

وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR-ANNABA UNIVERSITY  
UNIVERSITE BADJI MOKHTAR-ANNABA



جامعة باجي مختار - عنابة

Faculté des Sciences de L'Ingéniorat Année 2017-2018  
Département d'Informatique

## THESE

Présentée en vue de l'obtention  
du diplôme de Doctorat en Sciences

# Architecture basée contexte pour la médiation des propriétés des services Web

Option  
Intelligence Artificielle

par  
M<sup>me</sup> MECHERI Karima

### Directeur de Thèse

M<sup>me</sup> Labiba SOUICI-MESLATI Prof. Université Badji Mokhtar-Annaba

### Co-Directeur de Thèse

Mr Mahmoud BOUFAIDA Prof. Université Abdelhamid Mehri-Constantine2

### Président

Mr Nadir FARAH Prof. Université Badji Mokhtar - Annaba

### Examineurs

Mr Nacereddine ZAROOUR Prof. Université Abdelhamid Mehri -Constantine2

Mr Yacine LAFIFI Prof. Université 8 Mai 1945 -Guelma

# Remerciements

Je tiens à exprimer mes profonds remerciements à ma directrice de thèse madame **Souici-Meslati Labiba**, Professeur à l'université Badji Mokhtar - Annaba - pour ses conseils, ses orientations, ses encouragements et son soutien permanent tant au niveau des connaissances qu'au niveau humain. Je la remercie pour ses efforts considérables et pour le temps qu'elle a consacré à l'encadrement de ce travail de recherche.

Mes plus vifs remerciements à mon co-directeur de thèse monsieur **Boufaida Mahmoud**, Professeur à l'université Abdelhamid Mehri-Constantine2- pour ses conseils, son aide et son soutien tout au long de ce travail de recherche. Je le remercie pour le temps qu'il a consacré pour diriger cette thèse malgré ses nombreuses charges. Ses compétences sa clairvoyance et son expérience m'ont profondément marquées. Qu'il trouve ici, ma sincère reconnaissance et ma profonde gratitude.

Je tiens, également, à remercier les membres du jury qui m'ont fait l'honneur de bien vouloir évaluer mon travail, et plus précisément :

Monsieur **Nadir FARAH**, Professeur à l'université Badji Mokhtar -Annaba- pour l'honneur qu'il m'a fait, en acceptant de présider ce jury.

Monsieur **Nacreddine ZAROUR**, Professeur à l'université Abdelhamid Mehri- Constantine 2-, pour avoir accepté de juger le présent document.

Monsieur **Yacine LAFIFI**, Professeur à l'université 8 Mai 1945 -Guelma-, d'avoir accepté de faire partie de ce jury.

J'adresse mes sincères remerciements à monsieur **Meslati Djamel**, Professeur à l'université Badji Mokhtar -Annaba-, et chef du laboratoire LISCO, pour son aide précieuse et sa collaboration.

Je tiens à remercier mon amie madame **KLAI Sihem**, maître de conférences à l'université Badji-Mokhtar, ainsi que tous mes collègues, enseignants, employés et responsables du département d'informatique.

Mes vifs remerciements à tous mes collègues, amis et responsables de l'INSFP Didouche Mourad -Annaba-.

Enfin, mes remerciements à tous les membres de ma famille qui m'ont tant soutenue, il s'agit en particulier de mes fils Abderrahmane, Nacereddine et Mondher, ma sœur Sabah, son mari Sofiane, mon frère Samir et son épouse Nacira. Sans oublier mes chers neveux Iheb Hocine, Meriem, Ali et Camélia.

# Dédicace

A la mémoire de mon cher époux  
Azzeddine BERREHAIL

# Table des abréviations

Abréviation	Signification
ACID	Atomicité, Cohérence, Isolation et Durabilité.
ASM	Abstract State Machine.
BEEP	Blocks Extensible Exchange Protocol.
BPEL	Business Process Execution Language.
BPEL4WS	Business Process Execution Language for Web Services.
BPML	Business Process Modeling Language.
BSSM	Behavioural Semantic Sub-model.
DAML+OIL	DARPA Agent Markup Language + Ontology Inference Layer.
DAML-S	DARPA agent markup language for services.
DCO	Data Contextual Ontology
DDO	Data Domain Ontology
DE et DSD	DIANE Elements (DE) et DIANE Service Description (DSD).
DL	Description Logic.
DLO	Data Local Ontology.
DSSM	Data Semantic Sub Model.
EAI	Entreprise Application Integration.
ebXML	Electronic Business XML.
EJB	Entreprise Java Beans
ENIO	ENterprise Interoperability Ontology.
ESB	Entreprise Service Bus.
FSMs	Finite State Machines.
FSSM	Functional properties Semantic Sub-Model.
FTP	File Transfer Protocol.
HTTP	Hypertext Transfer Protocol.
IRS-III	Internet Reasoning Service.
JAX-WS	L'API Java pour les services Web XML.
MaaS	Mobility as a service.
MEPs	Message Exchange Patterns
NSSM	Non functional properties Semantic Sub-Model.
ODSOIA	Ontology Driven Service Oriented Architecture.
OWL	Ontology Web Language.
OWL-S	Ontology Web Language for Web Services.
P2P / MAS	Peer to Peer Multi Agent System.
PHP	Hypertext Preprocessor.

QoS	Quality of Service.
RDF	Ressource Description Framework.
RDFS	Ressource Description Framework Schema.
RPC	Remote Procedue Call.
SA4MaaS	Semantic Annotation for Mobility as a Service.
SAWSD4J	Semantic Annotation of Web Service Description Language for Java.
SAWSDL	Semantic Annotation of Web Service Description Language.
SemSynT	Semantic-Syntactic Transformations.
SESMA	SEmantic Service MARkup.
SGML	Standard Generalized Markup Language.
SHOE	Simple HTML Ontology Extensions.
SI	Systèmes d'Information.
SIC	Systèmes d'Information Coopératifs.
SMTP	Simple Mail Transfer Protocol.
SOA / AOS	Service Oriented Architecture / Architectures Orientées Services.
SPARQL	SPARQL Protocol and RDF Query Language.
SST	Semantic-Semantic Transformation.
SW	Services Web.
SWba	Service Web de réservation de billets d'avion.
SWca	Service Web pour calculer le prix total.
SWch	Service Web de réservation d'une chambre d'hôtel.
SWRL	Semantic Web Rule Language.
SWS	Services Web Sémantiques.
SynSemT	Syntactic-Semantic Transformations.
UDDI	Universal Description Discovery and Integration
UML	Unified Modeling Language.
URI	Uniform Resource Identifier.
URL	Uniform Resource Locator.
W3C	Word Wide Web Consortium.
WS-BPEL	Web Services Business Process Execution Language.
WSCI	Web Service Choreography Interface.
WSDL	Web Service Description Language
WSDL-S	Adding Semantics to WSDL.
WSDM	Web Service of Data Mediation.
WSFM	Web Service of Functional mediation.
WSMO	Web Service Modeling Ontology.
WSMX	Web Service Modeling eXecution environment.

WSNM	Web Service of Non functional Mediation
WSPM	Web Service of Process Mediation.
WWW	World Wide Web.
XHTML	eXtensible Hyper Text Markup Language
XML	eXtensible Markup Language.
XOL	XML Ontology Language.
Xpath	XML path Language.
XSL	eXtensible Stylesheet Language.
XSLT	eXtensible Stylesheet Language Transformations.

## ملخص

يندرج عملنا البحثي ضمن الإطار العام للمعمارية الموجهة نحو الخدمات من أجل العمل المشترك لنظم المعلومات في الشركات. وبما أن الخدمات المتفاعلة لم تصمم في البداية للعمل المتبادل معاً، يمكن أن تظهر أوجه عدم التجانس على عدة مستويات تؤثر على خصائصها الوظيفية (السلوكية، الوظائف المقدمة، الترميز، تبادل البيانات، الخ)، فضلاً عن خصائصها غير الوظيفية (المعاملات، نوعية الخدمات، الخ)، مما يحول دون تعاونها. ولذلك، فإن الهدف الرئيسي من هذا العمل هو اقتراح نهج يسمح بالوساطة لخصائص خدمات الويب من أجل حل التضاربات الدلالية التي قد تنشأ بين خدمة الويب المرسله وخدمة الويب المتلقية. نقوم بإجراء دراسة مقارنة للأعمال القائمة ومن ثم تقديم نهج وهيكله قائمين على السياق للوساطة لخصائص خدمات الويب. في الواقع، نحن نصف الهيكله الدلالية (نموذج مرجعي) بالأونولوجيات لالتقاط دلالات خصائص خدمات الويب. ونؤكد بشكل خاص على النموذج السلوكي. ويستند هذا الأخير على مفهوم السياق والتمثيل باستخدام آلات الحالات المجردة. نربط خدمات الويب بالنموذج الدلالي باستخدام التعليق التوضيحي الدلالي للغة وصف خدمة الويب SAWSDL. ويتم التوفيق بين متواليات تبادل الرسائل لخدمات الويب في بنية تحتية موجهة نحو الخدمات باستخدام خوارزمية الوساطة للكوريغرافيا باستخدام النماذج السلوكية للطرفين وكذلك قواعد الانتقال بين أونولوجيات هذين النموذجين وإشراك وساطة البيانات. ونوضح نهجنا مع تنفيذ دراستي حالة. يتميز الهيكل المقترح بالمرونة التي تعزز إعادة الاستخدام، مما يحافظ على الموجود بفضل اقترانه الضعيف.

**الكلمات المفتاحية:** العمل البيئي لنظم المعلومات، عدم تجانس خدمات الويب، الوساطة الدلالية، الأنطولوجيا، آلة الحالة المجردة.

# Abstract

Our research work is in the general framework of service-oriented architectures for the information systems interoperability of companies. Since interacting services are not initially designed to interoperate together, heterogeneities can appear at several levels affecting their functional properties (behavioral, functionalities offered, encoding, exchanged data, etc.) as well as their non-functional properties (transactional, quality of services, etc.), preventing their cooperation. Therefore, the main objective of this work is to propose an approach allowing the mediation of the Web service properties in order to resolve the semantic conflicts that may arise between a sender Web service and a receiver Web service. We carry out a comparative study of the existing works and then present a context-based approach and architecture for the mediation of Web services properties. Indeed, we describe a semantic architecture (reference model) with ontologies to capture the semantics of Web services properties. We emphasize in particular the behavioral model. The latter is based on the notion of context and a representation using abstract state machines. We connect the Web services to the semantic model using the Semantic Annotation of Web Service Description Language (SAWSDL). The reconciliation of the message exchange sequences of the Web services in a service oriented infrastructure is carried out using a choreography mediation algorithm using the behavioral models of the two parties as well as the transition rules between the ontologies of these two models, and involving the data mediation. We illustrate our approach with the implementation of two case studies. The proposed architecture is flexible, which promotes reuse, thus preserving the existing thanks to its weak coupling.

**Keywords:** Interoperability of information systems, heterogeneity of Web services, semantic mediation, ontologies, abstract state machine.

# Résumé

Notre travail de recherche se situe dans le cadre général des architectures orientées services pour l'interopérabilité des systèmes d'information des entreprises. Etant donné que les services en interaction ne sont pas conçus initialement pour inter-opérer ensemble, des hétérogénéités peuvent apparaître à plusieurs niveaux touchant aussi bien leurs propriétés fonctionnelles (comportementales, fonctionnalités offertes, encodage, données échangées, ...etc.) que leurs propriétés non fonctionnelles (transactionnelles, qualité des services, ...etc.), empêchant leur coopération. De ce fait, l'objectif principal de ce travail est de proposer une approche permettant la médiation des propriétés des services Web afin de résoudre les conflits sémantiques pouvant apparaître entre un service Web émetteur et un service Web récepteur. Nous effectuons une étude comparative des travaux existants puis nous présentons une approche et une architecture basées sur la notion de contexte pour la médiation des propriétés des services Web. En effet, nous décrivons une architecture sémantique (modèle de référence) avec des ontologies pour capturer la sémantique des propriétés des services Web. Nous insistons en particulier sur le modèle comportemental. Ce dernier est basé sur la notion de contexte et une représentation utilisant des machines d'état abstraites. Nous relient les services Web au modèle sémantique grâce au langage SAWSDL (Semantic Annotation of Web Service Description Language). La réconciliation des séquences d'échange des messages des services Web dans une infrastructure orientée services est effectuée à l'aide d'un algorithme de médiation de chorégraphie, utilisant les modèles comportementaux des deux parties ainsi que des règles de transition entre les ontologies de ces modèles, et faisant appel à la médiation des données. Nous illustrons notre approche à l'aide de l'implémentation de deux études de cas. L'architecture proposée est flexible, ce qui favorise la réutilisation permettant ainsi de conserver l'existant grâce à son couplage faible.

**Mots-clés :** Interopérabilité des systèmes d'information, hétérogénéité des services Web, médiation sémantique, ontologies, machine d'état abstraite.

# Table des matières

## Introduction générale

1. Contexte de la recherche .....	1
2. Problématique.....	3
3. Objectifs et contributions.....	4
4. Organisation du document.....	6

## Chapitre 1 : Interopérabilité des systèmes d'information via les services Web sémantiques

1.1. Introduction.....	9
1.2. Systèmes d'information et processus métiers.....	9
1.3. Systèmes d'information coopératifs (SIC) .....	11
1.3.1. Notion d'interopérabilité.....	12
1.3.2. Caractéristiques et intérêt des SOA et des services Web.....	13
1.3.3. Composition des services Web .....	14
1.3.3.1. Orchestration.....	14
1.3.3.2. Chorégraphie.....	15
1.3.4. Hétérogénéités des services Web en interaction .....	16
1.3.5. Discussion : besoin de la sémantique .....	16
1.4. Sémantique et ontologies.....	17
1.4.1. Notion de sémantique.....	17
1.4.2. Notion d'ontologie .....	18
1.4.2.1. Structuration des ontologies .....	19
1.4.2.2. Langages de représentation et outils de développement des ontologies .....	19
1.4.3. Intégration des ontologies.....	21
1.4.3.1. Hétérogénéités des ontologies .....	21
1.4.3.2. Différents types d'intégration des ontologies .....	22
1.5. Les services Web sémantiques.....	24
1.5.1. Approches utilisant les langages de description sémantique .....	25
1.5.2. Approches d'annotation des langages existants .....	26
1.5.3. Approches d'intégration des services Web .....	27
1.6. Classification et comparaison de travaux sur l'interopérabilité des SWS .....	28

1.6.1.	Travaux basés sur les langages sémantiques .....	29
1.6.1.1.	Architecture WSMX (2005).....	29
1.6.1.2.	Architecture IRS-III (2008).....	29
1.6.1.3.	Architecture ODSOIA (2008).....	30
1.6.1.4.	Framework de chorégraphie SOPHIE (2010).....	31
1.6.1.5.	Architecture P2P/MAS (2014).....	31
1.6.1.6.	Ingénierie des SWS pour l'automatisation des processus métiers gouvernementaux (2015).....	31
1.6.2.	Travaux basés sur les annotations sémantiques .....	32
1.6.2.1.	Architecture de médiation sémantique orientée contexte.....	32
1.6.2.2.	Combiner SAWSDL, OWL-DL et UDDI pour la découverte de services Web réhaussés sémantiquement (2008).....	32
1.6.2.3.	Le projet METEOR-S (2008).....	32
1.6.2.4.	Médiation dynamique de données dans des scénarios d'intégration des applications d'entreprises.....	33
1.6.2.5.	SAWSDL-iMatcher (2011).....	33
1.6.2.6.	Approche SA4MaaS (2016).....	33
1.6.3.	Comparaison des travaux sur l'interopérabilité des SWS .....	34
1.7.	Conclusion .....	35
<b>Chapitre 2 : Un framework pour la mise en œuvre de l'interopérabilité des services Web sémantiques</b>		
2.1.	Introduction .....	38
2.2.	Fondements de l'approche proposée.....	38
2.2.1.	Classification des hétérogénéités des services Web .....	39
2.2.2.	Classification des mécanismes de médiation des services Web.....	41
2.3.	Description de l'architecture globale .....	42
2.4.	Processus de coopération de la couche d'interopérabilité.....	44
2.4.1.	Rôle du service Web broker.....	46
2.4.2.	Publication sémantique.....	46
2.4.3.	Découverte sémantique.....	47
2.4.4.	Médiation sémantique.....	47
2.4.5.	Exécution de services.....	47
2.5.	Description du modèle sémantique .....	48
2.5.1.	Description du sous modèle sémantique des données (DSSM, Data Semantic Sub Model) .....	49
2.5.1.1.	Notion de contexte de données et d'objet sémantique.....	50
2.5.1.2.	Description du modèle sémantique des données.....	51

2.5.2.	Description du sous modèle sémantique comportemental (BSSM, Behavioural Semantic Sub Model) .....	52
2.5.3.	Approche de conception et de création des ontologies.....	55
2.5.4.	Mécanismes d'annotation sémantique .....	55
2.6.	Conclusion .....	57
<b>Chapitre 3 : Une approche de médiation basée contexte pour l'interopérabilité des services Web sémantiques</b>		
3.1.	Introduction .....	60
3.2.	Niveaux de médiation pour l'interopérabilité des SWS .....	60
3.3.	Médiation des processus métiers .....	63
3.3.1.	Approche conceptuelle de la médiation des processus .....	64
3.3.2.	Fonctionnement du WSPM .....	65
3.3.3.	Etapas de la médiation sémantique des processus métiers .....	67
3.3.4.	Structure du module SST.....	69
3.3.5.	Rôle du module SST dans la médiation des processus .....	71
3.3.6.	Description des modules SynSemT et SemSynT .....	72
3.4.	Conclusion .....	72
<b>Chapitre 4 : Mise en œuvre de l'architecture et approche proposées</b>		
4.1.	Introduction .....	74
4.2.	Objectifs et fonctionnalités du prototype.....	74
4.3.	Création de la couche des services Web et du modèle sémantique des données.....	77
4.3.1.	Présentation de la première étude de cas .....	77
4.3.2.	Etapas d'implémentation de la couche des services Web .....	81
4.3.3.	Etapas d'implémentation de la couche sémantique des données .....	85
4.4.	Création de l'infrastructure de médiation des processus métiers .....	88
4.4.1.	Présentation de la deuxième étude de cas .....	88
4.4.2.	Etapas d'implémentation de l'infrastructure de médiation des processus...	92
4.5.	Conclusion .....	95
<b>Conclusion générale et perspectives .....</b>		97
<b>Références bibliographiques.....</b>		101
<b>A propos de l'auteur.....</b>		110

# Table des illustrations

Figure 1. Les trois acteurs des services Web.....	2
Figure 2. Vue globale de l'architecture proposée.....	5
Figure 1.1. Exemple de processus inter organisationnels .....	11
Figure 1.2. Schéma général d'un SIC [BEN 00] .....	12
Figure 1.3. Vue générale de l'orchestration .....	15
Figure 1.4. Vue générale de la chorégraphie .....	16
Figure 1.5. Fusion d'ontologies [MEL 07] .....	23
Figure 1.6. Processus de mapping d'ontologies [13].....	24
Figure 1.7. Classification basée sur les approches de description des services Web .....	28
Figure 2.1. Conflits de messages résolubles [CIM 05] .....	40
Figure 2.2. Architecture générale proposée pour l'interopérabilité sémantique des systèmes d'information.....	43
Figure 2.3. Processus global proposé pour l'interopérabilité sémantique .....	45
Figure 2.4. Sous-modèles sémantiques .....	49
Figure 2.5. Structure du sous-modèle sémantique comportemental.....	54
Figure 2.6. Exemple d'annotation de données .....	57
Figure 2.7. Exemple d'annotation comportementale .....	57
Figure 3.1. Médiation à plusieurs niveaux pour plusieurs tâches .....	61
Figure 3.2. Médiation pendant la découverte .....	62
Figure 3.3. Médiation pendant l'exécution .....	63
Figure 3.4. Architecture conceptuelle du WSPM .....	64
Figure 3.5. Description du WSPM .....	65
Figure 3.6. Interface du WSPM .....	67
Figure 3.7. Etapes de médiation des processus métiers .....	68
Figure 3.8. Architecture du module SST .....	69
Figure 3.9. Les trois cas de transition .....	70
Figure 3.10. Module SST dans le WSPM .....	71
Figure 4.1. Diagramme des cas d'utilisation du prototype.....	75
Figure 4.2. Flux d'information 'planification de voyage' .....	78
Figure 4.3. Diagramme de classes de la réservation de billet d'avion .....	89
Figure 4.4. Diagramme de classes de la réservation de chambre d'hôtels .....	78
Figure 4.5. Diagramme d'activité -client/services- .....	80
Figure 4.6. Test du SWba .....	81
Figure 4.7. Extrait de la méthode calculant le prix d'un billet d'avion .....	82
Figure 4.8. Test du service de réservation de chambre .....	82
Figure 4.9. Extrait de la méthode calculant le prix de chambres .....	83
Figure 4.10. Service de calcul total .....	83
Figure 4.11. Service process_webService .....	83
Figure 4.12 page d'accueil du client .....	84
Figure 4.13. Concepts de l'ontologie vol .....	86
Figure 4.14. Propriétés des concepts de l'ontologie vol .....	86
Figure 4.15. Relations entre concepts de l'ontologie vol .....	86
Figure 4.16. Concepts de l'ontologie hôtel .....	86
Figure 4.17. Propriétés des concepts de l'ontologie hôtel .....	86
Figure 4.18 Relations entre concepts de l'ontologie hôtel .....	86
Figure 4.19. Annotation du SWba .....	87
Figure 4.20. Annotation du SWch .....	87
Figure 4.21. Conversion de yen en euro .....	88
Figure 4.22. Chorégraphie de la demande de devis .....	88
Figure 4.23. BSSM des services Web client et fournisseur .....	91
Figure 4.24. Ontologie du SW demandeur.....	92

Figure 4.25. Ontologie du SW fournisseur.....	92
Figure 4.26. Ontologie SST .....	93
Figure 4.27. Règles SWRL .....	93
Figure 4.28. Une règle SWRL à exécuter (Exemple) .....	94
Figure 4.29. Résultat de l'exécution de la règle .....	94
Figure 4.30. Chargement de l'ontologie du SW demandeur (extrait).....	95
Tableau 2.1. Tableau comparatif des travaux existants sur l'interopérabilité des SWS.....	35
Tableau 4.1. Séquence d'échange de messages avec hétérogénéités .....	89
Listing 3.1. Algorithme du WSPM .....	66
Listing 4.1. Quote request Web service Requester Interface .....	89
Listing 4.2. Quote request Web service Provider Interface .....	90



# **Introduction générale**

# Introduction générale

## 1. Contexte de la recherche

Les contraintes affrontées par les entreprises sont souvent directement répercutées sur leurs systèmes d'informations. Ces derniers doivent sans cesse être agiles et adaptables afin de pouvoir mieux soutenir la stratégie de l'entreprise. Pour faire face aux exigences évolutives de leurs environnements amont et aval, les entreprises font appel à l'interopérabilité qui est devenue une nécessité incontournable. Elle peut être entre les applications de la même entreprise (intra-entreprise) ou entre les applications d'entreprises différentes (inter-entreprise). L'interopérabilité inter-organisationnelle introduit le paradigme de systèmes d'information coopératifs qui constitue le domaine de notre recherche. Tout au long de ces dernières années, et avec l'évolution d'Internet, une panoplie de techniques ont été mises en œuvre pour permettre l'interopérabilité des systèmes d'information tels que les intergiciels (en anglais middleware), l'ingénierie des modèles, les architectures orientées services et les services Web.

Depuis leur apparition dans les années 2000, les architectures orientées services (AOS) ou Service Oriented Architecture (SOA), se sont imposées rapidement comme une solution très efficace aux problématiques que rencontrent les entreprises en termes d'interopérabilité, de flexibilité, de réutilisation, et de réduction de couplage entre les différents systèmes. L'AOS est une démarche méthodologique de construction des systèmes d'information et d'implémentation du logiciel sous la forme de services. 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). Un service expose une interface qui définit le traitement offert sous la forme d'un message d'entrée et d'un autre de réponse. Le service respecte cinq propriétés qui sont: couplage faible, activable à distance, interopérable, asynchrone, expose un contrat d'utilisation (interface), et respecte le pattern d'architecture SOA [BON 05].

Le service métier 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 [BON 05]. Le meilleur moyen d'implémentation du SOA est l'utilisation des services Web.

Les services Web sont des intergiciels, mais de haut niveau, car ils sont implantés sur des intergiciels de base tels que les serveurs d'applications. « 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 » [1].

Dans l'utilisation des services Web, il y a trois acteurs qui interagissent ensemble (voir Figure 1):

- *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.

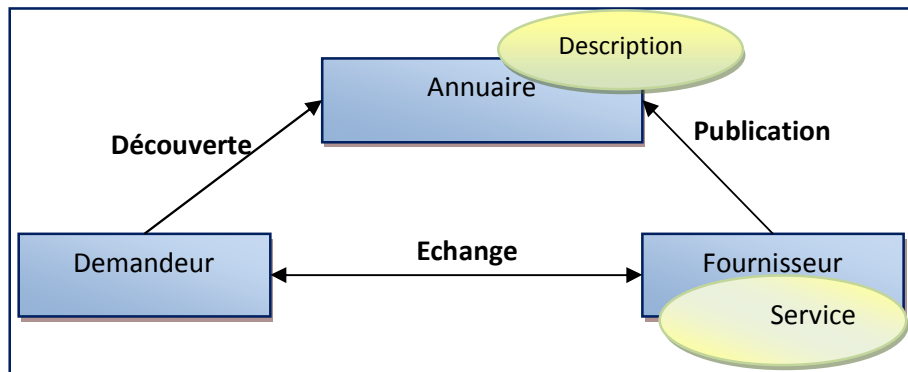


Figure 1. Les trois acteurs des services Web.

Les services Web s'appuient sur un ensemble de fonctions et de protocoles standards. Les fonctions permettent d'assurer notamment [MON 04]:

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

Ces fonctions sont supportées par des normes basées sur le langage XML (*Extensible Markup Language*) [MON 04, SIK 15, W3C 08] qui constitue une recommandation du W3C. C'est un métalangage informatique de balisage générique qui dérive du SGML (Standard Generalized Markup Language). La structure d'un document XML est définie et validée par un schéma. L'utilisation du langage XML a permis la création de nombreuses technologies, de langages, et de protocoles basés eux-mêmes sur ce langage (XHTML pour créer des pages Web ; Xpath, Xquery pour interroger des documents ; XSL, XSLT pour transformer des documents,...). De même, les standards des services Web (SOAP, WSDL, UDDI, SAWSDL, ...) sont basés XML. Tous ces langages dérivés permettent d'encoder d'une façon structurée l'information destinée à être partagée à travers Internet. La grande popularité du langage XML est due à sa simplicité, sa portabilité, son auto-description ainsi que son extensibilité. Il a été développé dans le but de permettre l'échange automatisé de contenus complexes entre systèmes hétérogènes [2].

Nous constatons donc que l'utilisation des architectures orientées services avec les différents standards des services Web offre plusieurs avantages, car réunis ensemble, ils favorisent la réutilisation, permettent l'interopérabilité, encapsulent la complexité, et conservent l'existant grâce à leur couplage faible.

## 2. Problématique

Les SOA et les services Web (SW) offrent des solutions pour l'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 services en interaction n'ont pas été conçus initialement pour inter-opérer ensemble. De ce fait, des hétérogénéités peuvent apparaître à plusieurs niveaux touchant leurs propriétés fonctionnelles (comportementales, fonctionnalités offertes, encodage, données échangées...etc.), ainsi que leurs propriétés non fonctionnelles (transactionnelles, de qualité des services, ...etc.). Ceci rend leur interopérabilité difficile, voire impossible. Pour remédier à ce problème une médiation sémantique pour réconcilier les parties en interaction est nécessaire.

Dans le cadre de cette thèse nous nous focalisons surtout sur la médiation des processus métiers. Fensel et Bussler [FEN 02] distinguent deux types de processus: des processus privés (orchestration) qui sont supportés intérieurement par une organisation et habituellement ne sont pas visibles à toute autre entité, et des processus publics (chorégraphie) qui définissent le comportement d'une entité métier (endpoint) en collaboration avec une autre entité qui est exprimé par échange de messages. Pour établir la communication, chaque entité doit comprendre le comportement des autres, et leurs comportements doivent correspondre [CIM 05]. Autrement dit, la communication entre SW demandeur et SW fournisseur peut avoir lieu seulement si leurs patterns d'échange de messages (MEPs : Message Exchange Patterns) correspondent précisément, quand un des deux envoie un message, l'autre est capable de le recevoir. Les hétérogénéités des MEPs (aspects comportementaux) sont plus complexes que celles des données. C'est à ces aspects comportementaux que nous nous intéressons dans notre recherche.

Fensel et Bussler [FEN 02] identifient trois cas de disparités qui peuvent apparaître pendant l'échange de messages: correspondance exacte, disparités de messages résolubles et disparités de messages non résolubles. Les disparités de messages résolubles peuvent susciter des problèmes d'ordre de messages, de combinaison ou éclatement de messages, des messages inattendus...etc.

Pour résoudre ces problèmes d'hétérogénéités des services Web et assurer leur intégration, il faudra répondre à ces deux principales questions :

***Comment modéliser et décrire la sémantique des services Web ?***

Sachant que les hétérogénéités sont diverses et concernent tous les aspects des services Web, cette tâche est difficile et constitue un domaine de recherche actif. La description sémantique des services Web permet de capturer explicitement les hétérogénéités dans le but de les traiter automatiquement. Malgré les avantages indéniables des ontologies pour capturer la sémantique, nous remarquons qu'elles présentent des limites dans le cadre des services Web. En effet, ces services sont déployés dans des environnements ouverts en perpétuels changements.

***Comment réconcilier les services Web en interaction ? Et avec quelle infrastructure ?***

Il s'agit de fournir une infrastructure basée services ainsi qu'un ensemble de mécanismes de médiation sémantique des séquences d'échange de messages, et des données véhiculées entre deux services Web en interaction pour permettre leur intégration. Cette médiation doit pouvoir utiliser les descriptions sémantiques pour qu'elle soit automatisable et dynamique.

### **3. Objectifs et contributions**

Nous rappelons que le travail de recherche envisagé se situe dans le cadre général de la coopération des systèmes d'information d'entreprises via les services Web sémantiques. Il a comme objectif principal la médiation des propriétés fonctionnelles et non fonctionnelles des services Web, dans le but de résoudre les conflits sémantiques pouvant apparaître entre le service Web émetteur et le service Web récepteur.

Notre approche globale a conduit à une architecture en couches [MEC 17] (voir Figure 2). La première couche représente les applications des systèmes d'information à l'aide des architectures orientées services, ce qui permet leur interopérabilité au niveau technique. Les services obtenus seront exposés aux organisations tierces à l'aide des standards des services Web. La deuxième couche concerne l'architecture permettant de décrire la sémantique des services Web obtenus précédemment. Elle est basée sur la notion d'ontologies et de contexte, et utilise les machines d'état abstraites (Abstract State Machine ou ASM) [BOR 08]. Elle constitue le modèle de référence pour annoter les descriptions des services Web à l'aide du langage SAWSDL (Semantic Annotation of Web Service Description Language) [HAU 09]. Ce modèle permet de capturer la sémantique des propriétés des services Web, aux niveaux fonctionnel (notamment les données et les caractéristiques comportementales) et non fonctionnel. 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 découverte, de médiation et d'exécution. Dans les chapitres 2 et 3 nous décrivons en détail la couche sémantique et l'infrastructure de médiation de la couche d'interopérabilité.

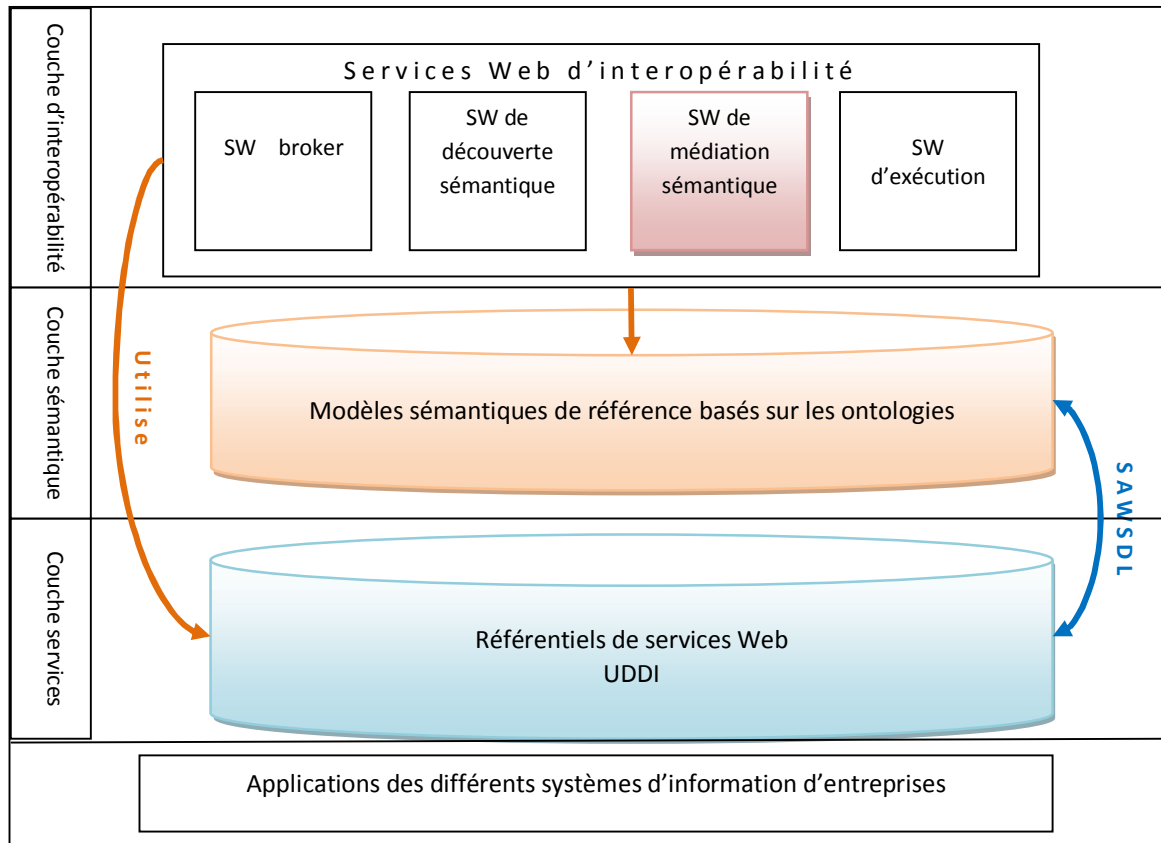


Figure 2. Vue globale de l'architecture proposée.

Pour répondre aux problématiques que nous avons citées, nos contributions, à travers cette architecture en couches, peuvent être résumées comme suit :

- Proposition d'un modèle sémantique de référence (couche sémantique) avec :
  - o des ontologies pour capturer la sémantique des propriétés fonctionnelles, comportementales, et non fonctionnelles des services Web,
  - o des ontologies des données échangées,
  - o définition de la notion de *contexte comportemental*, et
  - o utilisation des machines d'état abstraites (Abstract State Machines ou ASM).
- Proposition d'une approche d'annotation des descriptions des services Web par le modèle sémantique de référence en utilisant le langage SAWSDL (Semantic Annotation of Web Service Description Language), un standard qui fait partie des recommandations du consortium W3C. Il s'agit, en particulier, de l'annotation des données échangées entre

- services Web, et de leurs aspects comportementaux grâce aux attributs *'liftingSchemaMapping'*, *'loweringSchemaMapping'* et *'modelReference'*.
- Proposition d'une architecture basée services pour l'interopérabilité des systèmes d'information (couche d'interopérabilité), dont l'infrastructure de base est la médiation sémantique, orientée contexte, des données échangées entre services Web, et des processus métiers (chorégraphies).
  - Présentation d'une étude de cas illustrant notre approche et son implémentation.

#### 4. Organisation du document

Notre thèse est organisée en quatre chapitres en plus de cette introduction générale et d'une conclusion.

Le premier chapitre décrit, dans sa première partie, les notions de systèmes d'information coopératifs, selon l'angle des processus métiers, ainsi que les outils permettant leur interopérabilité technique, notamment les services Web. Nous abordons la composition de ces derniers ainsi que ses méthodes : l'orchestration et la chorégraphie. Des conflits entre les services Web en interaction peuvent apparaître, à cause de leurs hétérogénéités sémantiques. On note donc la nécessité d'une description sémantique de ces services en plus d'un système de médiation ce qui nous donne les services Web sémantiques. La deuxième partie explique d'abord la notion de sémantique ainsi que les approches de sa représentation, procédurale et déclarative. L'un des outils utilisés dans cette dernière approche est la notion d'ontologie. Quelques sections de cette partie sont consacrées, entre autres, à la structure et aux caractéristiques des ontologies, la classification des hétérogénéités de ces dernières et leurs différents types d'intégration, dont le mapping que nous utilisons dans notre approche. Les dernières sections sont dédiées aux langages de description sémantique des services Web, et les approches d'intégration de ces derniers. A la fin, nous effectuons une classification et une comparaison des travaux sur l'interopérabilité des services Web sémantiques pour positionner nos travaux par rapport aux recherches dans ce domaine.


Dans le chapitre 2, nous présentons l'architecture et l'approche globales que nous avons proposées pour l'interopérabilité des systèmes d'information via les services Web sémantiques dont une partie a été traitée dans nos travaux antérieurs [MEC 09, MEC 12]. La suite de ce chapitre est dédiée à la description de l'une des parties les plus importantes dans cette architecture, c'est le modèle sémantique dont le rôle consiste en la description des services Web, et la capture des hétérogénéités explicitement, pour pouvoir les traiter de façon semi-automatique par l'appel à la médiation. Ce modèle de référence est scindé en quatre sous modèles selon notre classification de la médiation. Notamment, le sous modèle sémantique des données [MEC 12] et le sous modèle sémantique des propriétés comportementales

(BSSM, Behavioural Semantic Sub Model). Pour décrire le BSSM, nous utilisons la méthode d'ASM pour sa force d'expressivité, et la notion de contexte que nous avons définie. Il comprend, pour chaque service Web, une ontologie locale comportementale et des ontologies contextuelles associées aux concepts de cette ontologie. L'annotation des services Web par ce modèle est prise en charge par le SAWSDL.

Le chapitre 3 montre d'abord, que la médiation qui constitue le noyau de notre architecture, est nécessaire à tous les niveaux et pour toutes les tâches des services Web. En effet, la médiation des propriétés fonctionnelles et celle des propriétés non fonctionnelles peuvent être utilisées pendant la découverte et la sélection. Cependant, la médiation des processus métiers et celle des données sont utilisées lors de l'exécution (invocation). Dans la suite de ce chapitre, nous décrivons la médiation des processus métiers qui est prise en charge par le service Web de médiation des processus WSPM (Web Service of Process Mediation). Ce dernier est implémenté par un algorithme de médiation de chorégraphie gérant la conversation entre les deux services Web en interaction. Il utilise les modèles comportementaux (BSSM) des deux parties et les règles de transition. Il fait appel au service Web de médiation de données WSDM (Web Service of Data Mediation) et à un raisonneur. Le cœur de ce médiateur est le module de transformations sémantiques-sémantiques basé sur la notion de mappings d'ontologies des BSSM et le raisonnement sur ces mappings.

Nous présentons dans le chapitre 4, l'implémentation de notre prototype de médiation basée contexte pour l'interopérabilité des SWS. Il est réalisé avec deux études de cas et en plusieurs étapes. Dans une première étape, nous effectuons la création de la couche des services Web et du sous modèle sémantique des données (DSSM : Data Semantic Sub Model). Dans la deuxième étape, nous procédons à la création du sous modèle sémantique des propriétés comportementales (BSSM : Behavioural Semantic Sub Model), avec des règles de transition. Ensuite, la création des mappings syntaxiques-sémantiques et sémantiques-syntaxiques. Et enfin, la création des mappings sémantiques-sémantiques par la création du module SST qui utilise, en plus des ontologies contextuelles, les règles de transitions prédéfinies.

Nous clôturons notre document par une conclusion et un ensemble de perspectives pour notre travail de recherche.



# Chapitre 1

## Interopérabilité des systèmes d'information via les services Web sémantiques

# Chapitre 1 : Interopérabilité des systèmes d'information via les services Web sémantiques

## 1.1. Introduction

Notre domaine de recherche est celui de l'interopérabilité des systèmes d'information à l'aide des architectures orientées services et des services Web. Les technologies de ces derniers ne permettent pas la découverte et l'exécution automatiques des services dans des environnements professionnels distribués et complexes. Les Services Web Sémantiques (SWS) surmontent cette limitation en ajoutant des descriptions sémantiques aux SW, ce qui permet la découverte automatique, la sélection, la composition et l'exécution de services pour des interactions machine-à-machine intelligentes et interopérables sur le World Wide Web (WWW).

Ce chapitre décrit dans une première partie, les systèmes d'information (SI), les processus métiers, la coopération des SI et les outils permettant leurs interopérabilités dont les services Web. Puis il détaille la notion de composition des services Web en processus métiers à l'aide de l'une des deux méthodes qui sont l'orchestration ou la chorégraphie. Quelle que soit la méthode de composition, des hétérogénéités peuvent apparaître entre les services en interaction, ce qui cause l'échec de leur interopérabilité. Dans la deuxième partie de ce chapitre, et pour introduire les SWS, nous expliquons d'abord la notion de sémantique et les approches de sa représentation, procédurale et déclarative. L'un des outils utilisés dans cette dernière approche est la notion d'ontologie. Nous abordons dans les sections qui suivent, la structure et quelques caractéristiques des ontologies, la classification des hétérogénéités de ces dernières et leurs différents types d'intégration, dont les correspondances (le mapping) que nous utilisons dans notre approche. La description des services Web par les ontologies conduit aux SWS. Ces derniers constituent l'objectif des sections qui présentent les langages de description sémantique des services Web ainsi que les approches de leur intégration. A la fin de ce chapitre, nous proposons une classification et une comparaison des travaux sur l'interopérabilité des services Web sémantiques afin de positionner nos travaux de recherche par rapport aux autres.

## 1.2. Systèmes d'information et processus métiers

Un système d'information est l'élément structurant d'une organisation. Ainsi, quand R. Reix le définit comme « *un ensemble organisé de ressources : matériel, logiciel, personnel, données, procédures permettant d'acquérir, traiter, stocker, communiquer des informations (sous forme de données, textes, images, sons, etc.) dans les organisations* » [REI 02], on

voit apparaître une notion essentielle, la procédure. Celle-ci décrit comment, quand et où le personnel est supposé utiliser matériel, logiciels et données pour que l'organisation soit informée. La diffusion de l'approche processus a conduit à intégrer la notion de processus dans la définition d'un système d'information [BIA 11] :

La définition de référence d'un processus métier est donnée par la norme ISO 9000:2000 : « *un ensemble d'activités corrélées ou interactives qui transforme des éléments d'entrée en éléments de sortie* » [BIA 11].

Un processus est un ensemble d'activités entreprises dans un objectif déterminé. La responsabilité d'exécution de tout ou partie des activités par un acteur correspond à un rôle. Le déroulement du processus utilise des ressources, et peut être conditionné par des événements, d'origine interne ou externe. L'agencement des activités correspond à la structure du processus [BIA 11].

Un processus est un ensemble d'activités visant à produire une sortie spécifique pour un client particulier, sur la base d'une entrée spécifique. Une activité est une fonction ou une tâche qui se produit au fil du temps et a des résultats reconnaissables [3].

A un niveau plus global, se pose la question des liens entre processus. La norme ISO 9000:2000 préconise d'avoir une vue globale de tous les processus et de leurs interactions : « Identifier, comprendre et gérer des processus corrélés comme un système contribue à l'efficacité et l'efficience de l'organisme à atteindre ses objectifs ».

Certains auteurs [HOF 00, WEI 02], ont proposé d'introduire la notion de contrat d'interface pour formaliser la communication entre processus relevant d'organisations différentes. En effet, les technologies de l'information permettent des organisations "globales", favorisées par l'ouverture des marchés et font coopérer des entités juridiques différentes. Cela signifie que l'on construit des systèmes de processus dont chaque partenaire n'a qu'une maîtrise partielle. L'explicitation de contrats contribue à une maîtrise élargie.

Soit, par exemple (voir Figure 1.1) un ensemble de processus se déroulant chez différents partenaires [BIA 11]. La vente en ligne s'effectue chez un hébergeur. Un processus de paiement, sous la responsabilité d'une banque, est appelé. Après conclusion de la vente, l'ordre est transmis au sous-traitant qui déclenche un processus d'expédition. Le commanditaire reçoit une image des transactions commerciales, notamment pour effectuer les règlements de son sous-traitant. Chacun des acteurs n'a qu'une vue limitée de l'activité des autres, d'où l'importance d'une explicitation claire des modalités de partage

d'information et de son contenu. Le contrôle du processus global ne peut qu'être réparti entre les acteurs.

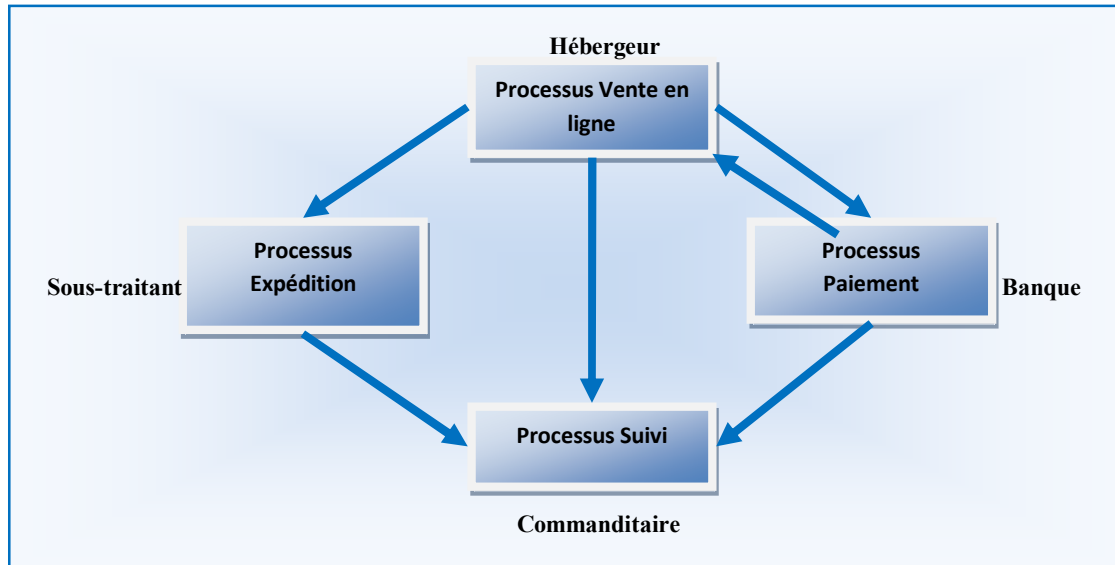


Figure 1.1. Exemple de processus inter organisationnels

Ce niveau global de processus inter-organisationnels introduit le paradigme de systèmes d'information coopératifs qui constitue le domaine de notre recherche. De ce fait, nous tenons à le définir, et à présenter la notion d'interopérabilité qui le caractérise.

### 1.3. Systèmes d'information coopératifs (SIC)

D'après Benslimane et al. [BEN 00], 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 1.2, 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 notions [BEN 00] :

**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.

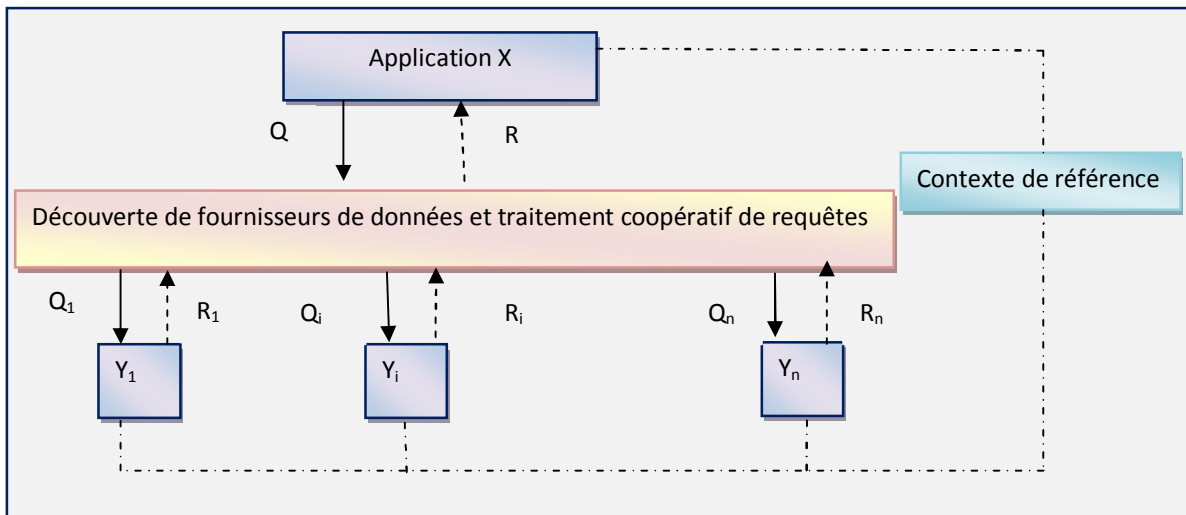


Figure 1.2. Schéma général d'un SIC [BEN 00].

### 1.3.1. Notions d'interopérabilité

Au cœur des systèmes d'information coopératifs, on trouve la notion d'interopérabilité à laquelle plusieurs définitions ont été données, par exemple, d'après Panetto et al. [PAN 05] « *L'interopérabilité des applications d'entreprises est définie comme la capacité pour un système d'échanger de l'information et des services dans un environnement technologique et organisationnel hétérogène* ». L'EIF (European Interoperability Framework) la définit comme « *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* » [EIF 04]. Dans ces définitions et d'autres, on souligne l'importance de la capacité d'échange d'information et de connaissance des systèmes d'information hétérogènes, et ceci, à plusieurs niveaux. L'EIF a proposé, notamment, trois niveaux d'interopérabilité [EIF 04] : interopérabilité technique ou des plates-formes, interopérabilité sémantique et interopérabilité organisationnelle [PAN 05, PAN 06, PAN 08, KLI 04].

Tout au long de ces dernières années, une panoplie de techniques ont été mises en œuvre pour permettre l'interopérabilité des systèmes d'information telles que les intergiciels, l'ingénierie des modèles, les architectures orientées services et les services Web. Dans le

cadre de nos travaux, nous avons retenu les deux dernières techniques en raison de leurs avantages multiples (voir section 1.3.2.).

### **1.3.2. Caractéristiques et intérêt des SOA et des services Web**

Les architectures orientées services (Service Oriented Architecture ou SOA), représentent une démarche méthodologique de construction des systèmes d'information et d'implémentation du logiciel sous la forme de services. 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). Un service expose une interface qui définit le traitement offert sous la forme d'un message d'entrée et d'un autre de réponse. Le service respecte cinq propriétés qui sont : couplage faible, activable à distance, interopérable, asynchrone, expose un contrat d'utilisation (interface), et respecte le pattern d'architecture SOA [BON 05]. Parmi les différents types de services construits à l'aide de SOA, nous nous intéressons au service métier, qui 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) [BON 05]. En pratique, il y a une correspondance directe entre le service Web et le service métier. De ce fait, la meilleure façon d'implémenter un service métier est d'utiliser les standards des services Web.

*D'après le W3C «Un service Web est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML. Sa définition peut être découverte par d'autres systèmes logiciels. Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet ».*

D'après cette définition, on note que les standards des services Web reposent essentiellement sur le langage XML pour la représentation des messages et des interfaces. De ce fait, un service Web possède les caractéristiques suivantes :

- Il est accessible via Internet.
- Son interface est publique et stockée dans un annuaire.
- Pour le transfert des messages XML il utilise principalement HTTP (sans écarter l'utilisation d'autres protocoles tels que : SMTP, FTP, BEEP...).
- L'intégration d'application à l'aide des services Web produit des systèmes faiblement couplés.

L'intérêt attribué aux services Web, est dû à leur compatibilité avec de nombreux autres environnements car ils sont indépendants des plates-formes, en raison de leur utilisation des standards XML et HTTP pour transférer les données. Les services Web permettent donc, aux entreprises, d'offrir des applications accessibles, à distance, par d'autres entreprises. Ils représentent la façon la plus efficace de partager des méthodes et des fonctionnalités. En outre, grâce à la réutilisation des services existants, ils réduisent considérablement le temps de réalisation. Pour récapituler, les services Web et les SOA présentent plusieurs avantages, ils :

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

### **1.3.3. Composition des services Web**

La composition consiste à combiner les fonctionnalités, de plusieurs services Web, dans un processus métier (business process) dans le but de répondre à des demandes complexes qu'un seul service ne peut pas satisfaire [MRI 07] [SHEN 14]. Les langages de composition sont par exemple, BPEL4WS (WS-BPEL), BPML...etc.

Fensel et Bussler [FEN 02] distinguent deux types de processus métiers: les processus privés (orchestration), qui sont réalisés en interne par une organisation, et ne sont généralement pas visibles à une autre entité, et les processus publics (chorégraphie), qui définissent le comportement des organisations, en collaboration avec d'autres entités.

#### **1.3.3.1. Orchestration**

Dans l'orchestration, un processus principal (service Web) prend le contrôle de la progression de la composition et coordonne les différentes opérations des différents services Web [REM 06] tel qu'un chef d'orchestre.

Les services Web invoqués ne connaissent pas, et n'ont pas besoin de savoir qu'ils sont impliqués dans un processus de composition, et qu'ils jouent un rôle dans la définition des processus métiers. Seul le processus central (coordinateur de l'orchestration) est conscient de cet objectif. Ainsi, l'orchestration est centralisée par des définitions comportementales explicites des opérations et l'ordre d'appel des services Web. Ces derniers décrivent les interactions entre services Web en identifiant les messages, et en spécifiant la logique et les séquences d'invocation. Le module exécutant les interactions de ces services Web est appelé un moteur d'orchestration (voir Figure 1.3). Ce module joue le rôle d'intermédiaire

entre les services Web, en les appelant suivant les interfaces comportementales de ces derniers [OUS 15].

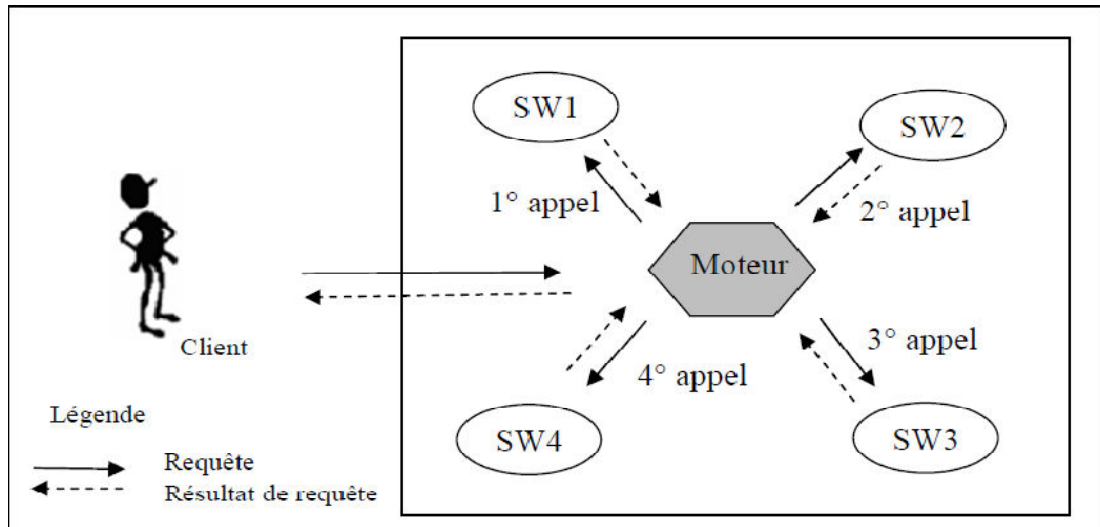


Figure 1.3. Vue générale de l'orchestration [OUS 15]

### 1.3.3.2. Chorégraphie

La chorégraphie ne repose pas sur un service Web principal. Chacun des services intervenants dans la composition sait précisément ce qu'il doit faire, quand il doit le faire et avec qui [REM 06].

La chorégraphie définit le comportement d'une entité métier, en collaboration avec une autre entité. Elle est généralement exprimée par un échange de messages. Pour établir la communication, chaque entité doit comprendre le comportement de l'autre et leurs comportements doivent correspondre [CIM 05].

La chorégraphie (*global model* in WSCI (Web Service Choreography Interface) and *multiparty collaboration* in ebXML) capture les processus de collaboration impliquant plusieurs services où les interactions entre ces services sont considérées dans une perspective globale. L'interface comportementale (*abstract process* in BPEL and *collaboration protocol profile* in ebXML) capture les dépendances comportementales des interactions dans lesquelles un service donné est impliqué.

La chorégraphie de services Web est une généralisation de l'orchestration qui consiste à concevoir une coordination décentralisée des services Web. Dans une chorégraphie, les interactions sont de type pair-à-pair (P2P) et décrites dans un langage de description de chorégraphie [OUS 15].

La chorégraphie est aussi appelée composition dynamique. En effet, l'exécution n'est pas régie de manière statique comme dans une composition de type orchestration. Dans une chorégraphie (voir Figure 1.4), à chaque pas de l'exécution, le service Web choisit le service Web qui lui succède et implémente ainsi une partie de la chorégraphie. La composition de type chorégraphie n'est pas connue, ni décrite à l'avance [KIL 10].

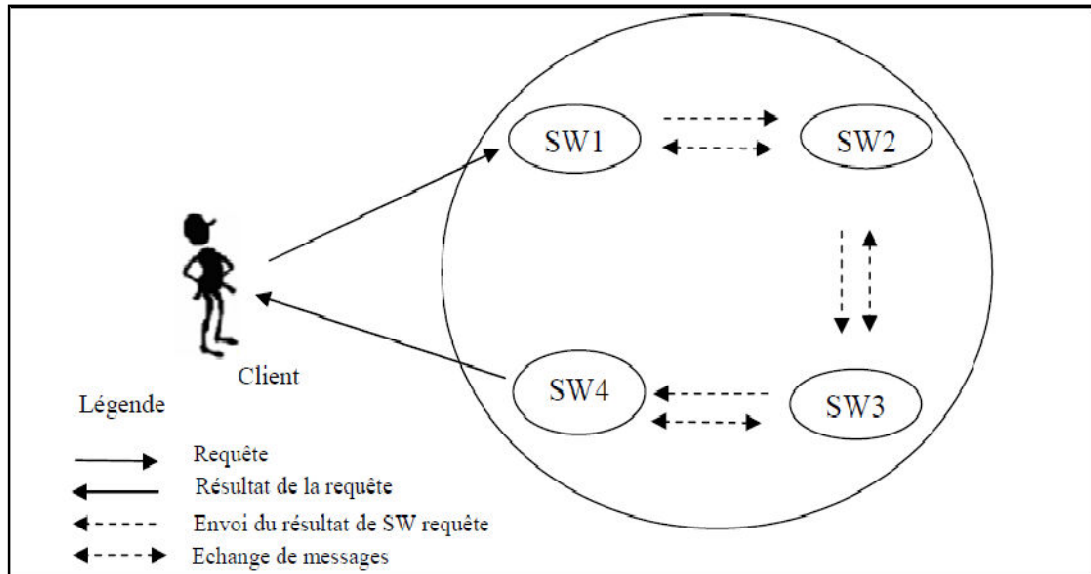


Figure 1.4. Vue générale de la chorégraphie [OUS 15]

### 1.3.4. Hétérogénéités des services Web en interaction

Vu la diversité des services Web, leur déploiement dans un environnement dynamique, ainsi que leur création indépendamment les uns des autres, des hétérogénéités apparaissent lors de leur interaction, et ceci à plusieurs niveaux. Dans le deuxième chapitre nous détaillons ces disparités et nous proposons une classification exhaustive couvrant tous les aspects des services Web. Dans ce qui suit nous nous limitons à les résumer en deux catégories [FAT 15, MRI 07]:

- **Hétérogénéités entre propriétés non fonctionnelles** : elles touchent les propriétés non décrites dans le WSDL tels que les propriétés transactionnelles, la qualité de service (QoS) et les performances des services Web.

- **Hétérogénéités entre propriétés fonctionnelles** : elles concernent les propriétés décrites dans un document WSDL, telles que les paramètres d'entrée/sortie, la fonctionnalité du service, le protocole d'échange utilisé, l'encodage, les données et les séquences d'échange de messages (MEPs, pour Message Exchange Patterns) représentant l'aspect comportemental du service.

### **1.3.5. Discussion : besoin de la sémantique**

L'adoption des services Web constitue une avancée fondamentale dans le développement des systèmes d'information interopérables, notamment, au niveau technique (plateformes). Néanmoins, lors de la communication des services Web, des hétérogénéités sémantiques, à plusieurs niveaux, peuvent apparaître, ce qui empêche leur interopérabilité ou leur composition. En effet, ces services ont été développés les uns indépendamment des autres, et ils n'ont pas été conçus pour interagir ensemble. Pour les réconcilier et permettre leur interopérabilité avec succès, une description sémantique et une médiation de ces services sont indispensables, ce qui nous donne les services Web sémantiques. Dans les sections suivantes, nous présentons la notion de sémantique et les ontologies qui permettent de la représenter, par la suite, nous décrivons les approches de description et d'intégration des services Web sémantiques.

Il est à noter que, dans nos travaux de recherche, nous nous intéressons à la médiation de la chorégraphie des services Web et leurs interfaces comportementales, c'est-à-dire les séquences d'échange de messages entre deux services.

## **1.4. Sémantique et ontologies**

### **1.4.1. Notion de sémantique**

La sémantique est définie comme étant l'étude du sens des mots, elle constitue une branche de la linguistique. En informatique, Woods [WOO 75] la définit comme « une forme formelle de représentation des connaissances humaines ».

Dans le domaine des systèmes d'information [IZZ 06], 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.

Uchold et Gruninger [USC 02] distinguent les différents types de sémantiques 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)** : 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** : 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** : 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.

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 qui étudie comment transformer l'expression du sens en une représentation formelle manipulable par une machine [IZZ 06]. Pour modéliser la sémantique, nous pouvons utiliser l'une des deux approches suivantes :

- 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.

La seconde est meilleure que la première du fait qu'elle présente des avantages tels que la standardisation, la capture, la réutilisation, l'inférence et la flexibilité [USC 02]. 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.

### 1.4.2. Notion d'ontologie

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 (Intelligence Artificielle). Il existe plusieurs définitions de la notion d'ontologie [GUA 95, SWA 97, GOM 00]. Nous retenons la définition de base de Gruber et celle d'Izza qui touche de près nos travaux de recherche :

- « *Une ontologie est une spécification explicite d'une conceptualisation* », [GRU 93].
- « *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* », [IZZ 06].

Les ontologies peuvent jouer divers rôles tels que [ZOU 06]:

- l'acquisition et représentation des connaissances ;
- la recherche et extraction des connaissances : inférer la connaissance qui est pertinente face à la requête de l'utilisateur ;
- le partage et intégration des connaissances : intégration des différentes sources d'information;
- la gestion des connaissances ;
- la simplification du dialogue homme-machine.

#### **1.4.2.1. Structuration des ontologies**

Une ontologie est constituée de plusieurs composants qui sont [GOM 00, GRU 93]:

**Les concepts** (termes ou classes) de l'ontologie, représentent 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. La description d'un ensemble d'objets, d'une tâche, d'une fonction, d'une stratégie, d'un processus de raisonnement, ...etc., correspondent à des concepts.

**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 : la relation de spécialisation (subsomption), la relation de composition (méronymie), 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 lesquelles le  $n$ ème élément (extrant) de la relation est défini de manière unique à partir des  $n-1$  éléments précédents (intrants). Formellement, les fonctions sont définies ainsi :  $F : C_1 \times C_2 \times \dots \times C_{n-1} \times C_n$ . 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.

#### 1.4.2.2. 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 jusqu'à nos jours, à la naissance d'une panoplie de langages de description d'ontologies et, parallèlement, de nombreux outils pour leur construction. Pour une ample description, nous trouvons dans la littérature abondante des listes exhaustives ainsi que des comparaisons de ces langages et outils [COR 03, GOM 00, IZZ 06, ONT 00, SIK 15] [4].

Les langages de représentation des ontologies basés Web ont pour socle de base le langage XML, et sont résumés comme suit :

- **SHOE** (Simple HTML Ontology Extensions).
- **XOL** (XML Ontology Language).
- **RDF et RDFS** (Resource Description Framework et RDF Schema) [5].
- **DAML+OIL** : DAML-ONT (DARPA Agent Markup Language) est associé à OIL (Ontology Inference Layer) et repose sur RDF.
- **OWL** (Ontology Web Language) est un vocabulaire XML basé sur RDF.

Pour décrire nos ontologies nous utilisons OWL, que nous détaillons dans ce qui suit : **OWL** : 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 (informations) 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 [OWL 04]:

- OWL Lite : c'est une version d'OWL aux fonctionnalités réduites, mais suffisant pour bien des usages, comme la constitution de taxonomies ou de thésaurus ;
- 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 [5,6].

OWL Lite 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,

ObjectTypeProperty, allValuesFrom, someValuesFrom, Cardinality, MinCardinality, MaxCardinality, Ontology, Imports ...etc. [OWL 04].

Comme nous l'avons déjà mentionné, il existe une panoplie d'outils de construction d'ontologies tels qu'Ontolingua, Ontosaurus, Tadzebao, WebOnto, et Protégé 2000 [COR 03, GOM 00, ONT 00]. Pour notre part, nous utilisons Protégé que nous présentons dans ce qui suit:

**Protégé 2000** : développé au département d'Informatique Médicale de l'Université de Stanford, appartient aux outils dits de la seconde génération [7]. Protégé-2000 est un éditeur 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 base 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) [8]. Des « plugin » pour les langages RDF, DAML+OIL et OWL ont été développés pour Protégé. On peut intégrer à Protégé des outils: un moteur d'inférence tel que JESS [9], ou des outils d'inférence spécifiques au Web sémantique basés sur des logiques de description [10] tel que RACER [11].

A présent, nous introduisons un bref aperçu sur les hétérogénéités des ontologies et leur intégration.

### **1.4.3. Intégration des ontologies**

Il existe principalement trois approches d'utilisation des ontologies [WAC 01] qui sont l'approche mono-ontologie, l'approche multi-ontologie et l'approche hybride. Les deux dernières sont à l'origine du problème d'intégration des ontologies à cause de diverses hétérogénéités.

#### **1.4.3.1. Hétérogénéités des ontologies**

Plusieurs classifications d'hétérogénéités existent [KLE 01, NOY 00, GOH 97, IZZ 06...]. Le projet Knowledge Web [12] distingue fondamentalement quatre niveaux d'hétérogénéités qui sont:

- les hétérogénéités de niveau syntaxique : liées aux caractéristiques des langages utilisés pour représenter les ontologies.

- les hétérogénéités de niveau terminologique : concernent toutes les différences liées au processus de nomination des entités (classes, propriétés, etc.), telles que la synonymie, la polysémie ...etc.
- les hétérogénéités de niveau conceptuel : concernent les différences de couverture de l'univers du discours, le degré de granularité et le point de vue de l'ontologie.
- des hétérogénéités de niveau pragmatique : il s'agit des hétérogénéités d'interprétation d'une ontologie. Ces hétérogénéités peuvent survenir lorsque des individus ou des communautés différentes interprètent différemment l'ontologie selon différents contextes.

#### **1.4.3.2. Différents types d'intégration d'ontologies**

La médiation d'ontologies est le processus qui permet de réconcilier différentes ontologies. Elle se base généralement sur le principe de la combinaison d'ontologies. Ce processus de combinaison est très souvent appelé intégration d'ontologies. Ceci signifie, à la fois, le fait que les ontologies peuvent être fusionnées en une ontologie unique nouvellement créée, et aussi le fait que les ontologies peuvent être mises en correspondance sans création de nouvelle ontologie [KLE 01]

Un certain nombre de techniques ont été proposées dans la littérature pour réaliser l'intégration des ontologies [BRU 06], [6]. Nous retenons les techniques suivantes: le mapping, l'alignement et la fusion d'ontologies. Dans le cadre de cette thèse nous nous focalisons sur les mappings.

##### *Alignement d'ontologies*

Le processus d'alignement d'ontologies permet d'amener deux ou plusieurs ontologies hétérogènes à un " accord mutuel" en les rendant ainsi consistantes et mutuellement cohérentes. L'alignement d'ontologies nécessite la transformation des ontologies impliquées par l'élimination des entités non pertinentes et le rajout des entités manquantes [IZZ 06].

##### *Fusion d'ontologies*

Le processus de fusion d'ontologies est fondé sur deux ontologies d'entrée (ou plus) et retourne une seule ontologie basée sur les ontologies sources. La fusion manuelle d'ontologies à l'aide d'outils conventionnels d'édition sans support est longue, difficile, et sujette à l'erreur. Par conséquent, plusieurs approches de fusion sont proposées [DRA 14, GAE 02] telles que PROMPT [NOY 00], CHIMAERA [MCG 00], FCAMerge [STU 01] et OntoMerge [DOU 05] (voir Figure 1.5).

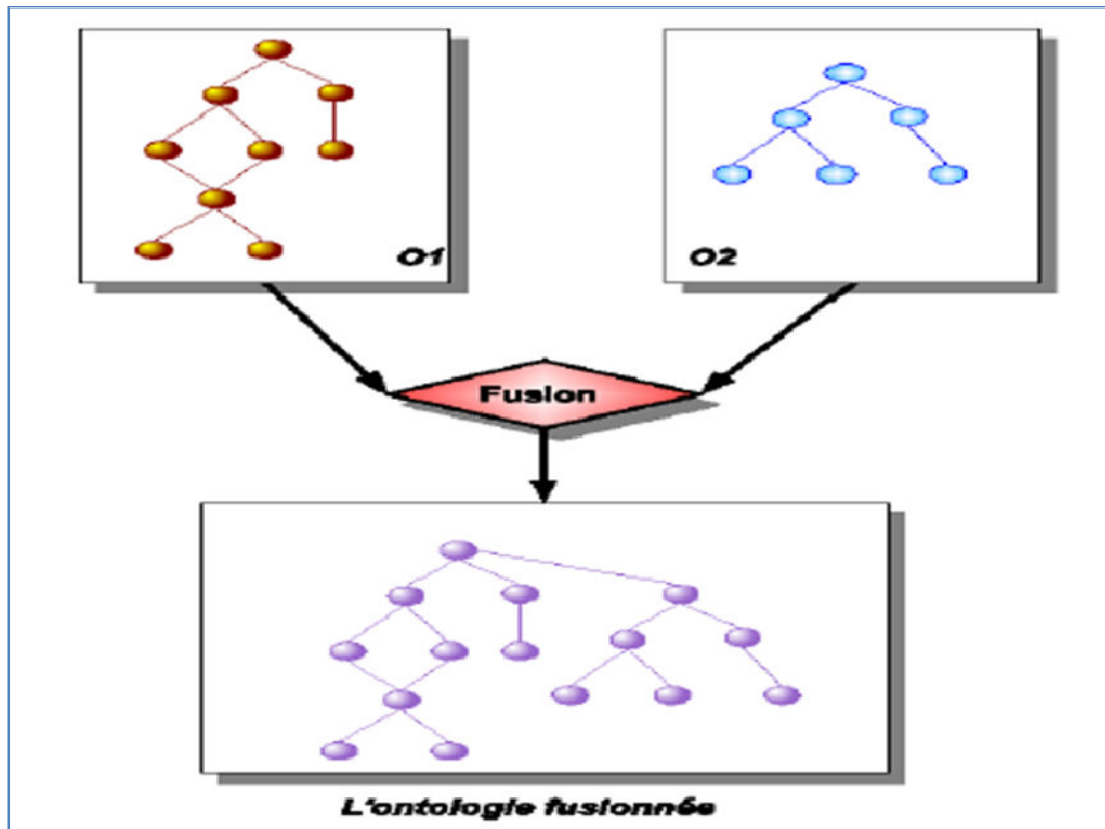


Figure 1.5. Fusion d'ontologies [MEL 07]

### *Mapping d'ontologies*

Le mapping d'ontologies peut être défini comme un ensemble de correspondances au sens mathématique entre un certain nombre d'ontologies [12]. 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 (voir Figure 1.6) [13]:

- L'importation des ontologies : qui consiste à charger les ontologies dans l'outil de mapping en effectuant, éventuellement, des translations de format ;
- La recherche de similarités : qui consiste à trouver, 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, éventuellement de façon semi-automatique, en utilisant un outil tel que PROMPT.

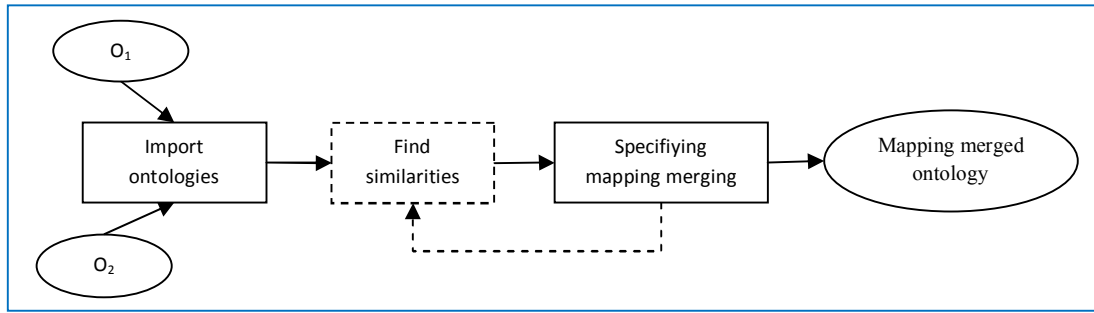


Figure 1.6. Processus de mapping d'ontologies [13].

Plusieurs outils de mapping ont été développés [DRA 14], nous citons à titre d'exemple : GLUE [DOA 02], FCA-Merge [STU 01], et PROMPT [NOY 03], DRAGO [14, 16], ServOMap [15, 16]. Mais, on note que le processus de mapping est semi-automatique, et il peut conduire parfois à des changements dans au moins une des ontologies sources.

Les mappings peuvent être représentés de plusieurs façons. Noy et Musen [NOY 03] proposent trois manières de les représenter. La première est une représentation déclarative des mappings en tant qu'instances dans une ontologie. La seconde consiste à définir des axiomes de transformation, alors que la troisième est l'utilisation de vues pour décrire des mappings entre une ontologie globale et une ontologie locale.

Les mappings sont définis pour être utilisés dans des raisonnements à des fins d'intégration diverses comme la médiation et la composition des services Web. Dans nos travaux de recherche, nous représentons les mappings à l'aide d'axiomes et de règles prédéfinis qui seront utilisés par des raisonneurs (Jena, Pellet).

## 1.5. Services Web sémantiques

Pour résoudre le problème d'hétérogénéités des services Web et assurer leur intégration, nous avons besoin, d'une part, d'une description sémantique de ces derniers à l'aide d'ontologies, ce qui nous donne les services Web sémantiques, et, d'autre part, d'une médiation de ces services. Cette dernière est alors automatisable et dynamique grâce à la description sémantique.

Pour atteindre l'interopérabilité sémantique, les services Web doivent être capables, d'une part, d'interpréter correctement la sémantique des données qu'ils envoient et reçoivent, et d'autre part, de décrire les fonctionnalités qu'ils fournissent en utilisant une sémantique explicite et compréhensible par les machines [MRI 07].

Nous distinguons principalement deux approches permettant de décrire sémantiquement les services Web [MRI 07] : les approches exploitant les langages de description sémantique

et les approches d'annotations de langages existants. En outre, plusieurs approches d'intégration des services Web existent. Nous les présentons dans les sections suivantes.

### 1.5.1. Approches utilisant les langages de description sémantique.

Les langages qui décrivent sémantiquement les services Web sont issus du Web sémantique. Ils tentent de remplacer les langages existants des services Web pour élever ces derniers au rang des services Web sémantiques. C'est le cas des langages OWL-S, DE, DSD ainsi que de l'architecture conceptuelle WSMO:

- **Le langage Ontology Web Language for Web Services (OWL-S)** est un sous ensemble du langage OWL (Ontologie Web Language) [McG 04] dédié à la description sémantique de services Web [MAR 07]. Une description OWL-S se compose de trois éléments [MAR 07] [Sik 15] : *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. La vue abstraite ou de haut niveau (service profile et process model), relie le service Web à des descriptions conceptuelles en OWL, décrites dans des ontologies. La vue concrète ou de bas niveau (le grounding) décrit la représentation physique du service Web.

- **Le langage Web Service Modeling Ontology (WSMO)** est une architecture conceptuelle dont le but est de décrire la sémantique des services Web [ARR 04]. Le méta-modèle du WSMO définit quatre éléments principaux : *les ontologies*, *les objectifs*, *les services Web* et *les médiateurs*. *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 [Sik 15].

La description concerne aussi la manière avec laquelle les services communiquent (chorégraphie) et celle avec laquelle ils sont composés (orchestration). *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.

- ***Les langages DIANE Elements (DE) et DIANE Service Description (DSD)***. 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 [KLE 05]. Ces langages utilisent les notions d'ensembles configurables et de logique floue pour améliorer la découverte sémantique de services.

### **1.5.2. Approches d'annotation des langages existants.**

Les travaux de cette catégorie 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) [PEE 05], 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** [PAO 02] [DOG 04] et **SAWSDL** [HAU 09], 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** [SHE 08] [MIL 04], soit en utilisant le **SAWSDL** qui constitue une recommandation du W3C.

WSDL-S [MIL 04] fournit des mécanismes pour annoter les descriptions des services Web avec des concepts sémantiques définis dans un modèle de domaine externe. En utilisant l'extensibilité des éléments et des attributs XML, les annotations sémantiques sur les éléments WSDL sont achevées par leur référencement aux concepts sémantiques d'un ou de plusieurs modèles de domaine externes (ontologie).

SAWSDL [HAU 09] définit des attributs d'extension qui peuvent être appliqués aux éléments du WSDL et du XML Schema pour annoter les interfaces, les opérations, et les messages d'entrée et de sortie du WSDL. Les extensions SAWSDL prennent deux formes: *model references* et *schema mappings*. Le *modelReference* est un attribut d'extension, *sawSDL:modelReference*, qui peut être appliqué à n'importe quel élément du WSDL ou XML Schema dans le but de pointer vers un ou plusieurs concepts sémantiques. SAWSDL fournit deux attributs pour attacher les mappings de schema XML, *sawSDL:liftingSchemaMapping* et *sawSDL:loweringSchemaMapping*. Lifting mappings transforment la donnée XML d'un message de service Web vers un modèle sémantique, alors que lowering mappings transforment la donnée d'un modèle sémantique vers un message XML.

### 1.5.3. Approches d'intégration des services Web

On distingue plusieurs types d'approches [MRI 07, SHE 14]:

- **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 [MED 04].
- **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 [TAH 06].
- **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 médiation utilisent ces descriptions améliorées pour réconcilier les services Web [DUM 06].
- **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 [CIM 05, DOM 08, HAL 05, IZZ 08, MID 16, VIT 09 ...].

## 1.6. Classification et comparaison de travaux sur l'interopérabilité des SWS

Après étude de plusieurs travaux de recherche, nous avons constaté que les approches orientées services et celles basées sur les descriptions étendues constituent actuellement les approches les plus prometteuses car elles bénéficient des avantages des services Web en utilisant les langages et les normes existants (voir section 1.5.3). C'est pour cette raison que nous avons choisi de discuter quelques travaux ayant adopté ces approches.

Dans cette section, nous proposons une classification et une comparaison des travaux relatifs à l'interopérabilité sémantique via les services Web en se basant sur des critères que nous avons sélectionnés [MEC 10a, MEC 17]. Nous distinguons deux principaux types de travaux, ceux basés sur les langages sémantiques, et ceux basés sur les annotations sémantiques des langages existants (voir Figure 1.7).

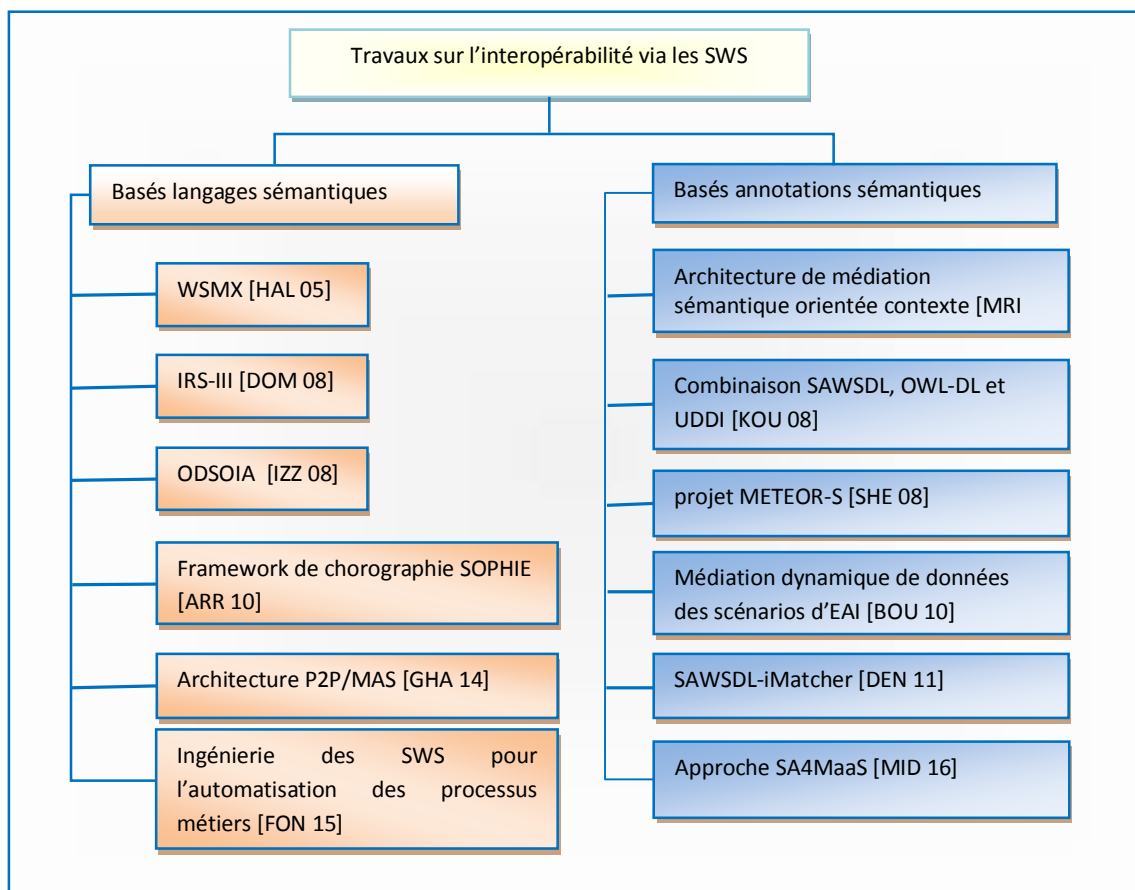


Figure 1.7. Classification basée sur les approches de description des services Web

## **1.6.1. Travaux basés sur les langages sémantiques**

### **1.6.1.1. Architecture WSMX (2005)**

L'architecture orientée services sémantiques WSMX de Haller et al. [HAL 05], [17], est un environnement d'exécution des services Web basé sur le Web Services Modelling Ontology WSMO (voir section 1.5.1). 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 des processus métier [CIM 05] 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 fonctionnelle 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 de deux composants. Le premier identifie 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 générées par le premier composant.

L'architecture WSMX est utilisée par Vitvar et al. [VIT 09] pour présenter la solution de DERI pour résoudre le scénario de médiation des SWS. Cependant, Vitvar et al. [VIT 09] ne donnent pas de sémantique pour l'interaction de messages, de ce fait, leur algorithme respecte seulement les modèles "messages combinés" et "arrêt d'un message inattendu".

### **1.6.1.2. Architecture IRS-III (2008)**

L'architecture IRS-III (IRS: Internet Reasoning Service) [CAB 05, CAB 12, DOM 08] 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 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 [Sik 15].

L'architecture IRS-III est utilisée par A. Gugliotta et al. [GUG 05] comme une infrastructure de base pour l'interopérabilité sémantique au sein d'une architecture d'un portail e-gouvernemental. Cette infrastructure utilise trois médiateurs appartenant à un même composant appelé "broker" (courtier).

Un médiateur de processus est utilisé 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 fonctionnelle 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. Ainsi, il est possible de convertir les données d'une ontologie à l'autre en passant par une représentation intermédiaire.

### **1.6.1.3. Architecture ODSOIA (2008)**

Izza [IZZ 08] 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 (Entreprise Service Bus) 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 OWL-S.

Les deux premières architectures sont utilisées par l'architecture d'intégration composées de services d'intégration qui sont : le service broker permettant la gestion du processus d'intégration des services fondamentaux, le service de description sémantique, le service de publication sémantique, le service de découverte, le service de médiation qui prend en charge la résolution des hétérogénéités sémantiques liées aux services fondamentaux, et le 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: service de médiation de données, service de médiation fonctionnelle et service de médiation de processus. Dans ces différentes médiations, plusieurs types d'ontologies et de mappings sont utilisés.

#### **1.6.1.4. Framework de chorégraphie SOPHIE (2010)**

SOPHIE [ARR 10] est un Framework conceptuel et une architecture pour la chorégraphie des services réalisée comme une SOA. Le modèle conceptuel décrit la structure, le comportement, le fonctionnement et les ontologies comme des préoccupations distinctes. Le modèle de comportement (behavioral model) est défini en utilisant les machines à états finis (Finite State Machines: FSM). Un ensemble d'algorithmes sont présentés pour générer des médiateurs pour surmonter les hétérogénéités des séquences d'échange de messages (MEP, Message Exchange Patterns) des parties en interaction. Cependant, ces algorithmes ne sont pas encore mis en œuvre et les hétérogénéités des MEPs qu'ils surmontent ne sont pas mentionnées.

#### **1.6.1.5. Architecture P2P / MAS (2014)**

P2P / MAS [GHA 14] est une architecture générique de Systèmes Multi-Agents (Peer to Peer Multi-Agents System) pour la découverte et la composition des services Web sémantiques. Les auteurs montrent comment les réseaux P2P (Peer to Peer ou pair à pair) peuvent mettre en œuvre des architectures de systèmes multi-agents ouverts (open-MAS architectures) pour construire un système distribué collaboratif. Pour la description sémantique des services Web ils utilisent OWL-S et les automates finis pour les comportements. Cependant, l'approche ne traite pas des hétérogénéités.

#### **1.6.1.6. Ingénierie des SWS pour l'automatisation des processus métiers gouvernementaux (2015)**

Fonou-Dombeu et al., [FON 15] proposent une infrastructure pour les services Web sémantiques du e-gouvernement qui intègre le Business Process Modelling (BPM) et les annotations sémantiques dans les solutions des SWS en vigueur comme des outils de

modélisation et d'ingénierie des SWS pour des opérations et des processus gouvernementaux non-automatisés. L'infrastructure proposée comprend, entre autres, trois types d'ontologies : de domaine, d'utilisateur et de service. Elle exploite le Web Service Modelling Ontology (WSMO) et peut être implémentée via IRS-III et WSMX.

## **1.6.2. Travaux basés sur les annotations sémantiques**

### **1.6.2.1. Architecture de médiation sémantique orientée contexte (2007)**

Mrissa [MRI 07] propose une architecture de médiation sémantique orientée contexte pour la composition de services Web. C'est une architecture comprenant 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 défini 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.

### **1.6.2.2. Combiner SAWSDL, OWL-DL et UDDI pour la découverte de services Web rehaussés sémantiquement (2008)**

Kourtesis et al. [KOU 08] présentent l'implémentation du registre sémantique du Middleware Fusion, un registre de services augmenté sémantiquement qui est constitué des spécifications UDDI et augmente ses services de publication et de découverte. Les auteurs combinent l'utilisation du SAWSDL pour créer les annotations sémantiques des descriptions des interfaces de service avec l'utilisation d'OWL-DL afin de modéliser les fonctionnalités du service et réaliser les matchmaking via un raisonnement basé sur la logique de description.

### **1.6.2.3. Le projet METEOR-S (2008)**

Dans le projet METEOR-S: Semantic Web Services and Processes, Sheth et al. [SHE 08] ont défini quatre types de sémantique pour les services Web : la sémantique des données, les sémantiques fonctionnelle, non-fonctionnelle et d'exécution qui sont utilisées dans le cycle de vie complet des processus du Web sémantique. L'environnement d'exécution gère les hétérogénéités aux niveaux protocole et données en utilisant des proxies avec des capacités de médiation des données et des protocoles. Pour la capture des protocoles d'interaction, des diagrammes d'activité UML sont utilisés. L'architecture de médiation de données [NAG 06] utilise les fonctionnalités d'extensibilité de WSDL et le moteur

populaire SOAP Axis2. METEOR-S met l'accent sur la composition et la médiation de données.

#### **1.6.2.4. Médiation dynamique de données dans des scénarios d'intégration des applications d'entreprises (2010)**

Bouras et al. [BOU 10] proposent une approche d'enrichissement sémantique pour la médiation dynamique des données dans des scénarios EAI. Elle se focalise sur la résolution d'hétérogénéités au niveau messages entre des services collaboratifs d'entreprise, et facilite la médiation automatique des données durant l'exécution en fournissant des transformations formelles des messages d'entrée et de sortie vers un modèle de référence commun de données métier : l'ENterprise Interoperability Ontology (ENIO). ENIO constitue une ontologie multicouches et multi-facettes, qui fournit une compréhension commune et partagée des données, des services et des processus dans les cas d'applications d'entreprises. Bouras et al. [BOU 10] utilisent un mécanisme d'annotation "modelReference" (SAWSDL attribute) pour pointer vers un concept ontologique de données. Le code de transformation XSLT créé, est stocké dans un registre commun. La référence vers ces attributs est ajoutée à la description sémantique du service.

#### **1.6.2.5. SAWSDL-iMatcher (2011)**

Dengping et al. [DEN 11], proposent un matchmaker personnalisé et efficace des services Web sémantiques. Il est basé sur iXQuery qui étend XQuery avec des jointures de similarité. Les auteurs ont montré comment iXQuery combine les requêtes structurées avec des jointures de similarité pour réaliser un service SAWSDL matchmaking et comment les utilisateurs peuvent facilement personnaliser leurs stratégies d'appariement préférées dans SAWSDL-iMatcher dans le but de découvrir les services Web adéquats. Les ontologies décrivant les fonctionnalités de ces services sont exprimées avec OWL.

#### **1.6.2.6. Approche SA4MaaS (2016)**

Midouni et al. [MID 16] définissent une approche de service complet, appelée SA4MaaS, afin de composer les services MaaS (Mobility as a service) pour la recherche de données multimédias. Cette approche repose sur un processus en quatre étapes: la description, le filtrage, le regroupement et la restitution. Plusieurs approches ont été définies, dans le but de décrire sémantiquement les services Web. Les auteurs proposent une nouvelle approche, Annotation sémantique pour les services MaaS (SA4MaaS), en tant que prolongement de la recommandation du W3C sur la sémantique pour les services Web (SAWSDL). Ils utilisent deux ontologies: une ontologie de domaine, dont les concepts définissent une sémantique

du domaine d'activité, et une ontologie multimédia, dont les concepts définissent un ensemble de propriétés d'annotation pour décrire le contenu multimédia.

### **1.6.3. Comparaison des travaux sur l'interopérabilité des SWS**

Dans le Tableau 1.1, nous comparons quelques travaux dans le domaine de l'interopérabilité sémantique via les services Web grâce à des critères que nous avons définis. Nous commençons par les différencier selon les approches et langages de description sémantiques (voir sections 1.5.1 et 1.5.2), et les approches d'intégration (voir section 1.5.3), puis selon les fonctions prises en compte (publication, découverte, invocation, ...), le niveau de médiation (données, fonctions, processus) (voir section 2.2.2) et la méthode de composition (chorégraphie/ orchestration) (voir section 1.3.3).

Nous distinguons donc, dans ce tableau, deux catégories importantes, celle des approches basées sur les langages de description sémantique dérivés du Web sémantique (cas des six premiers travaux dans le Tableau 1.1), et celle des approches basées sur les annotations des descriptions des services Web, qui exploitent les éléments d'extensibilité des langages existants (cas des six derniers travaux dans le Tableau 1.1). Ces annotations présentent plus d'avantages que les langages de description du moment qu'elles exploitent les normes existantes des services Web, et préservent leurs avantages, notamment la réutilisation et le couplage faible, en outre, elles sont relativement simples.

Pour pouvoir effectuer la médiation, tous ces travaux supposent l'adaptation de la sémantique locale des services Web à la sémantique de l'ontologie partagée. Donc, il n'y a pas une vraie médiation excepté celle proposée par Mrissa [MRI 07].

Dans nos travaux précédents [MEC 12], nous avons détaillé la médiation de données où nous utilisons la notion de contexte et les attributs SAWSDL. En ce qui concerne la représentation et la médiation des processus métiers, nous utilisons la méthode des machines à états abstraits (Abstract State Machines ou ASM) comme les travaux de Domingue et al. [DOM 08], Cimpian et Mocan [CIM 05] et Cabral et Domingue [CAB 05]. Cimpian et Mocan [CIM 05] décrivent l'aspect comportemental à l'aide des propriétés non fonctionnelles, cependant, dans notre approche, nous le décrivons en utilisant la notion de contexte comportemental que nous avons introduit.

Étant donné que nous adoptons l'approche orientée contexte, notre modèle conceptuel nous permet d'avoir une médiation sémantique dynamique, car il offre les avantages suivants:

- Il n'y a pas une représentation unique de l'information (ontologie commune), ainsi, les fournisseurs de services ne sont pas tenus d'adapter leur sémantique locale.

- Les hétérogénéités sémantiques sont explicitement représentées. Elles peuvent donc être interprétées et résolues automatiquement.

Critère Référence	Approche de description sémantique	Langage de description /annotation sémantique	Approche d'intégration des SW	Fonctions des SW	Niveau de médiation	Méthode de composition	Présentation comportementale
Haller et al. 2005 (WSMX) [CIM 05, HAL 05, VIT 09]	Langage de description sémantique	WSMO	Basé SW	Découverte Invocation Médiation	Processus Fonctions Données	Chorégraphie Orchestration	ASM
Dominge et al. 2008 (IRS_III) [DOM 08]	Langage de description sémantique	WSMO	Basé SW	Publication Médiation Exécution	Processus objectif données	Chorégraphie Orchestration	ASM
Izza et al. 2008 (ODSOIA) [IZZ 08]	Langage de description sémantique	OWL_S+	Basé SW	Publication Découverte Médiation Exécution	Processus Fonctions Données	Orchestration	---
Arroyo et Sicilia 2010 (SOPHIE) [ARR 10]	Ontologies	Ontologies	Basé service	Découverte	Processus Données	Chorégraphie	FSM
Gharzouli et Derdour 2014 [GHA 14]	Langage de description sémantique	OWL_S	Basé SW	Publication Découverte Exécution	Processus	Chorégraphie	Automates finis
Fonou-Dombeu et Huisman 2015 [FON 15]	Langage de description sémantique	WSMO	Basé SW	Découverte Invocation Médiation	Processus Fonctions Données	Chorégraphie Orchestration	ASM
Mrisa 2007 [MRI 07]	Annotation des langages existants	WSDL_S & Context	Basé SW & description étendue	Médiation	Données	Orchestration	---
Kourtesis et al. 2008 [KOU 08]	Annotation des langages existants	SAWSDL	Basé SW & description étendue	Publication Découverte Matchmaking	Fonctions.	---	---
SAWSDL-iMatcher 2011 [DEN 11]	Annotation des langages existants	SAWSDL	Basé SW & description étendue	Découverte Matchmaking	Fonctions	---	---
Sheth et al. 2008 (METEOR_S) [SHE 08]	Annotation des langages existants	WSDL_S	Basé description étendue	Publication Découverte Médiation Exécution	Données Protocoles	Orchestration	Diagrammes d'activités UML
Bouras et al. 2010 [BOU 10]	Annotation des langages existants	SAWSDL	Basé description étendue	Médiation	Données	---	---
Midouni et al. Approche SA4MaaS 2016 [MID 16]	Annotation des langages existants	SAWSDL	Basé SW & description étendue	Publication Découverte Exécution	---	---	---

Tableau 1.1. Tableau comparatif de travaux sur l'interopérabilité des SWS.

## 1.7. Conclusion

Ce chapitre a été consacré à l'interopérabilité des systèmes d'information via les SWS. Dans les premières sections, Nous avons d'abord présenté quelques définitions des concepts et outils dans le domaine de l'interopérabilité et la composition des services Web. Nous avons ensuite souligné le besoin de la sémantique, ce qui nous a menés à présenter des notions relatives à la sémantique et aux ontologies. Les dernières sections ont été

consacrées aux services Web sémantiques et leur intégration. A la fin de ce chapitre, nous avons proposé une classification et une étude comparative des travaux dans notre domaine. Les chapitres suivants seront dédiés à l'architecture et l'approche proposées pour la médiation des services Web sémantiques.

## Chapitre 2

Un Framework pour la mise  
en œuvre de  
l'interopérabilité des  
services Web sémantiques

# Chapitre 2 : Un framework pour la mise en œuvre de l'interopérabilité des services Web sémantiques

## 2.1. Introduction

Dans ce chapitre, nous présentons d'abord, des classifications des hétérogénéités des services Web, nous discutons des classifications de la médiation de ces derniers et, nous proposons la notre que nous jugeons pertinente. Ensuite, nous présentons l'architecture et l'approche globales que nous avons proposées pour faire inter-opérer les systèmes d'information via les SWS. Dans le cadre de cette thèse, nous nous intéressons à deux parties importantes de cette architecture qui sont la médiation des services Web et le modèle sémantique permettant de les décrire. Dans ce chapitre, nous détaillons ce modèle dont le rôle est la description des services Web ainsi que la capture des hétérogénéités explicitement pour pouvoir les traiter de façon semi-automatique par l'appel à la médiation. Notre modèle est scindé en quatre sous modèles selon la classification de la médiation que nous avons proposée et qui touche toutes les propriétés des services Web. Nous présentons le sous modèle sémantique des données [MEC 12] et nous détaillons le sous modèle sémantique des propriétés comportementales (processus métiers). Notre modèle utilise la méthode d'ASM pour sa force d'expressivité et la notion de contexte que nous avons définie. Il comprend, pour chaque service Web, une ontologie locale comportementale et des ontologies contextuelles associées aux concepts de cette ontologie. Pour relier les services Web aux modèles sémantiques, nous présentons les mécanismes d'annotation des descriptions des services en utilisant le standard SAWSDL par l'ajout de ses attributs aux documents WSDL. Il s'agit surtout de l'annotation des données et des séquences d'échange des messages (propriétés comportementales).

## 2.2. Fondements de l'approche proposée

Le problème principal qu'affrontent les systèmes d'information coopératifs via les SWS est celui des hétérogénéités de ces derniers. En effet, lors de l'interaction des services Web, plusieurs conflits peuvent apparaître à cause de diverses hétérogénéités touchant la quasi-totalité de leurs propriétés, ce qui mène à l'échec leur interopérabilité. Par conséquent, une médiation, pour résoudre ces conflits, est impérative. Dans ce sens, nous allons présenter les différentes hétérogénéités des services Web puis, des classifications des techniques de médiation de ces derniers et nous proposons la notre, qui prend en considération toutes les propriétés des services Web.

### 2.2.1. Classification des hétérogénéités des services Web

Etant donné que les services Web ont été créés indépendamment les uns des autres, des hétérogénéités apparaissent à plusieurs niveaux touchant toutes leurs propriétés, ce qui cause des conflits lors de leur composition ou de leur interopérabilité.

Il existe plusieurs classifications des hétérogénéités dans la littérature [BUS 03, CAB 05, CIM 05, MRI 07]. Lorsque l'on considère les caractéristiques des services Web, la classification suivante couvre toutes les hétérogénéités qui peuvent se produire.

**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, c'est-à-dire non décrites dans le Web Service Description Language (WSDL). Ces hétérogénéités comprennent [FAT 15]:

- Les propriétés transactionnelles ACID (atomicité, cohérence, isolation et durabilité).
- La qualité de service (QoS) et les performances des services Web telles que la disponibilité, la vitesse, le coût, la fiabilité, ...etc.

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

**Hétérogénéités des données échangées entre les services Web.** Les données échangées entre les services Web peuvent présenter des hétérogénéités que Mrissa [MRI 07] classe en trois niveaux:

- Le niveau syntaxique: concerne l'encodage des données. Les hétérogénéités de ce niveau sont traitées en utilisant le XML.
- Le niveau structurel: est relatif aux différentes représentations des données au niveau schéma.
- Le niveau sémantique: comprend la signification véhiculée par les données.

**Hétérogénéités entre les propriétés comportementales.** La communication entre le SW demandeur et le SW fournisseur ne peut avoir lieu que si leurs séquences d'échange de messages (Message Exchange Patterns ou MEPs) correspondent précisément, c'est-à-dire, quand l'un d'eux envoie un message, l'autre est en mesure de le recevoir. Les hétérogénéités comportementales sont plus complexes que celles des données.

Fensel et Bussler [FEN 02] identifient trois cas de discordances qui peuvent apparaître lors de l'échange de messages (chorégraphie): correspondance précise des messages, messages non-concordants résolubles (hétérogénéités résolubles) et messages non-concordants insolubles (hétérogénéités insolubles).

Les messages non-concordants résolubles sont :

- présence de messages supplémentaires,
- ordre différent de messages
- nécessité d'éclater un message,
- combiner des messages
- et absence de messages requis,

Cimpian et Mocan [CIM 05] identifient cinq modèles de médiation du processus pour résoudre ces conflits (voir Figure 2.1):

- Arrêt d'un message inattendu,
- Inversion de l'ordre de messages,
- Eclatement d'un message,
- Combinaison de plusieurs messages,
- Génération d'un message.

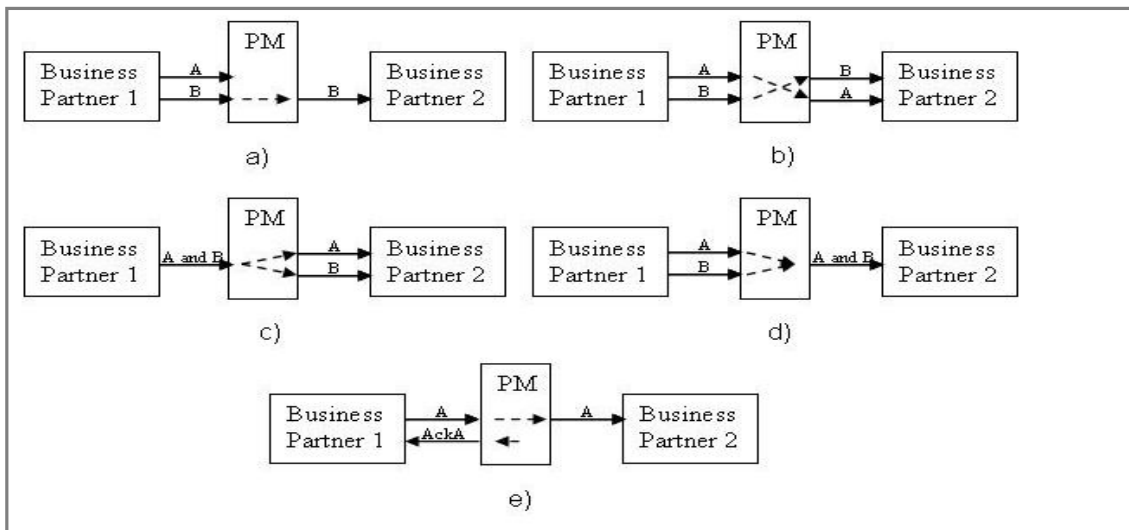


Figure 2.1. Conflits de messages résolubles [CIM 05].

## 2.2.2. Classification des mécanismes de médiation des services Web

Rappelons que la médiation des services Web est un aspect de la composition de ces derniers. En effet, des conflits peuvent apparaître entre les services Web en interaction qui présentent des hétérogénéités à plusieurs niveaux.

Dans la littérature, il existe plusieurs classifications des techniques de médiation entre les services Web, car il est possible de classer les tâches de médiation selon diverses perspectives.

Bussler [BUS 03] distingue deux types de médiation: médiation d'évènements métiers et médiation des processus métiers. La première vise à résoudre les hétérogénéités de données et la seconde vise à résoudre les hétérogénéités comportementales (behavioristes).

Cabral et Domingue [CAB 05] proposent trois niveaux de médiation : celle des données, celle des fonctionnalités et celle des processus métiers.

Mrissa [MRI 07] propose trois niveaux de médiation:

- Niveau d'intégration des services Web, visant à résoudre toutes les hétérogénéités entre les services Web, notamment celles entre les propriétés non décrites dans les documents WSDL (non fonctionnelles), telles que les propriétés relatives à la catégorisation, la qualité des services, les séquences d'échange, ...etc.
- Niveau d'adaptation de l'interface, visant à résoudre les hétérogénéités entre les propriétés fonctionnelles décrites dans les documents WSDL, comme les fonctionnalités offertes par le service Web, les protocoles et l'encodage utilisé (RPC/encoded, Document/Literal,...).
- Niveau de médiation des données, visant à résoudre les hétérogénéités des données échangées entre les services Web. Ce niveau est inclus dans le niveau de l'adaptation de l'interface qui est lui-même inclus dans le niveau d'intégration des services Web.

Les trois classifications ne traitent pas les différents aspects des services Web tels que la sécurité, la qualité de service (QoS), etc. La classification de Mrissa distingue les aspects fonctionnels et non fonctionnels des services Web, mais fusionne l'aspect comportemental avec l'aspect non fonctionnel.

En revanche, notre approche fournit une solution flexible pour faire interopérer les SWS à quatre niveaux: niveau des données, niveau fonctionnel, niveau des processus et niveau non fonctionnel. A cet égard, notre classification de la médiation des services Web est basée sur quatre niveaux, qui correspondent aux caractéristiques des SW:

- *Niveau de médiation de données*, pour résoudre les conflits de données qui peuvent se produire entre les services Web.
- *Niveau de la médiation fonctionnelle*, établit une correspondance entre la fonctionnalité fournie et celle demandée.
- *Niveau de la médiation des processus*, pour résoudre les hétérogénéités de comportement qui peuvent se produire entre les deux chorégraphies de services Web en interaction.
- *Niveau de médiation non fonctionnelle*, établit une correspondance entre les propriétés non fonctionnelles du fournisseur du service Web et le service Web demandeur.

L'infrastructure de médiation constitue le noyau de l'architecture pour la mise en œuvre de l'interopérabilité des systèmes d'information que nous présentons dans les sections qui suivent.

### **2.3. Description de l'architecture globale.**

Notre approche globale pour mettre en œuvre l'interopérabilité des systèmes d'information via les services Web sémantiques, a conduit à une architecture en couches comme le montre la Figure 2.2 [MEC 17]:

La première couche représente les applications des systèmes d'information à l'aide des services Web, ce qui permet leur interopérabilité au niveau technique et syntaxique. Pour cela, les entreprises doivent procéder à la migration de leurs systèmes d'information vers les architectures orientées services. En adoptant la méthodologie adéquate pour le faire, un catalogue de différents types de services sera dégagé, dont les services métiers [MEC 10c]. Ces derniers seront exposés aux organisations tierces à l'aide des standards des services Web (WSDL, UDDI...).

La deuxième couche concerne l'architecture permettant de décrire la sémantique des services Web obtenus précédemment. Elle est basée sur la notion d'ontologies et de contexte que nous avons défini, et utilise la méthode formelle d'ASM (Abstract State Machine) [BOR 08]. Elle constitue le modèle de référence pour annoter les descriptions des services Web à l'aide du SAWSDL (Semantic Annotation of Web Service Description Language) [HAU 09]. Ce modèle permet de capturer la sémantique des propriétés des services Web, aux niveaux non fonctionnel et fonctionnel (notamment les données et les caractéristiques comportementales).

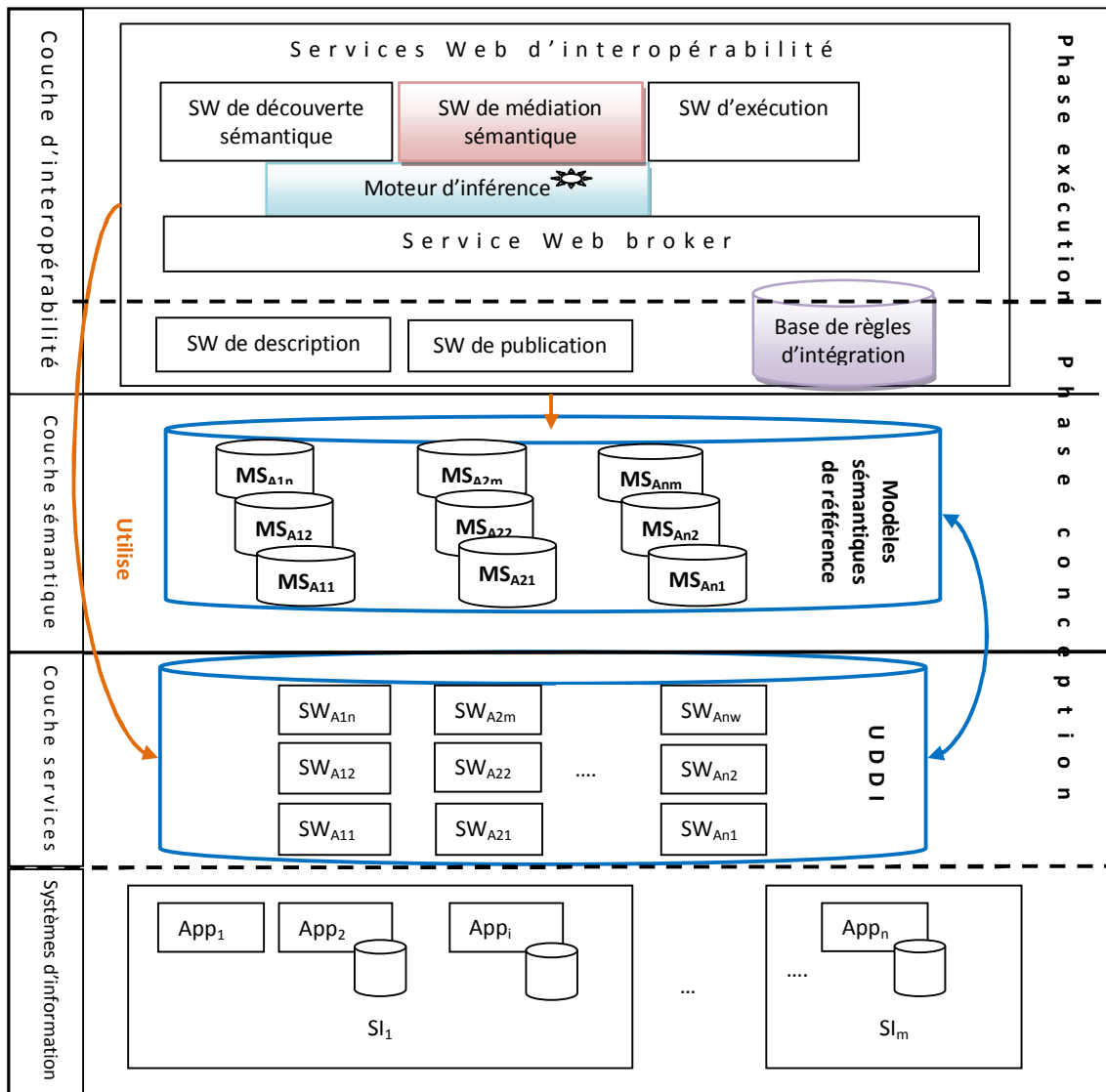


Figure 2.2. Architecture générale proposée pour l'interopérabilité sémantique des systèmes d'information.

La dernière couche est une architecture basée services pour l'interopérabilité sémantique des systèmes d'information. En exploitant les deux premières couches, elle permet 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. 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 découverte, de médiation, et d'exécution. Dans ce processus, une base de règles et un moteur d'inférence sont utilisés.

Comme le montre la Figure 2.2, les trois couches sont étroitement liées en raison des interactions qu'elles entretiennent. En effet, la couche supérieure est composée de plusieurs services d'interopérabilité qui fonctionnent pendant la phase d'exécution. Elle utilise les

modèles sémantiques de la deuxième couche pour faire inter-opérer les services de la première couche. Ces deux dernières sont créées en mode conception.

Notre approche présente de nombreux avantages, et à plusieurs niveaux :

- ✓ Notre architecture est au dessus des systèmes d'information des entreprises en adoptant les architectures orientées services et les services Web qui, constituent des paradigmes incontournables pour le développement des systèmes d'information interopérables, surtout en raison du couplage faible qu'ils procurent, de la conservation de l'existant et de la réutilisation.
- ✓ L'introduction de la notion de contexte ainsi que l'enrichissement sémantique des descriptions des services Web, apportent plusieurs avantages, tels que la description explicite des hétérogénéités sémantiques qui permet l'automatisation de la médiation à plusieurs niveaux. 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 chapitre, nous présentons, de façon générale, le rôle de chaque type de service et le processus de coopération de la couche d'interopérabilité. Puis nous décrivons, en détail, la couche sémantique au niveau des données et des processus pour construire le modèle sémantique de référence. Ce modèle sera exploité par l'infrastructure de médiation, constituant le noyau de la couche d'interopérabilité, qui sera détaillée dans le Chapitre 3.

## **2.4. Processus de coopération de la couche d'interopérabilité**

Nous rappelons que notre architecture est basée services Web, 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 ontologies ou du modèle sémantique (MS) ainsi que les règles,
- L'annotation des descriptions des services Web prise en charge par un service Web de description sémantique,
- La publication 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,
- Les services Web d'exécution.

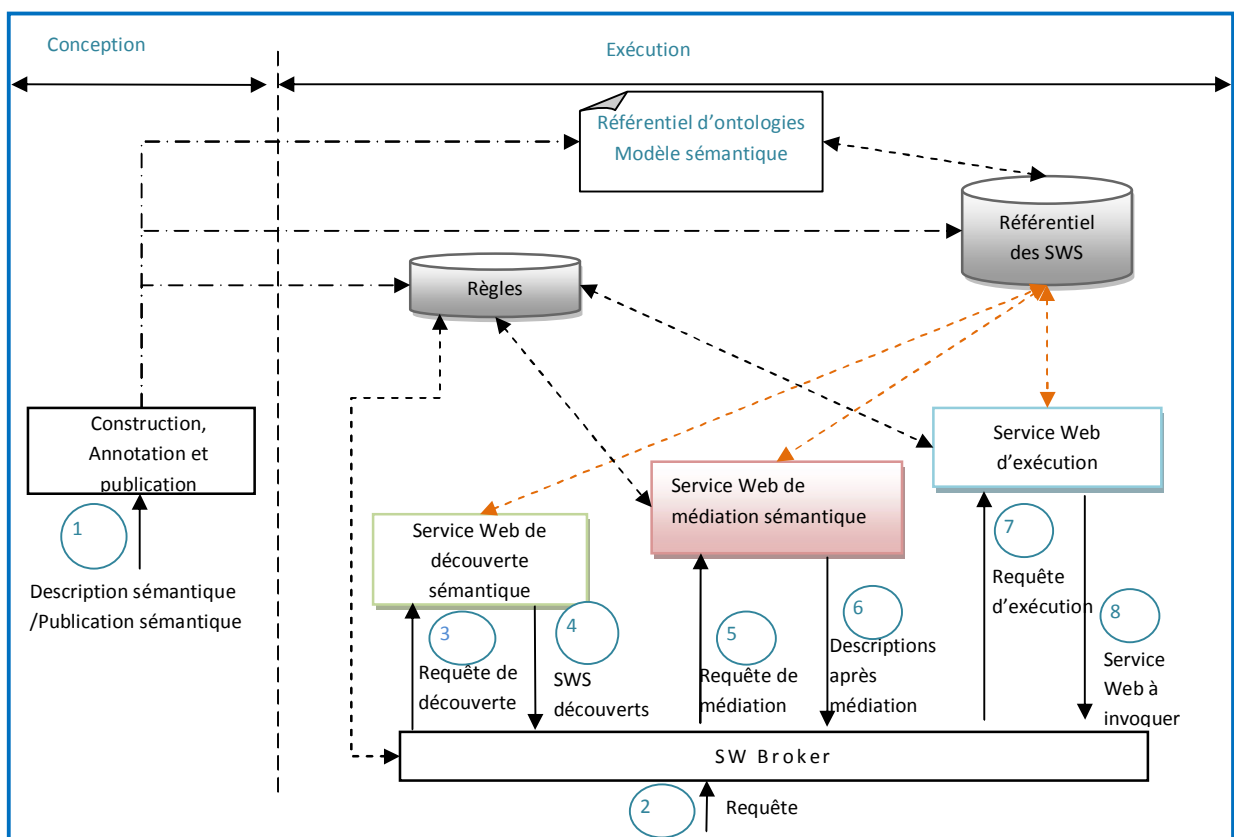


Figure 2.3. Processus global proposé pour l'interopérabilité sémantique.

Les étapes de ce processus d'interopérabilité sont décrites dans la Figure 2.3. Après la phase de conception, lorsque le SW broker reçoit une requête d'un utilisateur ou d'un SWS (2) la découverte sémantique commence (3, 4). Les SWS découverts passent alors par la médiation (5, 6) afin de résoudre les éventuels conflits sémantiques. Puis les SWS sont exécutés (7, 8) par le service Web d'exécution. Les services Web d'interopérabilité utilisent le référentiel des services Web sémantiques qui comprend quelques liens (annotations) au

référentiel des ontologies (modèle sémantique). Tous les services Web d'interopérabilité utilisent un ensemble de règles et de requêtes.

### **2.4.1. Rôle du service Web broker**

Notre processus d'interopérabilité est mis en œuvre par plusieurs types de 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.

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.

### **2.4.2. Publication sémantique**

La publication concerne les ontologies, ainsi que les services Web. Une fois les ontologies construites, et les descriptions 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).

Pour la publication des descriptions des services Web nous utilisons le référentiel UDDI3.0 [CLE 04] Avec cette solution, nous pouvons soit annoter d'abord les fichiers WSDL puis les publier, soit publier les descriptions de services, puis leur 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 de la requête suivante du service Web broker:

*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.

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

### 2.4.3. 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).

### 2.4.4. Médiation sémantique

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

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

Où :

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

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

### 2.4.5. 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.

Dans le cadre de cette thèse, nous nous intéressons à deux parties essentielles de notre architecture globale pour l'interopérabilité des SI. Il s'agit, d'une part, de la médiation sémantique des services Web, notamment au niveau des processus, et, d'autre part, du

modèle sémantique décrivant ces services. Dans ce qui suit, nous détaillons notre modèle sémantique pour décrire les services Web et permettre leur médiation automatique. Nous décrivons cette dernière dans le Chapitre 3.

## 2.5. Description du modèle sémantique

Les hétérogénéités sémantiques entre services Web peuvent être résolues seulement par une description sémantique de ceux-ci. La meilleure méthode pour capturer la sémantique consiste à utiliser des ontologies. Il s'agit d'une description explicite des hétérogénéités qui permet une interprétation automatique par des raisonneurs.

Le modèle sémantique est une partie très importante dans notre architecture parce qu'il a trois buts importants:

- Fournir un modèle de référence sémantique des données pour résoudre les hétérogénéités sémantiques des données échangées entre deux services Web.
- Fournir une représentation formelle des propriétés fonctionnelles et non fonctionnelles offertes par chaque service Web pour faciliter les processus de découverte et de sélection des services Web.
- Fournir un modèle comportemental formel et basé règles pour résoudre les hétérogénéités des séquences d'échange des messages.

L'ensemble de ces buts de notre modèle sémantique vise à supporter le processus semi-automatisé de médiation, et par conséquent, l'intégration de services Web au niveau sémantique.

Pour avoir une description sémantique complète des services Web, nous devons considérer leurs propriétés fonctionnelles (y compris les données) et non fonctionnelles, aussi bien que leurs propriétés comportementales.

Notre architecture conceptuelle est construite d'après les quatre niveaux de notre classification de médiation (voir section 2.2.2). Elle consiste en un modèle sémantique composé de quatre sous-modèles (voir Figure 2.4):

- ***Sous-modèle sémantique des propriétés non fonctionnelles (NSSM, Non functional properties Semantic Sub-Model)***: décrit les sémantiques des propriétés non fonctionnelles des services Web, comme les propriétés concernant la catégorisation, la qualité de services, disponibilité temporelle... etc.
- ***Sous-modèle sémantique des propriétés comportementales (BSSM, Behavioural Semantic Sub-model)***: décrit la sémantique de la manière avec laquelle les

fonctionnalités de service Web peuvent être accomplies. Dans notre cas, ceci est pris en charge par des ontologies comportementales.

- **Sous-modèle sémantique des propriétés fonctionnelles (FSSM, Functional properties Semantic Sub-Model):** décrit la sémantique des propriétés fonctionnelles comme les fonctionnalités offertes par le service Web, les protocoles et le codage (RPC/encoded, Document/Literal, ...).
- **Sous-modèle sémantique de Données (DSSM, Data Semantic Sub-Model):** décrit la sémantique des données échangées entre deux services Web [MEC 12].

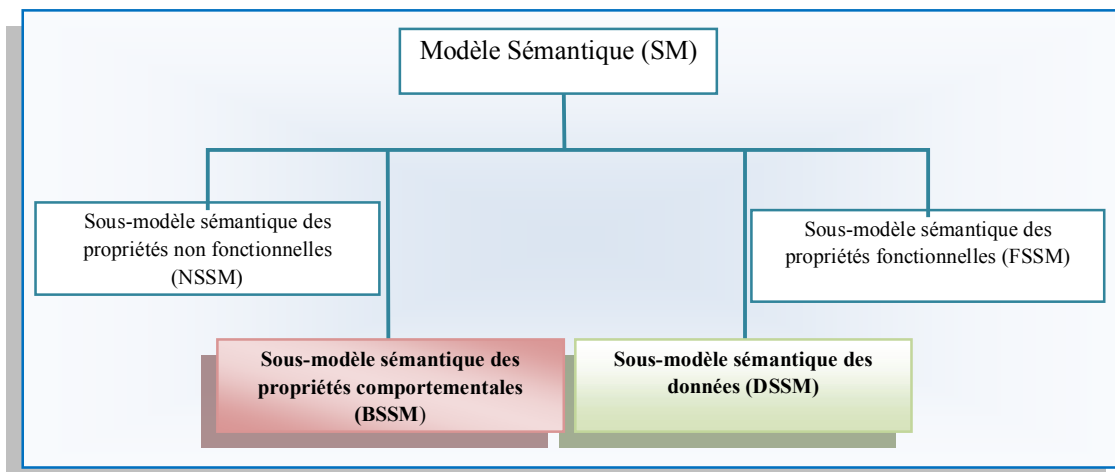


Figure 2.4. Sous-modèles sémantiques.

Ces sous modèles seront utilisés, respectivement, dans les médiations des propriétés fonctionnelles et non fonctionnelles, des processus métiers, et des données.

Dans ce qui suit, nous présentons le modèle sémantique des données [MEC 12] et le modèle sémantique des propriétés comportementales [MEC 17]. Les autres sous modèles feront l'objet de travaux futurs.

### 2.5.1. Description du sous modèle sémantique des données (DSSM, Data Semantic Sub Model)

Le DSSM sera le modèle de référence pour annoter les données échangées entre services Web. Le DSSM 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 [MRI 07].

### 2.5.1.1. Notion de contexte de données et d'objet sémantique

**Définition du contexte de données:** « 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 » [MRI 07].

**Définition de l'objet sémantique:** Une donnée dont la sémantique est explicitement décrite possède [MRI 07] :

*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 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 :

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

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 : Soit l'exemple de l'objet sémantique S concernant le concept « prix » avec son contexte.

Concept c : Prix ;

valeur v : 25 000 ;

type t : xsd double ;

contexte : pour le concept prix le contexte est formé de plusieurs autres concepts appelés modifieurs, comme le facteur multiplicateur (1/1000), la TVA (incluse oui/non), la devise (Euro), le pays (France), date de la devise (10/04/2016), le format de la date (jj/mm/aaaa).

Les modifieurs '*date*, *devise* et *format de date*' sont dynamiques. En effet, 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.

### 2.5.1.2. Description du modèle sémantique des données

Notre sous modèle sémantique (DSSM) inspiré de celui de Mrissa [MRI 07] qui 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 (DLO, Data Local Ontology)*** 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 (DDO, Data Domain Ontology)*** 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 (DCO, Data Contextual Ontology)*** 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 adhéré 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. Les valeurs et les noms des *modifieurs* statiques sont spécifiés dans les ontologies locales de données, contrairement à l'approche de Mrissa [MRI 07], qui les intègre dans les descriptions des services Web.

### **2.5.2. Description du sous modèle sémantique comportemental (BSSM, Behavioral Semantic Sub Model)**

Pour capturer les propriétés comportementales sémantiquement, nous utilisons la méthode des machines d'états abstraits (Abstract State Machine ou ASM) [BOR 08] en raison de ses avantages précieux surtout sa puissance d'expression et sa modélisation basée règles.

Une machine à états abstraits (introduite au milieu des années 1980 par Yuri Gurevich) est un automate fini dont les états ne portent pas simplement des noms, mais des structures au sens de la logique mathématique, c'est-à-dire des ensembles non vides munis de fonctions, d'opérations et de relations [BOR 08].

Il s'agit d'une méthode d'ingénierie des systèmes pratique et scientifiquement fondée qui rassemble les deux aspects du développement d'un système : l'analyse des exigences par une modélisation de haut niveau d'abstraction du domaine d'application, et le déploiement des solutions algorithmiques sur des plates-formes variées.

Notre BSSM est donc conçu selon les principes des ASMs :

- il est basé état,
- l'état est représenté par la signature  $\Sigma$  (ou vocabulaire),
- le changement d'état est modélisé par les règles de transition qui modifient les valeurs des fonctions et relations définies par la signature.

*L'état* est décrit par une ontologie exploitant la notion d'objet sémantique et de contexte comme nous les avons définis pour les données échangées. Les fonctions et relations de l'ASM sont représentées dans notre modèle par la notion d'objet sémantique. Nous adaptons la définition de contexte de données [MEC 12] pour définir la notion de contexte comportemental décrivant la catégorie de l'objet sémantique :

**Définition du Contexte Comportemental :** « Le contexte comportemental englobe l'information relative à l'interaction dans laquelle est engagé le service Web, cette information est nécessaire à l'interprétation des actions de l'interaction ».

**Définition de l'objet sémantique :** une donnée XML (échangée) est décrite formellement par un objet sémantique S défini comme suit :

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

Avec :

**c :** un concept c, définit la famille ontologique à laquelle cette donnée appartient, c'est-à-dire la classe abstraite dont cette donnée est une instance.

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

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

**C :** un contexte C, qui apporte des précisions sur l'interprétation de la donnée [MEC 12].

**B :** un contexte comportemental B, qui apporte des précisions sur l'interprétation de l'aspect comportemental. Il s'agit d'un lien à une ontologie contextuelle comportementale comportant, entre autre, les informations suivantes :

**Order Ord:** sa valeur est un nombre qui représente l'ordre du message (données échangées).

**Acknowledgement Ack:** dont la valeur est vraie si un message d'accusé de réception est nécessaire.

**Pattern P :** décrivant la catégorie de l'objet sémantique (donnée échangée au niveau sémantique). Selon la classification des fonctions d'ASM le contexte comportemental P peut avoir une des cinq catégories suivantes :

- **Static:** l'objet sémantique S ne peut pas être modifié pendant l'exécution.
- **In (monitored):** S ne peut être changé que par l'environnement, et peut être lu par la machine.
- **Out :** S ne peut être changé que par l'exécution des règles de transition, non lu par la machine et peut être lu par l'environnement.
- **Shared :** S peut être changé par la machine et par l'environnement.
- **Controlled :** S est directement mis à jour uniquement par la machine, il ne peut pas être lu ou mis à jour par l'environnement.

Comme le montre la Figure 2.5, notre BSSM contient deux types d'ontologies: une ontologie comportementale locale et des ontologies contextuelles comportementales associées. L'ontologie locale décrit la sémantique locale des messages XML échangés

(WSDL) au niveau du fournisseur du service Web. Les ontologies contextuelles comportementales sont associées aux concepts de l'ontologie locale. Une ontologie contextuelle permet de décrire le contexte d'un concept de l'ontologie locale. Ainsi, à chaque concept de l'ontologie locale, nous associons une ontologie contextuelle qui permet de clarifier les hétérogénéités sémantiques du comportement et de résoudre les conflits de processus liés aux contextes de l'utilisateur lors de l'exécution.

Le BSSM de chaque service Web est donc composé d'une ontologie locale comportementale (Behaviorist Local Ontology : **BLO**) et de plusieurs ontologies contextuelles comportementales (Behaviorist Contextual Ontology : **BCO**). Toutes ces ontologies sont locales car elles capturent la sémantique des messages XML échangés (WSDL) au niveau du service Web (demandeur /fournisseur) avec leurs contextes comportementaux. Elles sont exprimées en OWL [SCH 04]. Elles sont définies par le propriétaire (fournisseur) du service Web.

Dans notre approche, chaque fournisseur décrit la sémantique de ses propres services Web. Cela permet une fiabilité des annotations de la description du service Web et l'intégration des services existants sans avoir besoin d'une description supplémentaire du service. L'utilisation du langage SAWSDL rend cette solution compatible avec le WSDL, permettant, ainsi, l'interopérabilité avec les architectures existantes et l'adhésion de nouveaux fournisseurs.

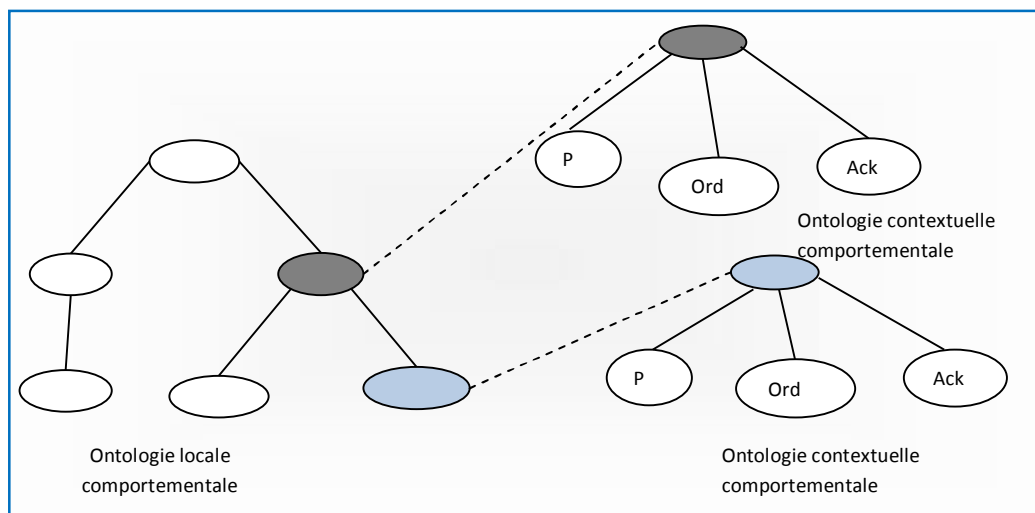


Figure 2.5. Structure du sous-modèle sémantique comportemental

**Les règles de transition** expriment les changements d'états par le changement de l'ensemble des instances. Chaque règle est exprimée par des conditions et des effets qui sont des expressions de logique de description (DL). Pour exprimer ces règles nous utilisons le SWRL (Semantic Web Rule Language) [HOR 04]. Il permet d'écrire des règles basées sur les concepts d'OWL DL en apportant plus d'expressivité au langage OWL.

### 2.5.3. Approche de conception et de création des ontologies

Pour le modèle des données (DSSM), nous adoptons pour la création des ontologies locales la méthode qui consiste à transformer le modèle de données en une ontologie. En effet, un modèle de classe UML peut être transformé en une ontologie pour le Web sémantique. A l'aide du plugin DataMaster de l'outil Protégé, on peut transformer une base de données en une ontologie.

En ce qui concerne l'ontologie de domaine qui constitue un accord sur le sens des termes et les ontologies contextuelles de données, 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.

En ce qui concerne l'ontologie locale du modèle comportemental (BSSM), elle est construite à partir du WSDL. Et pour ce qui est des ontologies contextuelles associées à cette ontologie locale il faut étudier pour chaque concept (classe) de l'ontologie locale, la possibilité du besoin d'une ontologie contextuelle, en énumérant les concepts qui constituent le contexte (Pattern P, Order Ord et *Acknowledgement* Ack).

Toutes nos ontologies sont décrites à l'aide du langage OWL DL 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.

### 2.5.4. Mécanismes d'annotation sémantique

L'enrichissement sémantique des descriptions des services Web par le modèle de référence (Modèle sémantique) est nécessaire pour permettre le raisonnement par machine et par conséquent la découverte et la médiation dynamiques des services Web sémantiques.

Nous utilisons les mécanismes d'annotations apportés par le SAWSDL pour enrichir sémantiquement, d'une part, les données, dans le but d'automatiser la médiation au niveau données, et, d'autre part, les séquences d'échange de messages dans le but d'automatiser la médiation des processus métiers.

SAWSDL [HAU 09] : Semantic Annotations Web service Description Language est l'un des standards récents du World Wide Web consortium W3C. SAWSDL définit comment ajouter des annotations sémantiques aux différents composants d'un document WSDL2.0 [CHI 04] (et aussi supporte le WSDL1.1 [CHR 01], comme les messages d'entrée / sortie, les interfaces et les opérations. Pour accomplir l'annotation sémantique, SAWSDL définit des attributs d'extension qui peuvent être appliqués aux éléments WSDL aussi bien qu'aux éléments de schéma XML. Ces attributs sont:

**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` and `xs: attribute`.

**LiftingSchemaMapping et LoweringSchemaMapping:** SAWSDL introduit les annotations de mapping de schémas qui dressent les solutions après découverte dans l'utilisation d'un service Web. Le mapping référencé par le premier attribut définit comment une instance XML, est transformée à une ou plusieurs données du modèle sémantique. Le mapping référencé par le deuxième attribut définit comment les données dans un modèle sémantique sont transformées aux données instances XML.

### **Annotation des données**

Pour annoter sémantiquement les données, des attributs SAWSDL sont ajoutés aux descriptions du service Web source et du service Web cible, il s'agit des schémas XML (.XSD). Ces attributs sont : `LiftingSchemaMapping` pour les mappings syntaxiques-sémantiques (voir Figure 2.6), et `LoweringSchemaMapping` pour les mappings sémantiques-syntaxiques.

Ces attributs font référence aux définitions des mappings des données XML de et vers le modèle de données (DSSM). Elles concernent 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. Pour compléter le contexte 'C', on fait appel à des règles logiques permettant d'inférer les valeurs des *modifieurs* dynamiques.

```

<wsdl: description...>
  <wsdl: types>
    <xs: schema...>
      <xs: element name="QuoteRequest">
        <xs: complexType sawsdl:liftingSchemaMapping=' '
http://.../QuoteRequest2Ont.xslt' '>
          <xs: sequence>
            <xs: element name="QuoteReqId" type="xs:integer" />
            <xs: element name="QuoteReqProduct" type="product" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>

```

Figure 2.6. Exemple d'annotation de données

## Annotation des séquences d'échange des messages

Notre approche pour l'annotation sémantique des aspects comportementaux (MEPs), consiste à ajouter des attributs ModelReference au niveau des opérations dans le document WSDL. Cet attribut fait référence au BSSM décrit précédemment (voir Figure 2.7).

```

<wsdl:interface name="quoteR">
  <wsdl:operation name="QuoteR" pattern="http://.../Out-In">
Sawsdl:modelReference="http://.../quoteR-BSSMr#QuoteRequest"
Sawsdl:modelReference="http://.../quoteR-BSSMr#QuoteResponse"
    <wsdl:output element="QuoteRequest"/>
    <wsdl:input element="QuoteResponse"/>
  </wsdl:operation>
</wsdl:interface>
</wsdl:description>

```

Figure 2.7 : Exemple d'annotation comportemental

Le modèle sémantique (notamment le DSSM et le BSSM) est une partie très importante dans la médiation, des données et des processus métiers, que nous décrivons dans le chapitre suivant.

## 2.6. Conclusion

Dans ce chapitre, nous avons présenté notre architecture globale ainsi que le processus d'interopérabilité. Nous nous sommes focalisés sur deux parties noyaux de cette architecture qui sont le modèle sémantique et l'infrastructure de médiation pour résoudre les conflits causés par les éventuelles hétérogénéités entre les services Web. Nous avons présenté le modèle sémantique permettant de décrire ces derniers en nous intéressant, en particulier, au modèle sémantique comportemental (BSSM) qui présente beaucoup d'avantages car:

- il est basé sur la méthode d'ASM, c'est donc un modèle à base d'état et de règles.
- ces règles sont exprimées grâce au puissant langage SWRL.
- il exploite la notion de contexte pour décrire les propriétés comportementales.
- ce modèle de référence est relié à la description du service Web concerné par le SAWSDL.

Dans le Chapitre 3 nous allons aborder, en détail, la médiation sémantique des services Web au niveau des processus métiers. Cette médiation exploite notre modèle sémantique comportemental (BSSM).



## Chapitre 3

Une approche de médiation  
basée contexte pour  
l'interopérabilité des services  
Web sémantiques

# Chapitre 3 : Une approche de médiation basée contexte pour l'interopérabilité des services Web sémantiques

## 3.1. Introduction

Lors de l'interaction entre deux services Web, des conflits peuvent apparaître à cause des hétérogénéités sémantiques. Pour être capable de communiquer directement, les deux services doivent avoir des sémantiques compatibles, ce qui est, dans la plupart du temps, impossible, car chacun est créé indépendamment de l'autre. De ce fait, la médiation sémantique est nécessaire pour garantir l'interopérabilité ou l'intégration des services Web.

Dans le chapitre précédent, nous avons proposé, entre autres, un modèle sémantique dont le rôle est de décrire explicitement les hétérogénéités des services Web. Ce modèle est la base de l'automatisation de la médiation. Nous avons décrit, en particulier, le sous modèle sémantique comportemental (BSSM, Behavioural Semantic Sub Model), pour lequel nous avons utilisé la méthode d'ASM et la notion de contexte comportemental que nous avons définie.

Au début de ce chapitre, nous montrons que la médiation, qui constitue le noyau de notre architecture, est nécessaire à tous les niveaux et pour toutes les tâches des services Web. En effet, la médiation des propriétés fonctionnelles et celle des propriétés non fonctionnelles peuvent être utilisées pendant la découverte et la sélection. Cependant, la médiation des processus métiers et celle des données sont utilisées lors de l'exécution (invocation).

Dans le cadre de cette thèse, nous décrivons la médiation des processus métiers qui fait appel à la médiation des données abordée dans nos travaux précédents [MEC 12].

Le service Web de médiation des processus WSPM (Web Service of Process Mediation) est implémenté par un algorithme de médiation de chorégraphie gérant la conversation entre les deux services Web. Il utilise les modèles comportementaux (BSSM) des deux parties et les règles de transition. WSPM fait appel au service Web de médiation de données WSDM (Web Service Data Mediation) et à un raisonneur. Le cœur de ce médiateur est le module de transformations sémantiques-sémantiques, basé sur la notion de mappings d'ontologies des BSSM et le raisonnement sur ces mappings.

## 3.2. Niveaux de médiation pour l'interopérabilité des SWS

Pour faire inter-opérer des SWS, nous avons besoin des tâches de découverte dynamique, de sélection, et d'invocation. Pour chacune de ces tâches, la médiation peut être exigée à

différents niveaux: au niveau de données, au niveau fonctionnel, au niveau processus métiers et au niveau propriétés non fonctionnelles.

Les quatre niveaux de médiation sont assurés par quatre types de services Web qui sont respectivement:

- **WSDM** : Web Service of Data Mediation ou Service Web de Médiation des Données,
- **WSFM** : Web Service of Functional mediation ou Service Web de médiation fonctionnelle,
- **WSPM** : Web Service of Process Mediation ou Service Web de Médiation des Processus et,
- **WSNM** : Web Service of Non functional Mediation ou Service Web de Médiation non fonctionnelle.

Comme illustré dans la Figure 3.1, la médiation est prise en charge par les quatre types de services Web ci-dessus, en exploitant les ontologies du modèle sémantique décrit dans le chapitre précédent (voir section 2.5), de la manière suivante :

- La médiation non fonctionnelle des SWS peut être effectuée par le WSNM en utilisant le NSSM (Non functional properties Semantic Sub-Model).
- L'adaptation d'interfaces c'est-à-dire, la médiation des propriétés fonctionnelles peut être effectuée par le WSFM en utilisant le FSSM (Functional properties Semantic Sub-Model). Le WSFM peut appeler le WSNM.
- Les données échangées peuvent être inter-opérées à l'aide du WSDM en utilisant le DSSM (Data Semantic Sub Model).
- La médiation des processus métiers peut être effectuée par le WSPM en utilisant le BSSM (Behavioural Semantic Sub Model). Le WSPM peut appeler le WSDM.

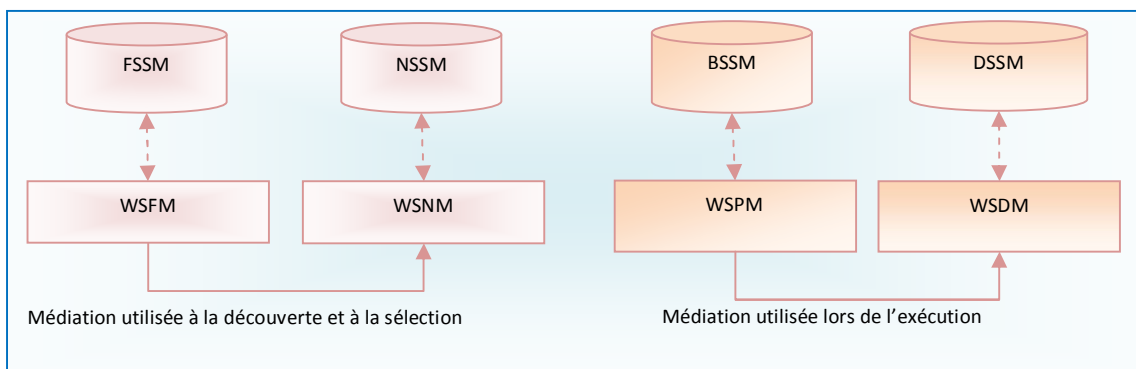


Figure 3.1. Médiation à plusieurs niveaux pour plusieurs tâches

La médiation peut être utilisée pendant la découverte et la sélection ainsi que l'exécution (invocation). Les organigrammes de la Figure 3.2 illustrent la médiation, aux niveaux fonctionnel et non fonctionnel, utilisée pendant les tâches de découverte et de sélection des services Web.

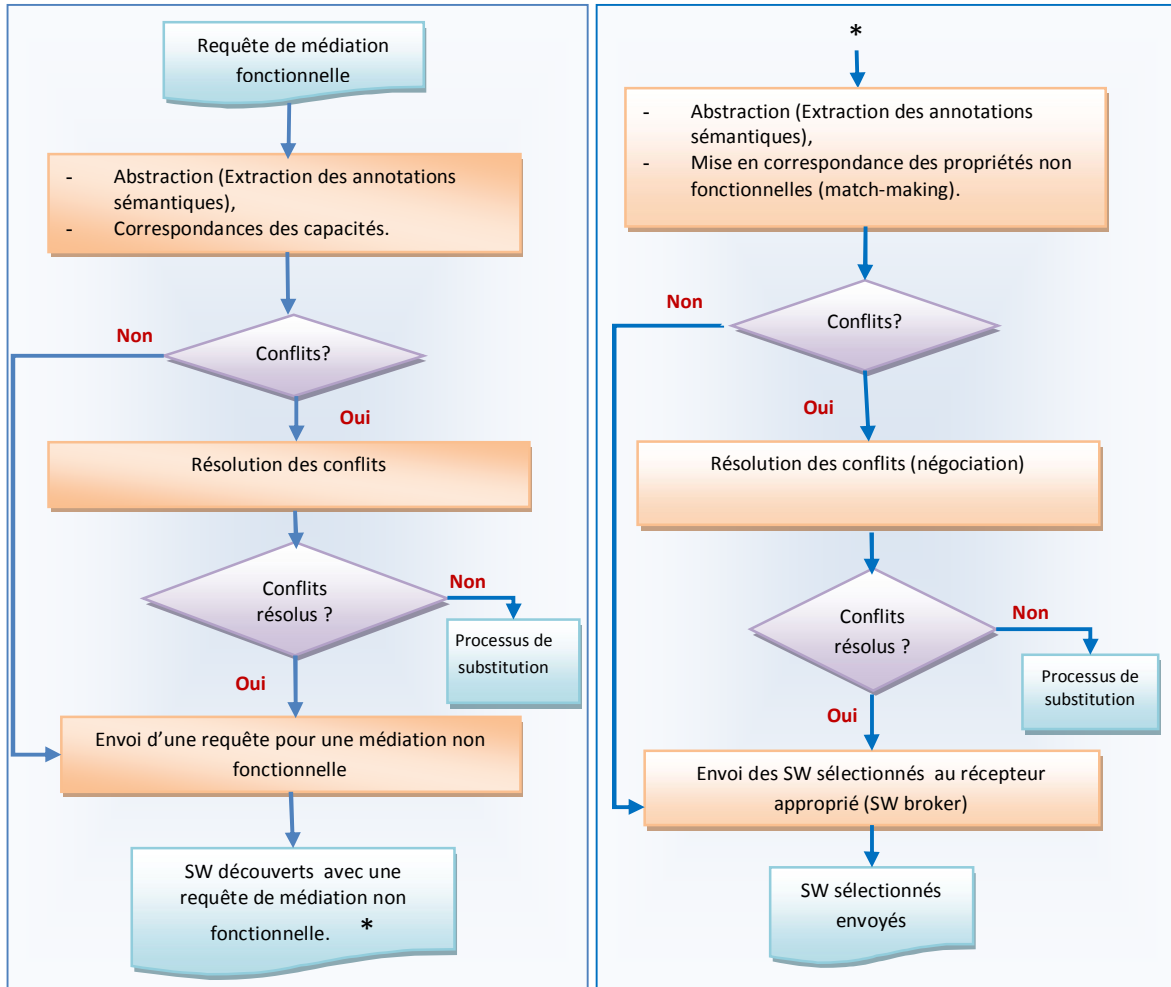


Figure 3.2. Médiation pendant la découverte

Les organigrammes de la Figure 3.3 illustrent la médiation, aux niveaux des processus et des données, utilisée pendant la tâche d'exécution des services Web.

Dans la suite de cette thèse, nous détaillons la médiation des processus qui fait appel à la médiation des données déjà décrite dans nos travaux précédents [MEC 12]. Elle permet de réconcilier les services Web sémantiques et permettre leur interopérabilité.

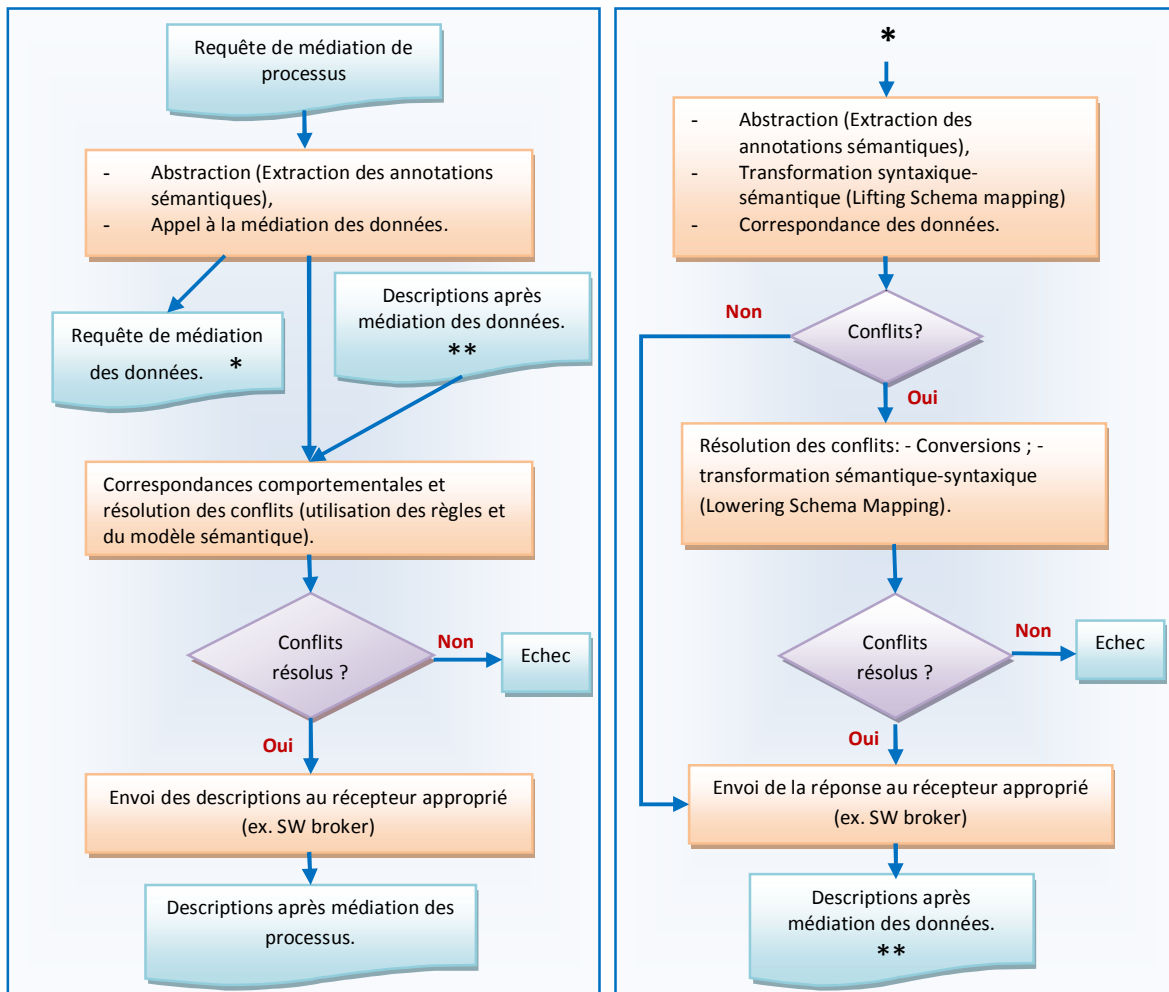


Figure 3.3. Médiation pendant l'exécution.

### 3.3. Médiation des processus métiers

Pour être capable de communiquer directement, les deux services Web doivent avoir des séquences d'échange de messages compatibles, ce qui généralement impossible, puisque chacun est créé indépendamment de l'autre. Par conséquent, l'usage d'un système de médiation externe est impératif. Le rôle de ce système est de transformer les messages du demandeur et/ou les messages du fournisseur pour permettre la communication entre les deux parties.

Dans le cadre de notre travail de recherche, nous nous intéressons, entre autres, à la réconciliation des conflits, entre services Web, causés par les hétérogénéités des séquences d'échanges de messages, pour lesquels les types de disparités sont: ordre différent de messages, présence de messages supplémentaires, absence de messages requis, nécessité d'éclater un message ou combiner des messages.

### 3.3.1. Approche conceptuelle de la médiation des processus

Notre approche pour réconcilier les séquences d'échange des messages des services Web s'appuie sur un type de services appelé service Web de médiation des processus ou WSPM (Web Service of Process Mediation). Ceux-ci peuvent résoudre les conflits par inversion de l'ordre des messages, arrêt d'un message inattendu, en combinant plusieurs messages ou en éclatant un message.

Ces médiateurs utilisent un ensemble de règles qui permettent d'identifier les équivalences entre les différents modèles sémantiques (BSSM : Behavioral Semantic Sub Models). Ces règles et modèles sont créés et stockés pendant la phase de conception et utilisés pendant la phase d'exécution.

Puisque les propriétés comportementales n'apparaissent pas dans le document WSDL, nous utilisons les éléments d'extensibilité de ce dernier et le SAWSDL pour y ajouter de l'information supplémentaire à propos des séquences d'échange de messages, et nous exploitons l'attribut ModelReference pour joindre la description étendue au modèle de référence (BSSM).

Le WSPM (voir Figure 3.4) est inséré entre le service Web demandeur et le service Web fournisseur, il intercepte leurs messages et procède aux éventuelles transformations en exploitant le modèle sémantique et les règles pour exécuter l'algorithme de médiation de processus métiers.

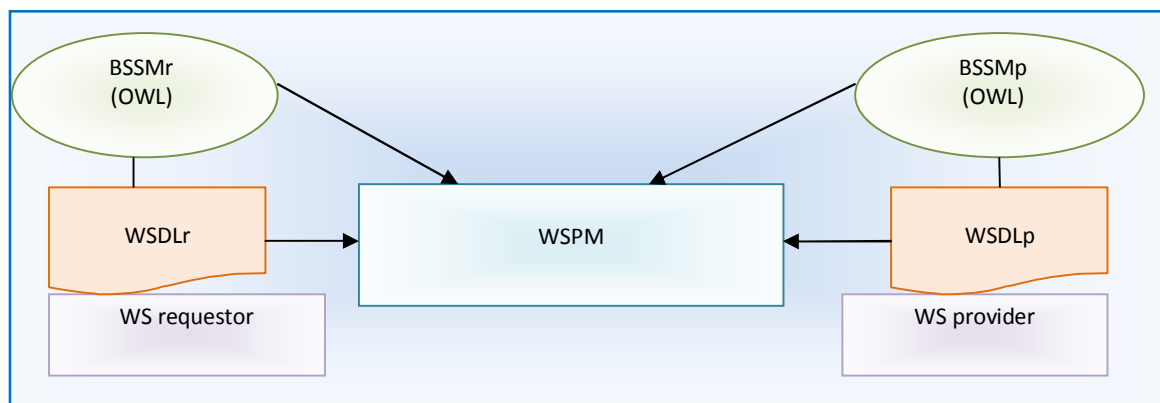


Figure 3.4. Architecture conceptuelle du WSPM.

### 3.3.2. Fonctionnement du WSPM

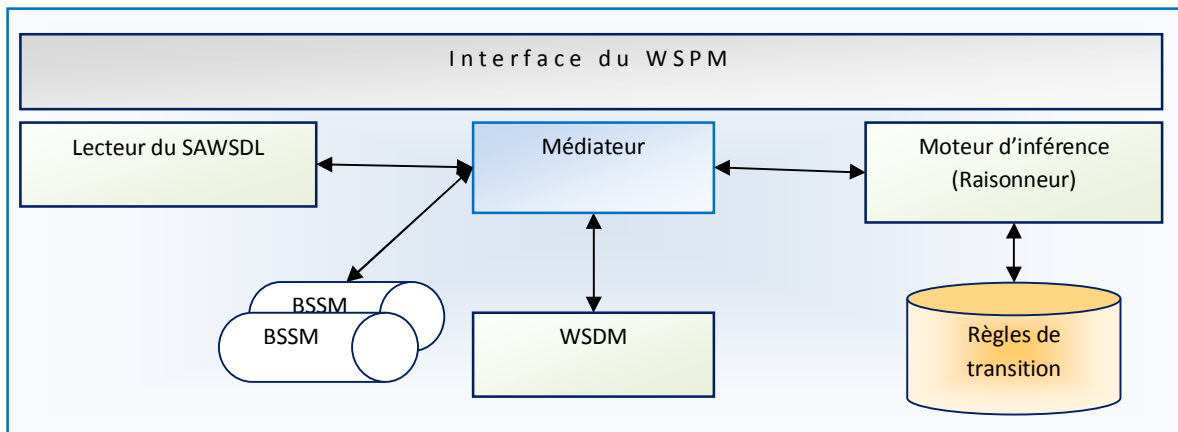


Figure 3.5. Description du WSPM

Le service Web de médiation des processus WSPM (Web Service of Process Mediation) (voir Figure 3.5) est implémenté par un algorithme de médiation de chorégraphie (voir Listing 3.1) gérant la conversation entre les deux services Web. Il utilise les ontologies comportementales des deux parties (BSSM) et les règles de transition. WSPM fait appel au service Web de médiation de données WSDM (Web Service of Data Mediation) et à un raisonneur. Les étapes de cet algorithme sont comme suit :

- Téléchargement des fichiers WSDL annotés des services Web demandeur et fournisseur (WS1 et WS2).
- Extraction des annotations des opérations et celles des données (ModelReference, lifting and LowringSchemaMappings attributs) en utilisant l'API SAWSDL4J [HAU 09]. Ces annotations se réfèrent aux concepts d'ontologies des BSSMs (BLO, BCO) et des mappings.
- Chargement des copies de celles-ci dans la mémoire.
- Lors de la réception de messages, des transformations syntaxiques-sémantiques sont réalisées en utilisant les attributs SAWSDL liftingSchemaMappings et en appelant le service Web de médiation des données (WSDM) pour transformer les instances du contexte du service Web source dans le contexte du service Web cible. Ces données conceptuelles sont enregistrées en mémoire (C).
- En utilisation les attributs ('P', 'Ord' et 'Ack'), des ontologies contextuelles (BCO) des BSSM, et les règles de transition, le WSPM effectue des transformations sémantiques: génération de messages, réorganisation des messages, combinaison de plusieurs messages dans un seul, et éclatement d'un message en plusieurs autres messages.
- Pour envoyer les données conceptuelles (C), l'algorithme effectue les transformations sémantiques-syntaxiques en utilisant l'attribut lowringSchemaMapping pour obtenir les données XML (msg). Ensuite, l'algorithme envoie le message (msg) au service Web en utilisant l'attribut modelReference.

---

**Listing 3.1.** Algorithmme du WSPM

---

**Inputs:** msg, URI of wsdl1 (WS1), URI of wsdl2 (WS2)

**Uses:** BSSM1, BSSM2, TR1, TR2, ModelReference, lifting & lowering mappings

**Outputs:** msg after mediation

**1. Message reception and syntactic-semantic transformations**

**Repeat**

```
c= receiveMessage(msg, WS1)
  {cm =liftingSchemaMapping(msg);
  c = dataMediation(Cm);
  return c;}
```

Add c to C;

**Until** no message;

**2. Semantic-Semantic transformations (reasoning)**

**For All** c ∈ C of BSSM1 **do**

Check contextual information (P, Ord, Ack);

Evaluate TR; // Transition Rules

**if** Ack=True **then**

Generate (dummyAck);

send (dummyAck, WS1);

**endif;**

**if** extraMessageMismatch **then**

**if** c is expected now /\*(P=in according BSSM2 and TR)\*/ **then**

add (c, BSSM2); /\*add c to BSSM2\*/

**endif;**

**endif**

**if** MergMismatch **then**

merging(c<sub>i</sub>, c); /\*merging c<sub>i</sub> from C of BSSM1 in c\*/

Add(c, BSSM2); /\* Add c to BSSM2 \*/

**endif;**

**If** splitMismatch (c is composed of c<sub>i</sub>) **then**

Generate(c<sub>i</sub>); /\* generate c<sub>i</sub> from C of BSSM1\*/

Add(c<sub>i</sub>, BSSM2); /\* Add c<sub>i</sub> to BSSM2\*/

**endif;**

**if** orderingMismatch **then**

add(c<sub>i</sub>, BSSM2) /\* add c<sub>i</sub> ∈ C of BSSM1 to BSSM2 according to context information (ord) \*/

**endif;**

**endfor;**

**3. semantic- syntactic transformations**

send(msg, WS2)

**{for all** c ∈ C of BSSM2 **do**

msg= loweringSchemaMapping(c);

send msg to WS2;

**endfor;**

}

**4. sending messages**

**for all** c ∈ C of BSSM2 **do**

msg= loweringSchemaMapping(c);

send(msg, WS2) /\*send msg to WS2\*/;

**endfor;**

---

Comme le montre la Figure 3.6 les actions de communication du WSPM via son interface est un ensemble de messages comme suit :

1. Réception d'une requête du WS requestor.
2. Envoi de cette requête au WS provider.
3. Réception d'une réponse du WS provider.
4. Envoie de cette réponse au WS requestor.

Le WSPM grâce à sa capacité de médiation peut effectuer des transformations de ces messages pour assurer la communication entre les services Web requestor et provider.

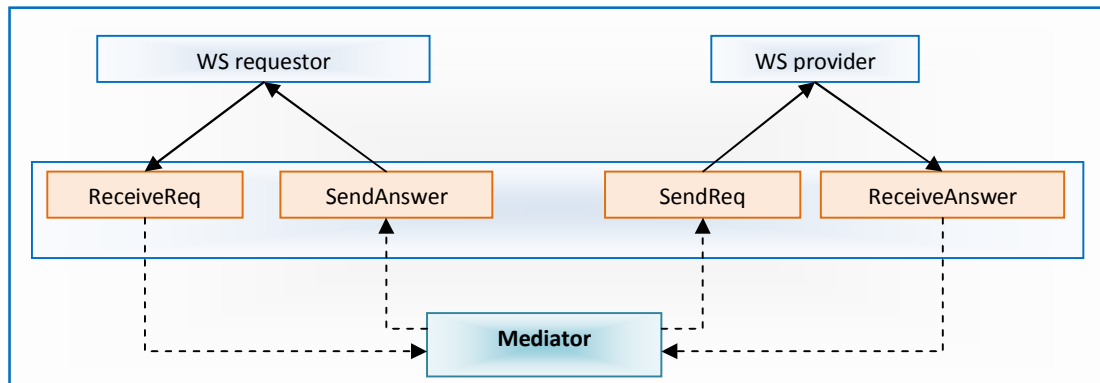


Figure 3.6. Interface du WSPM

### 3.3.3. Etapes de la médiation sémantique des processus métiers

Le processus de médiation comprend les trois grandes étapes suivantes (voir Figure 3.7) :

**1. Etape d'abstraction** : permet d'associer un message XML source/cible annoté, en un ou plusieurs concepts appartenant au modèle sémantique BSSM. Il s'agit des transformations syntaxiques-sémantiques (SynSemT).

**2. Etape de transformation sémantique** : permet de transformer sémantiquement les concept(s) de l'ontologie source et ceux de l'ontologie cible. Cette transformation est réalisée grâce aux mappings sémantiques et exploite les ontologies du BSSM, en faisant appel à l'algorithme de médiation et à un moteur d'inférence. Il s'agit des transformations sémantiques-sémantiques (SST).

**3. Etape de concrétisation syntaxique** : permet de transformer les concepts ontologiques résultants du BSSM, appartenant au niveau sémantique cible au format de messages XML cible du service concerné. Il s'agit des transformations sémantiques-syntaxiques (SemSynT).

Il est à noter que notre objectif principal est de concevoir et réaliser ces étapes de médiation par la création des trois modules de transformation, à savoir :

- **Le module SST** : responsable des transformations sémantiques-sémantiques (flèches 2 et 2' sur la Figure 3.7) se basant sur les mappings d'ontologies des BSSM.
- **Le module SynSemT** : responsable des transformations syntaxiques-sémantiques (flèches 1 et 1' sur la Figure 3.7).
- **Le module SemSynT** : responsable des transformations sémantiques-syntaxiques (flèches 3 et 3' sur la Figure 3.7).

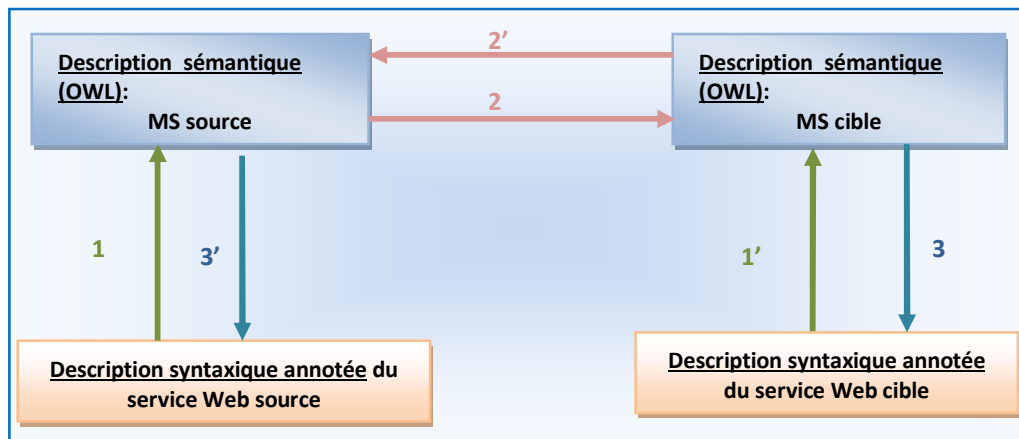


Figure 3.7: Etapes de médiation des processus métiers

## Fonctions de conversion et de transformation utilisées

Selon le principe de notre médiation des processus métiers 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, définissant comment une instance du document XML est transformée vers un ou plusieurs concepts du modèle sémantique. Il s'agit de transformations des données véhiculées par les messages. Pour les messages, on utilise les annotations SAWSDL « ModelReference » associant le message du WSDL à son concept dans le BSSM.
- ✓ **Les transformations sémantiques-syntaxiques**, elles concernent les données, en utilisant les annotations SAWSDL « loweringSchemaMapping » associant un concept du modèle sémantique à un mapping vers une donnée du document XML, définissant comment un concept sémantique est transformé en une donnée XML. Il s'agit aussi de transformations des données véhiculées par les messages.
- ✓ **Les transformations sémantiques-sémantiques**,
  - D'abord pour les données, en plus des mappings classiques d'ontologies nous utilisons des fonctions de conversion des valeurs des concepts grâce aux valeurs des modifieurs de l'ontologie contextuelle temporaire source et des valeurs des modifieurs de l'ontologie contextuelle temporaire cible. Ces mappings sont pris en charge par le WSDM [MEC 12].
  - Ensuite, pour les messages, ces mappings sont pris en charge par le WSPM précisément par le module SST, avec l'algorithme de médiation exploitant les ontologies du modèle comportemental BSSM et ses informations contextuelles

ainsi que les règles de transition prédéfinies. Le WSPM appelle le WSDM pour les mappings des données.

Dans ce qui suit, nous détaillons chaque module à part. Cependant, il est à noter que ces modules fonctionnent ensemble pour effectuer la médiation sémantique des processus métiers (et des données) comme le montre l'algorithme (voir Listing 3.1).

### 3.3.4. Structure du module SST

Notre approche de développement du module SST, responsable des transformations sémantiques-sémantiques, est basée sur la notion de correspondances d'ontologies (mappings) des BSSM et le raisonnement sur ces mappings. Elle peut être résumée comme suit (voir Figure 3.8):

- Utiliser les ontologies BLO et BCO du BSSM capturant la sémantique des deux SW hétérogènes en interaction (au niveau processus).
- Créer des mappings entre ces deux ontologies :
  - fusion des deux ontologies pour avoir l'ontologie SST,
  - création des règles de correspondance entre les concepts de l'ontologie SST.
  - Utiliser la technique de raisonnement sur ces mappings : exécution des règles par un moteur d'inférence pour faire inter-opérer ces deux ontologies et par conséquent résoudre les conflits sémantiques.

La Figure 3.8 illustre le schéma de l'architecture du module SST qui découle de cette approche :

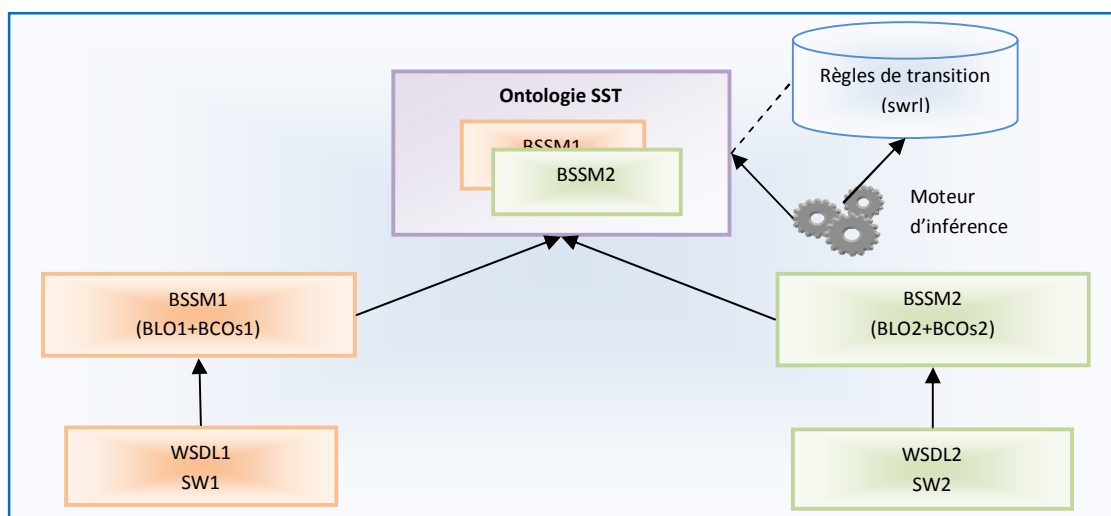


Figure 3.8. Architecture du module SST

## Modèle sémantique comportemental

Notre BSSM est une partie de l'architecture sémantique globale décrite dans le chapitre précédent (voir section 2.5.2). Il contient, pour chaque service Web, les ontologies locales comportementales (BLO) décrivant la sémantique des messages XML échangés entre services Web au niveau de leurs fournisseurs (propriétaires des SW), ainsi que les ontologies contextuelles comportementales (BCO) associées aux concepts de l'ontologie locale.

Il est à noter que nos ontologies seront compatibles sur le plan syntaxique et terminologique, mais, elles seront hétérogènes sur le plan conceptuel (sémantique) (voir section 1.4.3.1). L'approche utilisée est l'approche multi-ontologies (voir section 1.4.3).

### Règles de transition

Les règles de transition, ou de mappings, expriment les changements des instances des concepts ontologiques. Chaque règle est exprimée par des conditions et des effets à l'aide des expressions de la logique de description (DL). Nos règles sont écrites en utilisant le SWRL (Semantic Web Rule Language) [HOR 04] qui offre une forte expressivité pour enrichir les ontologies OWL, en plus de sa capacité à manipuler les instances des ontologies par des variables.

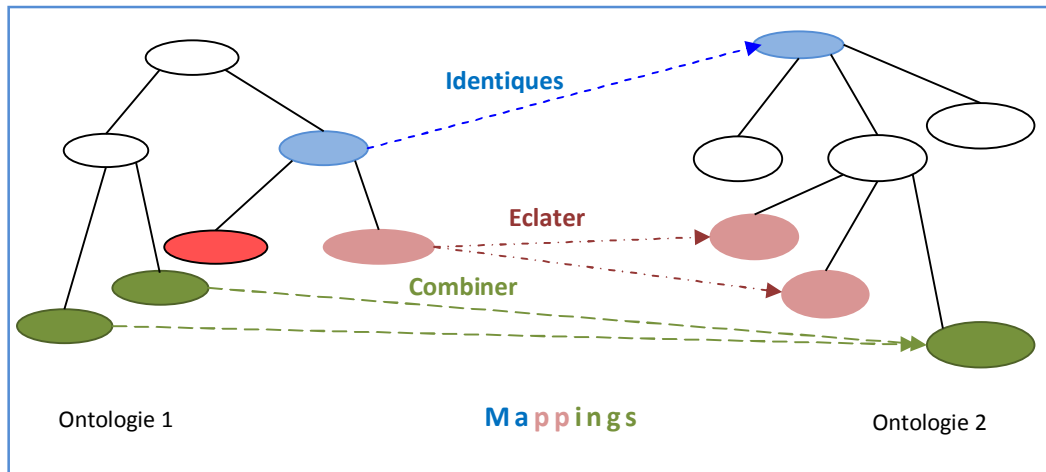


Figure 3.9. Les trois cas de transition

Dans notre cas, les règles permettent d'instancier des concepts d'une ontologie avec les instances des concepts d'une autre ontologie. Nous pouvons avoir cinq cas dont les trois premiers conduisent à des transitions (voir Figure 3.9) c'est-à-dire changements d'état :

- ✓ Un concept (P=out) de l'ontologie 1 instancie un concept (P=in) de l'ontologie 2 (concepts identiques).
- ✓ Un concept (P=out) de l'ontologie 1 instancie deux concepts (P=in) de l'ontologie 2 (split/éclater), c'est-à-dire un concept de l'ontologie 1 équivalent à deux concepts de l'ontologie 2.

- ✓ Deux concepts de l'ontologie 1 (P=out)instancient un concept (P=in) de l'ontologie 2 (merge/combiner), c'est-à-dire deux concepts de l'ontologie1 équivalents à un concept de l'ontologie 2.
- ✓ Un concept de l'ontologie 1 avec P=out mais qui n'instancie aucun concept de l'ontologie 2, c'est-à-dire ce concept de l'ontologie 1 n'est équivalent à aucun concept de l'ontologie2. Le médiateur ne fait rien pour ce concept.
- ✓ Un concept parmi les trois premiers cas de l'ontologie 1 possède son information contextuelle pour Ack=vraie, ceci signifie qu'il a besoin d'un accusé de réception, qui sera généré par le médiateur.

### 3.3.5. Rôle du module SST dans la médiation des processus

La médiation est prise en charge par le service Web pour la médiation des processus (Web Service of Process Mediation : WSPM) qui comprend trois modules : le module de transformations sémantiques-sémantiques (SST pour Semantic-Semantic Transformation), le module de transformations syntaxiques-sémantiques (SynSemT) et celui de transformations sémantiques-syntaxiques (SemSynT).

Le WSPM est inséré entre le service Web demandeur et le service Web fournisseur, il intercepte leurs messages et procède aux éventuelles médiations en exploitant les modules de transformations. Parmi ces modules nous avons notre SST qui utilise le modèle sémantique (BSSM) et fait appel à un moteur d'inférence pour exécuter les règles de transition permettant la médiation des chorégraphies des services Web en interaction.

La Figure 3.10 positionne notre module SST (en rouge) par rapport aux autres modules dans l'architecture du service Web de médiation WSPM.

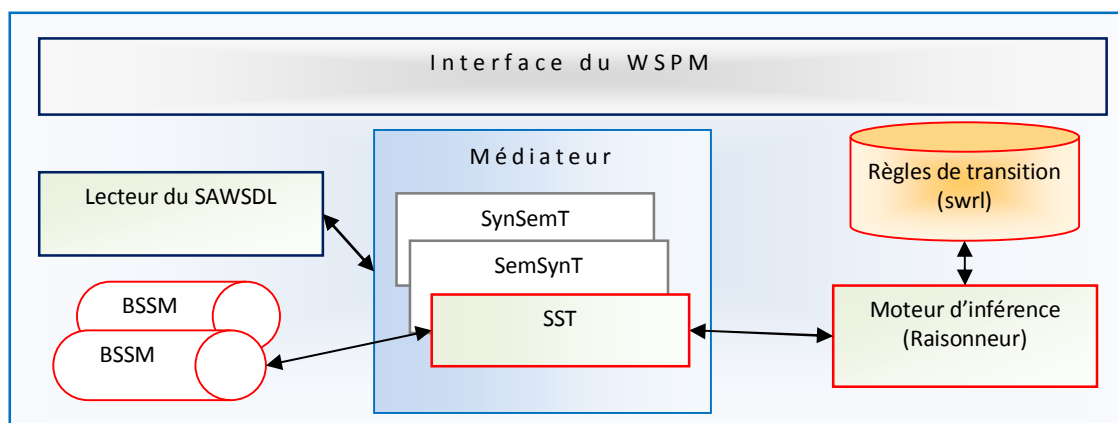


Figure 3.10. Module SST dans le WSPM

### 3.3.6. Description des modules SynSemT et SemSynT

Ils concernent les mappings des descriptions des services Web vers les concepts du modèle sémantique et vice versa.

- ✓ **Les transformations syntaxiques-sémantiques**, pour les données : on utilise 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. Pour les séquences d'échange de message, il s'agit simplement d'une association syntaxique aux concepts du modèle à l'aide de l'attribut SAWSDL « `ModelReference` ».
- ✓ **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.


### 3.4. Conclusion

Nous avons présenté, dans ce chapitre, l'approche et l'infrastructure de médiation des services Web sémantiques, notamment au niveau des processus métiers.

Dans notre approche, la réconciliation des séquences d'échange des messages des services Web est prise en charge par le WSPM (Web Service of Process Mediation). Il peut résoudre les conflits par inversion de l'ordre des messages, arrêter un message inattendu, combiner plusieurs messages et éclater un message. Il est implémenté par un algorithme de médiation de chorégraphies et un ensemble de règles qui permettent d'identifier les équivalences entre les différents modèles sémantiques (BSSM : Behavioral Semantic Sub Models).

Nous avons décrit le module noyau de ce médiateur. Il s'agit d'un module permettant les Mappings entre deux modèles capturant la sémantique behavioriste (BSSM) de deux services Web en interaction.

Dans le chapitre suivant, nous allons décrire l'implémentation de deux études de cas pour illustrer et valider l'architecture et l'approche que nous avons proposées.



# Chapitre 4

## Mise en œuvre de l'architecture et approche proposées

# Chapitre 4 : Mise en œuvre de l'architecture et approche proposées

## 4.1. Introduction

Les deux chapitres précédents ont permis de répondre à un certain nombre de préoccupations portant sur l'interopérabilité des systèmes d'information à travers les services Web sémantiques. En effet, dans le chapitre 2 nous avons présenté l'architecture globale en couches pour faire inter-opérer les applications de différents SI, où la couche de base concerne la représentation des applications à l'aide des services Web. La seconde couche représente la description sémantique des services obtenus. Et enfin, la troisième couche comprend plusieurs types de services d'intégration (découverte, médiation, exécution). Dans cette architecture nous nous sommes intéressés à deux parties essentielles qui sont le modèle sémantique et l'infrastructure de médiation. Dans la suite du chapitre 2, nous avons détaillé le modèle sémantique décrivant toutes les propriétés des services Web. Nous avons, en particulier, expliqué les sous modèles de données et des propriétés comportementales qui sont basés contextes. Le chapitre 3 a apporté des réponses à la problématique de médiation des propriétés des services Web de la première couche en utilisant les modèles sémantiques de la deuxième couche.

L'infrastructure de médiation des séquences d'échange de messages a été amplement détaillée. Dans ce chapitre, nous allons nous intéresser aux aspects liés à l'implémentation de notre prototype et aussi à son expérimentation à travers deux cas réels. Aussi, la section 4.2 exposera les objectifs du prototype à travers ses fonctionnalités à implémenter, ainsi que les outils utilisés. Les sections 4.3 et 4.4 présenteront les expérimentations que nous avons réalisées sur deux cas réels. La première étude de cas pour valider la création d'une part, de la couche des services et d'autre part, des ontologies du modèle sémantique des données échangées entre deux services Web, et ceci pour résoudre les conflits de données. La deuxième étude de cas représentant les séquences d'échange de messages entre deux services Web (chorégraphie) est utilisée pour valider, d'abord, la création du modèle sémantique comportemental, puis, valider la médiation des processus.

## 4.2. Objectifs et fonctionnalités du prototype

L'implémentation du prototype a pour but de valider les fonctionnalités décrites par les cas d'utilisation de la Figure 4.1. Ces cas d'utilisation existent en mode conception et en mode exécution.

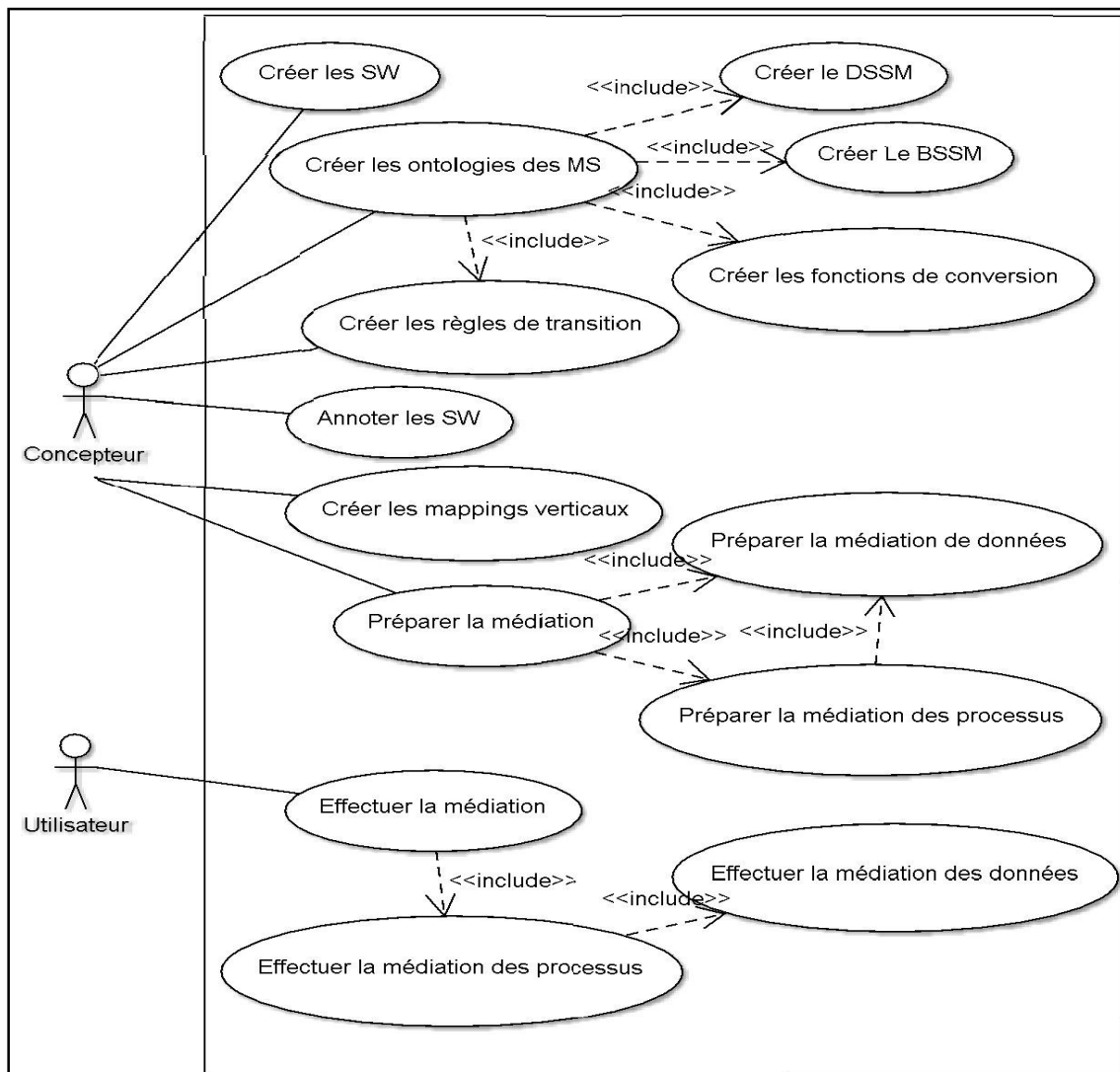


Figure 4.1 : Diagramme des cas d'utilisation du prototype

Lors de la conception nous avons les cas suivant :

1. La construction de la couche de services Web: il s'agit de définir les services métiers des différentes applications des systèmes d'information. Ces services sont à exposition externe via les services Web. Ils permettent l'interopérabilité inter-entreprises.
2. La construction de la couche sémantique par la création des différentes ontologies des sous modèles (voir section 2.5). Cette fonctionnalité comprend quatre sous cas d'utilisation :
  - a. Création du sous modèle des données (DSSM, Data Semantic Sub Model) échangées avec pour chaque service Web une ontologie locale de données (DLO, Data Local Ontology), des ontologies contextuelles de données (DCO, Data Contextual Ontology). Chaque DCO est associée à un concept de DLO. En plus d'une ontologie de domaine de données (DDO, Data Domain Ontology) commune pour tous les services en interaction.
  - b. Création des fonctions de conversion.

- c. Création du sous modèle comportemental (BSSM, Behavioral Semantic Sub Model) de chaque service Web qui est composé d'une ontologie locale comportementale (Behaviorist Local Ontology : **BLO**) et de plusieurs ontologies contextuelles comportementales (Behaviorist Contextual Ontology : **BCO**).
  - d. Création des règles de transition (mappings sémantiques-sémantiques).
3. Annotation des services Web à l'aide des attributs du SAWSDL.
  4. Création des mappings syntaxiques-sémantiques (liftingSchemaMapping) et vice versa (loweringSchemaMapping).
  5. Préparation de la médiation comprenant deux sous cas d'utilisation : médiation des données et médiation des processus. Cette dernière est réalisée par la création du service Web de médiation des processus (WSPM) implémenté par l'algorithme de chorégraphie (voir section 3.3) gérant la conversation entre deux services Web. Il utilise les ontologies des deux parties (BSSMs) et les règles de transition. WSPM fait appel au service Web de médiation de données WSDM (Web Service of Data Mediation) et à un raisonneur.

Lors de l'exécution, l'algorithme de médiation sera réalisé en plusieurs modules comme suit :

- D'abord, téléchargement des fichiers WSDL annotés des services Web demandeur et fournisseur. Extraction des annotations des opérations et celles des données (ModelReference, lifting & LoweringSchemaMappings attributs) en utilisant l'API SAWSDL4J. Ces annotations se réfèrent aux concepts d'ontologies des BSSMs et aux mappings verticaux (syntaxique-sémantique et vice versa). Chargement des copies des ontologies dans la mémoire.
- **le module SynSemT** : lors de la réception de messages, des transformations syntaxique-sémantiques sont réalisées en utilisant les attributs SAWSDL liftingSchemaMappings et en appelant le service Web de médiation des données (WSDM) pour transformer les instances du contexte du service Web source dans le contexte du service Web cible. Ces données conceptuelles sont enregistrées en mémoire (C).
- **le module SST** : en fusionnant les deux BSSMs en une ontologie SST et en utilisant les attributs ('P', 'Ord' et 'Ack') des ontologies contextuelles (BCO) et les règles de transition, le WSPM effectue des transformations sémantiques: génération de messages, réorganisation des messages, combinaison de plusieurs messages dans un seul, et éclatement d'un message en plusieurs autres messages.
- **le module SemSynT** : pour envoyer les données conceptuelles (C), l'algorithme effectue les transformations sémantiques-syntaxiques en utilisant l'attribut

lowringSchemaMapping pour obtenir les données XML (msg). Ensuite, l'algorithme envoie le message (msg) au service Web en utilisant l'attribut modelReference.

En ce qui concerne les outils de développement, il est à noter que notre prototype est implémenté en utilisant le langage Java sous l'environnement NetBeans qui est un projet open source ayant un succès et une base d'utilisateurs très large [18]. Plusieurs autres outils sont utilisés tels que :

- Le serveur glassfish [19] et les composants EJB (Entreprise Java Beans) [20] pour l'implémentation de l'application Web distribuée.
- L'API Java pour les services Web XML (JAX-WS). Elle fait partie de la plateforme Java EE et permet la création des services Web SOAP.
- Pour créer des services Web sur une plate-forme différente, nous utilisons WampServer sous Windows avec le serveur Apache2, le langage de scripts PHP5 [21] et d'une base de données MySQL [22].
- Protégé 2000 qui est un outil gratuit écrit en Java, est utilisé pour créer toutes nos ontologies [7].
- SWRL (Semantic Web Rule Language) [HOR 04] et JESS [9] pour créer et tester les règles de transition.
- L'API SAWSD4J pour gérer les annotations des WSDLs.
- L'API Jena [23] et l'outil Pellet [8] pour implémenter le moteur d'inférence qui exécute les règles de transition.

### **4.3. Création de la couche des services Web et du modèle sémantique de données**

Comme nous l'avons déjà souligné, pour valider la création de la couche des services et les ontologies du modèle sémantique des données (DSSM), nous présentons une étude de cas simple, mais complète, qui nécessite la création et l'interaction de plusieurs services Web appartenant à des SI et des plates-formes différents. Nous composons d'abord ces services, puis nous montrons qu'un échec de cette composition peut survenir à cause d'hétérogénéités dans les données échangées. Ce qui nous mène à décrire sémantiquement les données des services Web pour pouvoir résoudre ce conflit à l'aide d'une médiation.

#### **4.3.1. Présentation de la première étude de cas**

Pour illustrer notre problématique et valider notre approche nous présentons une étude de cas de planification d'un voyage au Japon, qui consiste en la réservation de billets d'avion et d'une chambre d'hôtel.

Cette planification nécessite la coordination de plusieurs services Web au sein d'une même composition à savoir :

- Un service Web de réservation de billets d'avion (**SWba**),
- Un service Web de réservation d'une chambre d'hôtel (**SWch**),
- Un service Web pour calculer le prix total (**SWca**) à partir des prix des deux services précédents.

Le processus de cette planification de voyage est représenté par la Figure 4.2.

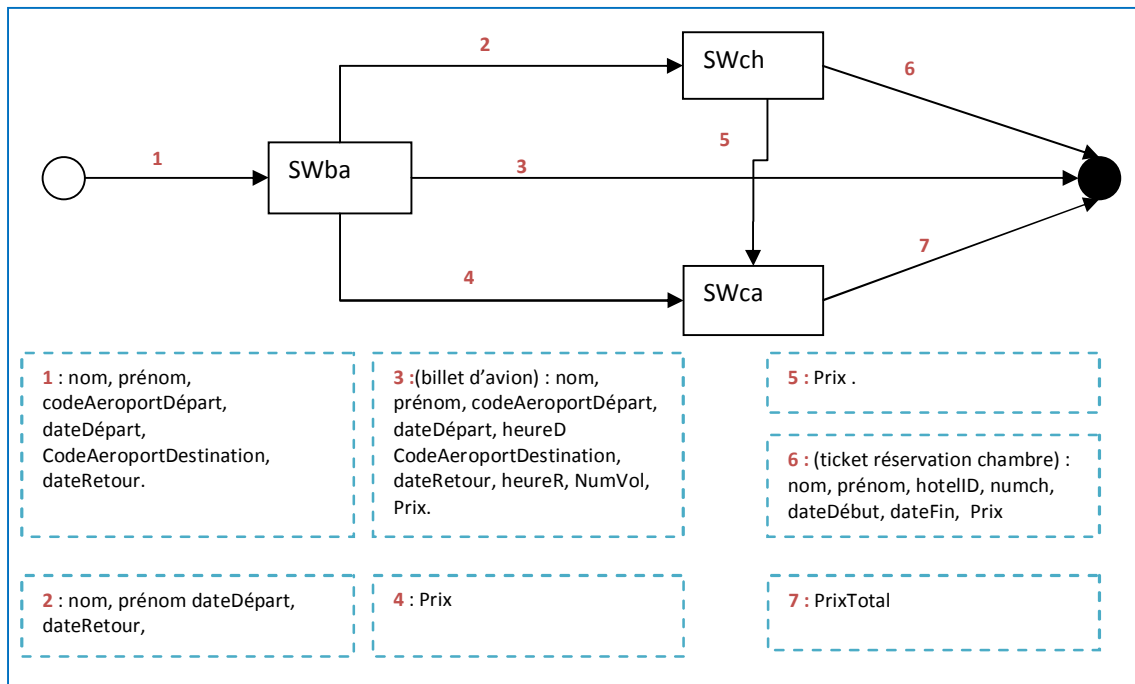


Figure 4.2. Flux d'information 'planification de voyage'.

### Service de réservation de billet

Nous supposons que le service de réservation de billet d'avion appartient à un système français, la Figure 4.3 nous illustre le diagramme de classes représentant la réservation de billet d'avion.

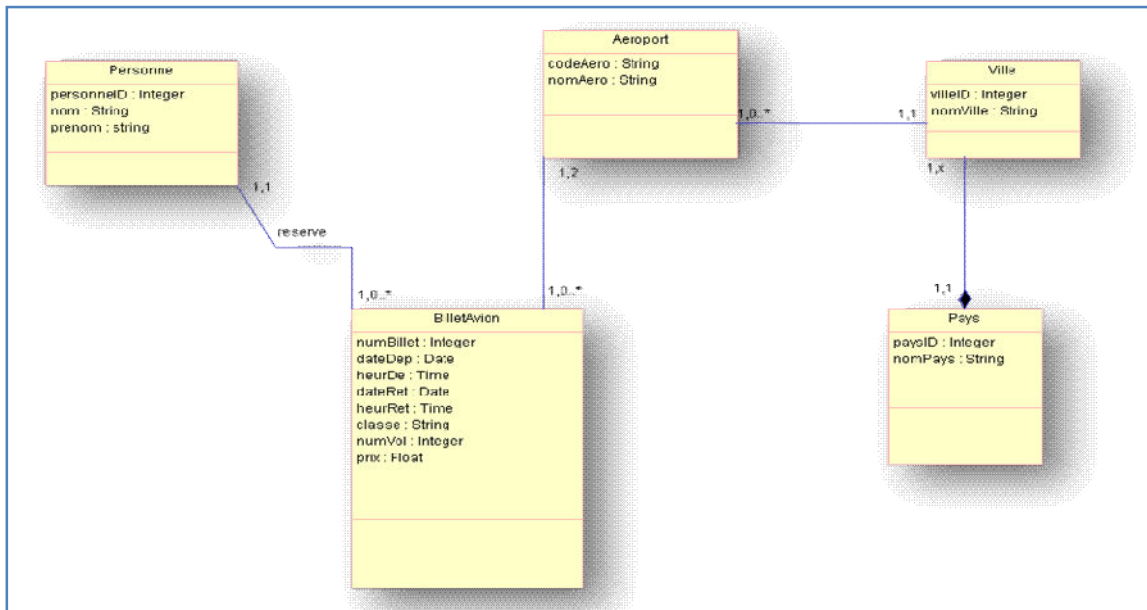


Figure 4.3. Diagramme de classes de la réservation de billet d'avion.

- Une personne effectue une ou plusieurs réservations de billet d'avion.
- Un billet est l'objet d'une seule réservation, d'un aéroport de départ et de destination.

### Service de chambre d'hôtel

Nous supposons que le service de réservation de chambre est un service de tri d'hôtels japonais permettant la réservation de chambre dans un hôtel choisi. Voici le diagramme de classe (voir Figure 4.4) décrivant la réservation de chambre d'hôtel.

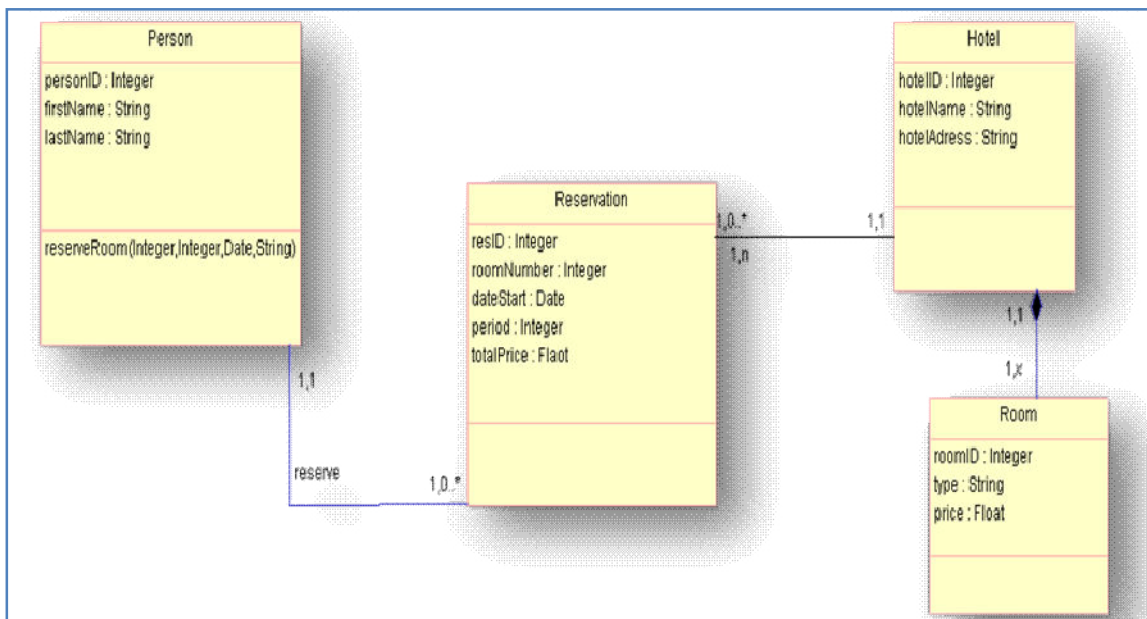


Figure 4.4. Diagramme de classes de la réservation de chambre d'hôtels.

- Une personne réserve un hébergement dans un hôtel.

- La réservation concerne un seul hôtel.

### Service de calcul du total

Le service de calcul permet de calculer le total des frais nécessaires au voyage, en additionnant les prix des deux services précédents.

### Composition des services

Pour réserver un voyage au Japon, nous avons créé un service Web supplémentaire jouant le rôle de coordinateur entre les trois services (SWba, SWch, SWca), ceci pour optimiser l’invocation des services par le client que nous avons construit. La Figure 4.5 représente le diagramme d’activité illustrant l’interaction entre client et services.

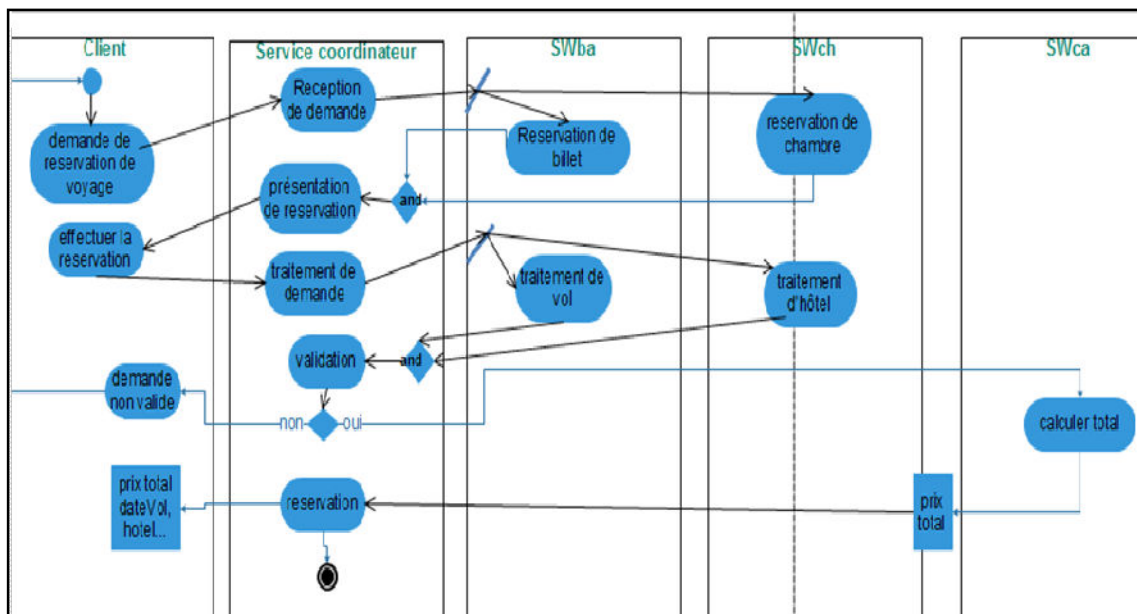


Figure 4.5. Diagramme d'activité -client/services-

Nous voyons que le processus global s’effectue à travers les étapes suivantes :

- **L’invocation** : le client invoque le service coordinateur pour effectuer une réservation complète d’un voyage au Japon (réservation de billets d’avion, réservation de chambres d’hôtel). Le service coordinateur, à son tour, « invoque » les services concernés pour exécuter la requête du client qui n’est pas censé savoir qu’il s’agit d’une collaboration de plusieurs SW.
- **La réservation d’un billet d’avion** : cette étape est assurée par le service de réservation de vol (SWba).



du vol ainsi que de sauvegarder la réservation dans la table « billetavion » contenue dans la base de données « vols ».

```
@WebMethod(operationName = "totalBillet")
public double totalBillet(@WebParam(name="classVol") String classVol, @WebParam(name="nbAdulte") int nbAdulte,
@WebParam(name="nbkids") int nbkids){

    return ((prixAdulte*nbAdulte)+(prixKids*nbkids));
}
```

Figure 4.7. Extrait de la méthode calculant le prix d'un billet d'avion

### Service de réservation de chambre (SWch)

← → 🏠 localhost:8080/Room\_WebService/Chambre\_WebService?Tester

## Room\_WebService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

**Methods :**

public abstract void services.ChambreWebService.room(java.lang.String,java.lang.String,java.lang.String,java.lang.String,java.lang.String)  
room (  |  |  |  |  )

public abstract double services.ChambreWebService.totalChambre(java.lang.String,int,int,int,int)  
totalChambre (  |  |  |  |  )

public abstract java.lang.String services.ChambreWebService.reservChambre(int,java.lang.String,int,double,int)  
reservChambre (  |  |  |  |  )

Figure 4.8. Test du service de réservation de chambre

La Figure 4.8 illustre le service permettant la réservation de chambres d'hôtels. C'est le service supposé appartenir à un système japonais. Il permet d'offrir une liste d'hôtels selon des critères donnés et d'établir la réservation dans l'hôtel choisi en se connectant à la base de données « hôtel ».

Il permet également de calculer le prix d'une réservation de chambre d'hôtel (voir Figure 4.9) en fonction du nombre d'adultes, d'enfants, de chambres, de jours et de types de chambres.

```

@WebMethod(operationName = "totalChambre")
public double totalChambre(@WebParam(name="classChambre") String classChambre,
    @WebParam(name="nbAdulte") int nbAdulte, @WebParam(name="nbKids") int nbKids,
    @WebParam(name="nbChambre") int nbChambre, @WebParam(name="periode") int periode) {

    return (((prixChambreAd*nbAdulte) + (prixChambreKid*nbKids))*nbChambre)*periode;
}

```

Figure 4.9. Extrait de la méthode calculant le prix de chambres

### Service de calcul de total (SWca)

## Total\_WebService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

#### Methods :

public abstract double services.TotalWebService.calculTotal(double,double)

calculTotal (  ,  )

Figure 4.10. Service de calcul total

Le service de calcul du total (voir Figure 4.10) assure la fonctionnalité de calculer le total entre la réservation de billets d’avion et l’hébergement dans un hôtel.

### Service ProcessWebService

Figure 4.11. Service process\_webService.

La Figure 4.11 représente le service ProcessWS, c'est le service de coordination entre les différents SW. Il joue le rôle d'un service main et encapsule un processus métier. C'est l'orchestrateur de nos autres services Web. Son rôle peut être résumé comme suit :

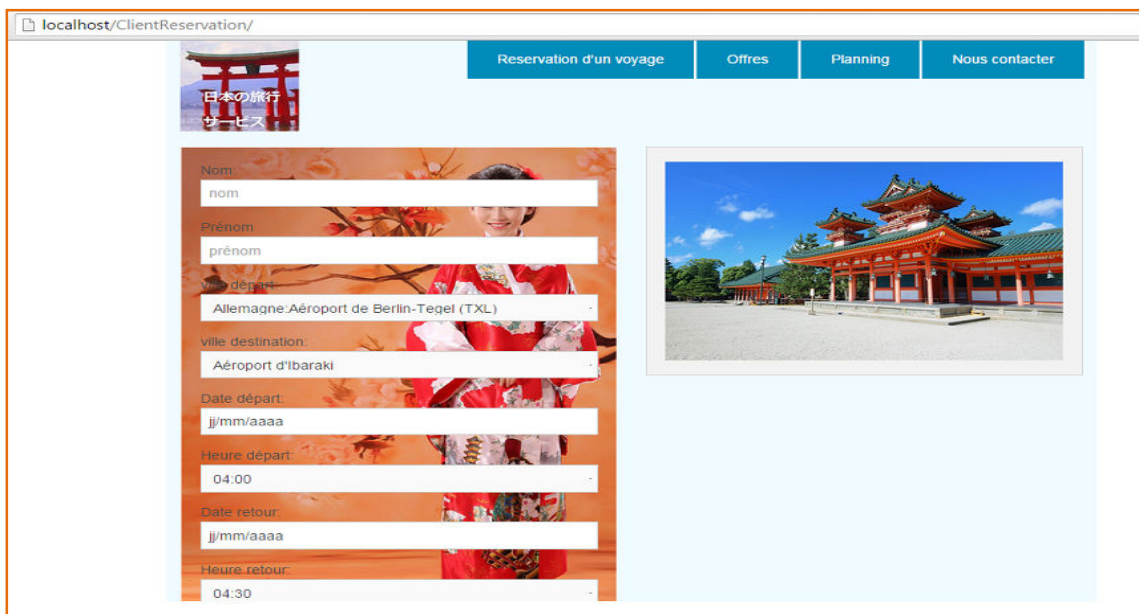
- C'est le service invoqué par le client.
- Il récupère les données saisies par l'utilisateur pour les transmettre comme paramètres aux différentes méthodes.
- Il fait appel aux méthodes publiées par les autres services Web afin d'élaborer la réservation de voyage.

ProcessWS fait appel à plusieurs autres méthodes publiées des services Web et effectue les traitements nécessaires pour rendre le résultat final au client.

### Le client

Nous avons développé un client pour consommer nos services Web (voir Figure 4.12). C'est une application Web en PHP, que nous avons nommée « CSV » (Client des Services de la réservation de Voyage).

Nous avons choisi de créer notre client en PHP pour montrer que l'exploitation des SW est indépendante des plateformes et technologies utilisées : nos SW sont en Java et notre client est en PHP. L'un des points forts des SW est qu'ils gèrent l'interopérabilité technique entre applications sous plateformes hétérogènes.



The screenshot shows a web browser window with the address bar displaying 'localhost/ClientReservation/'. The page has a light blue header with a navigation menu containing four items: 'Reservation d'un voyage', 'Offres', 'Planning', and 'Nous contacter'. Below the header, there is a main content area. On the left, there is a vertical form for booking a trip. The form fields are: 'Nom' (with 'nom' entered), 'Prénom' (with 'prénom' entered), 'ville départ' (with 'Allemagne:Aéroport de Berlin-Tegel (TXL)' selected), 'ville destination' (with 'Aéroport d'Ibaraki' selected), 'Date départ' (with 'jj/mm/aaaa' entered), 'Heure départ' (with '04:00' selected), 'Date retour' (with 'jj/mm/aaaa' entered), and 'Heure retour' (with '04:30' selected). The form background features a woman in a red kimono. To the right of the form is a large image of a traditional Japanese shrine building with a red facade and green roof. The browser's address bar and the navigation menu are highlighted with a blue border.

Figure 4.12. Page d'accueil du client

L'utilisateur doit remplir ce formulaire de réservation de voyage.

En validant le formulaire, le service ProcessWS récupère les données et effectue les traitements nécessaires pour proposer une liste d'hôtels disponibles dans la ville destination qu'offre le SWch. Le client choisit un hôtel et confirme la réservation ou bien l'annule. En confirmant la réservation, le SW ProcessSW récupère le choix de l'utilisateur et complète la réservation en contactant les SWba, SWch et SWca, puis informe l'utilisateur de la disponibilité de sa réservation.

### **Problème d'hétérogénéité**

Parmi les méthodes appelées par le processSW, nous avons :

- la méthode *totalChambre* (classChambre,nbAdulte,nbKids,nbChambre,periode) du SWch, illustrée dans la Figure 4.9, cette méthode rend le prix en Yen vu que le SWch est japonais,
- la méthode *totalBillet* (classVol, nbAdult, nbKids) du SWba, illustrée par la Figure 4.7, cette méthode rend le prix en Euro vu que le SWba est français.
- La méthode *calculTotal* (prixBillet, prixChambre) du SWca, cette méthode accepte deux entrées en Euro et rend le prix total en Euro.

Pour résoudre ce conflit, une capture de la sémantique des services Web est nécessaire ce qui permet une médiation par notre ProcessWS qui consiste en une conversion du prix du Yen en Euro.

### **4.3.3. Etapes d'implémentation de la couche sémantique des données**

Puisque les SW appartiennent à des systèmes d'informations différents, des conflits apparaissent entre les données échangées.

Pour résoudre les conflits causés par ce type d'hétérogénéités, une description et médiation sémantiques sont mises en place par notre architecture. Nous présentons par exemple, les modèles de données DSSM des services SWba et SWca (voir Figures 4.13 à 4.18).



Figure 4.13. Concepts de l'ontologie 'Vol'.

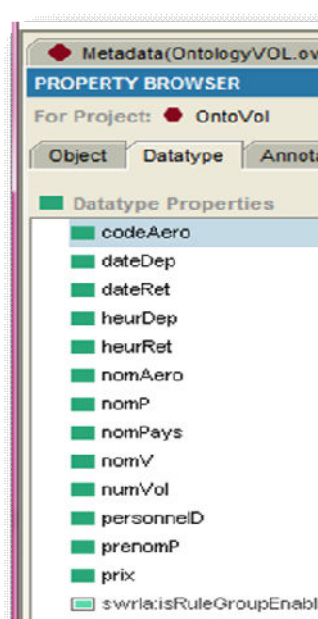


Figure 4.14. Propriétés des concepts de l'ontologie 'Vol'.



Figure 4.15. Relations entre concepts de l'ontologie 'Vol'.

Les Figures suivantes présentent l'ontologie « hôtel » relative au SWch.



Figure 4.16. Concepts de l'ontologie 'Hôtel'.

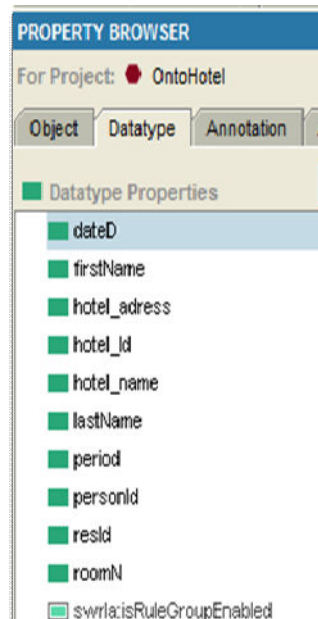


Figure 4.17. Propriétés des concepts de l'ontologie 'Hôtel'.

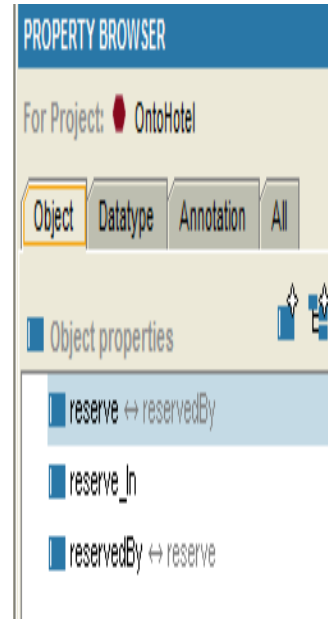


Figure 4.18. Relations entre concepts de l'ontologie 'Hôtel'.

## Annotation des SW

Nous avons notamment annoté les services Web comme présenté dans les Figures suivantes.

```

<xs:complexType name="operationReservBillet">
  <xs:sequence>
    <xs:element name="nom" type="xs:string" minOccurs="0"
      sawsdl:modelReference = "http://www.owl-ontologies.com/OntologyVOL.owl#nomP"
      sawsdl:liftingSchemaMapping= "http://www.owl-ontologies.com/mapping/operationReservBillet2Ont.xslt" />
    <xs:element name="prenom" type="xs:string" minOccurs="0"
      sawsdl:modelReference = "http://www.owl-ontologies.com/OntologyVOL.owl#prenomP"
      sawsdl:liftingSchemaMapping= "http://www.owl-ontologies.com/mapping/operationReservBillet2Ont.xslt" />
  </xs:sequence>
</xs:complexType>

```

Figure 4.19. Annotation du SWba

Le service SWba est le service source, c'est lui qui va envoyer des données au service SWca. Nous voyons qu'il a été annoté par la propriété `sawsdl:liftingSchemaMapping` (Figure 4.19) qui permet d'effectuer un mapping vertical : du service Web au concept d'ontologie. C'est un mapping syntaxique-sémantique.

```

<xs:complexType name="room">
  <xs:sequence>
    <xs:element name="hotele" type="xs:string" minOccurs="0"/>
    <xs:element name="lastname" type="xs:string" minOccurs="0"
      sawsdl:modelReference = "http://www.owl-ontologies.com/OntoHotel.owl#lastName"
      sawsdl:loweringSchemaMapping= "http://www.owl-ontologies.com/mapping/Ont2room.xslt"/>
    <xs:element name="firstname" type="xs:string" minOccurs="0"
      sawsdl:modelReference = "http://www.owl-ontologies.com/OntoHotel.owl#firstName"
      sawsdl:loweringSchemaMapping= "http://www.owl-ontologies.com/mapping/Ont2room.xslt"/>
    <xs:element name="dateStart" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

```

Figure 4.20. Annotation du SWca

La Figure 4.20 présente l'annotation du service SWca qui constitue la cible, ceci est décrit par la propriété `sawsdl:loweringSchemaMapping` qui permet aussi des mappings verticaux mais, cette fois-ci, du concept ontologique au SW (un mapping sémantique-syntaxique).

### Médiation des données

Grâce à cette annotation, le ProcessWS va jouer le rôle de médiateur de données (WSDM). Il utilise les informations contextuelles du concept prix des deux ontologies pour effectuer la médiation qui consiste en la conversion du prix du Yen en Euro comme le montre ce morceau de code illustré dans la Figure 4.21 suivante.

```

pHotel = pHotel*0.00748254;
DecimalFormat df = new DecimalFormat("#####.000");
String str = df.format(pHotel);
pHotel = Double.parseDouble(str.replace(',','.'));

```

Figure 4.21. Conversion de Yen en Euro

#### 4.4. Création de l'infrastructure de médiation des processus métiers

Pour évaluer comment notre approche résout les hétérogénéités entre les SWS au niveau processus, notre seconde étude de cas représente les séquences d'échange de messages entre deux services Web (chorégraphie). Elle est utilisée pour valider d'abord la création du modèle sémantique comportemental, puis, valider la médiation des processus.

##### 4.4.1. Présentation de la deuxième étude de cas

Il s'agit d'une simple étude de cas utilisant deux services Web chacun d'un système d'information différent de deux entreprises virtuelles A et B.

Les deux services Web sont :

- Le service Web Client (de l'entreprise A), demandant un devis relatif à un certain produit.
- Le service Web Fournisseur (de l'entreprise B), qui répond au devis du produit demandé.

Les séquences d'échange de messages entre ces deux services Web en interaction sont décrites dans la Figure 4.22 suivante :

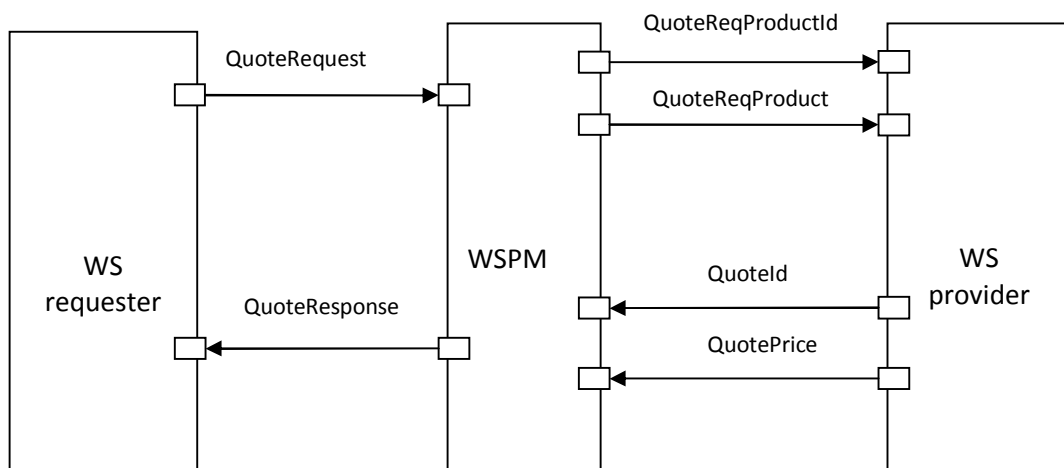


Figure 4.22. Chorégraphie de la demande de devis

Selon la chorégraphie présentée dans la Figure précédente, les interfaces comportementales des deux services Web sont présentées dans le Tableau 4.1.

Comportement du SW Client	Comportement du SW Fournisseur	Hétérogénéités
SendRequest(QuoteRequest)	ReceiveRequest(QuoteId) ReceieveRequest(QuotePrice)	Split message
ReceiveAnswer(QuoteResponse)	SendAnswer(QuoteId)	Combining messages
ReceiveAnswer(QuoteResponse)	SendAnswer(QuotePrice)	

Tableau 4.1. Séquence d'échange de messages avec hétérogénéités

Les Listings 4.1 et 4.2 montrent les descriptions des interfaces des services Web demandeur et fournisseur (documents WSDL annotés) pour la demande de devis.

**Listing 4.1:** Quote request Web service Requester Interface

```

<wsdl:description...>
  <wsdl:types>
    <xs:schema...>
      <xs:element name="QuoteRequest">
        <xs:complexType sawsdl:liftingSchemaMapping=' '
http://.../QuoteRequest2Ont.xslt''>
          <xs:sequence>
            <xs:element name="QuoteReqId" type="xs:integer" />
            <xs:element name="QuoteReqProduct" type="product" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="product">
        <xs:sequence>
          <xs:element name="ProductId" type="xs:integer">
          <xs:element name="ProductName" type="xs:string" >
        </xs:sequence>
      </xs:complexType>
      <xs:element name="QuoteResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="QuoteId" type="xs:integer" />
            <xs:element name="QuotePrice" type="xs:double" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </xs:types>

  <wsdl:interface name="quoteR">
    <wsdl:operation name="QuoteR" pattern="http://.../Out-In">
Sawsdl:modelReference="http://.../quoteR-BSSMr#QuoteRequest"
Sawsdl:modelReference="http://.../quoteR-BSSMr#QuoteResponse"
      <wsdl:output element="QuoteRequest"/>
      <wsdl:input element="QuoteResponse"/>
    </wsdl:operation>
  </wsdl:interface>
</wsdl:description>

```

**Listing 4.2:** Quote request Web service Provider Interface

```
<wsdl:description...>
  <wsdl:types>
    <xs:schema...>
      <xs:element name="QuoteReqId" type="xs:integer" />
      <xs:element name="QuoteReqProduct" type="product" />
    <xs:complexType name="product">
      <xs:sequence>
        <xs:element name="ProductId" type="xs:integer">
        <xs:element name="ProductName" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
    <xs:element name="QuoteId" type="xs:integer" />
      sawsdl:loweringSchemaMapping='' http://.../ Ont2QuoteId.lowering''
    <xs:element name="QuotePrice" type="xs:double" />
      sawsdl:loweringSchemaMapping='' http://.../ Ont2QuotePrice.lowering''
    </xs:schema>
  </xs:types>

  <wsdl:interface name="quoteR">
    <wsdl:operation name="QuoteRI" pattern="http://.../In-only">
      SawSDL:modelReference="http://.../quoteR-BSSMp#quoteReqId"
      <wsdl:input element="QuoteReqId"/>
    </wsdl:operation>
    <wsdl:operation name="QuoteRPr" pattern="http://.../In-only">
      SawSDL:modelReference="http://.../quoteR-BSSMp#quoteReqProduct"
      <wsdl:input element="QuoteReqProduct"/>
    </wsdl:operation>
    <wsdl:operation name="QuoteI" pattern="http://.../out-only">
      SawSDL:modelReference="http://.../quoteR-BSSMp#quoteId"
      <wsdl:output element="QuoteId"/>
    </wsdl:operation>
    <wsdl:operation name="QuoteP" pattern="http://.../out-only">
      SawSDL:modelReference="http://.../quoteR-BSSMp#quotePrice"
      <wsdl:output element="QuotePrice"/>
    </wsdl:operation>
  </wsdl:interface>
</wsdl:description>
```

Pour construire le modèle sémantique BSSM, nous utilisons ‘Protégé’ pour créer les deux ontologies locales et nous ajoutons, pour chaque concept, son ontologie contextuelle comportemental (voir Figure 4.23).

Les règles de transitions (TR) expriment des changements d’états. Dans notre cas, les changements concernent les instances ontologiques. Pour rendre la présentation simple, nous supposons qu’il n’y a pas de conflits dans les données (noms, types, ...etc.).

Voici les règles de transition utilisées dans notre étude de cas.

- **Règle de transition du SW client correspondant au quoteR-BSSMr**

**Règle 1:** Le SW Client crée une instance de ‘ QuoteRequest ‘ pour laquelle il a les instances de ‘ quoteReqId ‘ et ‘ quoteReqProduct ‘. Cette instance de ‘ QuoteRequest ‘ est envoyée au WSPM.

$$QuoteReqId (?QuoteReqId) \wedge QuoteReqProduct (?QuoteReqProduct) \rightarrow quoteRequest(?x[?QuoteReqId, ?QuoteReqProduct])$$

- **Règles de transitions du SW fournisseur correspondant au quoteR-BSSMp**

**Règle 2,3 :** Le service Web fournisseur crée les instances ‘ quoteId ‘ et ‘ quotePrice ‘ après avoir reçu les instances ‘ quoteReqId ‘ et ‘ quoteReqProduct ‘ du WSPM. Les instances créées sont envoyées au WSPM.

$$QuoteReqId (?QuoteReqId) \wedge QuoteReqProduct (?quoteReqProduct) \rightarrow quoteId (?x)$$

$$QuoteReqId (?QuoteReqId) \wedge QuoteReqProduct (?quoteReqProduct) \rightarrow quotePrice (?x)$$

- **Exemple d’une règle de mapping**, pour obtenir une réponse de devis, WSPM combine les sorties du service Web Fournisseur et les envoie au service Web Client :

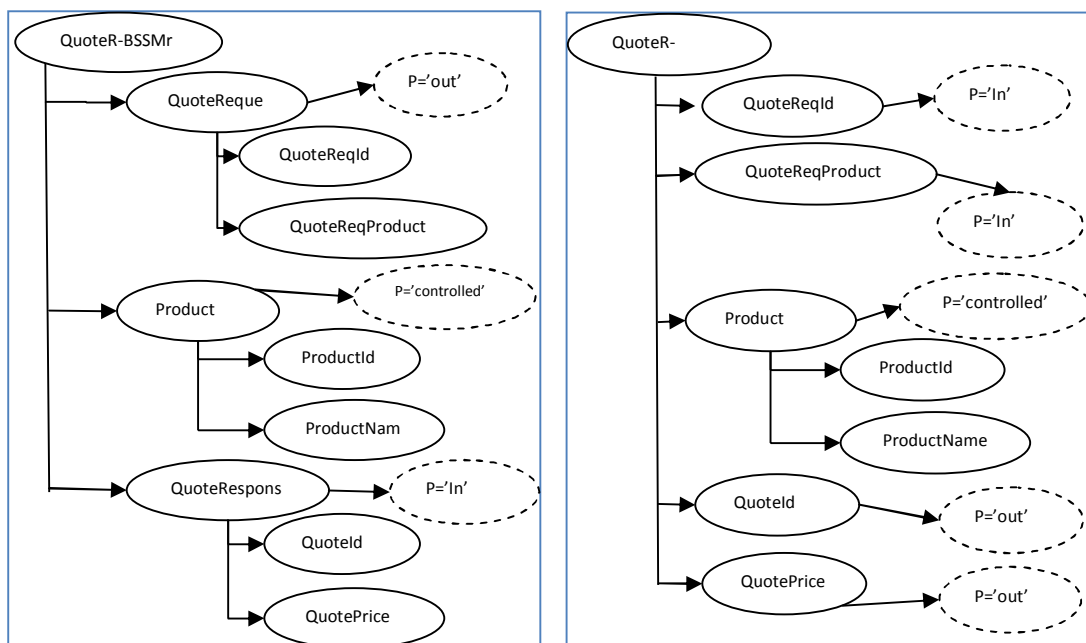
$$QuoteR-BSSMp:QuoteId(?qid) \wedge QuoteR-BSSMp: QuotePrice(?qprice) \wedge ?QResp = swrl:concat (?qid ?qprice) \rightarrow QuoteR-BSSMr: QuoteResponse(?QResp).$$


Figure 4.23. BSSM des services Web client et fournisseur.

#### 4.4.2. Etapes d'implémentation de l'infrastructure de médiation des processus

Nous allons, dans cette section, présenter notre implémentation étape par étape.

Les Figures 4.24 et 4.25 représentent les ontologies locales comportementales (BLO) des BSSM des services Web demandeur et fournisseur.

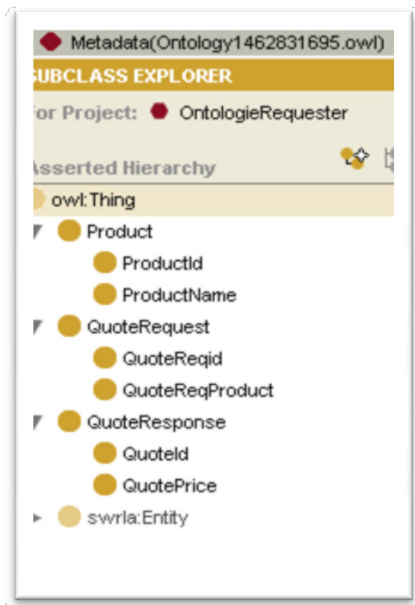


Figure 4.24. Ontologie du SW demandeur.

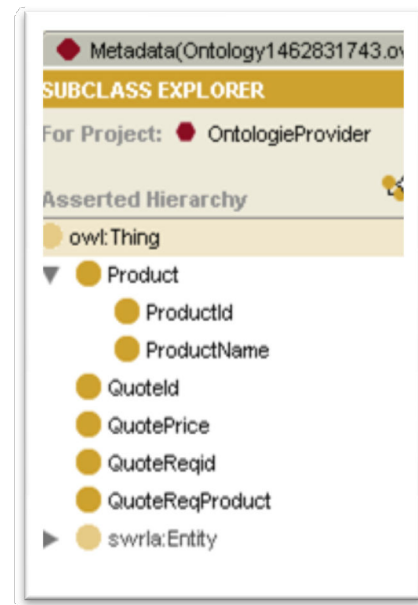


Figure 4.25. Ontologie du SW fournisseur.

#### Création de l'ontologie SST et des règles de mapping

Pour effectuer la médiation entre les deux ontologies 'Ontology Requester' et 'Ontology Provider' des deux modèles comportementaux (BSSMs), nous avons créé une troisième appelée 'SST Ontology' par fusion des deux premières. Ici, nous avons utilisé la fonction `owl:import` pour importer les deux ontologies 'Ontology Requester' et 'Ontology Provider' dans 'SST Ontology'. Ceci nous permet de manipuler les deux en même temps, en utilisant des règles de mappings SWRL que nous avons créés, puis exécutées avec succès (Figures 4.26, 4.27 et 4.28).

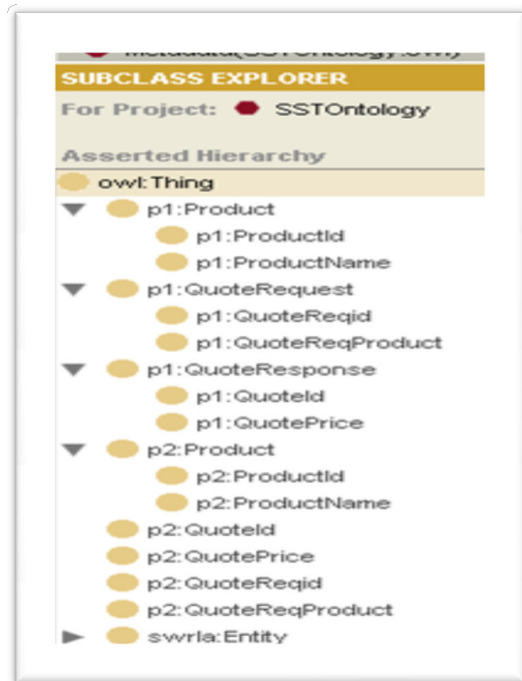


Figure 4.26. Ontologie SST

Enabled	Name	Exp
<input type="checkbox"/>	Rule_1	$\rightarrow p1:QuoteReqid(?x) \rightarrow p2:QuoteReqid(?x)$
<input type="checkbox"/>	Rule_2	$\rightarrow p1:QuoteReqProduct(?x) \rightarrow p2:QuoteReqProduct(?x)$
<input type="checkbox"/>	Rule_3	$\rightarrow p2:QuoteReqid(?QuoteReqid) \rightarrow p2:Quoteld(?QuoteReqid)$
<input type="checkbox"/>	Rule_4	$\rightarrow p2:QuoteReqProduct(?QuoteReqProduct) \rightarrow p2:QuotePrice(?QuoteReqProduct)$
<input type="checkbox"/>	Rule_5	$\rightarrow p2:Quoteld(?x) \wedge p2:QuotePrice(?y) \rightarrow p1:QuoteResponse(?x) \wedge p1:QuoteResponse(?y)$

Figure 4.27. Règles SWRL

Voici un exemple d'exécution d'une règle SWRL qui est sensée faire la transmission des données de l'ontologyProvider vers l'ontologyRequester, en combinant les instances de deux concepts de la première pour instancier un concept de la dernière.

Nous avons intégré le moteur de règle JESS avec l'éditeur du langage SWRL dans Protégé pour pouvoir exécuter ces règles.

```
Rule_8 → p2:QuoteId(?x) ∧ p2:QuotePrice(?y) → p1:QuoteResponse(?x) ∧ p1:QuoteResponse(?y)
```

Figure 4.28. Une règle SWRL à exécuter (Exemple)

Pour obtenir une réponse de devis, nous avons combiné les deux concepts ‘**QuoteId**’ et ‘**QuotePrice**’ de l’ontologieProvider préfixé par P2 pour les envoyer a l’ontologieRequester préfixé par p1 dans le concept ‘**QuoteResponse**’ (voir Figure 4.29 et 4.30). ceci signifie que les deux concepts **QuoteId** et **QuotePrice** de P2 sont équivalents à un seul concept **QuoteResponse** de P1.

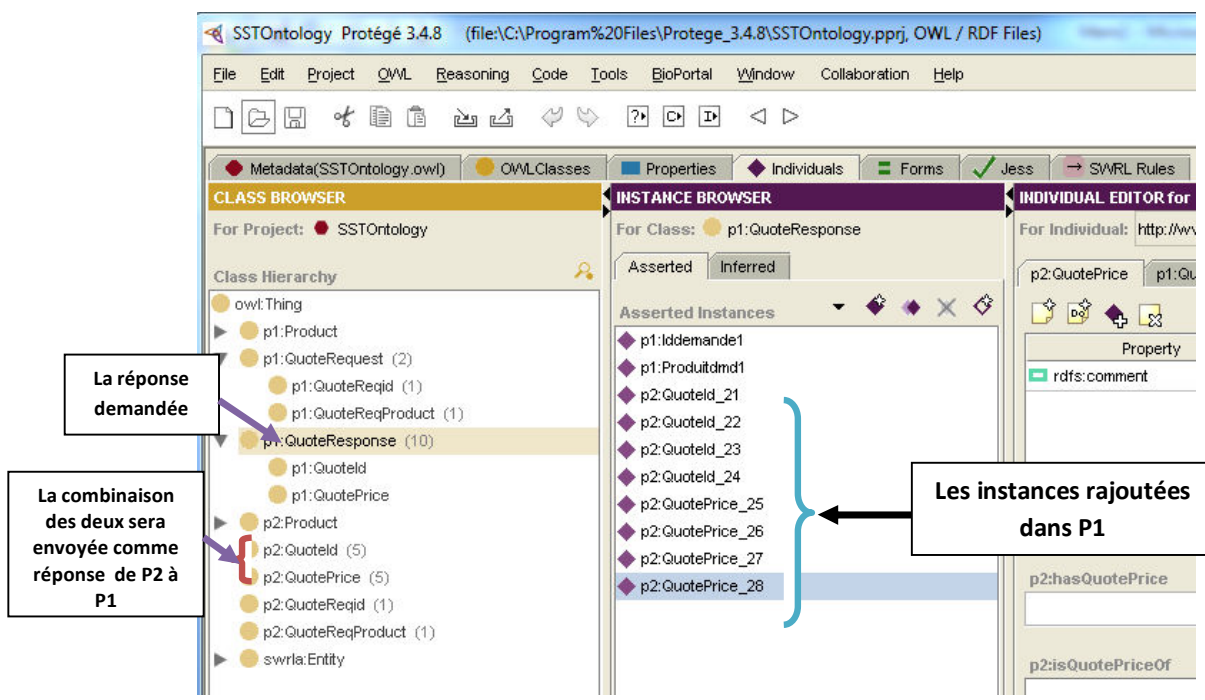


Figure 4.29. Résultat de l’exécution de la règle

## Module SST

Ce module est implémenté en Java sur l’IDE NetBeans. Il permet de charger les ontologies des deux services Web (Figure 4.30), l’ontologie SST, d’en extraire les concepts, les propriétés et d’exécuter les règles de mappings SWRL par un moteur d’inférence basé règles (Pellet et Jena) pour combiner ou éclater des instances de concepts d’une ontologie source pour instancier les concepts d’une autre ontologie cible, en créant des individus dans cette dernière.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns="http://www.owl-ontologies.com/Ontology1462831695.owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  <owl:Ontology rdf:about="http://www.owl-ontologies.com/Ontology1462831695.owl"/>
  <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1462831695.owl#QuoteReqProduct">
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1462831695.owl#QuoteRequest"/>
    </rdfs:subClassOf>
    <owl:disjointWith>
      <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1462831695.owl#QuoteReqid"/>
    </owl:disjointWith>
  </owl:Class>
  <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1462831695.owl#QuoteRequest">
    <owl:disjointWith>
      <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1462831695.owl#Product"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1462831695.owl#QuoteResponse"/>
    </owl:disjointWith>
  </owl:Class>
  <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1462831695.owl#ProductId">
    <owl:disjointWith>
      <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1462831695.owl#ProductName"/>
    </owl:disjointWith>
    <rdfs:subClassOf>
      <owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1462831695.owl#Product"/>
    </rdfs:subClassOf>
  </owl:Class>

```

Figure 4.30. Chargement de l'ontologie du SW Demandeur (extrait)

## 4.5. Conclusion

Dans ce chapitre, nous avons présenté l'implémentation de notre prototype de la médiation basée contexte pour l'interopérabilité des SWS. Il est réalisé en plusieurs étapes :

- ✓ Création de la couche des services Web.
- ✓ Création du modèle sémantique : sous modèle sémantique des données (DSSM : Data Semantic Sub Model) et le sous modèle sémantique des propriétés comportementales (BSSM : Behavioural Semantic Sub Model). En plus de la création des règles de transition.
- ✓ Annotation des services Web à l'aide du SAWSDL, qui lie les descriptions des services Web au modèle sémantique.
- ✓ Création des mappings syntaxiques-sémantiques et sémantiques-syntaxiques (modules SynSemT et SemSynT).
- ✓ Création des mappings sémantiques-sémantiques par la création du module SST. Il utilise en plus des ontologies contextuelles les règles de transitions prédéfinies.



# Conclusion générale et perspectives

## Conclusion générale et perspectives

Le travail de recherche effectué se situe dans le cadre général des architectures orientées services pour l'interopérabilité des systèmes d'information d'entreprises. L'adoption des SOA et des services Web constitue une avancée fondamentale dans le développement des systèmes d'information interopérables, notamment au niveau des plates-formes. Cependant, ces techniques ne traitent pas la problématique liée à l'interopérabilité sémantique qui, constitue l'enjeu majeur pour la coopération des systèmes d'information. En effet, dans le cadre des services Web, les hétérogénéités sont complexes et apparaissent à plusieurs niveaux, touchant leurs aspects fonctionnels, comportementaux et non fonctionnels, ceci rend leur interopérabilité difficile.

Dans le cadre de cette thèse nous nous focalisons sur la médiation des processus métiers publics c'est-à-dire la chorégraphie. En effet, la communication entre SW demandeur et SW fournisseur peut avoir lieu seulement si leurs Message Exchange Patterns (Patterns d'échange de messages) correspondent précisément. Ceci signifie que quand un des deux envoie un message, l'autre est capable de le recevoir, ce qui est impossible dans la plupart du temps. Les hétérogénéités des MEPs (aspects comportementaux) sont plus complexes que celles de données.

Notre travail de recherche a comme objectif principal la médiation des propriétés fonctionnelles et non fonctionnelles des services Web, dans le but de résoudre les conflits sémantiques pouvant apparaître entre le service Web émetteur et le service Web récepteur.

En effet, l'objectif du travail envisagé dès le départ est de proposer une approche et une architecture basée contexte pour la médiation des propriétés des services Web.

La première partie de notre travail a été consacrée à l'étude bibliographique relative à l'interopérabilité des systèmes d'information d'entreprises à l'aide des architectures orientées services et les services Web puis à l'aide des SWS [MEC 10b].

Cette étude nous a permis de constater que cette interopérabilité des services Web est réalisée par la composition de ces derniers en processus métiers, elle peut être une orchestration ou une chorégraphie. Nous avons relevé le problème d'hétérogénéités sémantiques des services Web en interaction, et la nécessité d'un système de médiation pour les résoudre. Notons que nous nous intéressons à la médiation des processus métiers et particulièrement la chorégraphie des services Web et leurs interfaces comportementales.

Nous avons débuté la deuxième partie de notre travail par la proposition d'une classification des travaux relatifs à l'interopérabilité sémantique via les services Web en nous basant sur des

critères que nous avons sélectionnés. Grâce à une étude comparative de cet état de l'art, nous avons positionné nos travaux de recherche par rapport aux autres.

Après avoir délimité le domaine de nos travaux de recherche, nous avons concentré nos efforts sur nos contributions pour répondre aux problématiques citées. L'architecture globale que nous proposons pour l'interopérabilité des systèmes d'information via les services Web sémantiques comprend plusieurs composants [MEC 17]. Nous nous intéressons à deux parties importantes de cette architecture qui sont l'infrastructure de médiation des services Web et le modèle sémantique permettant de les décrire.

Le rôle du modèle sémantique est la description des services Web et la capture des hétérogénéités explicitement pour pouvoir les traiter de façon semi-automatique par l'appel à la médiation. Notre modèle est scindé en quatre sous modèles selon la classification de la médiation que nous avons proposée et qui touche toutes les propriétés des services Web. Parmi ces sous-modèles, nous nous intéressons au :

- Sous modèle sémantique des données (DSSM : Data Semantic Sub Model) présenté dans nos travaux précédents [MEC 09, MEC 10a, MEC 12].
- Sous modèle sémantique des propriétés comportementales (BSSM : Behavioural Semantic Sub Model). Il comprend, pour chaque service Web, une ontologie locale comportementale et des ontologies contextuelles associées aux concepts de cette ontologie. Notre BSSM présente beaucoup d'avantages car:
  - il est basé sur la méthode d'ASM ce qui le rend un modèle à base d'état et de règles.
  - ces règles sont exprimées grâce au puissant langage SWRL.
  - il exploite la notion de contexte pour décrire les propriétés comportementales.
  - ce modèle de référence est relié à la description du service Web concerné par le SAWSDL.

La médiation, qui constitue le noyau de notre architecture, est nécessaire à tous les niveaux et pour toutes les tâches des services Web. En effet, la médiation des propriétés fonctionnelles et celle des propriétés non fonctionnelles peuvent être utilisées pendant la découverte et la sélection alors que la médiation des processus métiers et celle des données sont utilisées lors de l'exécution (invocation).

Dans notre approche, la réconciliation des séquences d'échange des messages des services Web est prise en charge par le WSPM (Web Service of Process Mediation). Il est implémenté par un algorithme de médiation de chorégraphie gérant la conversation entre les deux services Web. Il utilise les modèles comportementaux (BSSM) des deux parties et les règles de transition. WSPM fait appel au service Web de médiation de données WSDM (Web Service

of Data Mediation) et à un raisonneur. Le cœur de ce médiateur est le module de transformations sémantiques-sémantiques (SST) basé sur la notion de mappings d'ontologies des BSSM et le raisonnement sur ces mappings. Il comprend aussi les modules de transformations syntaxiques-sémantiques et sémantiques-syntaxiques (SynSemT et SemSynT).

L'implémentation de notre prototype de médiation basée contexte pour l'interopérabilité des SWS est réalisée selon deux études de cas, une pour la couche des services et l'autre pour la médiation des processus. Elle comporte plusieurs étapes :

- Création de la couche des services Web.
- Création du modèle sémantique : le sous modèle sémantique des données (DSSM : Data Semantic Sub Model) et le sous modèle sémantique des propriétés comportementales (BSSM : Behavioural Semantic Sub Model), en plus de la création des règles de transition.
- Annotation des services Web à l'aide du SAWSDL, qui lie les descriptions des services Web au modèle sémantique.
- Création des mappings syntaxiques-sémantiques et sémantiques-syntaxiques (modules SynSemT et SemSynT).
- Création des mappings sémantiques-sémantiques par la création du module SST, qui utilise, en plus des ontologies contextuelles, les règles de transitions prédéfinies.

Dans nos travaux futurs nous envisageons d'entreprendre:

- Une extension du modèle sémantique pour prendre en compte les propriétés fonctionnelles et non fonctionnelles nécessaires dans la découverte et la sélection sémantiques.
- Une amélioration de l'infrastructure de médiation des processus et celle des données en les reliant par l'appel de la première à la deuxième.
- Le parachèvement de l'infrastructure de médiation, en prenant en compte la médiation sémantique fonctionnelle et non fonctionnelle.



# **Références bibliographiques**

## Références bibliographiques

- [ARR 04] Arroyo S., Stollberg M., “WSMO Primer”, WSMO Deliverable D3.1, DERI Working Draft, Technical report, Web Service Modelling Ontology, 2004.
- [ARR 10] Arroyo S., Sicilia M. A., “Architecture and algorithms of the SOPHIE choreography framework”, *J Intell Inf Syst*, Vol. 34, pp. 193-221, 2010.
- [BEN 00] Benslimane D., Yétongnon K., Chraïbi S., Leclercq E., Abdelwaheb E., “Les systèmes d’information coopératifs : le projet DECA”. *Technique et science informatiques*. Vol. 19, No. 7, pp. 1009-1043, 2000.
- [BIA 11] Bia-Figueiredo M., Gillette Y., Morley C., “Processus métiers et S.I. - Gouvernance, management, modélisation”, 3e édition, Dunod, 2011.
- [BON 05] Bonnet P., “Cadre de référence Architecture SOA, Meilleures Pratiques”, Rapport technique, Montreuil, France, 2005.
- [BOR 08] Börger E., Bernhard T., “Modeling workflows, Interaction Patterns, Web services and Business Process: the ASM-Based Approach”, *Abstract State Machines B and Z*, first international conference ABZ’08, pp 24-38, London, UK, 2008.
- [BOU 10] Bouras T., Gouvas P., Mentzas G., “Dynamic Data Mediation in Enterprise Application Integration Scenarios”, In Mentzas G., Friesen A. (Eds.), “Semantic Enterprise Application Integration for Business Processes: Service-Oriented Frameworks”, *Business science reference*, pp. 19-35, Hershey, New York, , 2010.
- [BRU 06] Bruijn J., Ehrig M., Feier C., Recuerda F. M., Scharffe F., Weiten M., “Ontology Mediation, Merging, and Aligning”, In Davies J., Studer R., Warren P. (Eds.), “Semantic Web Technologies: Trends and Research in Ontology-based Systems”, WILEY, pp.95-113, 2006.
- [BUS 03] Bussler C., “The role of Semantic Web Technology in Enterprise Application Integration”, *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, Vol. 26, No. 4, pp. 62-68, 2003.
- [CAB 05] Cabral L., Domingue J., “Mediation of semantic Web services in irs-iii”, 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, 2005.
- [CAB 12] Cabral L., Domingue J., “Ontology Based Discovery of Semantic Web Services with IRS-III”, In Blake B., Cabral L., König-Ries B., Küster U., Martin D. (eds), “Semantic Web Services”, pp.191-202, Springer, 2012.

- [CHI 04] Chinnici R., Gudgin M., Moreau J.J., Schlimmer J., Weerawarana S. (eds.), "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language", Technical Report, W3C, 2004.
- [CHR 01] Christensen E., Curbera F., Meredith G., Weerawarana S., "W3C Notes, Web Service Description Language WSDL 1.1", Technical Report, W3C, 2001.
- [CIM 05] Cimpian E., Mocan, A., "D13.7 v0.1 Process Mediation in WSMX". In Cimpian, E. "WSMX Working Draft", Technical Report, WSMO, 2005.
- [CLE 04] Clement L., Hatley A., Riegen C. V., Rogers T. (eds.), "UDDI Version 3.0.2", Organization for the Advancement of Structured Information Standards (OASIS), 2004.
- [COR 03] Corcho O., Fernandez-Lopez M., Gomez-Pérez A., "Methodologies, tools and languages for building ontologies : Where is their meeting point ?", Data & Knowledge Engineering, Vol. 46, No. 1, pp. 41-64, 2003.
- [DEN 11] Dengping Wei, Ting Wang, Ji Wang, Abraham Bernstein., "SAWSDL-iMatcher: A customizable and effective Semantic Web Service matchmaker". Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 9, No. 4, pp. 402-417, 2011.
- [DOA 02] Doan A., Madhavan J., Domingos P., Halevy A., "Learning to Map Between Ontologies on the Semantic Web", 11th International WWW Conference (WWW'02), Hawaii, US, 2002.
- [DOG 04] Dogac A., Kabak Y., Laleci G., "Enriching ebxml registries with OWL ontologies for efficient service discovery", 14<sup>th</sup> international Workshop on Research Issues on Data Engineering: Web Services for E-Commerce an E-Government, RIDE-ECEG'04, pp. 69-76, Boston, USA, 2004.
- [DOM 08] Domingue J., Cabral L., Galizia S., Tanasescu V., Gugliotta A., Norton B., Pedrinaci C., "IRS-III: A broker-based approach to semantic Web services", Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 6, No. 2, pp. 109-132, Elsevier, 2008.
- [DOU 05] Dou D., Mcdermott D., Qi P., "Ontology translation by ontology merging and automated reasoning", In Tamma V., Cranefield S., Finin T.W., Willmott S., "Ontologies for Agents: Theory and Experiences", pp.73-94, Springer, 2005.
- [DRA 14] Dragisic Z., Eckert K., Euzenat J., Faria D., et al., "Results of the Ontology Alignment Evaluation Initiative 2014", The 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC-2014), Trentino, Italy, 2014.

- [DUM 06] Dumas M., Spork M., ‘Adapt or perish : Algebra and visual notation for service interface adaptation’, In Dustdar S., Fiadeiro J. L., Sheth A. P. (eds.), ‘Business Process Management’, LNCS 4102, pp. 65-80, Springer, 2006.
- [EIF 04] European Interoperability Framework for pan-European eGovernment Services, Office for official Publications of the European Communities ISBN 92-894-8389-X, Luxembourg, 2004.
- [FAT 15] Fatih K., Lars F., Dogan K., ‘Service composition with consideration of interdependent security objectives’, Science of Computer Programming, Vol. 97, Part 2, pp. 183-201, Elsevier, 2015.
- [FEN 02] Fensel D., Bussler C., ‘The Web Service Modeling Framework WSMF’, Electronic Commerce Research and Applications, Vol.1, No.2, pp. 113-137, Elsevier, 2002.
- [FON 15] Fonou-Dombeu J. V., Huisman M., ‘Engineering Semantic Web Services for Government Business Processes Automation’, In Ko A., Francisconi E., ‘Electronic Government and the Information Systems Perspectives’, LNCS 9265, pp. 40-54, Springer, 2015.
- [GAE 02] Lortal G., ‘Etat de l’art: Ontologies et Intégration/Fusion d’ontologies’, Rapport de stage, Laboratoire Dialogue et Intermédiations Intelligentes de la Direction des Interactions Humaines DIH/D2I au centre de Recherche et Développement de France Télécom (FTR&D), France, 2002.
- [GHA 14] Gharzouli M., Derdour M., ‘To implement an open-MAS architecture for semantic Web services discovery, what kind of P2P protocol do we need?’, International Journal of Agent Technologies and Systems (IJATS), Vol. 6, No. 3, pp. 57-72, IGI Global, 2014.
- [GOH 97] Goh C.H., ‘Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources’, Phd Thesis, MIT, 1997.
- [GOM 00] Gomez-Perez A., ‘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, Réseau international de néologie et de terminologie, Rint, 2000.
- [GRU 93] Gruber T., ‘A Translation Approach to Portable Ontology Specifications’, Knowledge Acquisition, Vol. 5, No. 2, pp. 199-220, Elsevier, 1993.
- [GUA 95] Guarino N., ‘Formal Ontology, Conceptual Analysis and Knowledge Representation’, International journal of Human and Computer Studies, Vol. 43, No. 5/6, pp. 625-640, Elsevier, 1995.

- [GUG 05] Gugliotta A., Gabral L., Domingue J., Roberto V., “A Semantic Web Service-based Architecture for the Interoperability of E-government Services”, Web Information Systems Modeling Workshop (WISM’05), 5th International Conference on Web Engineering (ICWE’05), Sydney, Australia, 2005.
- [HAL 05] Haller A., Cimpian E., Mocan E., Oren E., Bussler C., “Wsmx - a semantic service-oriented architecture”, International Conference on Web Services, Orlando, Florida, USA, 2005.
- [HAU 09] Hausberger T., “Semantic Annotations for WSDL”, Semantic Web Services PS, Slides, Departement of Computer Science Leopold Franzens University Innsbruck, 2009.
- [HOF 00] Hoffner Y., Lwdwig H., Gulcu C., Grefen P., “An architecture for cross-organisational business processes”, Second International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, WECWIS, Milpitas, CA, USA, 2000.
- [HOR 04] Horrocks I., Patl-Schneider P. F., Boley H., Tabet S., Grosopf B., Dean M., “SWRL: A Semantic Web Rule Language Combining OWL and RuleML”, W3C, Technical Report, 2004.
- [IZZ 06] Izza S., “Intégration des systèmes d’information industriels, une approche flexible basée sur les services sémantiques”, Thèse de doctorat, Ecole Nationale Supérieure des Mines, Saint-Etienne, France, 2006.
- [IZZ 08] Izza S., Vincent L., Burlat P., “Exploiting semantic Web services in achieving flexible application integration in the microelectronics field”, Computers in Industry, Vol 59, pp. 722-740, Elsevier, 2008.
- [KIL 10] Kil H., “Efficient Web service composition: from signature-level to behavioral description-level”, Doctoral dissertation, Pennsylvania State University, USA, 2010.
- [KLE 01] Klein M., “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.
- [KLE 05] Klein M., König-Ries B., Müssig M., “What is needed for semantic service descriptions - a proposal for suitable language constructs”, International Journal on Web and Grid Services (IJWGS), Vol. 1, No. 3/4, pp. 328-364, Inderscience, 2005.

- [KLI 04] Klischewski R., "Information integration or process integration: How to achieve interoperability in administration", International Conference on E-Government (EGOV'04), DEXA, Zaragoza, Spain, 2004.
- [KOU 08] Kourtisis D., Paraskakis I., "Combining SAWSDL, OWL-DL and UDDI for Semantically Enhanced Web Service Discovery". In Bechhofer S., Hauswirth M., Hoffmann J., Koubarakis M. (Eds.), ESWC, LNCS 5021, pp. 614-628, Berlin/Heidelberg, Germany, Springer, 2008.
- [MAR 04] Martin D., Burstein M., McDermott D., McIlraith S., Paolucci M., Sycara K., McGuinness D. L., Sirin E, N. Srinivasan N., "Bringing Semantics to Web Services with OWL-S", World Wide Web, Vol.10, pp. 243-277, Springer, 2007.
- [McG 00] McGuinness D.L., Fikes R., Rice J., Wilder S., "The chimaera ontology environment", Association for the Advancement of Artificial Intelligence (AAAI), Austin, Texas, USA, 2000.
- [McG 04] McGuinness D.L., Van Harmelen F., "OWL Web Ontology Language Overview", Technical Report, W3C, 2004.
- [MEC 09] Mecheri K., "Une architecture orientée services pour la mise en oeuvre de mécanismes d'interopérabilité des systèmes d'information d'entreprises", Thèse de Magister, Université Badji Mokhtar, Annaba, Algérie, 2009.
- [MEC 10a] Mecheri K., Boufaïda M., Souici-Meslati L., "A semantic Web Service Oriented Architecture for the Information Systems Interoperability", Proceedings of the third International Conference on Web and Information Technologies (ICWIT'2010), Marrakech, Morocco, pp. 15-26, 2010.
- [MEC 10b] Mecheri K., Souici-Meslati L., "Semantic interoperability of Web services: A survey", First International Conference on Machine and Web Intelligence (IEEE ICMWI'10), Algiers, Algeria, pp. 55-60, 2010.
- [MEC 10c] Mecheri K., Boufaïda M., Souici-Meslati L., "Une architecture orientée services pour l'interopérabilité des SI dans la SETA", Proceedings of the Second Workshop on Web Services (WWS'10), pp. 37- 48, CERIST, Alger, Algérie, 26-27 décembre 2010.
- [MEC 12] Mecheri K., Boufaïda M., Souici-Meslati L., "Data mediation towards semantic Web service interoperability", Int. J. Web Science, Vol. 1, No. 3, pp. 163-176, Inderscience, 2012.
- [MEC 17] Mecheri K., Boufaïda M., Souici-Meslati L., Meslati D., "Etude des mécanismes d'interopérabilité des systèmes d'information basés sur les services Web sémantiques", Revue des Sciences et de la Technologie Synthèse, No. 35, pp. 1-13, 2017.

- [MED 04] Medjahed B., Benatallah B., Bouguettaya A., Elmagarmid A.K., ‘‘Webbis : An infrastructure for agile integration of Web services’’, *Int. J. Cooperative Inf. Syst.*, Vol. 13, No. 2, pp.121-158, World Scientific, 2004.
- [MEL 07] Mellal N., ‘‘Réalisation de l’interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d’information’’, Thèse de Doctorat, Université de Savoie, France, 2007.
- [MID 16] Midouni S. A. D., Amghar Y., Chikh A., ‘‘A semantic annotation for MaaS services: the SA4MaaS approach’’, *Int. J. of Metadata, Semantics and Ontologies*, Vol.11, No.2, pp.82-92, Inderscience, 2016.
- [MIL 04] Miller J., Verma K., Rajasekaran P., Sheth A., Aggarwal R., Sivashanmugam K., ‘‘ WSDL-S : Adding Semantics to WSDL’’, White Paper, Technical report, Large Scale Distributed Information Systems, 2004.
- [MON 04] Monfort V., Goudeau S., ‘‘Web services et interopérabilité des systèmes d’information’’, Dunod, Paris, France, 2004.
- [MRI 07] Mrissa M., ‘‘Médiation Sémantique Orientée Contexte pour la Composition de Services Web’’, Thèse de Doctorat, Université Claude Bernard Lyon I, 2007.
- [NAG 06] Nagarajan M., Verma K., Sheth A. P., Miller J., Lathem J., ‘‘Semantic Interoperability of Web Services – Challenges and Experiences’’, *Proceedings of the IEEE International Conference on Web Services, ICWS'06*, Chicago, USA, 2006.
- [NOY 00] Noy N.F., Musen M. A., ‘‘PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment’’, *Seventeenth National Conference on Artificial Intelligence*, Austin, Texas, USA, pp. 450-455, 2000.
- [NOY 03] Noy N.F., Musen M. A., ‘‘The Prompt suite: Interactive tools for ontology merging and mapping’’, *International Journal of Human Computer Studies*, Vol. 59, No. 6, pp. 983-1024, Elsevier, 2003.
- [ONT 00] 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.
- [OUS 15] Oussalah Y., ‘‘Spécification Formelle des Services Web à Interface Flexible’’, Thèse de Doctorat, Université Constantine 2- Abdelhamid Mehri, Algérie, 2015.
- [OWL 04] OWL Coalition, ‘‘OWL - Web Ontology Language’’, Technical Report OWL guide, W3C, 2004.

- [PAN 05] Panetto H, Chen D. et al., "D6.1 : Practices, principles and patterns for interoperability", Technical Report, INTEROP-NOE, Interoperability Research for Network enterprises Applications and Software Network of Excellence, N° IST 508-011, 2005.
- [PAN 06] Panetto H., "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, France, 2006.
- [PAN 08] Panetto H., Molina A., "Enterprise integration and interoperability in manufacturing systems: Trends and issues", Computers in Industry, Vol. 59, No. 7, pp. 641-646, Elsevier, 2008.
- [PAO 02] Paolucci M., Kawamura T., Payne T., Sycara K., "Importing the Semantic Web in UDDI", International Workshop on Web Services, E-Business, and the Semantic Web (WES'02), Toronto, Canada, 2002.
- [PEE 05] Peer J., "Semantic Service Markup with SESMA", Web Service Semantics Workshop (WSS'05) at the Fourteenth International World Wide Web Conference (WWW'05), Chiba, Japan, 2005.
- [REI 02] Reix R., "Systèmes d'information et management des organisations", 4ème édition, Vuibert, 2002.
- [REM 06] Rempacek S., "Sémantique, interactions et langages de description des services Web complexes", Thèse de Doctorat, Université de Reims Champagne-Ardenne, France, 2006.
- [SCH 04] Schreiber G., Dean M., "OWL Web ontology language reference", Technical Report, W3C, 2004.
- [SHE 08] Sheth P.A., Gomadam K., Ranabahu A., "Semantics Enhanced Services: METEOR-S, SAWSDL and SA-REST", Data Engineering Bulletin Issues, Vol. 31, pp. 8-12, IEEE Technical Committee, 2008.
- [SHE 14] Sheng Q. Z., Qiao X., Vasilakos A. V., Szabo C., Bourne S., Xu X., "Web services composition: A decade's overview", Information Sciences, Vol. 280, pp. 218 -238, Elsevier, 2014.
- [Sik 15] Sikos L. F., "Mastering Structured Data on the Semantic Web", 1<sup>st</sup> edition, Apress, 2015.
- [STU 01] Stumme G., Maedche A., "FCA-Merge: Bottom-up merging of ontologies", 7<sup>th</sup> International Conference on Artificial Intelligence, IJCAI'01, pp. 225-230, Seattle, USA, 2001.

- [SWA 97] Swartout B., Patil R., Knight K., Russ T., "Use of Large-Scale Ontologies", Spring Symposium Series on Ontological Engineering, Stanford, USA, pp. 138-148, 1997.
- [TAH 06] Taher Y., Benslimane D., Fauvet M.-C., Maamar Z., "Towards an approach for Web services substitution", In Ghodous P., Dieng-Kuntz R., Loureiro G. (eds), IDEAS, pp. 166-173, IOS Press, 2006.
- [USC 02] Uschold M., Gruninger M., "Creating Semantically Integrated Communities on the World Wide Web", Semantic Web Workshop, Honolulu, Hawaii, 2002.
- [VIT 09] Vitvar T., Zaremba M., Mocan M., Mocan A., "Mediation using WSMO, WSML and WSMX", In Petrie C. et al. (Eds.), "Semantic Web Services Challenge Results from the first year", pp. 31-49, Springer, 2009.
- [WAC 01] Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., Hübner S., "Ontology-based Integration of Information - A Survey of Existing Approaches", Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing, pp. 108-117, Seattle, USA, 2001.
- [WEI 02] Weigand H., Van Den Heuvel W. J., "Cross-organizational workflow integration using contracts", Decision Support Systems, Vol. 33, No. 3, pp. 247-265, Elsevier, 2002.
- [WOO 75] Woods W.A., "What's in a link: foundations of semantic networks", National Technical Information Service (NTIS), USA, 1975.
- [ZOU 06] Zouggar N., Vallespir B., Chen D., "Enrichissement de la Modelisation d'Entreprise par les Ontologies", 6<sup>ème</sup> Conférence Francophone de MODélisation et SIMulation, MOSIM'06, Rabat, Maroc, 2006.

## Webographie

- [1] Technologies des Web services  
[http://www.softeam.fr/technologies\\_web\\_services.php](http://www.softeam.fr/technologies_web_services.php). (Accessed in April 2017)
- [2] W3C Recommendation: Extensible Markup Language (XML) 1.0 (Fifth Edition), 2008, <http://www.w3.org/TR/2008/REC-xml-20081126/>. (Accessed in July 2017)
- [3] Glossary, Business Process Trends BPT, 2013,  
[http://www.bptrends.com/resources\\_glossar.cfm](http://www.bptrends.com/resources_glossar.cfm). (Accessed in August 2017)
- [4] IDEAS Project Deliverables (Wp1-Wp7), Public reports, 2003. [www.ideas-roadmap.net](http://www.ideas-roadmap.net). (Accessed in August 2017)
- [5] RDF Primer, W3C Working Group, 2014, <http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/>. (Accessed in April 2017)
- [6] INTEROP, “State of the art and state of the practice including initial possible research orientations”, 2004, Deliverable D8.1, <http://www.interop-noe.org/>. (Accessed in 2016).
- [7] <http://protege.stanford.edu/>. (Accessed in April 2017)
- [8] Pellet, <https://github.com/stardog-union/pellet>. (Accessed in April 2017)
- [9] JESS, <http://herzberg.ca.sandia.gov/jess/>. (Accessed in April 2017)
- [10] Description Logics, <http://dl.kr.org/>. (Accessed in April 2017)
- [11] Racer Semantic Web Inference Engine website,  
<http://www.sts.tuharburg.de/~r.f.moeller/racer/>. (Accessed in April 2017)
- [12] KnowledgeWeb, Deliverables of KWEB Project, 2004,  
<http://knowledgeWeb.semanticWeb.org/>. (Accessed in April 2017)
- [13] Sekt, “Deliverables of SEKT Project: D4.2.2 State-of-the-art survey on Ontology Merging and Aligning V2”, 2006, <http://www.sekt-project.com/>. (Accessed in 2017).
- [14] DRAGO, <http://drago.fbk.eu/system-description.html>
- [15] ServOMap, <https://code.google.com/archive/p/servo/downloads>
- [16] 50 Ontology Mapping and Alignment Tools,  
<http://www.mkbergman.com/1769/50-ontology-mapping-and-alignment-tools/>
- [17] Web Service Modeling eXecution environment, DERI and STI2, 2008,  
[www.wsmx.org](http://www.wsmx.org). (Accessed in April 2017)
- [18] NetBeans, [https://netbeans.org/index\\_fr.html](https://netbeans.org/index_fr.html). (Accessed in April 2017)
- [19] Glass-fish, <http://glass-fish.joydownload.com/>. (Accessed in April 2017)
- [20] EJB, <http://www.jmdoudoux.fr/java/dej/chap-ejb3.htm>. (Accessed in April 2017)
- [21] PHP tutorials, <http://fr.html.net/tutorials/php/lesson1.php>. (Accessed in April 2017)
- [22] Wamserver, <http://www.wampserver.com/>. (Accessed in April 2017)
- [23] La bibliothèque Jena2, <http://jena.sourceforge.net/>. (Accessed in April 2017)

# A PROPOS DE L'AUTEUR

## BIOGRAPHIE DE L'AUTEUR

---



Karima MECHERI, est née en 1968 à Annaba (nord-est algérien), où elle a poursuivi ses études jusqu'à l'obtention du baccalauréat (option mathématique) en 1987. Elle rejoint l'université d'Annaba la même année pour suivre une formation d'ingénieur d'état en Informatique. En 1992, elle obtient son ingéniorat, option Systèmes d'information. De 1996 à 2010, elle est professeur spécialisé dans l'enseignement professionnel grade II (PSEPII) au sein de l'INSFP Didouche Mourad d'Annaba. Pendant cette période, elle encadre des dizaines de projets de fin d'études et enseigne aux Techniciens Supérieurs en informatique dans les options : Systèmes d'information et Bases de données.

A partir de 2006, elle accède aux études de post-graduation pour préparer un mémoire de Magister en informatique à travers lequel elle propose une architecture orientée services pour la mise en œuvre de mécanismes d'interopérabilité des systèmes d'information d'entreprises. En 2009, elle soutient son Magister et devient Maître-assistant B en 2010, au département d'informatique de l'université Badji-Mokhtar d'Annaba, puis Maître-Assistant A à partir de 2013. Elle a encadré et dirigé plusieurs projets de fin d'études de Licence en Informatique et de Master option : Ingénierie des Logiciels Complexes.

Karima MECHERI est actuellement membre du laboratoire LISCO. Dans le cadre de ses activités de recherche scientifique, elle a collaboré, en qualité de membre, à deux projets de recherche CNEPRU: Modélisation et Evolution des Systèmes Complexes (MESYCO-1 et MESYCO-2) et a participé à des conférences internationales spécialisées dans le domaine de la coopération des systèmes d'information, des architectures orientées services et des services Web sémantiques.

## 1. Publications dans des journaux internationaux

---

1. Mecheri K., Boufaïda M., Souici-Meslati L., "Data mediation towards semantic Web service interoperability", **Int. J. Web Science (IJWS), Vol. 1, No. 3, pp. 163-176, 2012.**

### Informations sur le journal IJWS

**ISSN: 1757-8795.**

**Fréquence de publication:** 4 numéros par an.

**Processus de reviewing:** double-blind review.

**Publié par:** Inderscience publishers

**Site Web:** <http://www.inderscience.com/jhome.php?jcode=IJWS>

**Indexé par :** Academic OneFile (Gale), cnpLINKer (CNPIEC), DBLP Computer Science Bibliography, Google Scholar, Info Trac (Gale), Inspec (Institution of Engineering and Technology), ProQuest Advanced Technologies Database with Aerospace.

2. Mecheri K., Boufaïda M., Souici-Meslati L., Meslati D., "Etude des mécanismes d'interopérabilité des systèmes d'information basés sur les services Web sémantiques", **Revue des Sciences et de la Technologie Synthèse, No. 35, pp. 1-13, 2017.**

### Informations sur la revue Synthèse

**ISSN:** 1111 – 4924.

**Fréquence de publication:** 2 numéros par an.

**Processus de reviewing:** peer reviewed.

**Publié par:** Université Badji Mokhtar - Annaba

**Site Web:** <http://dpubma.univ-annaba.dz/>

**Indexé par :** African Journals On Line (AJOL), Research Bible, Academic Keys, Directory of Research Journals Indexing, Cite Factor, Advanced Science Index, International Impact Factor Services, Scientific Indexing Services, The Journals Impact Factor.

3. Mecheri K., Souici-Meslati L., Boufaïda M., Meslati D., "Context Based Mediation for Semantic Web Services Interoperability". (Article soumis pour publication dans un journal international spécialisé, **Date de soumission : 13/11/2015**).
4. Mecheri K., Boufaïda M., Meslati D., Souici-Meslati L., "Context based interoperability of Semantic Web Services". (Article soumis pour publication dans un journal international spécialisé. **Date de soumission : 17/05/2017**).

## **2. Communications avec comité de lecture international**

---

1. Mecheri K., Boufaida M., Souici-Meslati L., "A semantic Web Service Oriented Architecture for the Information Systems Interoperability", Proceedings of the third International Conference on Web and Information Technologies (**ICWIT'2010**), pp. 15-26, Marrakech, Morocco, 16-19 June 2010.
2. Mecheri K., Souici-Meslati L., "Semantic interoperability of Web services: A survey", Proceedings of the First International Conference on Machine and Web Intelligence (**IEEE ICMWI'10**), pp. 55-60, Algiers, Algeria, 3-4 October 2010.
3. Mecheri K., Boufaida M., Souici-Meslati L., "Une architecture orientée services pour l'interopérabilité des SI dans la SETA", Proceedings of the Second Workshop on Web Services (**WWS'10**), pp. 37- 48, CERIST, Alger, Algérie, 26-27 décembre 2010.