Web. Le Middleware est le cœur de cette architecture et comprend l'infrastructure pour l'interopérabilité sémantique (qui utilise IRS-III) et le modèle conceptuel constitué de l'ontologie de services, l'ontologie de domaine E-government et l'ontologie de domaine des événements de la vie (Life Event). Ce portail est utilisé pour la publication de nouveaux services et l'invocation des objectifs.

### ***L'architecture ODSOIA***

Izza [1] quant à lui propose une architecture orientée services pour l'intégration des applications des systèmes d'information industriels appelée ODSOIA (Ontology Driven Service Oriented Architecture), c'est une solution intra-entreprise qui augmente le bus ESB de deux couches architecturales en plus de la couche orientée services : la couche sémantique et la couche d'intégration.

Pour ce qui est de l'architecture de services il propose une démarche de construction de services fondamentaux : services IT (Informatique) et services métier, et d'urbanisation des services obtenus avant leur publication syntaxique à l'aide d'un service de publication (syntaxique).

L'architecture sémantique comprend un ensemble d'ontologies globales, de domaines et locales, des ontologies de services et de mappings concernant les processus, les fonctions et les données, puis vient une publication sémantique des descriptions des services en utilisant un service de publication (sémantique). Pour décrire sémantiquement les services Web il utilise le OWL-S.

Les deux premières architectures sont utilisées par l'architecture d'intégration composées de services d'intégration qui sont : service broker permettant la gestion du processus d'intégration des services fondamentaux, service de description sémantique, service de publication sémantique, service de découverte, service de médiation qui prend en charge la résolution des hétérogénéités sémantiques liées aux services fondamentaux, et service

d'exécution permettant la supervision de l'exécution des services fondamentaux. Cette architecture utilise aussi des référentiels de services et des référentiels d'ontologies.

Pour la médiation il existe trois catégories de services qui sont : service de médiation de données, service de médiation fonctionnelle et service de médiation de processus [104].

Dans ces différentes médiations, plusieurs types d'ontologies et de mappings sont utilisés.

### *Architecture de médiation sémantique orientée contexte*

Mrissa [34] propose une architecture de médiation sémantique orientée contexte pour la composition de services Web. C'est une architecture comprenant les couches suivantes : une couche fournisseur, une couche composition et une couche description. Il s'agit d'une médiation de données prise en charge par un service Web médiateur inséré entre le service Web émetteur et le service Web récepteur lors de l'exécution de la composition. Ces services sont préalablement décrits sémantiquement en utilisant les éléments d'extensibilité du WSDL et une ontologie de domaine à laquelle sont associées des ontologies contextuelles. Mrissa définit l'objet sémantique grâce à la notion de contexte lui-même définit comme un objet sémantique, il obtient ainsi une définition récurrente de la notion de contexte qui constitue tout élément nécessaire à l'interprétation correcte de la donnée.

Faisons le point de tous les travaux présentés par le tableau comparatif suivant (Tableau I.2.) :

Architecture Critère	Protocole PRESTO	Plate forme Mélusine	WSMX/IRS-III	ODSOIA	Architecture de médiation sémantique orientée contexte
Niveau d'interopérabilité	Technique	Technique	Sémantique	Sémantique	Sémantique
Approche de description sémantique	/	/	WSMO	OWL-S	WSDL-S (SW) Contexte
Approche d'intégration des services Web	Basée services Web	Basée adaptateurs	Basée services Web	Basée services	Basée services Web Descriptions étendues
Médiation	/	Lien entre les activités du workflow et leurs implémentations	Processus Fonctions Données	Processus Fonctions Données	Données

*Tableau comparatif des travaux existants*

### **II.7.3. Discussion**

Comme nous l'avons déjà mentionné les approches orientées services sont plus intéressantes que les autres, de ce fait, nous avons présenté quelques travaux de ces approches qui se réunissent dans l'utilisation des services (voir tableau I.2), et la description sémantique de ces

derniers ou bien par un langage de description sémantique (cas des travaux : [102], [84], [1]) ou une annotation des langages existants en exploitant leurs éléments d'extensibilité (cas des travaux de : [34]). Ces annotations présentent plus d'avantages que les langages de description du moment qu'elles exploitent les normes existantes des services Web, en outre, elles sont relativement simples.

Notons qu'à part les travaux de Mrissa [34], tous les autres supposent l'adaptation de la sémantique locale des services Web à la sémantique de l'ontologie partagée, pour pouvoir effectuer la médiation.

En ce qui nous concerne, nous optons pour les annotations des descriptions des services Web à l'aide du récent langage SAWSDL qui constitue une recommandation du W3C, et un modèle sémantique à base d'ontologies et la notion de contexte introduite par Mrissa. Notre approche et architecture sont inspirées des travaux de Mrissa et Izza.

## **II.8. Conclusion**


Dans ce chapitre, nous avons présenté la notion de sémantique et sa représentation notamment par les ontologies, qui constituent le meilleur outil conceptuel de capture et de description des connaissances. Vue leur importance et leur large utilisation dans plusieurs domaines, une panoplie de définitions ont été dressées par beaucoup d'auteurs depuis leur apparition dans le domaine de la philosophie jusqu'à leur adoption en IA (Intelligence Artificielle), aussi, plusieurs méthodes de conception, outils et langages de description des ontologies ont été présentés.

Ce chapitre a notamment présenté des classifications des hétérogénéités sémantiques des services Web, des approches de description de la sémantique de ces derniers, des médiations résolvant les confits inhérents, et une classification des approches d'interopérabilité et d'intégration des services Web pour dresser l'interopérabilité sémantique des systèmes d'information. A l'issue de cette classification, nous avons retenu les approches orientées services et nous avons présenté une étude des travaux existants.

Le chapitre suivant est consacré à notre contribution qui concerne les niveaux d'interopérabilité technique et sémantique, en proposant une approche et une architecture flexible et orientées services Web sémantiques.



## Chapitre III



Une Architecture orientée  
services sémantiques pour la  
mise en œuvre de  
l'interopérabilité des  
systèmes d'information

# Chapitre III

## Une architecture orientée services sémantiques pour l'interopérabilité des systèmes d'information

### III.1. Introduction

Dans les chapitres précédents nous avons présenté entre autres les systèmes d'information, la nécessité de leur coopération et, les outils conceptuels et technologiques pour la mise en œuvre des mécanismes d'interopérabilité au niveau technique, et au niveau sémantique. Nous avons aussi discuté quelques approches et architectures. Pour assurer la coopération des systèmes d'information, les outils permettant l'interopérabilité technique se sont avérés insuffisants à cause de l'existence de l'hétérogénéité sémantique située à plusieurs niveaux des services Web.

Nous allons consacrer ce chapitre pour décrire l'approche et l'architecture proposées pour mettre en œuvre les mécanismes de l'interopérabilité des systèmes d'information au niveau technique et sémantique.

Pour assurer l'interopérabilité au niveau technique et syntaxique, nous avons retenu l'approche orientée services, les architectures dirigées par les modèles (MDA) et les services Web. Et pour ce qui est de l'interopérabilité au niveau sémantique nous avons proposé l'enrichissement des descriptions des services Web par l'information sémantique décrite par une architecture conceptuelle basée sur la notion d'ontologies, et la notion de contexte. Les ontologies obtenues et les annotations des descriptions, seront utilisées pour la découverte sémantique et la médiation des services Web pour assurer leur interopérabilité. L'architecture qui en découle est structurée en couches dont le noyau est une infrastructure de médiation. Dans le cadre de cette thèse nous détaillons le sous modèle sémantique des données et la médiation des données échangées entre les services Web.

Ce chapitre est structuré en plusieurs parties, en premier lieu, nous décrivons l'approche globale pour l'interopérabilité des systèmes d'information qui mettra en exergue notre architecture, que nous détaillons par la suite.

Dans la deuxième partie nous traitons la migration des systèmes d'information vers les architectures orientées services en utilisant le MDA et les services Web.

La troisième partie sera consacrée à la description de l'architecture conceptuelle en exploitant la notion d'ontologies et de contexte pour détailler le sous modèle sémantique des données.

Ensuite, nous présentons les mécanismes d'annotation des descriptions des services en utilisant le standard SAWSDL et le sous modèle sémantique de données. Il s'agit surtout de l'annotation des données.

La dernière partie décrit l'infrastructure pour l'interopérabilité des systèmes d'information et détaille l'approche de médiation des données entre les services Web.

## III 2. Approche globale

Notre contribution porte sur la coopération des systèmes d'information. L'approche préconisée est basée sur les services Web sémantiques et peut être résumée comme suit :

- ✓ ***Construction d'une architecture orientée services :*** Nous avons retenu les architectures orientées service (SOA) en raison de leurs avantages multiples. Or, leur principale limite, est le manque de méthodologies permettant de les construire de façon efficace. De ce fait, nous adoptons une méthodologie de migration des systèmes d'information vers une architecture orientée services basée sur le MDA d'où un catalogue de différents types de services est dégagé dont les services métiers. Ces derniers seront exposés aux organisations tierces à l'aide des principaux standards des services Web via le bus de services d'entreprise ESB (& I. 5.3.5), qui permet aux services métiers de communiquer et de coopérer. Notre approche consiste à étendre l'ESB par une architecture conceptuelle et une architecture d'interopérabilité sémantique.
- ✓ ***Construction d'une architecture conceptuelle décrivant l'architecture orientée services déjà construite :*** Les standards utilisés par les services Web supportent l'interopérabilité au niveau technique. Cependant les hétérogénéités sémantiques entre les services Web sont complexes et rendent leur interopérabilité difficile à cause de la non prise en compte de la sémantique par la pile des standards des services Web. De ce fait, nous proposons une architecture conceptuelle permettant de capturer la sémantique des services Web. Cette architecture est basée sur la notion d'ontologies locales, contextuelles et de domaine.
- ✓ ***Annotation de l'architecture orientée services:*** Cette étape consiste en la proposition d'une méthodologie pour enrichir sémantiquement les descriptions des services Web en se basant sur le Semantic Annotations for Web Service Description Language (SAWSDL), un des récents standards du World Wide Web Consortium W3C [97], et en prenant les ontologies de l'architecture conceptuelle comme référence pour cette annotation.
- ✓ ***Architecture pour l'interopérabilité sémantique des services Web :*** A ce niveau se dégage notre architecture orientée services Web sémantiques comprenant les éléments précédents et permettant la publication, la découverte l'invocation et l'exécution des services Web hétérogènes en faisant appel à la médiation pour résoudre les conflits qui peuvent exister entre les services Web et permettre leur interopérabilité. C'est une architecture en couches comme le montre la figure III.1.  
Dans cette architecture le processus de coopération est pris en charge par un service Web broker (courtier) qui orchestre les autres types de services d'interopérabilité et qui sont les services Web : de description, de publication, de découverte, de médiation, et d'exécution.  
Dans le cadre de cette thèse nous présentons de façon générale le rôle de chaque type de service d'interopérabilité et le processus de coopération, et nous détaillons la médiation sémantique des données nécessaire à la résolution des éventuels conflits pour garantir des échanges de données consistants et permettre l'interopérabilité des services Web sans échec.

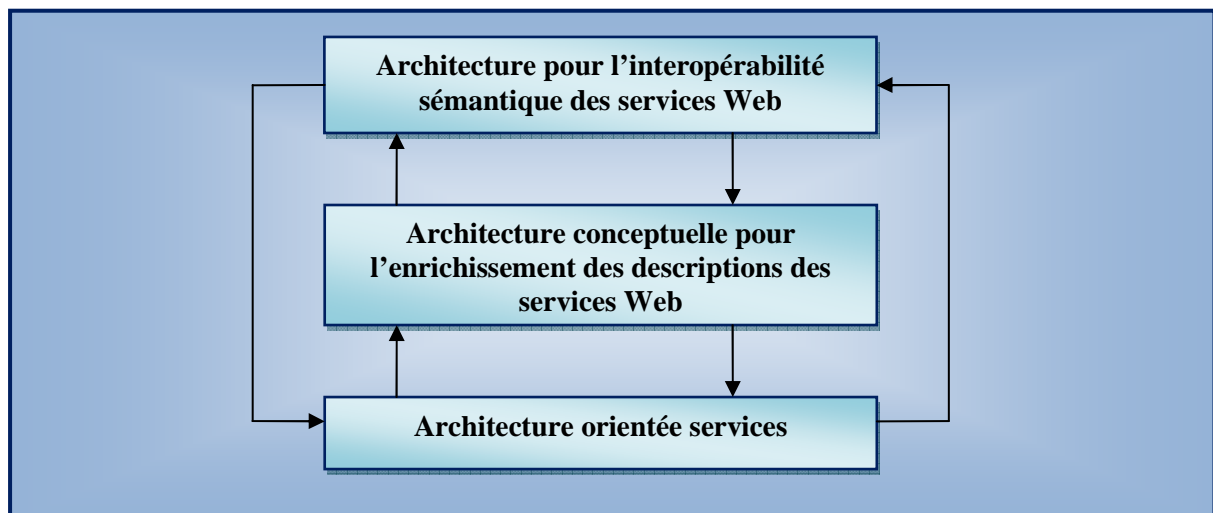


Figure III.1. Différentes couches pour l'interopérabilité sémantique des systèmes d'information.

D'après la figure III.1 les trois couches sont étroitement liées en raison des interactions qu'elles entretiennent : la première couche concerne l'architecture orientée services, est décrite sémantiquement par la deuxième couche représentant l'architecture conceptuelle. Cette dernière est utilisée par la couche supérieure pour faire interopérer les services de la première couche.

Notre approche présente de nombreux avantages et à plusieurs niveaux :

Les architectures orientées services et les services Web constituent des paradigmes incontournables pour le développement des systèmes d'information interopérables, en raison surtout du couplage faible qu'ils procurent. L'approche adoptée permet de faire migrer les systèmes d'informations vers une architecture orientée services de façon progressive et en intégrant l'existant.

L'introduction de la notion de contexte et l'enrichissement sémantique des descriptions des services Web, apportent plusieurs avantages tels que la description explicite des hétérogénéités sémantiques qui permet l'automatisation des conversions des données, et par conséquent, facilite la médiation de ces dernières. Aussi, elle évite aux fournisseurs d'adapter leurs sémantiques locales. L'utilisation du SAWSDL qui constitue une recommandation du W3C permet l'interopérabilité avec les architectures existantes, et l'adhésion de nouveaux fournisseurs à notre architecture.

L'infrastructure d'interopérabilité sémantique proposée est basée services, ce qui la rend flexible. Aussi, elle est semi centralisée du fait que le processus de coopération est pris en charge par un service Web courtier qui supervise les autres services d'interopérabilité.

Dans ce qui suit nous allons décrire les étapes de notre approche en détail.

### III.3. Passage vers une architecture orientée services

Nous traitons dans cette section une démarche de passage des systèmes d'informations vers une architecture orientée services (SOA). Il s'agit d'une démarche de construction d'une

SOA, qui se déroule selon trois cycles : un cycle d'abstraction, un cycle de vie et un cycle de décision (figure III.2).

**Cycle d'abstraction**, qui se déroule selon une approche MDA pour relever le niveau d'abstraction, et modéliser le domaine métier de l'entreprise.

**Cycle de vie**, permettant la construction des services et leur publication.

**Cycle de décision**, tout au long des deux cycles précédents des décisions sont prises et par la maîtrise d'ouvrage et par la maîtrise d'œuvre du fournisseur ainsi que la maîtrise d'ouvrage du consommateur (client).

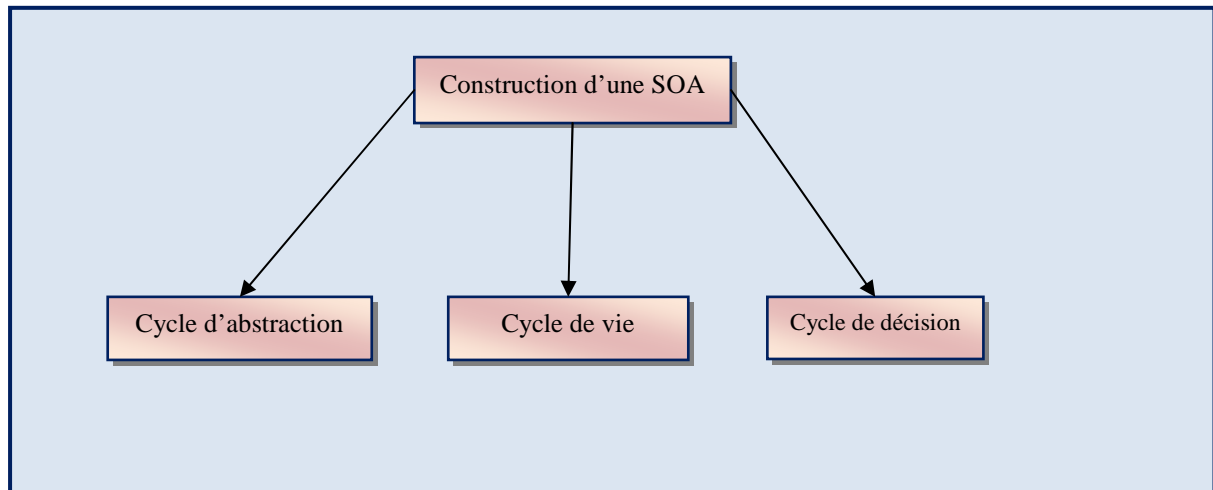


Figure III.2 Cycles de construction d'une SOA.

Notons d'abord que les architectures orientées services sont décomposées en deux architectures : la SOA IT (ou SOA applicative) et la SOA métier.

Contrairement à la SOA métier, il existe plusieurs outils et méthodes pour définir les services applicatifs de la SOA IT tels que : IBM WebSphere [122], Apache Beehive [123] ...etc.

Dans le cadre de nos travaux de recherche nous nous intéressons à la SOA métier qui constitue une solution essentielle pour assurer l'interopérabilité des systèmes d'information.

### III.3.1. Approche de construction de la SOA métier

Deux approches fondamentales existent pour la construction de logiciels sont : l'approche descendante (Top-down) et l'approche ascendante (bottom-up) [34]. L'approche descendante est généralement plus complexe mais complète, tandis que l'approche ascendante est plus pragmatique mais plus restreinte. L'approche descendante favorise les services à forte granularité. Cependant l'approche ascendante favorise les services à fine granularité.

La démarche que nous adoptons (figure III.3) combine les deux approches pour les raisons suivantes :

- Le choix du grain pertinent.
- L'accélération de l'approche descendante par le mécanisme de réutilisation introduit par l'approche ascendante.

- La prise en compte des deux architectures qui sont la SOA métier (exposition de services métiers) et la SOA applicative (architecture applicative orientée services).

L'approche que nous adoptons comporte trois phases principales (voir figure III.3):

- ✓ Construction des services métiers.
- ✓ Construction des services applicatifs.
- ✓ Publication des services fonctionnels et métiers.

### III.3.2. Phase de construction des services métiers

Sachant que nous avons opté pour une approche hybride, la construction des services métiers se base sur une abstraction (approche descendante) et une réutilisation (approche ascendante).

En ce qui concerne l'abstraction, il s'agit de la modélisation habituelle des processus à l'aide d'un diagramme d'activité UML, l'impact de la SOA métier apparaît ici d'une part en la spécification, des services métiers par la découverte des opérations<sup>10</sup>, et d'autre part en la spécification des services fonctionnels par la découverte des phases<sup>11</sup>.

En ce qui concerne la réutilisation, elle consiste en la construction de services métiers (ou fonctionnels) à partir des services applicatifs issus de la construction de la SOA applicative.

Cette phase nécessite l'intervention de la maîtrise d'ouvrage et la maîtrise d'œuvre du fournisseur (et parfois la maîtrise d'ouvrage du client), pour décider de promouvoir :

- des services fonctionnels en services métiers, ou
- des services applicatifs en services fonctionnels ou métiers.

### III.3.3. Phase de construction des services applicatifs

Dans cette phase l'approche descendante consiste d'abord en la modélisation habituelle des données à l'aide d'un diagramme de classe UML. L'impact de la SOA applicative ici consiste en une cartographie du modèle de données obtenu en catégories<sup>12</sup>, tout en favorisant l'action de réutilisation des catégories existantes déjà (approche ascendante).

Suite à cette modélisation vient l'étape de construction des services respectant le pattern d'architecture SOA. En exploitant les modèles précédents, les traitements issus de la modélisation des processus sont décomposés sous forme de services rattachés à des catégories obtenues de la cartographie du modèle de données. La réutilisation des services déjà existant est favorisée par le pattern même de la SOA.

Les services obtenus sont réifiés sous forme de composants.

L'approche ascendante consiste en l'encapsulation des composants existants par des services (métiers, fonctionnels, ou applicatifs).

10 : sous-ensemble du processus d'origine qui va être exposé à des organisations externes.

11 : sous-ensemble du processus d'origine à exposition interne.

12 : sous-ensembles de classes appelés objets métiers ou sujets métiers

Grâce à ces étapes nous obtenons un catalogue de services de différents types à savoir les services métiers, les services fonctionnels, les services applicatifs rattachés aux catégories, et les services techniques utilisés par les services applicatifs.

### Choix des technologies des services Web

Dans ce qui suit nous nous intéressons aux services qui ont un sens orienté métier. Les services fonctionnels et métiers exposent à des organisations (internes ou externes) des données et des fonctionnalités, sachant que ces organisations peuvent avoir des plates formes hétérogènes, et sachant aussi qu'il y a une correspondance directe entre le service métier et le service Web, le choix des standards des services Web s'impose.

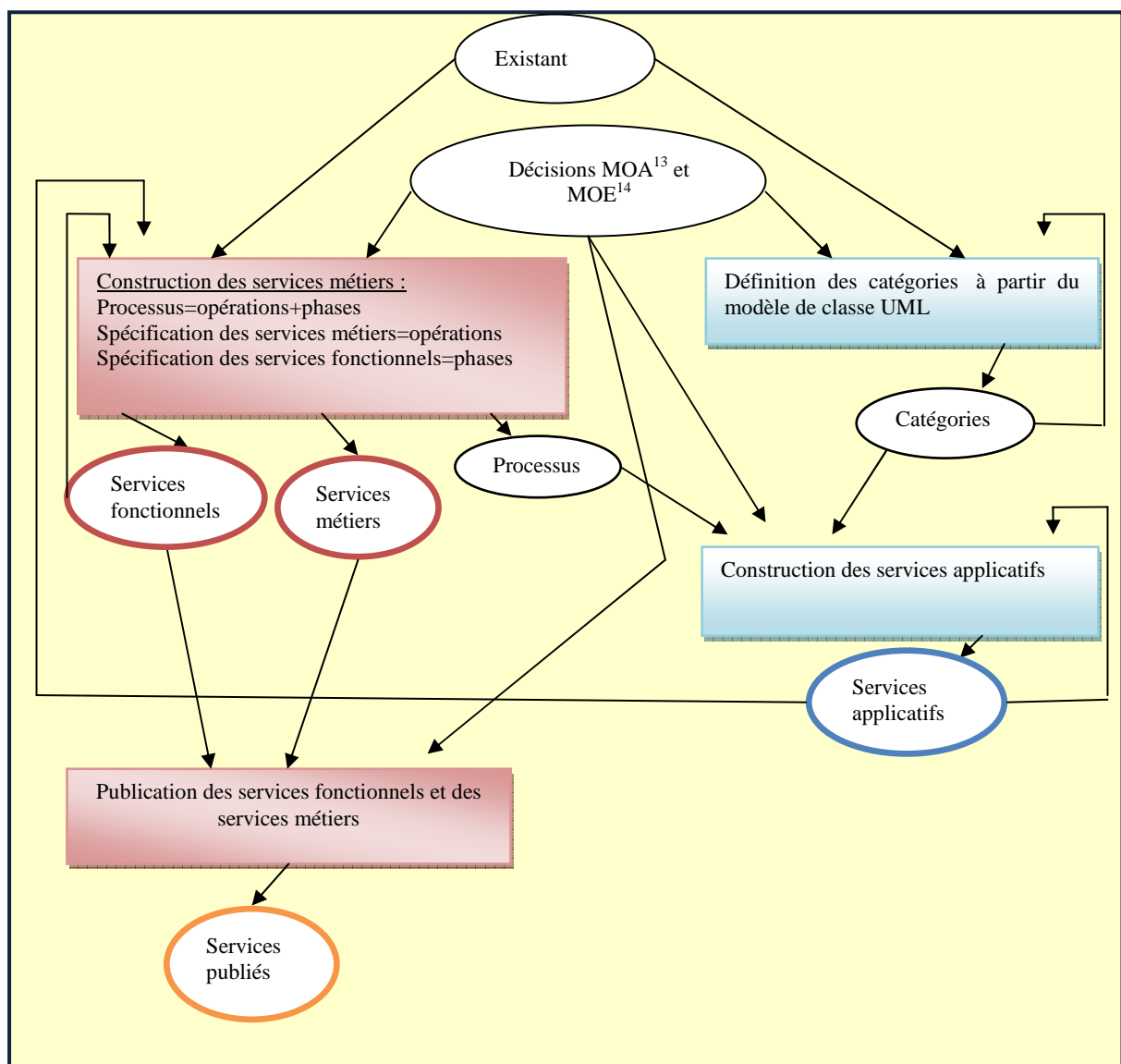


Figure III.3 Approche de construction d'une SOA métier.

13 : Maîtrise d'ouvrage, il s'agit des gens métiers.

14 : Maîtrise d'œuvre, il s'agit des gens informatiques.

### III.3.4. Phase de publication

Les services à exposition interne ou externe doivent faire l'objet d'une publication dans un référentiel adéquat. Nous proposons pour cela d'utiliser la version UDDI3.0 [109] qui présente plusieurs avantages tels que la notion de service privé et de service public et la notion d'annuaires distribués.

Avec ce référentiel les services fonctionnels seront publiés en tant que services Web privés, c'est-à-dire, ils seront utilisés par les organisations internes de l'entreprise. Tandis que, les services métiers seront publiés en tant que services Web publics, c'est-à-dire, ils seront utilisés par les organisations tierces.

Puisque nous avons opté pour une approche orientée services, la publication sera prise en charge par un service Web de publication (voir § III.6.).

### III.4. Construction de l'architecture sémantique

Nous avons traité dans la section précédente une approche de passage des systèmes d'information vers une architecture orientée services. Tous les concepts et techniques utilisés (SOA, MDA, services Web) permettent l'interopérabilité au niveau technique et syntaxique. Mais l'interopérabilité au niveau sémantique est toujours un domaine de recherche actif.

Nous avons vu dans le chapitre précédent que les hétérogénéités sémantiques des services Web concernent leurs propriétés non fonctionnelles, et fonctionnelles y compris les données échangées.

Les hétérogénéités sémantiques à l'origine des conflits entre services Web ne peuvent être résolues que par une description sémantique de ces derniers. La meilleure méthode pour capturer la sémantique consiste en l'utilisation des ontologies (§ II.3).

Notre architecture sémantique est construite selon les trois niveaux d'intégration et d'interopérabilité vus au chapitre précédent, et est issue d'un modèle conceptuel (figure III.4) composé de trois sous modèles:

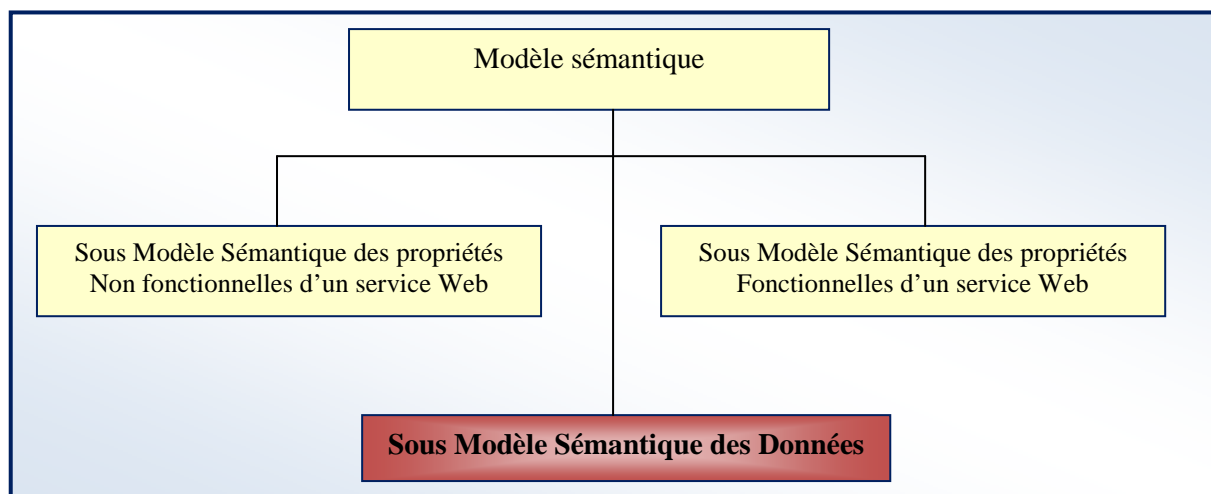


Figure III.4. Sous modèles sémantiques.

**Sous Modèle Sémantique des propriétés Non fonctionnelles (SMSN):** Il décrit la sémantique des propriétés non fonctionnelles des services Web telles que les propriétés concernant la catégorisation, la qualité des services, les séquences d'échange ...etc.

**Sous Modèle Sémantique des propriétés Fonctionnelles (SMSF) :** Il décrit la sémantique des propriétés fonctionnelles telles que les fonctionnalités offertes par le service Web, les protocoles et l'encodage (RPC, Litteral Document...) utilisés.

**Sous Modèle Sémantique des Données (SMSD):** décrit la sémantique des données échangées entre services Web.

Ces sous modèles seront utilisés respectivement dans la médiation sémantique :

- des propriétés non fonctionnelles,
- des propriétés fonctionnelles, et
- des données.

Pour la découverte sémantique, nous proposons l'utilisation du SMSN et du SMSF.

Dans le cadre de nos travaux nous traitons la résolution des conflits dus aux hétérogénéités des données. Par conséquent, nous détaillons le sous modèle sémantique des données, les autres sous modèles feront l'objet de travaux futurs.

### III.4.1. Construction du Sous Modèle Sémantique des Données (SMSD)

Le SMSD sera le modèle de référence pour annoter les données échangées entre services Web. Le SMSD et les annotations sémantiques seront utilisés pour la médiation sémantique des données échangées pour résoudre les conflits qui peuvent exister.

Notre modèle exploite la notion de contexte et d'objet sémantique introduits par Mrissa [34].

#### III.4.1.1. Notion de contexte et d'objet sémantique

**Définition du contexte :** « Le contexte d'une donnée englobe tout élément interne ou externe, relatif à la donnée ou même complètement extérieur, qui est nécessaire à l'interprétation correcte de la donnée » [34].

**Définition de l'objet sémantique [34] :** Une donnée dont la sémantique est explicitement décrite possède :

*Un concept (c)*, qui définit la famille ontologique à laquelle cette donnée appartient, c'est-à-dire la classe abstraite dont cette donnée est une instance.

*Un type (t)*, qui définit le genre de contenu de la donnée (schéma XML simple / complexe).

*Une valeur (v)*, qui est la donnée elle-même, c'est une instance du type de la donnée.

*Un contexte (C)*, qui apporte des précisions sur l'interprétation de la donnée.

Donc d'après [3] un objet sémantique S est défini comme suit :

$$S = (c, v, t, C)$$

Le contexte (C) est lui-même composé d'objets sémantiques appelés *modifieurs*. Ces derniers ont la capacité de modifier la signification de l'unique objet sémantique auquel ils sont associés.

Une description formelle d'un contexte C est :

$$C = \{(c_1, v_1, t_1, C_1), \dots, (c_n, v_n, t_n, C_n)\}, n \in \mathbb{IN}$$

Où  $(c_i, v_i, t_i, C_i)$ ,  $1 \leq i \leq n$ , sont *les modifieurs* qui décrivent les différentes propriétés sémantiques de S. Ceci permet de former des descriptions récursives et de représenter le contexte des objets sémantiques avec des structures arborescentes composées de *modifieurs*. Ces derniers peuvent être statiques ou dynamiques.

*Modifieur statique* : est indépendant des autres modifieurs. Il possède une valeur explicite qui apporte une information quant à l'interprétation de l'objet sémantique.

*Modifieur dynamique* : est dépendant d'un ou de plusieurs autres modifieurs. Il possède une valeur qui peut être déduite, par une fonction ou un ensemble de règles logiques des valeurs prises par d'autres modifieurs appartenant au même contexte.

Exemple : Prenant l'exemple de l'objet sémantique S concernant le concept « prix » avec son contexte (figure III.5).

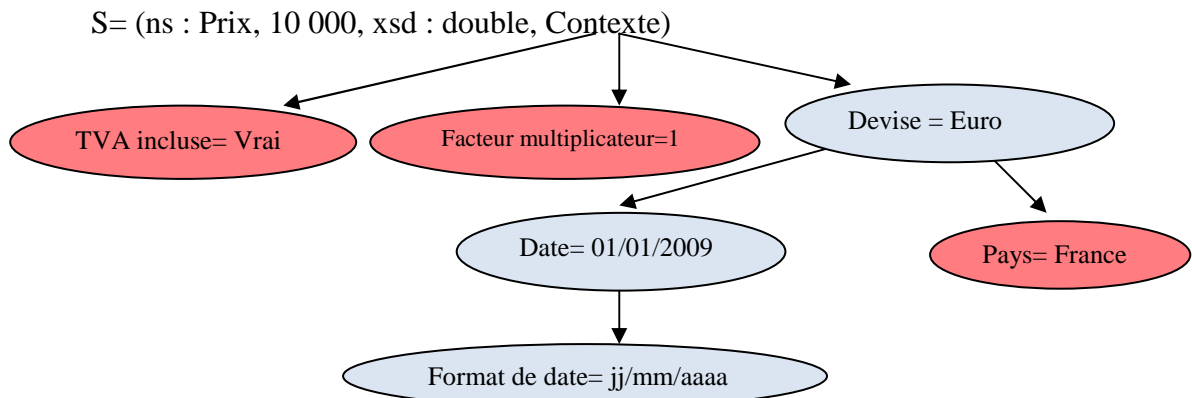


Figure III.5. Exemple d'un objet sémantique et son contexte

Les modifieurs en bleu sont dynamiques.

La valeur du modifieur *Format de date* est déduite à partir de la valeur du modifieur *pays* appartenant au même contexte. La règle logique permettant cette déduction peut être décrite comme suit :

« Si pays= France, alors Format de date= jj/mm/aaaa ».

Cependant les autres modifieurs sont statiques.

#### III.4.1.2. Description du modèle sémantique des données :

Notre sous modèle sémantique (SMSD) inspiré de celui de [34] où il intègre la notion de contexte sous forme d'ontologies contextuelles liée aux concepts de l'ontologie de domaine. Il en diffère dans l'utilisation des ontologies locales et la spécification des valeurs des modifieurs pour pouvoir utiliser le SAWSDL. Il comprend trois types d'ontologies : des ontologies locales, une ontologie de domaine et, des ontologies contextuelles.

*Des ontologies locales de données (OLD)* décrivant la sémantique locale des données au niveau de chaque fournisseur. Ce dernier est le plus apte à décrire la sémantique de ses

propres services, ce qui permet une fiabilité des annotations de la description du service Web, et l'intégration des services existants sans le recours à une description supplémentaire du service.

**Une ontologie de domaine de données (ODD)** commune qui constitue un accord concernant les noms des concepts sémantiques utilisés pour permettre l'interopérabilité. Donc chaque fournisseur doit adhérer à cette ontologie.

**Des ontologies contextuelles de données (OCD)** auxquelles sont associés les concepts de l'ontologie de domaine. Chaque ontologie contextuelle permet de décrire le contexte d'un concept de l'ontologie de domaine. Ceci permet de décrire clairement les hétérogénéités sémantiques. Après avoir adhérer l'ontologie de domaine, le fournisseur doit mettre à jour les ontologies contextuelles associées à sa sémantique locale. Ceci permet d'établir les correspondances avec les représentations des autres fournisseurs.

Les ontologies contextuelles fournissent les vocabulaires qui permettent de spécifier les différentes représentations structurelles et sémantiques des modifieurs. En ce qui concerne les valeurs et les noms des modifieurs statiques, sont spécifiés dans les ontologies locales de données (contrairement à l'approche de Mrissa [34], qui les intègre dans les descriptions des services Web).

### III.4.2. Approche de conception des ontologies

Dans le chapitre précédent nous avons discuté plusieurs approches de conception des ontologies.

Puisque notre modèle SMSD concerne les données, nous adoptons pour la conception des ontologies locales la méthode qui consiste à transformer le modèle de données en une ontologie (§ II.3.5.). En effet, un modèle de classe UML peut être transformé en une ontologie pour le Web sémantique (à l'aide de l'outil Protégé).

En ce qui concerne l'ontologie de domaine qui constitue un accord sur le sens des termes et les ontologies contextuelles, leur développement se base aussi sur le modèle conceptuel, et respecte les étapes suivantes ; d'abord pour l'ontologie de domaine :

- Définir son domaine et sa portée ;
- Etudier la possibilité d'utiliser des ontologies existantes ;
- Enumérer les termes importants dans l'ontologie ainsi que leurs propriétés ;
- Définir les classes (concepts) et leur hiérarchie ;
- Définir les propriétés des classes (attributs), et leurs facettes (types de valeurs, valeurs possibles, nombre de valeurs,...) ; et pour les ontologies contextuelles :
- Etudier pour chaque concept (classe) de l'ontologie de domaine, la possibilité du besoin d'une ontologie contextuelle, en énumérant les concepts qui constituent le contexte (modifieurs statiques, et modifieurs dynamiques).
- Les modifieurs statiques et leurs valeurs existent dans les ontologies locales.

Nos ontologies sont classiques construites à partir du modèle conceptuel des données. Elles sont toutes décrites à l'aide du langage OWL qui offre d'importantes possibilités de description comme les relations d'équivalence, d'identité et de différence entre deux ressources, des propriétés de transitivité...etc. (§II.3.6.).

### III.5. Mécanismes d'annotation sémantique

L'enrichissement sémantique des descriptions des services Web par le modèle de référence est nécessaire pour permettre la découverte et la médiation sémantiques et par conséquent l'interopérabilité sémantique des services Web.

Nous utilisons les mécanismes d'annotation apportés par le SAWSDL pour enrichir sémantiquement les données échangées par les services Web (§ II.5.2.1).

#### Annotations et le contexte

Nous annotons les éléments XSD d'une description WSDL avec les attributs SAWSDL. Cette annotation concerne la valeur (l'instance XML) et son type décrits dans un document WSDL pour les associer à un concept 'c' de l'ontologie locale de données qui comporte les modifieurs statiques et leurs valeurs, ce qui nous donne ainsi, l'objet sémantique S (voir § IV.5). Pour compléter le contexte 'C', on fait appel à des règles logiques permettant d'inférer les valeurs des modifieurs dynamiques.

Ces annotations permettent une éventuelle médiation des services Web. Les annotations des autres éléments du document WSDL seront traitées dans des travaux futurs.

### III.6. Architecture pour l'interopérabilité sémantique des services Web

Notre architecture est basée sur les concepts déjà évoqués, elle permet la publication sémantique, la découverte sémantique, l'invocation et la médiation sémantique des services Web.

Notre processus d'interopérabilité sémantique a pour noyau le service Web broker (courtier) qui orchestre tous les autres services. Ce processus comprend deux phases ; la première est la phase de conception (design time), et la deuxième est la phase d'exécution (Run time).

En mode conception nous avons:

- La construction des services Web (services fonctionnels et services métiers);
- La construction des ontologies ;
- L'annotation des descriptions des services Web prise en charge par un service Web de description sémantique; et
- La publication sémantique des descriptions des services Web, et des ontologies par un service Web de publication.

En mode exécution le service Web broker orchestre les services suivants :

- Les services Web de découverte sémantique;
- Les services Web de médiation sémantique; et
- Les services Web d'exécution.

Les étapes de ce processus d'interopérabilité sont décrites par la figure III.6 et les sections qui suivent.

### III.6.1. Rôle du service Web broker

Notre processus d'interopérabilité est mis en œuvre par plusieurs services Web broker, ils permettent de superviser et orchestrer les autres services d'interopérabilité, ainsi, ils assurent tous ensemble un travail de coopération. Chaque service broker expose ses services d'interopérabilité (Services Web : de publication, de découverte, de médiation et d'exécution). Un service Web broker local peut être exposé par le service Web broker de domaine (figure III.7).

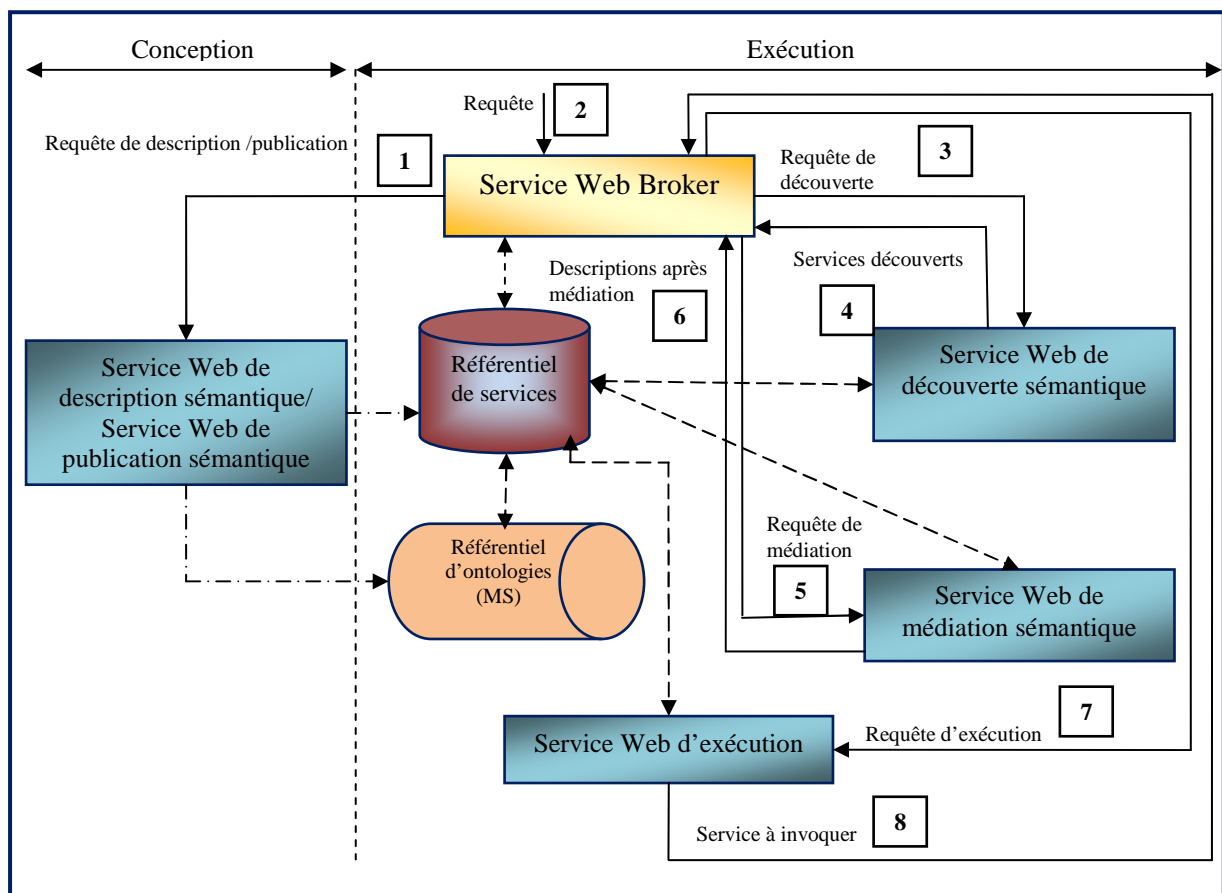


Figure III. 6. Processus général d'interopérabilité.

Le service Web broker utilise le référentiel des services qui comporte des liens (annotations) au référentiel des ontologies. Il supervise le processus d'interopérabilité en utilisant un ensemble de règles et de requêtes.

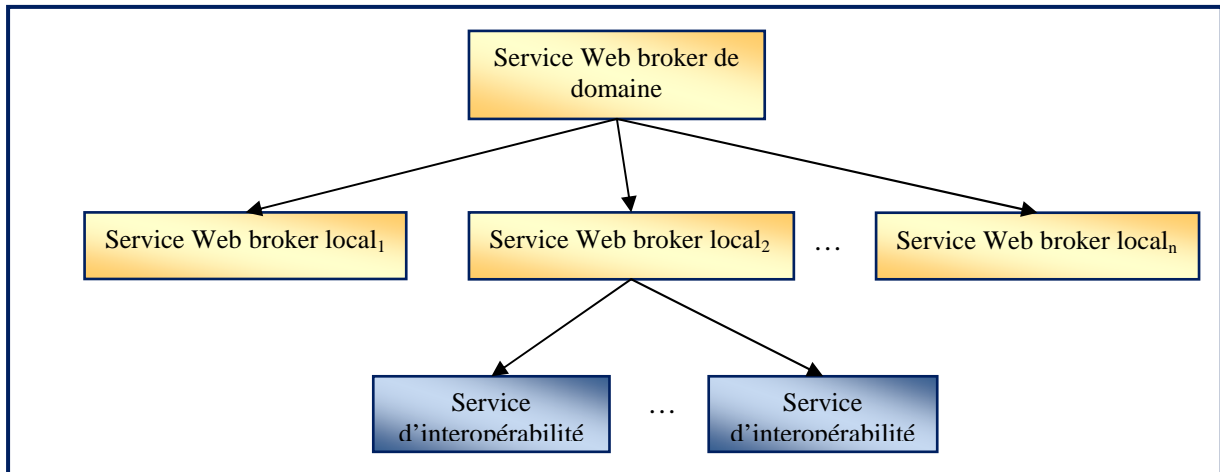


Figure III.7. Architecture des services Web broker.

### III.6.2. Publication sémantique

La publication concerne les ontologies, ainsi que les services Web. Une fois les ontologies construites, et les descriptions sont annotées vient une étape importante qui est la publication. Nous procédons à la publication des ontologies dans un référentiel spécialisé sous forme de bases de données (MySQL).

Et pour la publication des descriptions des services Web nous utilisons le référentiel UDDI3.0 comme nous l'avons montré dans la section III.3. Avec cette solution nous pouvons soit annoter d'abord les fichiers WSDL puis les publier, soit publier les descriptions de services puis leurs faire une mise à jour pour les annoter. Ce qui permet à des architectures orientées services existantes d'adhérer à notre architecture.

Cette publication est prise en charge par le service Web de publication sémantique, qui s'active à la réception du service Web broker de la requête suivante :

*Request (# Publish, FilePublication)*

Où FilePublication désigne le fichier à publier et qui peut être le fichier XML de la description du service Web annotée ou non, ou le fichier OWL (figure III.8).

Donc pour les descriptions des services Web on peut les publier annotées ou non encore.

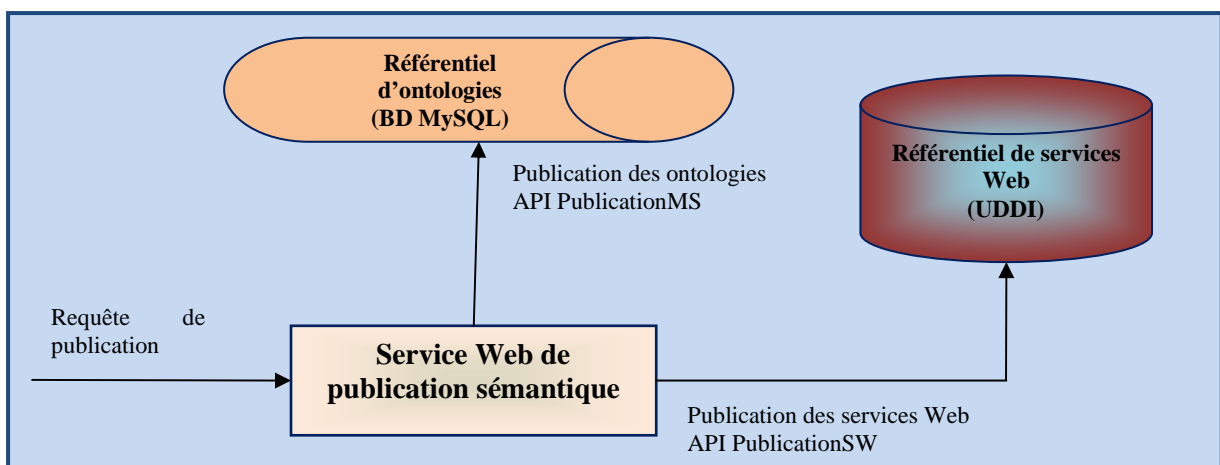


Figure III.8. Principe de publication sémantique.

### III.6.3. Découverte sémantique

La découverte sémantique des services publiés est dynamique, et elle est prise en charge par un service Web de découverte, dans le but de réaliser une requête émanant d'un utilisateur ou d'un autre service Web et qui arrive au service Web de découverte par l'intermédiaire du service Web broker. Pour la recherche et la sélection des services Web publiés, les services Web de découverte utilisent des algorithmes et un moteur d'inférence qui exploite les sous modèles sémantiques des propriétés fonctionnelles SMSF et des propriétés non fonctionnelles SMSN (non décrits dans le cadre de cette thèse).

### III.6.4. Médiation sémantique

La médiation sémantique concerne les services sélectionnés, et elle est prise en charge par un service de médiation, qui s'active à la réception du service Web broker de la requête suivante :

*Request (#Mediate, MediateType, DescriptionWS1, DescriptionWS2) ;*

Où :

'MediateType' désigne le type de médiation (des propriétés fonctionnelles/ des propriétés non fonctionnelles/ des données).

DescriptionWS<sub>i</sub> désigne les URLs des descriptions WSDL.

Nous détaillons la médiation des données dans la section (§ III. 7.).

### III.6.5. Exécution des services

Une fois les services sélectionnés, et après leur médiation, ils seront exécutés par des services d'exécution. Le service d'exécution est activable à la réception du service Web broker de la requête suivante :

*Request(# Execute, WS)*

Où WS est le service concerné par l'exécution.

Les services exécutés peuvent à leur tour invoquer le service Web broker en formulant une requête.

## III.7. Description de la médiation des services Web

Cette médiation constitue le noyau de notre architecture, elle est basée sur trois niveaux de médiation :

- la médiation des propriétés non fonctionnelles ;
- la médiation des propriétés fonctionnelles ; et
- la médiation des données.

Chaque niveau inclut l'autre, et le niveau le plus bas concerne la médiation des données.

Chaque niveau de médiation est pris en charge par un service Web médiateur, de ce fait nous avons :

- Un service Web de médiation des propriétés non fonctionnelles ;
- Un service Web de médiation des propriétés fonctionnelles ; et
- Un service Web de médiation des données.

Dans le cadre de notre travail de recherche nous nous intéressons à la médiation des données échangées entre services Web et, les autres niveaux de médiation seront traités ultérieurement.

Lors de l'échange de données entre deux services Web des conflits peuvent apparaître à cause des hétérogénéités sémantiques. En effet, la sémantique des données envoyées par le service Web émetteur peut être différente de celle attendue par le service Web destinataire. De ce fait, la médiation sémantique est nécessaire pour garantir des échanges de données consistants et permettre l'interopérabilité des services Web.

Pour résoudre les conflits sémantiques lors de l'échange de données, notre architecture exploite le modèle sémantique de référence SMSD et les annotations SAWSDL des descriptions des services Web.

Dans ce qui suit nous présentons notre principe de médiation de données, l'architecture de médiation et, les mappings et fonctions de conversion utilisées.

### III.7.1. Principe de médiation sémantique des données

Notre approche est inspirée de celle de [34], mais elle diffère de cette dernière dans l'utilisation du SAWSDL, le modèle sémantique et les transformations de schémas.

Notre processus de médiation comprend les trois grandes étapes suivantes (figure III.9) :

**1. Etape d'abstraction :** permet de traduire une donnée XML source/cible (définition de type ou élément) annotée, en un ou plusieurs concepts appartenant au modèle sémantique SMSD (OLD source/cible). Cette abstraction est prise en charge par les mappings de schémas dans la spécification SAWSDL via l'attribut *liftingSchemaMapping*, il s'agit des transformations syntaxiques-sémantiques.

**2. Etape de transformation sémantique :** permet de transformer sémantiquement tout concept de l'ontologie source en un autre concept ontologique équivalent appartenant à une ontologie cible. Cette transformation est réalisée grâce aux mappings sémantiques et exploite les ontologies du SMSD (locales, de domaine et contextuelles). Il s'agit des transformations sémantiques-sémantiques.

**3. Etape de concrétisation syntaxique :** permet de transformer les concepts ontologiques résultants appartenant au niveau sémantique cible au format de données XML cible du service concerné. Cette étape est prise en compte par la spécification SAWSDL via l'attribut *loweringSchemaMapping*, il s'agit des transformations sémantiques-syntaxiques.

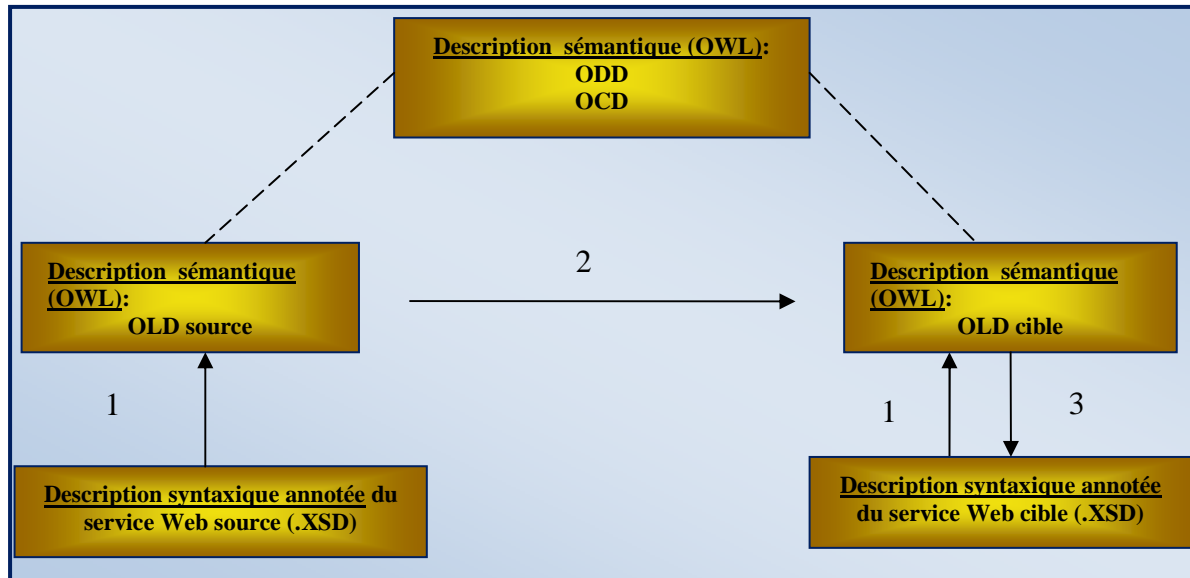


Figure III.9. Processus de médiation des données.

### III.7.2. Description de l'architecture de médiation

Nous avons vu dans la section précédente que la médiation des données échangées entre deux services Web est prise en charge par un service Web médiateur inséré entre le service Web source et le service Web cible (figure III.10.).

Après extraction des annotations du WSDL (source/cible) et les mappings syntaxiques (*liftingSchemaMapping*), vient l'extraction des modifieurs statiques et de leurs valeurs des ontologies locales. Grâce à l'ontologie contextuelle, deux autres ontologies temporaires de contextes seront construites, la première concerne le service Web source, et la deuxième concerne le service Web cible. Le service Web médiateur communique alors avec un moteur d'inférence (exemple Jena) pour, d'une part déduire les valeurs des modifieurs dynamiques, et d'autre part effectuer les correspondances sémantiques (exemple relation de subsomption...) entre les concepts des ontologies contextuelles temporaires (source et cible). Puis vient l'étape de conversion de la valeur du concept source vers la valeur du concept cible. Le résultat sera transformé au format de données cible (*loweringSchemaMapping*).

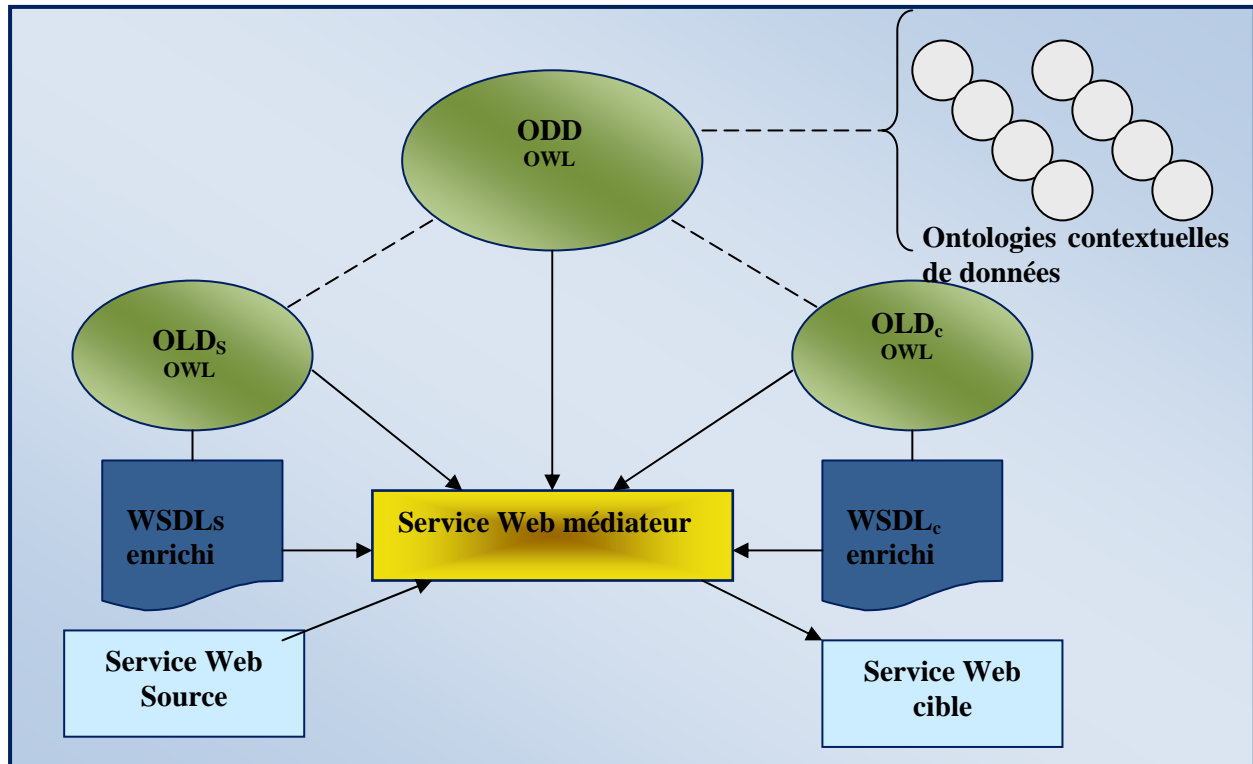


Figure III. 10. Architecture conceptuelle de la médiation proposée.

### III.7.3. Fonctions de conversion et de transformation utilisées

Selon le principe de notre médiation de données plusieurs types de transformations sont définis à savoir :

- ✓ *Les transformations syntaxiques\_sémantiques*, en utilisant les annotations SAWSDL « liftingSchemaMapping » associant un type ou élément du schéma XML à un mapping vers le modèle sémantique (en utilisant le XSLT), définissant comment une instance du document XML est transformée vers un ou plusieurs concepts du modèle sémantique.
- ✓ *Les transformations sémantiques\_syntaxiques*. en utilisant les annotations SAWSDL « loweringSchemaMapping » associant un concept du modèle sémantique à un mapping vers une donnée du document XML (en utilisant le SPARQL suivi du XSLT), définissant comment un concept sémantique est transformé en une donnée XML.
- ✓ *Les transformations sémantiques\_sémantiques* en plus des mappings classiques d'ontologies nous utilisons des fonctions de conversion des valeurs des concepts grâce aux valeurs des modificateurs de l'ontologie contextuelle temporaire source et des valeurs des modificateurs de l'ontologie contextuelle temporaire cible.

### **III.8. Conclusion**

Nous avons présenté dans ce chapitre notre architecture pour l'interopérabilité des systèmes d'information qui est une architecture structurée en couches :

- La couche de l'architecture orientée services.
- La couche sémantique et enrichissement des services.
- La couche d'infrastructure pour l'interopérabilité des services Web.

Nous avons préconisé l'utilisation les concepts de SOA, MDA et services Web pour le passage des systèmes d'information aux architectures orientées services.

Nous avons construit un modèle sémantique avec lequel nous avons enrichi les descriptions des services Web. Nous avons détaillé le sous modèle des données qui est construit par trois types d'ontologies : des ontologies locales qui véhiculent la sémantique locale de chaque fournisseur, une ontologie de domaine qui présente un accord des fournisseurs sur des concepts communs, et des ontologies contextuelles chacune associée à un concept de l'ontologie de domaine ce qui permet de résoudre les conflits liés aux contextes des utilisateurs lors de la mise à jour des ontologies contextuelles.

Nous avons présenté le processus de coopération qui est pris en charge par le service Web broker. Ce dernier orchestre les autres services Web d'interopérabilité sémantique: de publication, de découverte, de médiation et d'exécution.

Nous avons aussi évoqué la médiation des données qui est une partie des autres médiations à savoir la médiation des propriétés fonctionnelles et celle des propriétés non fonctionnelles.

Nous allons dans le chapitre suivant présenter une étude de cas à travers laquelle nous illustrons notre approche et l'architecture qui en découle.



Chapitre IV  
Etude de cas

## IV.1. Introduction

Dans le chapitre précédent nous avons présenté notre architecture pour l'interopérabilité des systèmes d'information des entreprises. Pour arriver à cette architecture nous avons évoqué plusieurs étapes que nous allons appliquer à une étude de cas qui fera l'objet de ce chapitre.

Pour illustrer le passage des systèmes d'information vers les architectures orientées services, nous prenons comme exemple une seule entreprise sur laquelle on appliquera l'approche de la section 3 du chapitre III.

Nous montrerons la nécessité et la faisabilité de l'interopérabilité dans l'architecture présentée dans la section 4 du chapitre III, nous traitons les relations intra et inter entreprise.

Une fois les services métiers définis, nous construisons les ontologies du sous modèle conceptuel de données. Avec lesquelles nous enrichissons les descriptions de deux services Web exemples pour illustrer la médiation sémantique des données échangées.

## IV.2. Enoncé de l'étude de cas

Nous allons aborder le système d'information d'une entreprise publique économique et autonome en l'occurrence la SETA (Société d'Etudes Techniques d'Annaba). Cette entreprise dont le siège social est fixé à Annaba est organisée sous forme de départements et antennes répartis sur l'est du territoire national.

L'activité de l'entreprise se caractérise essentiellement par le contrôle et le suivi de chantiers, l'expertise et les études:

- de conception géométrique et structurale des routes,
- d'aménagement routier, ferroviaire, aéroportuaire et portuaire,
- de conception et de calcul des ouvrages d'art,
- géologiques et géotechnique,
- hydrauliques concernant les ouvrages de travaux publics,
- de V.R.D pour les nouvelles zones à urbaniser de l'habitat et de l'industrie.

*Principaux clients concernés* : Toute prestation est généralement effectuée pour le compte de l'état ainsi que des administrations locales (Wilaya, APC, et autres).

*Autres partenaires* : La SETA développe de multiples relations avec par exemple les laboratoires d'analyses et fournisseurs nationaux et internationaux, les banques, les assurances ...etc.

Pour réaliser ces prestations l'entreprise et son client passent un contrat qui peut être selon son montant une convention ou un marché, s'il y a des travaux supplémentaires un avenant est établi.

Le système d'information comprend entre autres les applications suivantes (figure IV.1) :

1. **Calcul des prix de vente unitaires (Calcul-P)** ; application relevant de la comptabilité des coûts qui grâce à des informations relatives aux coûts de la main d'œuvre et des matériaux, à l'amortissement des matériels et aux montants de la sous-traitance ... (flux1) permet l'établissement du catalogue des prix de vente unitaires qui sera utilisé par l'application *Suivi- CTR* (flux2).
2. **Suivi des contrats (Suivi-CTR)** ; application qui prend en charge les informations sur les contrats (flux3), sur les situations évolutives (flux4) et la restitution des données d'évolution physique (flux5) qui seront utilisées par l'application *Facturation-recouvrement*. Elle utilise le catalogue des prix de vente unitaires émanant de l'application *Calcul-P* (flux2) pour établir les offres (flux3). Dans le cas où on a besoin d'analyses une relation externe est établie avec des laboratoires spécialisés (flux6), il s'agit d'une sous-traitance.
3. **Facturation-recouvrement (Fact-Recouv)** ; application qui se charge du suivi de la facturation et du recouvrement. Elle établit les factures (flux7) grâce aux informations d'évolution physique émanant de l'application *Suivi des contrats* (flux5), et produit périodiquement un état financier (flux5). Cette application développe aussi des relations avec les banques et autres partenaires (flux8).

La figure IV.1 laisse apparaître l'architecture applicative future simplifiée de notre système d'information et son articulation avec les systèmes d'information de ses partenaires.

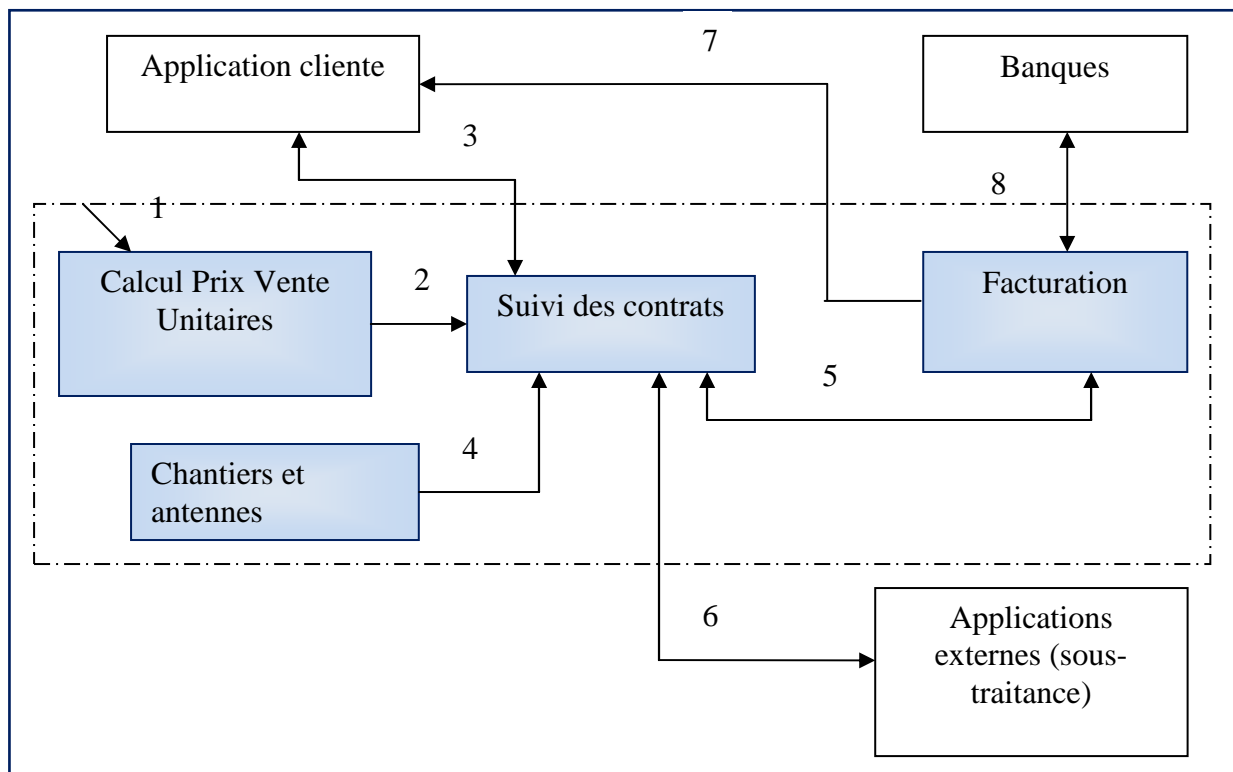


Figure IV 1. Articulation des systèmes d'information (SETA et ses partenaires).

L'objectif principal de ce travail est de faire inter-opérer ces applications. Il s'agit d'une interopérabilité intra-entreprise et inter-entreprise.

Pour cela nous allons appliquer les étapes de notre approche (chap. III), que nous rappelons ci-dessous :

1. La construction d'une architecture orientée services.
2. La construction d'un modèle sémantique décrivant l'architecture orientée services déjà construite.
3. L'annotation de l'architecture orientée services à l'aide du modèle sémantique.
4. L'architecture orientée services Web sémantiques pour l'interopérabilité des systèmes d'information.

### IV 3. Construction de l'architecture orientée services de l'entreprise SETA

Comme nous l'avons montré dans le chapitre III notre approche de passage vers une architecture orientée services est principalement hybride car elle exploite l'approche ascendante et descendante, ceci nous permet de bénéficier des avantages des deux approches. Nous utilisons l'approche descendante pour les applications encore manuelles bien sûr en exploitant les composants ou services existants (notion de réutilisation introduite par l'approche ascendante).

Pour les applications existantes, nous utilisons l'approche ascendante par l'encapsulation des composants par les services applicatifs qui peuvent être exposés par des services fonctionnels (à exposition interne). Ces derniers peuvent être promus en services métiers (à exposition externe). En cas de besoin nous ferons appel à la modélisation (introduite par l'approche descendante).

Selon la typologie des services obtenus, nous avons deux niveaux, la construction de la SOA applicative et la construction de la SOA métier.

#### IV.3.1. Construction de la SOA applicative

En appliquant notre approche hybride au système d'information simplifié de la SETA nous obtenons les services applicatifs suivants :

Application	Composant applicatif	Service applicatif
Calcul- P	Calcul du montant de la sous-traitance	SA_Calcul_P_Mt_ST
	Calcul des coûts directs et indirects	SA_Calcul_P_Coût_Direct
	Définition des prix de vente unitaires	SA_Calcul_P_Déf_PVU
	Etablissement des offres	SA_S_CTR_Etab_Offres
	Etablissement des contrats	SA_S_CTR_Etab_Contrats
	Lancement des travaux	SA_S_CTR_Lanc_Travaux

Suivi-CTR	Affectation des équipes	
	Suivi des équipes par projet	SA_S_CTR_Etab_Offres
	Suivi de l'évolution physique des projets	SA_S_CTR_Suivi_Equipes
	Suivi de la sous-traitance des analyses	SA_S_CTR_Suivi_Sous_Trt
Fact-Recouv	Etablissement des factures	SA_Fact_Recouv_Facturation
	Suivi des paiements	SA_Fact_Recouv_Paiement
	Etablissement de l'état de créances	SA_Fact_Recouv_Créances
	Clôture des contrats	SA_Fact_Recouv_Clôture

Tableau IV.1. Services applicatifs de la SETA

### IV.3.2. Construction de la SOA métier

Sur La base des services applicatifs obtenus, et suite à une décision de la maîtrise d'ouvrage de la SETA (les gens métiers), et à l'application de notre approche, des services métiers se sont dégagés. Ils sont bien entendu de deux types : les services fonctionnels et les services métiers. Tous nos services fonctionnels et métiers seront implémentés par des services Web.

#### IV.3.2.1. Définition des services fonctionnels

Les services fonctionnels (services Web) à exposition interne sont (TableauIV.2.) : SW\_Calcul\_P\_PVU, SW\_Calcul\_P\_Sous\_Traitance, SW\_S\_CTR\_Etab\_Contrats, SW\_S\_CTR\_Suivi\_Travaux, SF\_S\_CTR\_Suivi\_Equipes, SF\_S\_CTR\_Suivi\_Sous\_Trt, SF\_Fact\_Recouv\_Facturation, SF\_Fact\_Recouv\_Créances, SW\_Fact\_Recouv\_paiements

Application	Service Fonctionnel	Entrées	Sorties
Calcul- P	SW_Calcul_P_Sous_Traitance	Devis Laboratoire	Montants avec bénéfiques
	SW_Calcul_P_PVU	Coût directs Coûts indirects Montants avec bénéfiques	Catalogue des PVU
Suivi-CTR	SW_S_CTR_Offres	Cahier des charges (cas appel d'offre)/ Bon de commande Catalogue des PVU	Offre
	SW_S_CTR_Etab_Contrats	Offre accepté	Contrat
	SW_S_CTR_Suivi_Travaux	Date de lancement Matériel affecté Situation d'évolution	Etat d'évolution physique

	SW_S_CTR_Suivi_Equipes	Equipes affectées Pointage journalier	Mise à jour de la base de données
	SW_S_CTR_Suivi_Sous_Trt	Résultats d'analyses	Mise à jour de la base
Fact- Recouv	SW_Fact_Recouv_Facturation	Etat d'évolution physique	Facture
	SW_Fact_Recouv _paiements	Pièce de paiement (Chèque/Avis de crédit)	Mise à jour de la base
	SW_Fact_Recouv _Créances	Demande/période	Etat de créances

Tableau IV.2. Services fonctionnels de la SETA

Comme nous le constatons dans ce tableau les sorties d'un service fonctionnel (SW) sont des entrées de l'autre. Ceci montre bien le besoin des entreprises à faire interopérer des applications.

Les services fonctionnels assurent l'interopérabilité intra entreprise c'est à dire la coopération des applications internes (qui peuvent avoir des plates formes différentes) du système d'information de la SETA.

#### IV.3.2.1. Définition des services Métiers

Les services métiers seront exposés aux partenaires de la SETA, en revanche, cette dernière va bénéficier des services offerts par ses partenaires, il s'agit donc d'une coopération de systèmes d'information.

Nous supposons que les partenaires de la SETA ont procédé à ce passage vers le SOA, et ils offrent eux aussi des services métiers.

Nous résumons dans le tableau IV.3 quelques services métiers (SW) offerts par la SETA, un laboratoire Y et une banque X.

Acteur	Service Métier	Entrées	Sorties
SETA	SW_SETA_Offres	Informations Commande	Offre
	SW_SETA_ConsultatinFactures	Code de login Autres informations d'accès	Factures
Laboratoire Y	SW_LABY_Devis	Demande d'analyses	Devis
	SW_LABY_Résultats	Demande résultats	Résultats
Banque X	SW_BNQX_Avis_Crédit	Informations d'accès et Demande avis de crédit	Facture

Tableau IV.3. Services métiers de la SETA et ses partenaires.

Les services métiers assurent l'interopérabilité inter entreprise c'est à dire la coopération des systèmes d'information dont les plates formes sont hétérogènes.

### IV.3.3. Publication des services Web

Comme nous l'avons indiqué au chapitre précédent, les services fonctionnels feront l'objet d'une publication interne à l'entreprise et les services métiers feront l'objet d'une publication externe. Ceci est possible en utilisant un seul référentiel qui publie les services fonctionnels en tant que services privés et les services métiers en tant que services publics. Nous utilisons à cette fin l'UDDI3.0.

Rappelons que cette publication concerne les wsdl annotés ou non. Pour le besoin de notre architecture tous les wsdl doivent être annotés, d'où la nécessité de mettre à jour les descriptions non encore enrichies de l'information sémantique (§ IV.5.).

### IV.3.4. Hétérogénéités sémantiques

Les services Web décrits ci-dessus assurent l'interopérabilité des systèmes d'information de la SETA et ses partenaires. Mais comme nous l'avons signalé aux précédents chapitres cette interopérabilité est au niveau technique et syntaxique grâce aux technologies utilisées (MDA, SOA, et services Web).

Tous ces services Web (fonctionnels ou métiers) peuvent présenter des hétérogénéités sémantiques tant au niveau non fonctionnel (non décrit dans le document WSDL), qu'au niveau fonctionnel notamment au niveau des données échangées.

#### Exemple d'hétérogénéités sémantiques :

Nous prenons un service Web du laboratoire Y et un autre de la SETA. Ces deux services échangent des données comme suit:

- 1) SW\_LABY\_Devis : ce service Web fournit les prix offerts par le laboratoire Y, lesquels seront des entrées pour le service Web de la SETA suivant.
- 2) SW\_Calcul\_P\_Sous\_Traitance : qui grâce aux prix offerts par le service précédent, il calcule le montant de la sous-traitance en incluant le bénéfice.

Seulement le laboratoire Y peut être étranger (Japonais) et son service Web présente donc des données non conformes à celles exigées par le service Web de la SETA. A titre d'exemple nous présentons les sorties de l'un et les entrées de l'autre pour monter ces hétérogénéités :

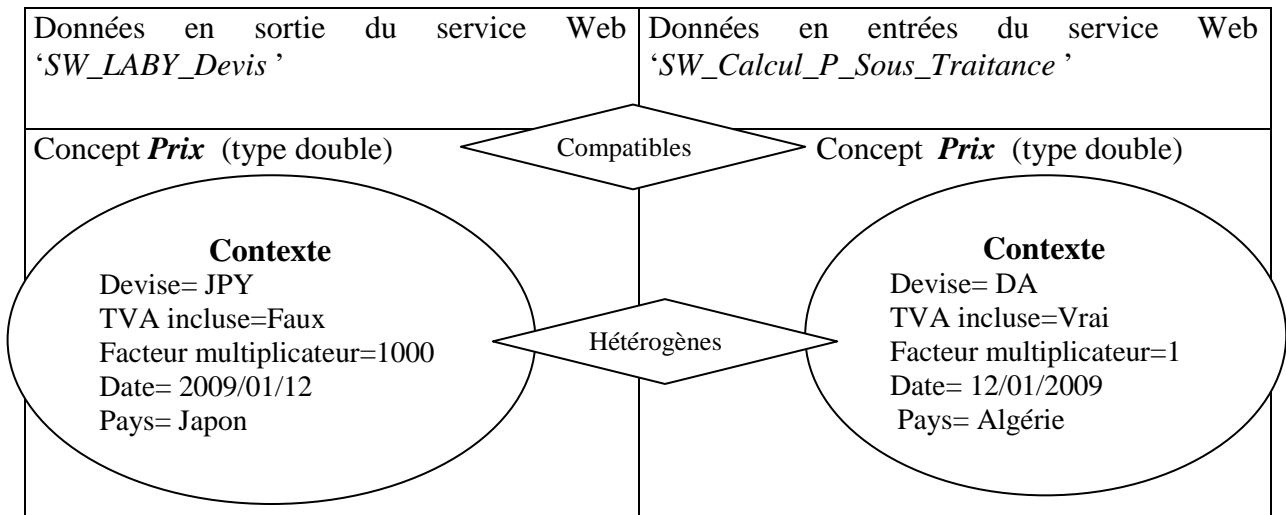


Tableau IV 4. Hétérogénéités sémantiques des données des deux services Web exemples.

Nous remarquons que ces hétérogénéités touchent le niveau structurel et sémantique des données échangées.

Pour résoudre les conflits causés par ce type d'hétérogénéités, une description et médiation sémantiques sont mises en place par notre architecture.

#### IV 4. Construction du modèle sémantique

Ce modèle est composé de trois sous modèles sémantiques (§ III.4.) :

1. Sous modèle sémantique des propriétés non fonctionnelles (SMSN).
2. Sous modèle sémantique des propriétés fonctionnelles (SMSF).
3. Sous modèle sémantique des Données (SMSD).

Dans le cadre de cette recherche nous avons détaillé la construction du troisième sous modèle SMSD, pour traiter la médiation des données.

##### IV 4.1. Construction du sous modèle sémantique de données (SMSD)

Notre sous modèle sémantique se compose de trois types d'ontologies à savoir :

- a. Des ontologies locales de données.
- b. Une ontologie de domaine de données.
- c. Des ontologies contextuelles de données.

Dans ce qui suit nous allons présenter des extraits de ces ontologies que nous avons créés à l'aide de l'outil Protégé (figure IV.2.).

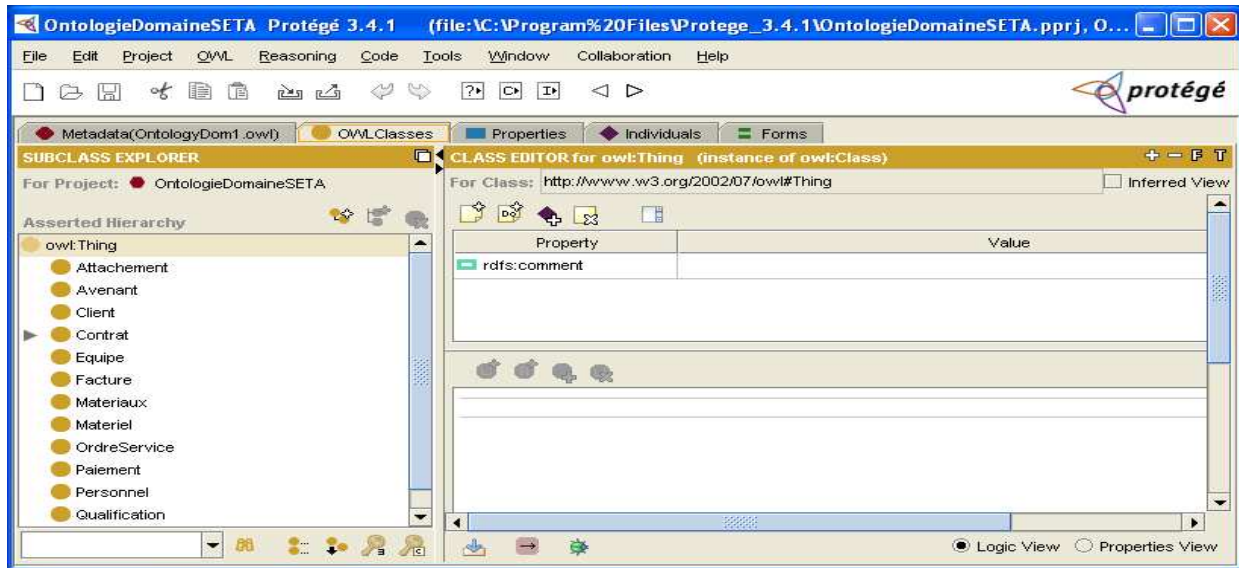


Figure IV.2 . Outil Protégé 2000.

Les ontologies locales de données (figures IV.3 et IV.4) véhiculent la sémantique locale (de la SETA, et du laboratoire) et spécifient les valeurs des modificateurs statiques.

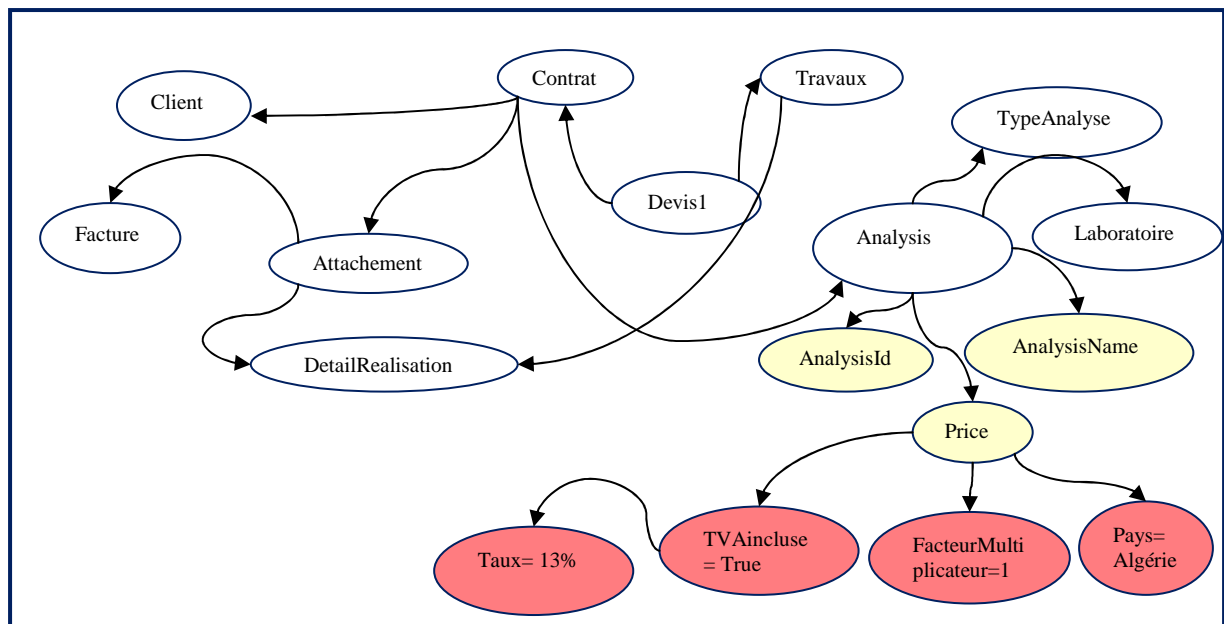


Figure IV. 3. Extrait de l'ontologie locale de données (OLD) de la SETA avec les concepts en rouge qui représentent les modificateurs statiques (contexte) du concept 'Price'.

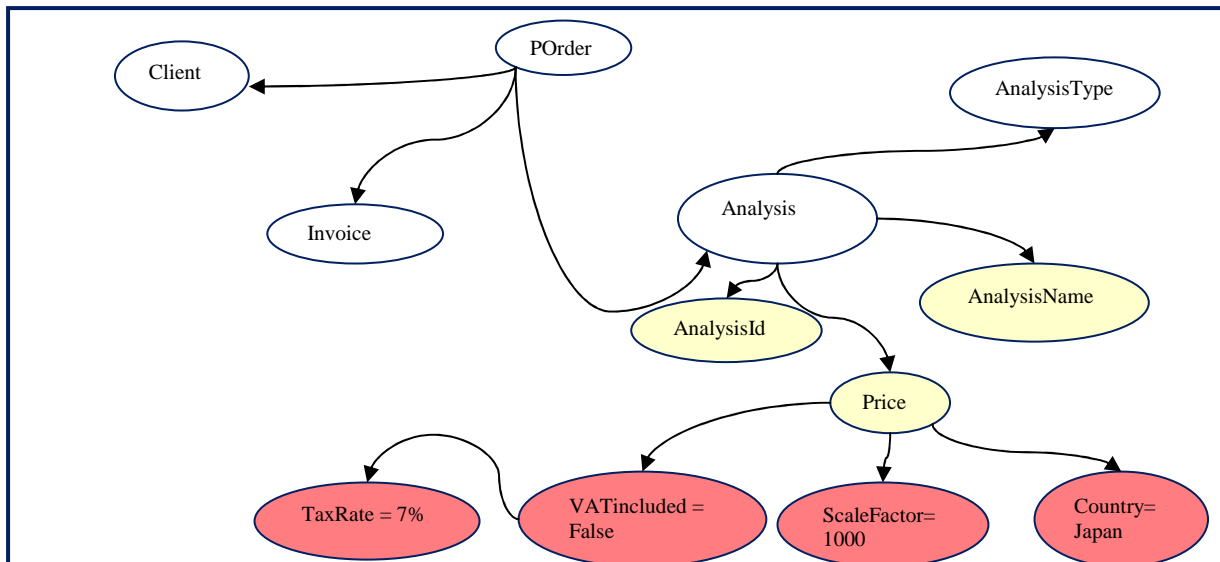


Figure IV.4. Extrait de l'ontologie locale de données (OLD) du laboratoire avec les concepts en rouge qui représentent les modificateurs statiques (contexte) du concept 'Price'.

Chaque fournisseur après avoir adhérer l'ontologie de domaine doit mettre à jour les ontologies contextuelles (figure IV.5) grâce à sa sémantique locale, il va établir par exemple les différentes relations d'équivalence, de subsumption ...etc. (ex. relation d'équivalence entre les concepts 'pays' et 'country'). Rappelons que l'ontologie contextuelle spécifie la structure et la sémantique des modificateurs.

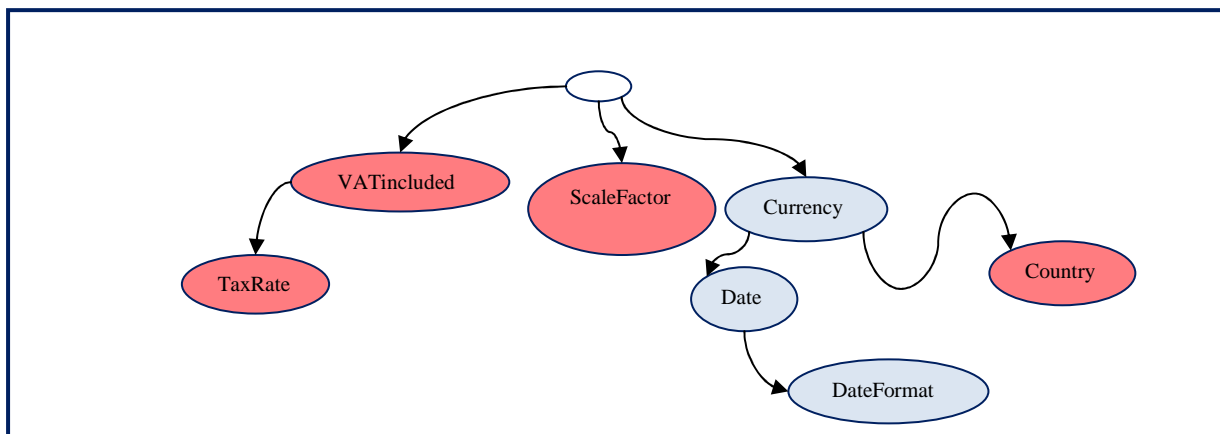


Figure IV.5. Ontologie contextuelle de données (OCD) associée au concept 'Price' de l'ontologie de domaine de données (ODD).

#### IV 4.2. Publication du modèle sémantique des données

Cette étape est primordiale pour les services de médiation sémantique des données. Une fois que les ontologies sont construites, nous devons alors les publier. Pour cela un référentiel commun est établi sous forme d'une base de données relationnelle en utilisant MySQL.

### IV 5. Annotation des descriptions des services par le SMSD

Sur la base du modèle sémantique obtenu, et grâce aux mécanismes d’annotation décrits au chapitre précédent, nous procédons à l’enrichissement des schémas XML de la description des services Web de l’exemple de la section IV.3.4.

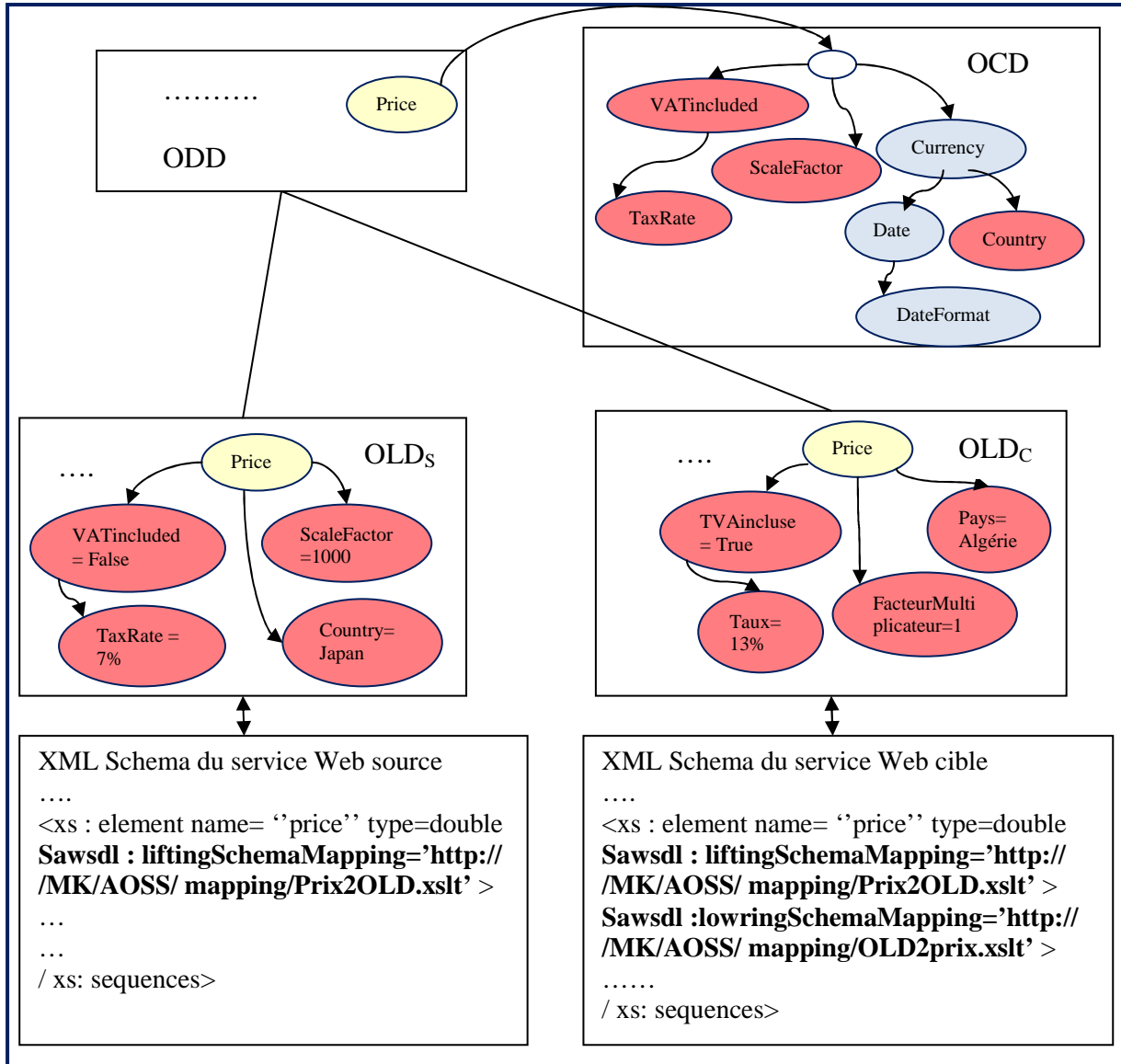


Figure IV .6. Annotation des descriptions relatives aux services Web de l’exemple.

Comme le montre la figure IV .6, des attributs SAWSDL sont ajoutés aux descriptions des services Web source et service Web cible, il s’agit des schémas XML.

Ces annotations (en gras) font référence aux définitions de mappings des données XML de et vers les ontologies locales de données dans lesquelles nous trouvons les modifieurs statiques et leurs valeurs liés au concept concerné (price).

Suite à cette annotation vient l’étape de publication des descriptions annotées des services Web (voir section IV3.3.)

## IV 6. Description de la médiation sémantique de données

La médiation sémantique des données échangées dans l'exemple est assurée par un service Web qui utilise les référentiels des ontologies, les descriptions annotées, et les définitions des mappings syntaxiques, et fait appel à un moteur d'inférence et une base de règles et de fonctions de conversion pour exécuter la médiation selon les étapes suivantes (figure IV.8):

- Abstraction sémantique : Le service Web médiateur télécharge les fichiers wsdl des SW source et SW cible, et appelle les définitions de mappings relatives aux annotations `liftingSchemaMapping` pour exécuter les transformations syntaxiques-sémantiques, des schémas XML vers les ontologies locales.
- Transformation sémantique : comprend plusieurs opérations:
  - o Correspondances des concepts utilisés par les deux ontologies locales grâce à l'ontologie de domaine.
  - o A partir de l'ontologie contextuelle associée au concept de l'ontologie de domaine, le service Web médiateur construit une ontologie contextuelle temporaire pour chacun des services Web source et cible. Les valeurs des modifieurs statiques sont identifiées à partir des ontologies locales (figure IV.7).
  - o Déduction des valeurs des modifieurs dynamiques grâce aux règles pour compléter les ontologies contextuelles temporaires.
  - o Conversion des valeurs grâce aux fonctions de conversion.

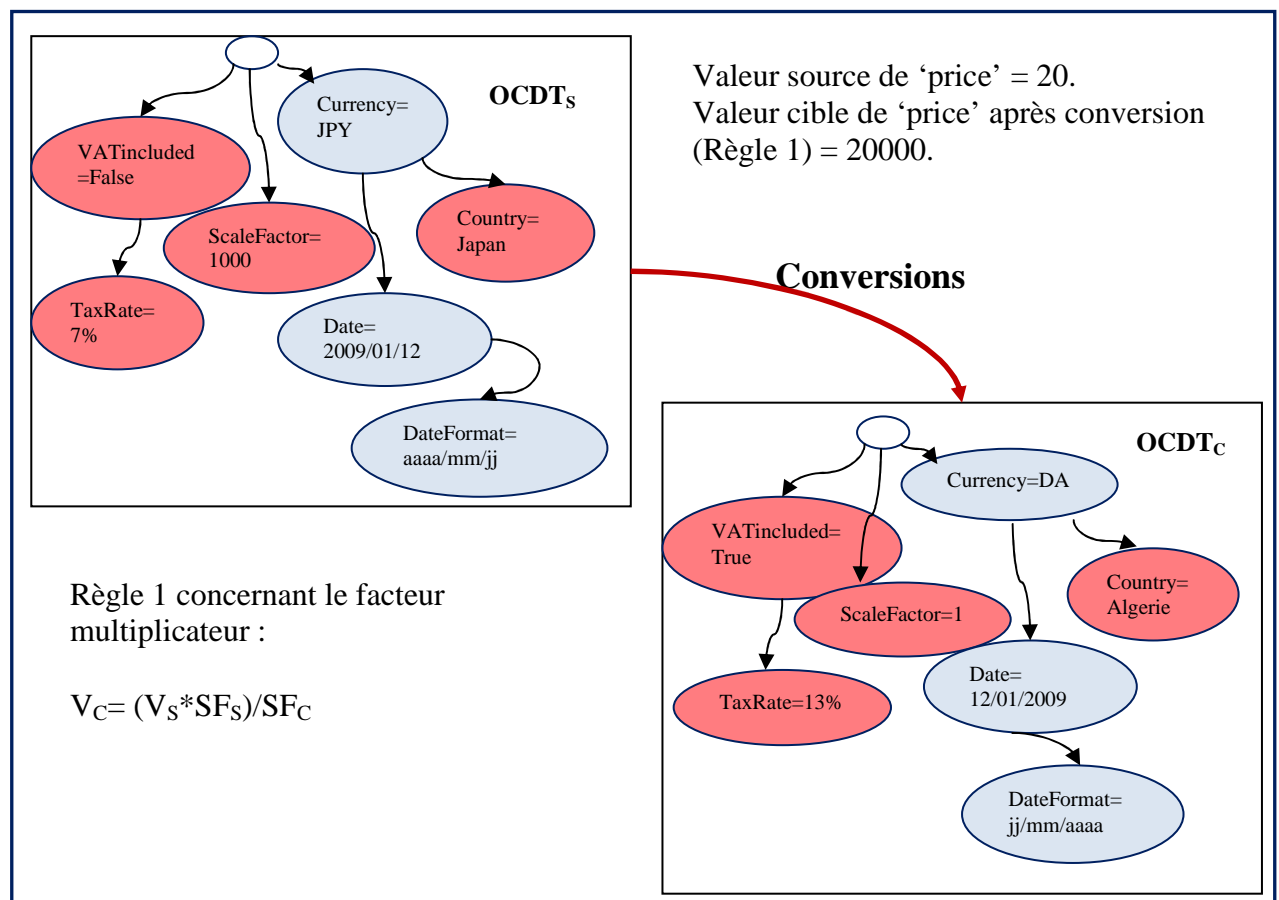


Figure IV.7. Ontologies contextuelles de données temporaires et conversions.

- Concrétisation syntaxique : grâce à l'attribut `lowringSchemaMapping` qui référence la définition de transformation des données sémantiques vers des données XML.

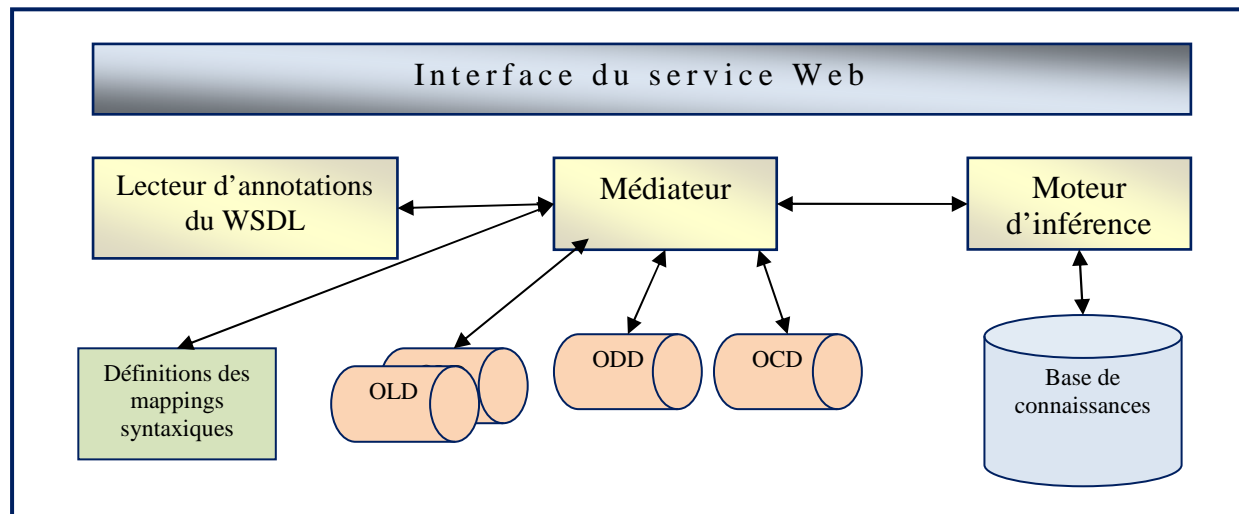


Figure IV.8. Description du service Web médiateur.

## IV 7. Conclusion

Nous avons présenté dans ce chapitre une étude de cas qui a illustré les différents concepts et étapes de notre approche. Nous avons en particulier, construit l'architecture de services métiers à exposition interne et ceux à exposition externes, lesquels sont publiés dans un référentiel UDDI en utilisant respectivement la clause privée ou publique.

Nous avons aussi, construit le sous modèle sémantique de données à l'aide de l'outil Protégé en transformant les modèles de données en ontologies locales, puis sont construites l'ontologie de domaine et les ontologies contextuelles.

A l'aide de ce sous modèle sémantique, et les mécanismes d'annotation du standard SAWSDL, nous avons enrichi les descriptions des services Web que nous avons choisi comme exemple.

Nous avons enfin, procédé à la médiation des données échangées entre ces deux services Web en exploitant le modèle sémantique et les annotations des descriptions des deux services.

# Conclusion générale

## Bilan du travail effectué

Nous rappelons que notre travail de recherche s'inscrit dans le cadre des systèmes d'information coopératifs, et dont l'objectif principal est la proposition d'une architecture orientée services pour la mise en œuvre des mécanismes de l'interopérabilité des systèmes d'information des entreprises. Notre ambition à travers cette architecture est de répondre aux problématiques d'interopérabilité technique par le passage à une architecture orientée services, et d'interopérabilité sémantique.

Une étude de l'état de l'art nous a permis de constater que les technologies permettant l'interopérabilité des systèmes d'information ont pris une énorme importance de part leur création, normalisation et standardisation par de grands constructeurs et organisations (exemples : W3C, OASIS ...). Dans cette étude, nous avons présenté une panoplie d'outils techniques et conceptuels permettant l'interopérabilité au niveau technique et syntaxique ainsi qu'au niveau sémantique, ce qui nous a permis de retenir les paradigmes des architectures orientées services (SOA), les services Web et l'ingénierie des modèles (MDA) pour dresser les mécanismes de l'interopérabilité technique. Comme nous l'avons mentionné, la principale limite de ces paradigmes est leur manque en ce qui concerne la présentation de l'aspect sémantique, principale cause de l'échec de l'interopérabilité des systèmes d'information. Pour remédier à ce problème, nous avons retenu l'approche déclarative de représentation de la sémantique, notamment les ontologies. L'approche que nous avons proposée a conduit à une architecture en couche. Nos principales contributions sont résumées comme suit :

- L'adoption d'une méthodologie basée sur l'ingénierie des modèles (MDA pour Model Driven Architecture) qui permet d'une part une migration progressive et adéquate des systèmes d'information existants vers des architectures orientées services, et d'autre part la construction de nouveaux systèmes d'information orientés services, qui tirent profit des avantages de SOA et des standards des services Web (SOAP, WSDL, UDDI).
- La proposition d'un modèle sémantique basé sur la notion d'ontologies pour capturer la sémantique des propriétés fonctionnelles et non fonctionnelles des services Web. Notre modèle conceptuel est scindé en trois sous modèles qui sont, le sous modèle sémantique des propriétés fonctionnelles, le sous modèle sémantique des propriétés non fonctionnelles et, le sous modèle sémantique des données. Nous avons détaillé le dernier sous modèle qui comprend trois types d'ontologies à savoir, des ontologies locales qui véhiculent la sémantique locale de chaque fournisseur, une ontologie de domaine qui présente un accord des fournisseurs sur des concepts communs, et des ontologies contextuelles. Notre sous modèle exploite la notion de contexte que nous avons dissociée de l'ontologie de domaine sous forme d'ontologies contextuelles, chacune liée à un concept de l'ontologie de domaine, dans le but de décrire

explicitement le contexte des données échangées entre services Web, et résoudre les conflits lors de la mise à jour des ontologies contextuelles.

- L'enrichissement sémantique des descriptions des services Web par une annotation grâce au modèle sémantique en utilisant le 'Semantic Annotation of Web Service Description Language', un standard récent qui fait partie des recommandations du consortium W3C. Il s'agit en particulier de l'annotation des données échangées entre services Web grâce au sous modèle sémantique des données.
- Et enfin, la proposition d'une architecture basée services pour l'interopérabilité des systèmes d'information, et la définition des mécanismes d'interopérabilité sémantique basés sur les concepts précédents. Nous avons présenté le processus de coopération qui est pris en charge par le service Web broker. Ce dernier orchestre les autres services Web d'interopérabilité sémantique qui sont chargés de la publication, de la découverte, de la médiation et de l'exécution des services Web en interaction. Notons que nous avons détaillé en particulier la médiation sémantique.

## Perspectives

Dans le but d'une continuité de nos travaux de recherche, nous avons envisagé les perspectives suivantes :

- Compléter le modèle sémantique par la construction des sous modèles des propriétés fonctionnelles et non fonctionnelles, en étudiant la possibilité d'exploiter la notion de contexte introduite pour le sous modèle des données.
- Annotation des descriptions des services Web grâce aux sous modèles précédents, pour permettre d'une part la découverte sémantique et d'autre part la médiation des propriétés fonctionnelles et celle des propriétés non fonctionnelles.
- Introduction de ces mécanismes de médiation dans un processus de composition de services Web.
- Administration des services, par des mécanismes permettant de contrôler leur intégrité et leur réutilisation.
- Maintenance des ontologies qui est une tâche très importante pour l'architecture sémantique.

## Références Bibliographiques

- [1] S. Izza ; ‘‘ Intégration des systèmes d’information industriels, une approche flexible basée sur les services sémantiques’’. Thèse de doctorat de l’Ecole Nationale Supérieure des Mines de Saint –Etienne, France, 20 Novembre 2006.
- [2] B. Leroux ; ‘‘ Urbanisation et Modernisation du Système d’information’’. Lavoisier, 2004.
- [3] R. Reix ; ‘‘ Systèmes d’information de gestion’’. Editions d’Organisation, 1995.
- [4] B. Manouvrier ; ‘‘ EAI- Intégration des applications d’entreprise’’. Hermes, 2001.
- [5] H. Panetto ; ‘‘Meta-modèles et modèles pour l’intégration et l’interopérabilité des applications d’entreprises de production’’. Habilitation à diriger des recherches. Université Henri Poincaré – Nancy I, 2006.
- [6] J.L. Lemoigne ; ‘‘La théorie du système général’’. Editions Presse Universitaire Française, 1984.
- [7] H. Tardieu, A. Rochfeld, C. Rolland ; ‘‘ La méthode Merise Principes et Outils’’. Edition d’organisation, 2002.
- [8] C. Tessier ; ‘‘ La pratique des méthodes en informatique de gestion : Typologie, analyse comparative, choix et mise en œuvre’’. Editions d’organisation, 1995.
- [9] C. Bussler ; ‘‘ The role of Semantic Web Technology in Entreprise Application Integration’’. In the Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 26 (N° 4), pp. 62-68, 2003.
- [10] W. Hasselbring ; ‘‘ Information System Integration’’. Communication of the ACM, Vol. 43 (N° 6), pp. 33-38, 2000.
- [11] M. Singh, M. Huhns ; ‘‘ Service-Oriented Computing, Semantics, Processes, Agents’’. Wiley, 2004.
- [12] M.T. Ozsü, P. Valduriez ; ‘‘ Principles of Distributed Database Systems’’. Prentice Hall, 1999.
- [13] G. Weiderhold ; ‘‘ Mediators in the Architecture of future Information Systems’’. IEEE Computer, Vol. 25 (N° 3), pp. 38-49, 1992.
- [14] D. Benslimane, K. Yétongnon, S. Chraïbi, E. Leclercq, E. Abdelwaheb ; ‘‘ Les systèmes d’information coopératifs : le projet DECA’’. Technique et science informatiques. Vol. 19 (N° 7/2000), pp. 1009-1043, 2000.
- [15] European Interoperability Framework for pan-European eGovernment Services, Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens (IDABC), November 2004, Luxembourg.
- [16] IEEE (Institute of Electrical and Electronics Engineers). Standard Computer Dictionary- A compilation of IEEE Standard Computer Glossaries NY. 610-1990. ISBN : 1559370793.

- [17] IEEE. Standards Information Network. IEEE 100, “ The Authoritative Dictionary of IEEE Standards Terms”, Seventh Edition. Newyork, NY, IEEE, 2000.
- [18] D. Chen, H. Panetto, et al. “D6.1 : Practices, principles and patterns for interoperability”. INTEROP-NOE, Interoperability Research for Network enterprises Applications and Software Network of Excellence, N° IST 508-011, Jun 2005.
- [19] “ European Interoperability Framework, - ICT Industry Recommendation-”. Brussels, 18 february 2004, WWW. Comptia.org
- [20] J.P. Meinadier ; “ Le métier d’intégration des systèmes”. Lavoisier-Hermes, 2002.
- [21] IDEAS ; “ IDEAS Project Deliverables (Wp1-Wp7)”. Public reports, [www.ideas-roadmap.net](http://www.ideas-roadmap.net), 2003.
- [22] D.S. Linthicum ; “ Next Generation Application Integration”. Addison-Wesley, 2004.
- [23] S. Krakowiak ; “ Introduction à l’intergiciel”. Université Joseph Fourier, Grenoble. Version du 19 janvier 2007.
- [24] P. Bonnet ; “ Cadre de référence Architecture SOA. Meilleures Pratiques”. Montreuil, 23 Février 2005. [http://pie.bonnet.ifrance.com/ON-guideSOA-2005-02-23%20\(part1\).pdf](http://pie.bonnet.ifrance.com/ON-guideSOA-2005-02-23%20(part1).pdf)
- [25] R. Klischewski ; “Information integration or process integration: How to achieve interoperability in administration”. EGOV04 at DEXA, 30 August - 3 September, 2004, Zaragoza, Spain.
- [26] M. Champion, C. Ferris, E. Newcomer, Iona, and D. Orchard ; “Web Services Architecture”. 14 Novembre 2002. <http://www.W3.org/TR/2002/WD-ws-arch-20021114/>.
- [27] V. Monfort, S. Goudeau ; “ Web services et interopérabilité des systèmes d’information”. Dunod, Paris, 2004.
- [28] D. Chappell ; “ Entreprise Service Bus”. O’Reilly media Inc., 2005.
- [29] B. Lublinsky and D. Tyomkin ; “ Dissecting Service Oriented Architectures”. Business Integration Journal 2003.
- [30] A. Naumenko and A. Wegman ; “ Two Approches in System Modelling and their Illustration with MDA and RM-ODP”. In ICEIS 2003 , the 5th International Conference on Enterprise Information Systems, p.p. 398-402.
- [31] R. Lemesle ; “ Transformation Rules Based on Meta-Modeling”. IEEE EDOC’98, La Jolla.
- [32] T. Levendovszky, G. Karsai, M. Maroti, A. Ledeczi and H. Charaf ; “ Model Reuse with Metamodel-Based Transformations”. In Proceedings of the 7th International Conference on Software Reuse : Methods, Techniques, and Tools table of contents. April 15-19, 2002. p.p. 166-178.
- [33] D. Berardi, D. Clavanese, G. D. Giacomo, M. Lenzerini, and M. Mecella ; “ A foundational vision of e-services”. In C. Bussler, D. Fensel, M. E.Orlowska, and J. Yang,

editors, WES, Volume 3095 of Lecture Notes in Computer Science, pages 28-40. Springer, 2003.

[34] M. Mrissa ; " Médiation Sémantique Orientée Contexte pour la Composition de Services Web". Thèse de doctorat de l'université Claude Bernard Lyon I. 15 Novembre 2007.

[35] S. Rampacek ; "Sémantique, interactions et langages de description des services web complexes". Thèse de doctorat. Université de Reims Champagne-Ardenne. Le 10 Novembre 2006.

[36] F. Leyman ; " Web Services Flow Language (WSFL 1.0)". Mai 2001. <http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>.

[37] I. A. Arkin ; "Business Process Modeling Language", 13 Novembre 2002. <http://www.bpml.org/bpml-spec.esp>.

[38] S. Thatte ; "XLANG - Web Services For Business Process Design", 2001.[http://www.gotdotnet.com/team/xml\\_wsspecs/xlang-c/default.htm](http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm).

[39] T. Andrews, F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana ; "Business Process Execution Language for WebServices (BPEL4WS), Version 1.1". May 2003.

[40] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau ; " Extensible Markup language (XML) 1.0". Editors : World Wide Web Consortium, 4 February 2004. <http://www.w3.org/TR/2004/REC-xml-20040204/>.

[41] "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language". Editors: R. Chinnici, Sun Microsystems ; M. Gudgin, Microsoft ; J.J. Moreau, Canon ; J. Schlimmer, Microsoft ; S. Weerawarana, IBM Research. W3C Working Draft 26 March 2004. <http://www.w3.org/TR/2004/WD-wsd120-20040326>

[42] J. Miller, K. Verma, P. Rajasekaran, A. Sheth, R. Aggarwal, K. Sivashanmugam ; " WSDL-S : A Proposal to W3C WSDL 2.0 Committee". In " METEOR-S : Semantic Web Services and Processes". LSDIS Lab, University of Georgia. <http://lsdis.cs.uga.edu/Projects/METEOR-S/>.

[43] W.A.Woods ; "What's in a link: foundations of semantic networks". In D.G.Bobrow & A. M. Collins (Eds.), 1975.

[44] M. Kavouras ; "A unified ontological framework for semantic integration". International Workshop on Next Generation Geospatial Information, Cambridge, UK, 2003.

[45] M. Uschold, and M. Gruninger ; "Creating Semantically Integrated Communities on the World Wide Web". In Semantic Web Workshop, Honolulu, Hawaii-Invited Talk, 2002.

[46] T. Gruber T ; "A Translation Approach to Portable Ontology Specifications". Knowledge Acquisition, Vol. 5 (No. 2) pp. 199-220, 1993.

- [47] W. N. Borst ; “Construction of Engineering Ontologies for Knowledge Sharing and Reuse”. PhD thesis, Tweente University, Enschede, NL- Centre for Telematica and Information Technology, 1997.
- [48] N. Guarino ; “Formal Ontology, Conceptual Analysis and Knowledge Representation”. International journal of Human and Computer Studies, Vol. 43 (No. 5/6), pp. 625-640,1995.
- [49] B. Swartout, R. Patil, K. Knight and T. Russ ; “Use of Large-Scale Ontologies”. Spring Symposium Series on Ontological Engineering. Stanford University, CA, pp. 138-148, 1997.
- [50] A. Gomez-Perez ; “Développements récents en matière de conception, de maintenance et d'utilisation d'ontologies”. 3èmes rencontres Terminologie et Intelligence Artificielle TIA, Vol 19, pp. 9-20, 2000.
- [51] N. Zouggar, B. Vallespir, D. Chen ; “Enrichissement de la Modelisation d’Entreprise par les Ontologies”. LAPS/GRAI Université Bordeaux1, ENSEIRB, UMR CNRS 5131. 6e Conférence Francophone de MODélisation et SIMulation - MOSIM’06 - du 3 au 5 avril 2006 – Rabat- Maroc. « Modélisation, Optimisation et Simulation des Systèmes : Défis et Opportunités ».
- [52] R. Mizoguchi ; “Ikeda: Task Ontologies for reuse of Problem Solving Knowledge”. In N. J. I. Mars (editors), Towards Very Large Knowledge Bases, IOS Press, 1995.
- [53] G. Van Heijst, A. Th. Schreiber and B. J. Wielinga ; “Using explicit ontologies in KBS development”. International Journal of Human-Computer Studies, Vol. 46 (No. 2/3) pp.183-292, 1997.
- [54] R. T. Gruber ; “Towards Principles for the Design of Ontologies Used for Knowledge Sharing”. International Journal of Human Computer Studies, Vol. 43, (No 5/6), pp. 907-928, 1995.
- [55] X. Blanc, O. Salvatori ; “MDA en action : Ingénierie logicielle guidée par les modèles”. Eyrolles, 2005.
- [56] O. Corcho, M. Fernandez-Lopez, and A. Gomez-Pérez ; “Methodologies, tools and languages for building ontologies : Where is their meeting point ?”. Data & Knowledge Engineering Vol. 46 (No. 1) pp. 41-64, Elsevier, 2003.
- [57] OntoWeb Consortium ; “Deliverable 1.3: A survey on ontology tools -OntoWeb Ontology-based information exchange for knowledge management and electronic commerce”. IST Project IST-2000-29243, 2000.
- [58] WonderWeb, Deliverables of WonderWeb project, <http://wonderweb.semanticweb.org/>, 2005.
- [59] K. Falkovych, M. Sabou, H. Stuckenschmidt ; “ UML for the Semantic Web : Transformation-Based Approaches”. Dans B. Omelayenko and M. Klein, editors, Knowledge Transformation for the Semantic Web, pp. 92-106. ISO Press, 2003.

- [60] La bibliothèque Jena2, <http://jena.sourceforge.net/>
- [61] INTEROP, 2004. State of the art and state of the practice including initial possible research orientations. Deliverable D8.1. <http://www.interop-noe.org/>
- [62] ‘‘RDF Primer’’. Recommendation RDF du W3C. <http://www.w3.org/>
- [63] OWL Coalition ; ‘‘OWL - Web Ontology Language - (W3C Recommendation 10 February 2004)’’. W3C, <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>, 2004.
- [64] Protégé website, <http://protege.stanford.edu/>
- [65] JESS website, <http://herzberg.ca.sandia.gov/jess/>
- [66] site Web sur « description logics », <http://dl.kr.org/>
- [67] Racer Semantic Web Inference Engine website, <http://www.sts.tu-harburg.de/~r.f.moeller/racer/>
- [68] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner ; ‘‘Ontology-based Integration of Information - A Survey of Existing Approaches’’. In: Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, 2001, Vol. pp. 108-117.
- [69] E. Mena, A. Illarramendi , V. Kashyap, and A.P.Sheth ; ‘‘Observer: An approach for query processing in global information systems based on interoperation across pre-existing ontologies’’. Distributed and Parallel Databases, Vol. 8 (No. 2) pp. 223-271, 2000.
- [70] C.H. Goh ; ‘‘Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources’’. Phd Thesis, MIT, 1997.
- [71] H. Wache, Th. Scholz, H. Stieghahn, and B. König-Ries ; ‘‘An integration method for the specification of rule-oriented mediators’’. Proceedings of DANTE’99, pp. 109-112, Kyoto, Japan, 1999.
- [72] H. Stuckenschmidt ; ‘‘Ontologies for Semantic Information Integration: Opportunities and Open Problems’’. EKAW-00, <http://www.cs.vu.nl/~heiner/public/EKAW-00.pdf>, 2000.
- [73] U. Visser, and H. Stuckenschmidt ; ‘‘Interoperability in GIS - Enabling Technologies’’. In Proceeding of 5th AGILE Conference on GIS, Spain, 2002.
- [74] M. Klein ; ‘‘Combining and relating ontologies: an analysis of problems and solutions’’. International Joint Conferences on Artificial Intelligence (IJCAI), Workshop on Ontologies and Information Sharing, Seattle, USA, 2001.
- [75] INTEROP ; ‘‘Deliverables of INTEROP Project’’. [www.interop-ne.org](http://www.interop-ne.org), 2005.
- [76] KnowledgeWeb ; ‘‘Deliverables of KWEB Project’’. EU-IST-2004-507482, <http://knowledgeweb.semanticweb.org/>, 2004.
- [77] Sekt, ‘‘Deliverables of SEKT Project’’. <http://www.sektproject.com/>, 2004.

- [78] J. Bruijn, H. Lausen, R. Krummenacher, A. Polleres, M. Kifer and D. Fensel ; “The Web Service Modeling Language WSMML”. Deliverable D16.1v0.2, Final Draft, WSMO project, <http://www.wsmo.org/TR/d16/d16.1/v0.2/20050320/>, 2005.
- [79] A. Doan, J. Madhavan, P. Domingos, and A. Halevy ; “Learning to Map Between Ontologies on the Semantic Web”. In The Eleventh International WWW Conference (WWW'02), Hawaii, US, <http://citeseer.ist.psu.edu/doan02learning.html>, 2002.
- [80] G. Stumme, and A. Madche ; “FCA-Merge: Bottom-up merging of ontologies”. In 7th International Conference on Artificial Intelligence (IJCAI'01), pp. 225-230, Seattle, WA, 2001.
- [81] N. F. Noy, and M. A. Musen ; “The Prompt suite: Interactive tools for ontology merging and mapping”. International Journal of Human Computer Studies, Vol.59(6): pp. 983-1024, 2003.
- [82] G. Athanasopoulos, A. Tsalgatidou, and M. Pantazoglou. Interoperability among heterogeneous services. In Society [69], pp. 174–181.
- [83] C. Bussler ; “Semantic web services : Reflections on web service mediation and composition”. In WISE, pp. 253–260. IEEE Computer Society, 2003.
- [84] L. Cabral and J. Domingue. Mediation of semantic web services in irs-iii. In First International Workshop on Mediation in Semantic Web Services (MEDIATE 2005) held in conjunction with the 3rd International Conference on Service Oriented Computing (ICSOC 2005), Amsterdam, The Netherlands., December 12th 2005.
- [85] G. Schreiber and M. Dean. Owl web ontology language reference. <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>, February 2004.
- [86] D. L. Martin, M. Paolucci, S. A. McIlraith, M. H. Burstein, D. V. McDermott, D. L. McGuinness, B. Parsia, T. R. Payne, M. Sabou, M. Solanki, N. Srinivasan, and K. P. Sycara ; “Bringing Semantics to Web Services : The OWL-S Approach”. In J. Cardoso and A. P. Sheth, editors, SWSWPC, volume 3387 of Lecture Notes in Computer Science, pp. 26–42. Springer, 2004.
- [87] S. Arroyo and M. Stollberg ; “WSMO Primer”. WSMO Deliverable D3.1, DERI Working Draft. Technical report, WSMO, 2004. <http://www.wsmo.org/2004/d3/d3.1/>.
- [88] M. Klein, B. König-Ries, and M. Müssig. “What is needed for semantic service descriptions - a proposal for suitable language constructs”. International Journal on Web and Grid Services (IJWGS), vol. 1(3/4) pp.:328–364, 2005.
- [89] J. Peer ; “Semantic Service Markup with SESMA”. In Web Service Semantics Workshop (WSS'05) at the Fourteenth International World Wide Web Conference (WWW'05), 2005.
- [90] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara ; “Importing the Semantic Web in UDDI”. In Proceedings of E-Services and the Semantic Web Workshop, 2002.

- [91] A. Ankolekar, M. Burstein, J. Hobbs, O. Lassila, D. Martin, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, and K. Sycara ; “DAML-S : Web Service Description for the Semantic Web”, 2002.
- [92] A. Dogac, Y. Kabak, and G. Laleci ; “Enriching ebxml registries with owl ontologies for efficient service discovery”. In RIDE, pp. 69–76. IEEE Computer Society, 2004.
- [93] D. Kourtesis and I. Paraskakis ; “Combining SAWSDL, OWL-DL and UDDI for Semantically Enhanced Web Service Discovery”. South East European Research Centre (SEERC), Research Centre of the University of Sheffield and CITY College. Research project FUSION (Business process fusion based on semantically-enabled service-oriented business applications), 2008.
- [94] P. A. Sheth, K. Gomadam, A. Ranabahu ; “ Semantics Enhanced Services: METEOR-S, SAWSDL and SA-REST”. Services Research Lab, kno.e.sis center, Wright state University, Dayton, OH. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2008.
- [95] J. Miller, K. Verma, P. Rajasekaran, A. Sheth, R. Aggarwal, and K. Sivashanmugam ; “ WSDL-S : Adding Semantics to WSDL”. White Paper. Technical report, Large Scale Distributed Information Systems, 2004. <http://lsdis.cs.uga.edu/library/download/wSDL-s.pdf>.
- [96] J. Miller, K. Verma, P. Rajasekaran, A. Sheth, R. Aggarwal, K. Sivashanmugam ; “WSDL-S: A Proposal to W3C WSDL 2.0 Committee”. LSDIS Lab, University of Georgia. METEOR-S: Semantic Web Services and Processes <http://lsdis.cs.uga.edu/Projects/METEOR-S/>
- [97] “Semantic Annotations for WSDL”. W3C Working Draft 28 September 2006. Editors: Joel Farrell, IBM ; Holger Lausen, DERI Innsbruck. <http://www.w3.org/TR/2006/WD-sawSDL-20060928/>
- [98] Erik Christensen, Francisco Curbera, Greg Meredith and Sanjiva Weerawarana ; “W3C Notes, Web Service Description Language WSDL 1.1”. <http://www.w3.org/TR/2001/NOTE-wSDL-20010315>
- [99] “ XML Schema Part 1 : Structures”. H. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, Editors. World Wide Web Consortium, 28 October 2004. <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028>.
- [100] “XMLSchema Part 2: Datatypes”. P. Byron and A. Malhotra, Editors. World Wide Web Consortium, 28 October 2004. <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>.
- [101] T. Hausberger ; “Semantic Annotations for WSDL”. Departement of Computer Science Leopold Franzens University Innsbruck. Semantic Web Services PS. 5 May 2009.

- [102] A. Haller, E. Cimpian, A. Mocan, E. Oren, and C. Bussler ; “Wsmx - a semantic service-oriented architecture”. In I. C. Society, editor, ICWS, pages 321–328. IEEE Computer Society, 2005.
- [103] A. Gugliotta, L. Gabral, J. Domingue, V. Roberto, M. Rowlatt, and R. Davies ; “ A Semantic Web Service-based Architecture for the Interoperabilité of E-government Services”. Italy 2006.
- [104] S. Izza, L. Vincent and P. Burlat ; “A Framework for Semantic Enterprise Integration”. In Proceedings of INTEROP-ESA'05, Geneva, Switzerland, pp. 78-89, 2005.
- [105] M. Chevassus ; “Enterprise Application integration : EAI”. Techniques de l'ingénieur, H2915, pp.1-20, 2005.
- [106] H. Kadima ; “MDA : Conception orientée objet guidée par les modèle”. Dunod,2005.
- [107] Y. Arens, H. Chun-Nan and A. Craig ; “Query processing in the sims information mediator”. In Advanced Planning Technology. AAAI Press, California, USA, 1996.
- [108] “ UDDI Version 3.0.2”. L. Clement, A. Hately, C. V. Riegen, and T. Rogers, Editors. Organization for the Advancement of Structured Information Standards (OASIS). [http://uddi.org/pubs/uddi\\_v3.htm](http://uddi.org/pubs/uddi_v3.htm).
- [109] “ Unified Modeling Language (UML) Version 2.0 “. Object Management Group. <http://www.omg.org/technology/documents/formal/uml.htm>.
- [110] “XSL Transformations (XSLT) Version 2.0“. M. Kay, Editor. World Wide Web Consortium, 3 Nov 2005. <http://www.w3.org/TR/2005/CR-xslt20-20051103/>.
- [111] “ XQuery 1.0 : An XML Query Language”. D. Chamberlin , A. Berglund, S. Boag, *et al.*, Editors. World Wide Web Consortium, 3 Nov 2005. <http://www.w3.org/TR/2005/CR-xquery-20051103/>.
- [112] “SPARQL Query Language for RDF“. E. Prud'hommeaux and A. Seaborne, Editors. World Wide Web Consortium, 6 April 2006. <http://www.w3.org/TR/2006/CR-rdf-sparql-query-20060406/>.
- [113] SAWSDL4J : Java API for manipulating and creating annotated SAWSDL documents. <http://www.schneier.com/essay-037.pdf>.
- [114] WSMO Studio : Semantic Web Service and Semantic Business Process modelling environment, Set of Eclipse plugins. <http://www.wsmostudio.org>.
- [115] Radiant : WSDL-S/SAWSDL Annotation Tool, Eclipse plugin. <http://lsdis.cs.uga.edu/projects/meteor-s/downloads/index.php?page=1>.
- [116] M. Dumas, M. Spork ; “Adapt or perish : Algebra and visual notation for service interface adaptation”. In S. Dustdar, J. L. Fiadeiro, and A. P. Sheth, editors, Business Process

Management, volume 4102 of Lecture Notes in Computer Science, pp. 65–80. Springer, 2006.

[117] B. Medjahed, B. Benatallah, A. Bouguettaya, and A. K. Elmagarmid ; “Webbis : An infrastructure for agile integration of web services”. *Int. J. Cooperative Inf. Syst.*, vol. 13(2) pp .121–158, 2004.

[118] Y. Taher, D. Benslimane, M.-C. Fauvet, and Z. Maamar ; “ Towards an approach for web services substitution”. In P. Ghodous, R. Dieng-Kuntz, and G. Loureiro, editors, *IDEAS*, pp. 166–173. IOS Press, 2006.

[119] A.Doan, J. Madhavan , P. Domingos, and A. Halevy ; “Learning to Map Between Ontologies on the Semantic Web". In *The 11th International WWW Conference (WWW'02)*, Hawaii, US, <http://citeseer.ist.psu.edu/doan02learning.html> , 2002.

[120] N. F. Noy, and D. McGuinness ; “OntologyDevelopment 101: A Guide to Creating our FirstOntology”. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*, mars 2001.

[121] Softeam ; [http://www.softeam.fr/technologies\\_web\\_services.php](http://www.softeam.fr/technologies_web_services.php).

[122] IBM ; “Web Services Coordination”. <http://www128.ibm.com/developerworks/library/specification/ws-tx/> , 2005.

[123] Apache ; “Apache Beehive”. <http://beehive.apache.org/>, 2005.

[124] PRESTO ; « PRESTO, Protocole d’échange standard et ouvert de l’administration ». 7 juillet 2007, [http://solutionslinux.fr/document\\_conferencier/45caf51e92153.pdf](http://solutionslinux.fr/document_conferencier/45caf51e92153.pdf)

[125] A. Tebib, M. Boufaïda ; “ Une architecture pour l’interopérabilité et la sécurité des services de l’E-gouvernement”. *CRiSIS’07*, co-located with *IEEE GIIS*, Marrakech, pp : 15-22, 2007.

[126] S. Sanlaville, J. Estublier ; “ Mélusine : un environnement de modélisation et de coordination de services”. *Revue d’Ingénierie des Systèmes d’Information*. Volume 12, N° 3/2005.