

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mohamed Khider – BISKRA  
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie  
Département d'informatique

N° d'ordre :.....

Série :.....



## Thèse

En vue de l'obtention du diplôme de Doctorat En Sciences

*Spécialité : Informatique*

*Option : Informatique*

Titre :

---

# Une approche sémantique pour l'interopérabilité des ERPs

---

Présentée et soutenue par :

**Merouane ZOUBEÏDI**

le : 05/11/2018

Devant le Jury composé de :

<b>Président :</b>	Allaoua CHAOUI	PROF,	Université de Constantine 2
<b>Rapporteur :</b>	Okba KAZAR	PROF,	Université de Biskra
<b>Examineurs :</b>	Mohamed BENMOHAMMED	PROF,	Université de Constantine 2
	Hammadi BENNOUI	PROF,	Université de Biskra
	Khaled REZEG	MCA,	Université de Biskra
	Brahim LEJDEL	MCA,	Université d'El Oued

## Dédicaces

*A mes défunts parents,  
que Dieu leur accorde sa miséricorde et  
leur ouvre la porte du paradis.*

## Remerciements

Tout d'abord, je remercie Dieu Tout Puissant de m'avoir donné la force et la patience nécessaire pour achever ce travail de thèse.

Mes remerciements les plus chaleureux sont destinés à ceux qui de près ou de loin m'ont soutenu et ont contribué à l'accomplissement de ce modeste travail. Je leur demande de trouver ici, l'expression de ma profonde reconnaissance. Je m'adresse plus particulièrement :

A M. Okba KAZAR, Professeur à l'Université de Biskra et Directeur du laboratoire LINFI, pour avoir guidé et dirigé avec brio cette thèse, pour son appui, ses conseils et sa constante disponibilité ;

A M. Allaoua CHAOUI, Professeur à l'Université de Constantine 2, qui m'a fait l'honneur de présider le jury et que je remercie très sincèrement ;

A MM. Mohamed BENMOHAMMED, Professeur à l'Université de Constantine 2, Hammadi BENNOUI, Professeur à l'Université de Biskra, Rezeg KHALED, Maître de Conférences A à l'Université de Biskra et Brahim LEJDEL, Maître de Conférences A à l'Université d'El Oued, qui ont daigné siéger dans ce jury. Je les remercie pour accepter de juger ce travail ;

A M. Saber BENHARZALLAH, Maître de Conférences A à l'Université de Batna 2 que je suis très heureux de remercier pour m'avoir fait bénéficier de ses commentaires, ses suggestions et sa précieuse documentation ;

A ma famille et plus particulièrement ma femme, mes enfants, mes sœurs et mes frères, un grand merci pour leur soutien permanent et pour les intérêts qu'ils ont portés pour mes études, ainsi que mes chers frères et amis MM. L. Mammeri, H. Fettous, I. Leghrissi, le Dr N. Mesbahi, le Dr A. Merizig, A. Alloui, D. Rezki et le Pr. Hasseine.

**Merouane ZOUBEIDI**

## ملخص

أصبحت اليوم، نظم تخطيط موارد المؤسسات ERP تعمل في بيئة ديناميكية عالية جدًا. لتزدهر في هذه البيئة، تواجه ERP حاجة متزايدة لتقاسم المعلومات والتعاون. علاوة على ذلك، يتميز نظم تخطيط موارد المؤسسات (ERP) بشكل عام بالاستقلالية والتوزيع وعدم التجانس. في مثل هذا السياق، يعتبر التشغيل المتبادل أحد الاهتمامات الرئيسية. كيف يمكننا التأكد من أن أنظمة المعلومات المختلفة القائمة على نظم تخطيط موارد المؤسسات يمكن أن تعمل معاً نحو نهاية مرغوبة للطرفين. في الواقع، تحدث مشاكل التشغيل البيئي على مستويات مختلفة (النحوية، الدلالي، إلخ). يوفر الوكلاء وخدمات الويب والأنطولوجيات حلاً تكميليًا جديدًا لمشاكل التشغيل البيئي. تقدم هذه الأطروحة إطار العمل تصميم وهندسة تعتمد على هذه التقنيات من أجل التشغيل البيئي الدلالي لنظم تخطيط موارد المؤسسات ERP.

**الكلمات المفتاحية:** نظم تخطيط موارد المؤسسات (ERP) ، الأنظمة المتعدد الأعوان (SMA) ، الهندسة الموجهة للخدمات (SOA) ، الأنطولوجيات ، التشغيل المتبادل الدلالي

## Résumé

Actuellement, les systèmes d'information de type ERPs opèrent dans un environnement dynamique très élevé. Pour prospérer dans cet environnement, les ERPs font face à un besoin croissant de partager des informations et à collaborer. Par ailleurs, ces ERP en général, sont caractérisées par l'autonomie, la distribution et l'hétérogénéité. Dans un tel contexte, l'interopérabilité est une préoccupation essentielle. Comment pouvons-nous assurer que divers systèmes d'information à base d'ERPs peuvent travailler ensemble vers une fin mutuellement souhaitable. En effet, les problèmes de l'interopérabilité se produisent à différents niveaux (syntaxique, sémantique, etc.). Les agents, les services web et les ontologies fournissent une nouvelle solution de complémentarité aux problèmes de l'interopérabilité. Cette thèse présente un cadre de travail conceptuel, architectural et implémentation basé sur ces technologies pour l'interopérabilité sémantique des ERPs.

**MOTS-CLES :** Entreprise Resource Planning (ERP), Systèmes Multi-Agents (SMA), Architectures Orientées Services, Ontologie, Interopérabilité sémantique.

## Abstract

Currently, Information Systems of ERP type operate in a high dynamic environment. To thrive in this environment, ERP face a growing need to share information and collaborate. In addition, these ERP in general, characterized by autonomy, distribution and heterogeneity. In this context, interoperability is a key concern. How can we ensure that different ERP systems can work together toward a mutually desirable end? Indeed, interoperability problems occur at different levels (syntactic, semantic, etc.). Agents, Service Oriented Architecture (SOA) and ontologies provide a new complementary solution to problems of interoperability. This thesis presents a conceptual and architectural framework based on these technologies for interoperability semantic of ERP.

**Keywords:** Enterprise Resource Planning (ERP), Multi-agent system, Web services, Ontology, Semantic Interoperability.

---

---

## Liste des publications

La liste suivante représente les articles publiés dans le cadre de cette thèse.

### A.1 Revues Internationales

- Merouane Zoubeidi, Okba Kazar, Saber Benharzallah, Nadjib Mesbahi. Une approche d'interopérabilité sémantique via les services web dans les progiciels de gestion intégrée. *Revue courrier du savoir*, Volume 26, Mars 2018, Pages 381-392.
- Nadjib Mesbahi, Okba Kazar, Saber Benharzallah, Merouane Zoubeidi. A Cooperative Multi-Agent Approach-Based Clustering in Enterprise Resource Planning. *International Journal of Knowledge and Systems Science*, Volume 6 Issue 1, January 2015, Pages 34-45, IGI Publishing Hershey, PA, USA.
- Nadjib Mesbahi, Okba Kazar, Saber Benharzallah, Merouane Zoubeidi, Samir Bourekkache. Multi-Agents approach for data mining based k-Means for improving the decision process in the ERP systems. *International Journal of Decision Support System Technology*. Volume 7 Issue 2, April 2015, Pages 1-14 - IGI Publishing Hershey, PA, USA.

### A.2 Conférences Internationales

- Merouane ZOUBEIDI, Okba KAZAR, Saber BENHARZALLAH and Nadjib MESBAHI. ISPGI : Une architecture sémantique à base d'agents pour l'interopérabilité d'un PGI. *Proceeding of the International Conference on Artificial Intelligence and Information Technology (ICA2IT 2014)*. Ouargla, Algérie.
- Merouane ZOUBEIDI, Okba KAZAR, Nadjib MESBAHI ET Saber BENHARZALLAH. Vers une architecture d'intégration de la sémantique via les Services Web : Etude de cas du Progiciel de Gestion Intégré. *La conférence francophone sur les Systèmes Collaboratifs (SysCo'2014)*. Hammamet, Tunisie, pp. 101-114.
- Merouane ZOUBEIDI, Okba KAZAR, Saber BENHARZALLAH and Nadjib MESBAHI. Toward an architecture for the Semantic Integration in an Enterprise Resource Planning. *Proceeding of the 3rd IEEE International Workshop on Advanced Information Systems for Enterprises (IWAISE'14)*. Tunis, Tunisia. <http://www.lifl.fr/iwaise14/program.html>.
- Nadjib MESBAHI, Okba KAZAR, Merouane ZOUBEIDI and Saber BENHARZALLAH. An agent-based modeling for an enterprise resource planning (ERP), *ACIIDS 2014*, Bangkok, Thailand LNCS Volume 551, 2014, pp. 225-234, Springer.

- Nadjib MESBAHI, Okba KAZAR, Saber BENHARZALLAH and Merouane ZOUBEIDI. A cooperative multi-agent approach for knowledge discovery from the enterprise resource planning. Proceeding of the 3rd IEEE International Workshop on Advanced Information Systems for Enterprises (IWAISE 2014). Tunis, Tunisia. [Http://www.lifl.fr/iwaise14/program.html](http://www.lifl.fr/iwaise14/program.html).
- Nadjib MESBAHI, Okba KAZAR, Saber BENHARZALLAH et Merouane ZOUBEIDI. Approche Multi-Niveaux à base d'agents pour la modélisation d'un progiciel de gestion intégré : Etude du cas de l'Entreprise National de Services aux Puits. Proceeding of the International Conference on Artificial Intelligence and Information Technology (ICA2IT 2014). Ouargla, Algérie.

### **A.3 Chapitre du livre**

- Nadjib MESBAHI, Okba KAZAR, Saber BENHRZALLAH, Merouane ZOUBEIDI, Djamil REZKI. A clustering approach based on cooperative agents to improve decision support in ERP. Technological Innovations in Knowledge Management and Decision Support. 2018, IGI Publishing Hershey, PA, USA. pp. 01-18.



## Liste des figures

Figure I-1 : Architecture technique de l'ERP .....	8
Figure I-2 : Architecture générale d'un ERP .....	9
Figure I-3 : Architecture modulaire ERP .....	10
Figure I-4 : Marché mondial des ERP .....	14
Figure I-5 : Processus d'implantation d'un ERP .....	17
Figure II-1: Pile des langages de représentation des ontologies .....	36
Figure II-2 : Structure d'un message FIPA-ACL.....	46
Figure II-3 : Architecture des services web .....	50
Figure III-1: Modélisation à l'ERP basée sur les agents et ontologies .....	79
Figure III-2: Architecture Fonctionnelle : Agent interface utilisateur .....	82
Figure III-3:Architecture Fonctionnelle : Agent domaine Achat/Appros .....	83
Figure III-4:Architecture Fonctionnelle : Agent Achat.....	86
Figure IV-1:Architecture d'interopérabilité sémantique.....	89
Figure IV-2:La structure interne de l'agent de mise à jour et exploitation .....	90
Figure IV-3:La structure interne de l'agent de correspondance et de mapping.....	91
Figure IV-4:Exemple d'un objet sémantique et son contexte .....	93
Figure IV-5 : Syntaxe d'un message FIPA-ACL .....	94
Figure IV-6: Articulation de système d'information du Groupe ENSP (Scénario) .....	95
Figure IV-7: Diagramme de séquence AUML.....	96
Figure IV-8: Processus d'interopérabilité sémantique .....	97
Figure IV-9:Extrait de l'ontologie de l'ERP1 & l'ERP2 (les concepts à l'intérieur en zone grisées représentant le contexte du concept .....	98
Figure IV-10:Format OWL de l'ontologie de médiation .....	100
Figure IV-11:Extrait de l'Ontologie de médiation.....	101
Figure IV-12:Annotation des descriptions relatives aux services web de l'exemple.....	102
Figure V-1:Scénario d'interopérabilité. ....	106
Figure V-2:Création des agents du système .....	110
Figure V-3:Simulation du protocole d'interaction entre les agents entre agents .....	110

---

---

## Liste des tableaux

Tableau I-1: Comparaison entre ERP1 et ERP2 .....	27
Tableau II-1: Typologie des conflits sémantiques .....	31
Tableau II-2: Comparaison des méthodologies de construction d'ontologies .....	36
Tableau II-3: Evaluation des approches ontologiques .....	40
Tableau II-4: Actes de communication FIPA-ACL, regroupés par catégories .....	46
Tableau II-5: Comparaison entre les services web et les SMAs .....	53
Tableau III-1: Comparaison de solutions d'interopérabilité .....	71
Tableau III-2: Tableau comparatif des travaux existants basés Service Web .....	76
Tableau IV-1: Hétérogénéités sémantiques des données des deux services Web exemples... ..	105

# Table des matières

<b>Dédicaces</b> .....	<b>II</b>
<b>Remerciements</b> .....	<b>III</b>
<b>ملخص</b> .....	<b>IV</b>
<b>Résumé</b> .....	<b>V</b>
<b>Abstract</b> .....	<b>VI</b>
<b>Liste des publications</b> .....	<b>VII</b>
<b>Liste des figures</b> .....	<b>IX</b>
<b>Liste des tableaux</b> .....	<b>X</b>
<b>Introduction Générale</b> .....	<b>1</b>
<b>Chapitre I : Coopération des systèmes d'information à base d'ERP</b> .....	<b>5</b>
I.1. Introduction.....	5
I.2. Historique de l'ERP .....	5
I.3. Définitions de l'ERP.....	6
I.4. Des caractéristiques d'un ERP .....	7
I.5. Les architectures d'un ERP .....	7
I.5.1. Architecture technique d'un ERP .....	7
I.5.2. Architecture modulaire d'un ERP .....	8
I.6. Avantages d'un ERP .....	10
I.7. Intérêts de l'ERP dans l'entreprise .....	11
I.8. Inconvénients des ERP .....	12
I.9. Topologies des ERP .....	12
I.9.1. ERP généralistes.....	12
I.9.2. Les ERP spécialisés (ERP métiers) .....	12
I.9.3. Les ERPs verticaux .....	13
I.10. Marché des ERP.....	13
I.10.1. Les ERP propriétaires.....	14
I.10.2. Les ERP open source.....	15
I.11. Implantation de l'ERP.....	15
I.11.1. Processus d'implantation de l'ERP.....	15
I.11.2. Facteurs clés de succès de l'implantation de l'ERP .....	17
I.12. Impact de l'implantation de l'ERP sur la performance dans l'entreprise .....	19
I.13. Méthodes d'intégration des ERPs .....	21
I.14. Implantation multi sites de l'ERP .....	22

I.15. ERP étendu vers une nouvelle génération .....	23
I.16. Coopération et interopérabilité des systèmes d'information de type ERP.....	27
I.17. Interopérabilité et intégration.....	28
I.18. Conclusion .....	29
<b>Chapitre II : Des outils conceptuels et technologiques pour la mise en œuvre des mécanismes de médiation sémantique.....</b>	<b>30</b>
II.1. Avant-propos .....	30
II.2. La notion de sémantique.....	30
II.2.1. Définition .....	30
II.2.2. De la typologie des hétérogénéités sémantiques.....	30
II.2.3. De la classification de la sémantique .....	31
II.2.4. De la représentation de la sémantique .....	32
II.3. Ingénierie Ontologique .....	32
II.3.1. Définition .....	32
II.3.2. Rôle et apport des ontologies .....	33
II.3.3. Méthodologies de construction d'ontologies .....	35
II.3.4. Langages de représentation et outils de développement des ontologies.....	36
II.3.5. Approches d'utilisation des ontologies : .....	39
II.3.6. Approches de mapping des ontologies .....	41
II.4. De l'apport des SMA pour la médiation sémantique .....	41
II.4.1. De la généralité sur les SMA .....	41
II.4.2. Différents Types d'agent.....	42
II.4.3. Communication entre agents.....	44
II.4.4. De l'interopérabilité des SMA .....	46
II.5. De l'apport de services web dans la médiation sémantique .....	47
II.5.1. Des éléments de définition.....	48
II.5.2. De l'architecture orientée service .....	49
II.5.3. De l'architecture de Service Web .....	49
II.5.4. Technologies des Web Services.....	50
II.5.5. De l'avantage des services Web .....	51
II.5.6. Les services Web et les SMA .....	52
II.5.7. De description sémantique de Services Web .....	53
II.5.8. Des médiations des Services Web .....	56
II.5.9. De l'interopérabilité et de l'intégration des Services web .....	57
II.6. Conclusion .....	57
<b>Chapitre III : Etat de l'art sur les approches d'interopérabilité sémantiques.....</b>	<b>59</b>
III.1. Introduction .....	59

III.2. Définition des critères de comparaisons.....	59
III.3 de l'Approche fédérée .....	59
III.3.1 Des composants d'un système fédéré .....	60
III.3.2. De l'Architecture .....	61
III.3.3. Des Types de fédération .....	62
III.4. Des Approches de type médiation.....	63
III.4.1. De la Caractéristique .....	63
III.4.2. Du Type de médiation .....	64
III.4.3. Des Prototypes réalisés dans l'approche de médiation.....	64
III.5. Comparaison entre les solutions d'interopérabilité :.....	67
III.5.1. Le Modèle d'interrogation et de représentation des données .....	67
III.5.2. De la Représentation de la sémantique.....	68
III.5.3. De la Transparence .....	69
III.5.4. De l'Extensibilité :.....	70
III.5.5. De l'Architecture .....	70
III.6. De la Classification des Approches d'interopérabilité par rapport aux techniques et outils utilisés.....	72
III.7. Modélisation de l'ERP par les SMA et ontologies .....	77
III.7.1 Présentation générale de l'architecture.....	78
III.7.2 Agents et leurs rôles .....	79
III.7.3. Architecture fonctionnelle des agents du système.....	82
<b>Chapitre IV : Une approche sémantique pour l'interopérabilité des ERPs .....</b>	<b>87</b>
IV.1. Introduction.....	87
IV.2. Motivations et objectifs de l'approche proposée .....	87
IV.3. Architecture multicouche de l'approche proposée .....	88
IV.3.1. Couche ERP .....	88
IV.3.2 Couche de l'interopérabilité .....	88
IV.4. Architecture fonctionnelle des agents :.....	90
IV.4.1. Agent de mise à jour et exploitation de l'ontologie locale de l'ERP .....	90
IV.4.2. Agent de correspondance et de mapping (Agent de routage) .....	91
IV.4.3 Construction du Modèle Sémantique des Données .....	92
IV.4.4. Mécanisme de coopération dans l'approche.....	93
IV.4.5. Communication des agents dans l'approche .....	94
IV.4.6. Scénario de fonctionnement des agents du système.....	94
IV.5. Processus d'interopérabilité sémantique .....	97
IV.5.1. Récupération des ontologies locales de domaines fonctionnels.....	97
IV.5.2. Construction de l'ontologie locale de l'ERP .....	98

---

---

IV.5.3. Construction de l'ontologie globale du domaine relatif aux ERPs et celle des aspects conflictuels.....	98
IV.5.4. Annotation de services Web :.....	101
IV.5.5. Service Web Médiateur .....	102
IV.6. Conclusion .....	103
<b>Chapitre V : Etude de cas et implémentation.....</b>	<b>104</b>
V.1. Introduction .....	104
V.2. Etude de cas.....	104
V.2.1. Problématique .....	105
V.2.2. Construction du modèle sémantique.....	106
V.2.3. Description de la médiation sémantique de données .....	107
V.3. Outils et aspects techniques de l'implémentation .....	107
V.3.1. Développement des agents .....	107
V.3.2. Simulation de l'interaction entre les agents.....	110
V.3.3. Développement lié à la médiation sémantique .....	111
V.4. Conclusion.....	112
<b>Conclusion générale .....</b>	<b>114</b>
<b>Bibliographie.....</b>	<b>116</b>

# **Introduction générale**

## **Contexte et problématique de recherche**

Depuis le début de 21 ème siècle, le monde économique a connu des bouleversements profonds dans tous les domaines, mais particulièrement dans celui des systèmes d'information. En effet, l'évolution des technologies de l'information, la prolifération de l'internet et la mondialisation de l'économie ont forcé les entreprises à faire face à de multiples problèmes tels que des marchés saturés, une compétitivité accrue, des clients plus exigeants et moins fidèles... Pour surmonter tous ces aléas, les entreprises doivent s'adapter, autrement dit intégrer des technologies nouvelles dans les systèmes de gestion des ressources humaines, de la comptabilité et des finances, de l'aide à la prise de décision mais aussi de la production qualitative, de la distribution, de l'approvisionnement et du commerce électronique.

C'est pourquoi les entreprises cherchent à développer leurs capacités de communication interne et externe pour mettre à la disposition des différents acteurs économiques les moyens d'agir et/ou de réagir en temps réel. L'entreprise ou la multinationale ou même la PME doit être capable de communiquer même avec des entités géographiquement dispersées, qu'il s'agisse des établissements propres, des clients ou des fournisseurs, tout en raccourcissant le cycle des processus, autrement dit allier réaction rapide, souplesse et flexibilité. Ceci impose une disponibilité de l'information en temps réel, une capacité de réaction à partir de tout poste de l'organisation.

Donc, l'accès des entreprises aux nouvelles technologies tend à modifier la communication entre les différents acteurs du monde des affaires, notamment entre l'entreprise et ses clients (Business To Consumer, B2C). Le fonctionnement interne de l'entreprise (Business To Employees, B2E) et la relation de l'entreprise avec ses différents partenaires et fournisseurs (Business To Business, B2B). On appelle aussi "e-Business" l'intégration au sein de l'entreprise d'outils basés sur les technologies de l'information et la communication, en l'occurrence les Progiciels de Gestion Intégrée (ERP). Ce progiciel permet de gérer l'ensemble des processus d'une entreprise en intégrant l'ensemble des fonctions de cette dernière comme nous l'avons souligné précédemment. Ces modules accèdent à des ressources communes, en particulier une base de données unique [1].

Dans cette optique et aussi afin d'éviter faillite et/ou disparition, les entreprises adhèrent à des groupes ou se fusionnent sans pour autant changer leur situation géographique, ni leurs infrastructures et ni leur ERP pour protéger et préserver leurs intérêts communs et ainsi faire face aux nouvelles contraintes imposées par le marché. Cette réorganisation structurelle a un impact direct sur les ERP. L'obligation est d'effectuer des développements applicatifs complémentaires afin de pouvoir intégrer ces solutions à des applications tierces, répondant à des besoins plus spécifiques des organisations et non couverts par les solutions de gestion intégrée en place. Pour que ces entreprises puissent communiquer et coopérer de manière efficace, fiable et performante, il est utile de rendre ces ERP interopérables [2] en premier lieu entre eux-mêmes, puis avec des solutions applicatives périphériques (module de gestion non

couvert par un ERP en place) offrant ainsi plus de fluidité pour l'information et enfin avec les autres solutions du marché à savoir les CRM (Customer Relationship Management) et les SCM (Supply Chain Management). L'interdépendance s'accroît et entraîne des besoins forts d'intégration en assurant un partage et échange d'informations et de services dans un environnement ouvert et dynamique [3]. Donc, il s'agit d'une interopérabilité intra-entreprise et interentreprises.

Dans ce cadre, le traitement de l'interopérabilité est une tâche coûteuse parce qu'il doit considérer l'hétérogénéité des données, de services (application et fonction) et processus. L'hétérogénéité peut apparaître dans l'utilisation de différents langages de programmation, le fonctionnement sur des plateformes et des systèmes d'exploitation variés. Ainsi, l'évolution des standards dans ce domaine permet de résoudre beaucoup de problèmes. L'hétérogénéité apparaîtra ensuite au niveau du modèle utilisé pour représenter et gérer les informations ; ce qui implique des conflits de représentation. Cependant, ces progiciels ont été développés indépendamment et ne partagent pas ainsi la même sémantique pour la terminologie de leurs modèles d'applications. De cela, l'hétérogénéité sémantique se produit parce qu'il y a un désaccord du sens. Certains types de conflits sémantiques apparaissent : les conflits de nom (problèmes taxonomiques et linguistiques), les conflits de valeurs (problèmes d'unités, d'échelles) et les conflits cognitifs (problèmes de signification). Plusieurs solutions existantes ont mis l'accent sur les aspects techniques et syntaxiques. Néanmoins, elles ont évolué vers une prise en compte de plus en plus importante de la sémantique. Elles offrent des avantages au détriment d'autres puisque chacune d'elles ne traite qu'une partie des conflits sémantiques et ne propose pas de combinaison adaptée de technologie (agent, ontologies, etc.) permettant un partage réel, efficace et évolutif d'informations et de services [2].

## **Contributions et objectifs du travail**

Comme vous le constaterez tout au long du développement de la thèse, nous avons insisté sur l'utilisation des systèmes d'information interopérables à base des ERP. Il est utile de souligner que nous nous sommes intéressés dans le cadre de cette thèse aux points suivants :

- ✓ La modélisation de chaque ERP participant par un ensemble de différents agents qui disposent des ontologies en tant que cadres conceptuels pour mettre en place des services Web sophistiqués. N'oubliant pas que la particularité des services Web réside dans le fait qu'il utilise la technologie Internet comme infrastructure pour la communication entre les composants logiciels et ceci en mettant en place un cadre de travail basé sur un ensemble de standards.
- ✓ La proposition d'une architecture sémantique pour l'interopérabilité des ERP basée sur la notion d'ontologies et d'agents pour traiter les problèmes inhérents à l'hétérogénéité des données et les fonctionnalités échangées entre les services Web. Par voie de conséquence, il est impératif de capturer la sémantique liée aux informations échangées, et de résoudre les conflits causés par les hétérogénéités sémantiques.
- ✓ L'intégration de cette architecture d'une approche globale d'interopérabilité par l'annotation des descriptions des services web par un modèle sémantique en utilisant le « Semantic Annotation of Web Service Description Language », un standard récent qui fait



partie des recommandations du consortium W3C. Il s'agit en particulier de l'annotation des données échangées entre services WEB. Cette approche a pour objectif essentiel la réalisation d'une médiation sémantique ayant les caractéristiques suivantes :

- ✓ Permettre aux différents systèmes participant à la coopération de ne pas modifier leurs fonctionnalités habituelles en respectant l'autonomie de l'ERP ;
- ✓ Permettre à un utilisateur, client et fournisseur d'accéder de manière transparente à des ressources fournies par les ERPs ;
- ✓ Construire un système de médiation sémantique flexible, permettant l'ajout de fonctionnalités et de services ;
- ✓ Résoudre la majorité des conflits sémantiques par le recours aux nouvelles technologies, à savoir, les ontologies ;
- ✓ Permettre à un nouveau module fonctionnel -non couvert par l'ERP- ou bien un autre ERP qui fournit son contexte, de trouver l'ERP, les données et les services partagés par le module fonctionnel, de les intégrer dynamiquement à l'ERP et de les utiliser de manière transparente.

## **Organisation du document**

Le travail présenté dans ce manuscrit est divisé en cinq chapitres. Le premier chapitre est consacré aux concepts liés aux ERPs, les architectures des ERPs, leurs avantages et leurs inconvénients. Il décrit les différentes topologies existantes des ERPs et l'évolution du marché ERP. Il présente par la suite les processus d'implantation d'un ERP et les facteurs clés pour faire réussir cette implantation. Aussi, Il aborde l'impact de l'implantation de l'ERP sur la performance des entreprises ainsi que sur des sites dispersés. Puis, il présente les différentes approches récemment parues pour conduire les ERPs vers une nouvelle génération. Enfin, il se termine par une synthèse sur les différents types de l'interopérabilité et leurs relations avec les ERPs

Le deuxième chapitre se rapporte à la présentation des concepts liés à notre contexte et nécessaires pour la bonne compréhension de notre travail. Il décrit aussi les outils permettant l'interopérabilité aux niveaux technique, syntaxique et sémantique.

Le troisième chapitre aborde les approches existantes inhérentes à l'interopérabilité sémantique. Il scinde ces approches en deux catégories selon les techniques utilisées. La première catégorie s'articule sur les systèmes multi-agents et ontologies. La deuxième catégorie se focalise sur les web services et ontologies. Il les évalue sur un certain nombre de critères relatifs à la modélisation de la sémantique et dresse un bilan. Il expose aussi une synthèse et le positionnement de notre contribution. Enfin, il présente la modélisation de l'ERP par les SMA et les ontologies.

Le quatrième chapitre est dédié à la description de notre architecture. Dans un premier temps, nous présentons l'approche globale de notre architecture en couches. Puis, nous donnons toutes les définitions nécessaires afin de rendre concrète l'architecture et détaillons chaque point de cette approche, en présentant d'abord une méthodologie de construction de la couche de base qui est une architecture orientée services. Ensuite, nous décrivons le modèle sémantique auquel nous enrichissons sémantiquement les descriptions des services Web. Tous ces concepts et mécanismes seront exploités par la dernière

couche en l'occurrence l'architecture d'interopérabilité basée services.

Le dernier chapitre présente une étude de cas et les aspects d'implémentation pour mettre en évidence les concepts et les mécanismes de l'architecture proposée.

Finalement, cette thèse est clôturée par une conclusion dans laquelle nous dressons un bilan et une synthèse des travaux abordés, et nous citons les développements envisagés relatifs à l'amélioration de l'architecture et les perspectives de recherches possibles.

# Chapitre I : Coopération des systèmes d'information à base d'ERP

---

## I.1. Introduction

Les activités de toutes les entreprises économiques dans le monde se développent actuellement dans un environnement concurrentiel, complexe et difficile. Pour faire face à cet environnement dont la concurrence est impitoyable, les entreprises économiques doivent assurer une amélioration incessante de leurs services, de leurs savoir-faires, veiller à l'augmentation de leurs productions et à la qualité tout en réduisant les coûts pour faire face aux défis du marché mondial. Pour y faire face, les sociétés économiques doivent impérativement faire appel à la créativité et utiliser de nouveaux outils des technologies de l'information et de la communication, en l'occurrence les progiciels de gestion intégrés (PGI) ou Entreprise Resource Planning (ERP). Ces progiciels de gestion permettent le pilotage des systèmes d'information de l'entreprise et des communications cohérentes entre les différents acteurs métiers. Etant l'ERP est le centre nerveux du Système d'information de l'entreprise, l'ERP permet la gestion de l'ensemble des secteurs tels que les ressources humaines, matérielle et financières [4], [5].

Dans ce chapitre, et avant de définir l'ERP, ses caractéristiques et ses architectures ainsi que leurs avantages et leurs inconvénients, il nous paraît utile de parler même succinctement de son historique.

## I.2. Historique de l'ERP

La gestion de la production assistée par ordinateur, la GPAO a vu le jour dans les entreprises industrielle au cours des années 60-70. Elle était fondée surtout sur la planification de ressources de production (MRP-Material Requirements Planning en anglais) qui est utilisée pour assurer le calcul des besoins et établir un tableau de planification des délais. En effet, le principal but de la gestion de production Assistée par Ordinateur était de transformer les données commerciales relatives aux ventes en données de production.

Cependant, les progiciels commencent à s'imposer face aux développements spécifiques dans les débuts des années 1980 où les MPR I sont remplacés par le concept MPR II. Tout en leur intégrant d'autres importantes fonctions telles que le calcul des coûts, le suivi de production et la planification à long, moyen et court termes, car MPR II ont été développés par les moyennes et grandes entreprises économiques. A la même période, le concept de Computer Integrated Manufacturing ou CIM a été utilisé pour intégrer des activités de production [6].

Avec la maturité de méthodes de gestion MPR, les ERP (Entreprise Resource Planning) ont vu le jour tout au début des années 1990 grâce à la généralisation du concept d'intégration des systèmes informatique et des processus métiers dans tous les domaines de l'entreprise. On peut affirmer donc que les ERP sont des progiciels de gestion et d'analyse possédant une forte cohérence de diffusion en interne, ce qui permet d'améliorer les processus de gestion et d'automatiser les tâches, ce qui augmentera énormément la réactivité des entreprises

économiques. Cependant, au début des années 2000, et grâce à la diffusion de l'internet, l'ERP a intégré d'autres extensions d'affaires telles que le Supply Chain Management (SCM) pour la gestion de la chaîne logistique, le Product lifecycle Management (ou PLM) pour la gestion du cycle de vie du produit, le E-Business ou le business intelligence et enfin le Customer Relationship Management (ou CRM) pour la gestion des relations clients [7].

De nos jours, d'autres fonctions plus récentes ont été intégrées par les fournisseurs d'ERPs ; on peut citer par exemple l'intégration de l'exécution des opérations et l'analytique en temps réel. On peut citer d'autres intégrations telles que l'utilisation des outils mobiles ainsi que le déploiement de la solution ERP dans une plateforme cloud [8]. On peut dire que ce progiciel fournit à l'ensemble des acteurs de l'entreprise, une image unifiée, intégrée, cohérente et homogène de l'ensemble des informations dont ils ont besoin [9].

En conclusion, les grands éditeurs qui regroupent des offres complètes sont Microsoft avec sa gamme DYNAMICS, SAP, et Oracle Business suite. D'autres éditeurs tel que SAGE proposent aussi des suites de gestion intégrées qui sont considérées comme ERP [10].

### I.3. Définitions de l'ERP

Comme on peut le relever, le mot « définitions » est mis en pluriel. Cependant avant de passer outre, l'ERP est le terme le plus couramment utilisé.

Il existe donc plusieurs définitions de l'ERP que nous vous présentons ci-dessous :

1. L'auteur REIX en 2002 [11] a défini l'ERP comme « une application informatique paramétrable, modulaire et intégrée, qui vise à fédérer et à optimiser les processus de gestion de l'entreprise en proposant un référentiel unique et en s'appuyant sur des règles de gestion standard ». Cette définition souligne le caractère standard de ce progiciel.
2. Jean-Louis Lequeux, l'auteur de l'ouvrage « Manager avec les ERP » [10] définit l'ERP comme étant : « un sous ensemble du système d'information capable de prendre en charge la gestion intégrale de l'entreprise, incluant la gestion comptable et financière, la gestion de la production et de la logistique, la gestion des ressources humaines, la gestion administrative ainsi que la gestion des ventes et des achats ». Pour lui, la vocation des ERP est une vision intégrale de l'entreprise économique qu'il considère comme un outil auquel il faut assurer une productivité optimale dans le but de pouvoir réduire au maximum les cycles de mise sur le marché des produits et/ou des services.
3. Cependant, il semble que la définition de l'ERP donnée par Willes et al. en 2003 [12], la plus juste et la plus complète. Pour eux « L'ERP est un système intégré qui permet à l'entreprise de standardiser son système d'information pour relier et automatiser ses processus de base. Il fournit ainsi aux employés les informations nécessaires pour diriger et contrôler les activités essentielles de l'entreprise telles que le long de la chaîne logistique, de l'approvisionnement à la production/ exploitation jusqu'à la vente et à la livraison au client final. Les employés n'entrent qu'une seule fois les informations, qui sont alors mises à la disposition de tous les systèmes de l'entreprise.» [13].
4. Quant à JONE [14], un autre auteur émérite, propose une toute autre définition. Pour lui, l'ERP renvoie à l'infrastructure logicielle : celle-ci assure non seulement la cohésion

interne dans l'ensemble de l'entité économique mais elle soutient aussi les processus commerciaux externes dont l'entreprise est partie prenante.

#### **I.4. Des caractéristiques d'un ERP**

L'ERP est considérée comme le centre nerveux vital du système d'information selon la référence [10], et qui inclut les données et caractéristiques globales suivantes :

1. existence d'un référentiel unique des données ainsi que les indications nécessaires pour retrouver les données elles-mêmes sur la base de données ;
2. adaptation des règles de fonctionnement (professionnelles, légales et règles dictées par le marché) ;
3. gestion effective de plusieurs domaines de l'entreprise par des modules intégrés ou des progiciels susceptibles d'assurer une collaboration des processus ;
4. unicité d'administration du sous-système applicatif (les applications) ;
5. existence d'outils de développement ou de personnalisation de compléments applicatifs.
6. uniformisation des Interfaces Homme-Machine (IHM) : même ergonomie des écrans, mêmes boutons, même famille de barres des menu, mêmes touches de fonctions et de raccourcis.

Cependant, Il est important de souligner que toute solution intégrée doit répondre aux trois premiers critères pour être un ERP [10].

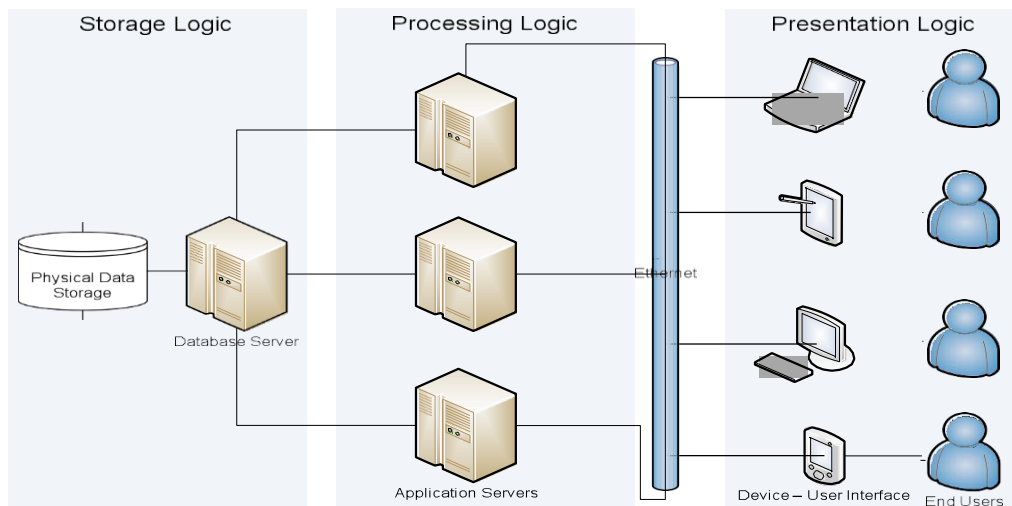
#### **I.5. Les architectures d'un ERP**

Nous avons constaté dans notre documentation et recherche bibliographiques l'existence de 02 types d'architectures pour un progiciel de gestion intégrée :

- ✓ L'architecture technique ;
- ✓ L'architecture modulaire.

##### **I.5.1. Architecture technique d'un ERP**

Les systèmes ERP tels que SAP et Oracle utilisent aujourd'hui une architecture de type Trois-Tiers Client/serveur avec une base de données centralisée de type relationnelle [3].



**Figure I-1 :** Architecture technique de l'ERP [3]

Comme illustrée dans la Figure I-1, l'architecture technique comporte trois couches. La première est la couche de « Serveur de base de données » illustrée pour gérer et stocker un volume important de données à travers une base de données relationnelle. La deuxième couche est le « Serveur d'application » où l'on trouve l'application serveur de l'ERP qui permet d'implémenter le logique métier, les processus d'affaires, les règles métier, l'authentification et la gestion des utilisateurs. La dernière couche est celle de « Présentation » qui est illustrée sous la forme d'une interface utilisateur. Elle est employée sur plusieurs périphériques allant de postes de travail jusqu'aux appareils mobiles. Nous constatons que cette architecture technique présente une grande flexibilité et évolutivité au progiciel ERP, comme par exemple l'extensibilité de l'architecture qui est obtenue grâce à l'introduction d'une nouvelle couche (Tiers) pour un autre serveur (serveur application, serveur web ... etc.)

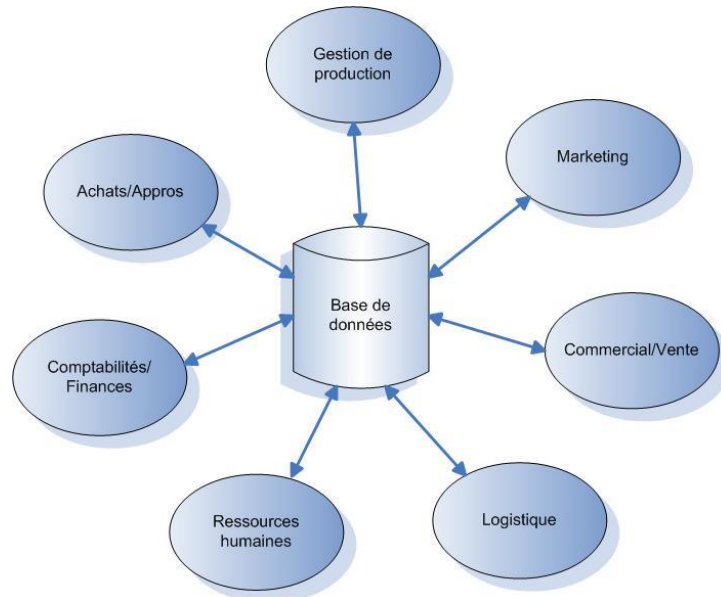
Plus récemment, une autre architecture technique est apparu dans le monde des ERP, pour donner une autre alternative de mettre en place du système ERP qui est « l'architecture en mode hébergé », possible soit auprès d'une solution Cloud de l'éditeur lui-même, par exemple comme la solution « SAP HANA Enterprise Cloud (HEC) », soit auprès des partenaires commerciaux de l'éditeur de progiciel dans leurs environnements respectifs [8].

### **I.5.2. Architecture modulaire d'un ERP**

D'après Jean-Louis Lequeux [10], un ERP est défini comme une application ayant au minimum les modules suivants :

- 1- GPAO (Gestion de Production Assistée par Ordinateur),
- 2- gestion comptable et financière,
- 3- gestion commerciale,
- 4- ressources humaines,
- 5- gestion planning,
- 6- gestion client,

Il existe plusieurs classifications de l'architecture modulaires d'un ERP dans la littérature des ERP. Cependant un accord tacite fait que toutes ces classifications s'accordent sur les modules essentiels pour un ERP et qui sont montrés dans la figure suivante :

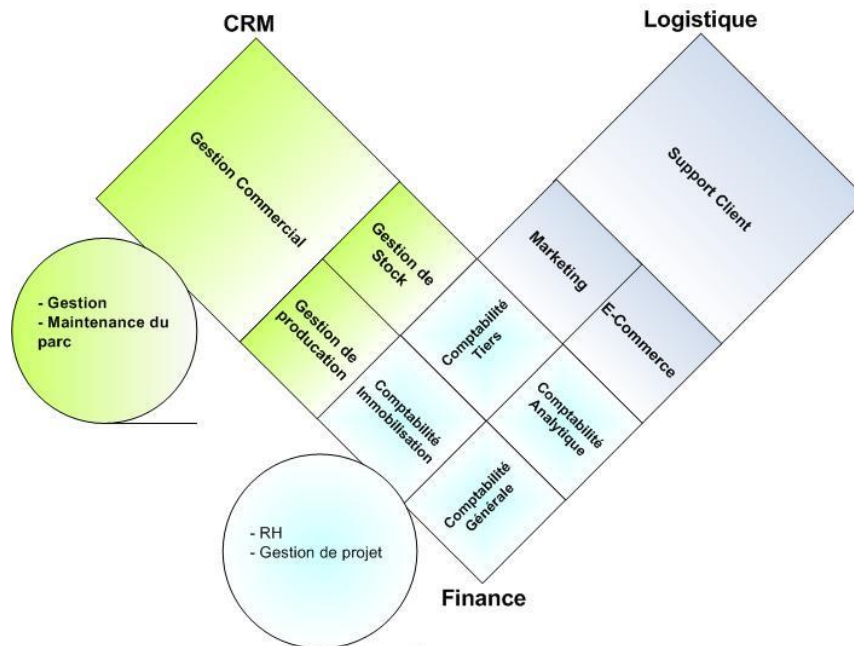


**Figure I-2 :** Architecture générale d'un ERP

Ainsi, l'utilisation d'une même base de données assure la cohérence de toute information dans l'organisation qui permettra donc une gestion homogène et sensée du système d'information de l'entreprise. Comme indiqué sur le schéma I-2, il existe un grand nombre des modules autour de cette base de données. Nous pouvons citer autres modules : la gestion de production, la gestion comptable et financière, la logistique, la gestion des ressources humaines, sans oublier la gestion Appro/Achats, la gestion du marketing et enfin le module commercial/vente.

D'après [15], un système ERP est dit modulaire dans le sens où il nous permet d'avoir une ou plusieurs applications en même temps, ou module par module. Les applications modulaires telles que les ERP, nous assurent de la pertinence et de l'efficacité de la compatibilité des modules entre eux, ces ERP s'imbriquent entre eux comme des blocs de des blocs de Lego et fonctionnent ensemble, d'où la garantie de la sûreté de la comptabilité à effectuer entre eux.

La figure I.3 représente l'architecture modulaire d'un ERP :



**Figure I-3 : Architecture modulaire ERP [15]**

Plusieurs modules touchant aux grandes activités de l'entreprise sont intégrés dans cette architecture modulaire. Ces modules rendent le système intégré et unifié.

On a donc une autre classification d'après la référence [16] basée sur les domaines fonctionnels de l'entreprise, et qui permet de sérier les modules de l'ERP en trois domaines distincts et qui sont :

- 1- **Le domaine métier :** il réunit un ensemble de modules qui prennent en charge la réalisation des processus situés sur la chaîne de valeurs de l'entreprise, tout comme, les modules de gestion des achats, de gestion de production, ou de gestion des stocks ...etc.
- 2- **Le domaine du pilotage :** toutes les fonctions facilitant l'analyse et la synthèse du fonctionnement du système de l'entreprise sont prises en charge par les modules de ce domaine. On peut citer entre autres les prévisions et les évaluations de la performance, les activités financières et comptables, la relation client, les tableaux de bord du contrôle de gestion...etc.
- 3- **Le domaine du support :** il s'agit là des fonctions responsables de l'administration du système informatique. Ainsi l'entretien de la structure de données et la modification des paramètres de configuration de la solution logicielle sont couverts par ce domaine.

## I.6. Avantages d'un ERP

Par rapport à un nouveau développement spécifique, on peut affirmer que l'ERP présente plusieurs avantages que nous citerons dans les points suivants [1], [15] et [17] :

1. L'intégrité et l'unicité du SI : autrement dit un ERP assure une logique et une ergonomie unique via sa base de données et cette dernière est également unique au sens « logique ». Pour être bref, on dira qu'un ERP permet d'éviter la redondance d'informations entre différents SI de l'entreprise.



2. Les communications interne et externe sont facilitées par le partage du même système d'information
3. Une meilleure coordination des services, donc un meilleur suivi de commande ou meilleure maîtrise des stocks par exemple.
4. La normalisation de la gestion des ressources humaines pour, par exemple, les entreprises qui gèrent de nombreuses entités économiques dispersées parfois géographiquement.
5. La possibilité pour l'utilisateur de récupérer des données quasi immédiate et/ou encore de les enregistrer.
6. Les mises à jour dans la base de données sont effectuées en temps réel et émises aux modules concernés.
7. Un ERP étant un outil multilingue et multidevise, il est de facto adapté au marché mondial, et en particulier aux multinationales.
8. Il n'existe point d'interface entre les modules, et donc il y a synchronisation des traitements et optimisation des processus de gestion.
9. La maintenance corrective est assurée directement par l'éditeur et non plus par le service informatique de l'entreprise. Cette maintenance corrective est donc simplifiée et permet surtout la maîtrise des coûts et des délais de mise en œuvre et de déploiement.
10. Des indicateurs nettement plus fiables sont mis à profit par les cadres supérieurs, alors qu'ils étaient extraits de plusieurs systèmes différents.

## **I.7. Des Intérêts de l'ERP dans l'entreprise**

Différents et d'innombrables études, enquêtes et ouvrages ont été faits en Europe en France surtout et aux USA sur la mise en œuvre des ERP. Ils ont permis surtout de mettre en avant les axes les plus fréquents d'amélioration cités par les entreprises dans le but de justifier leurs investissements dans l'ERP – Ainsi, on citera [18] :

1. Le pilotage par des indicateurs de performances et une meilleure connaissance des structures de coûts, et cela au niveau de l'ensemble des activités de l'entreprise.
2. Au niveau des ventes s'opère une augmentation du nombre d'offres qui permettra de générer éventuellement, des revenus complémentaires.
3. L'optimisation des fonds propres au niveau financier avec, par exemple, une amélioration des délais de paiement et une diminution des stocks.
4. Au niveau des achats, on notera une optimisation des conditions des achats grâce à une meilleure capacité à négocier, avec une diminution sensible des temps d'approvisionnement.
5. On notera, au niveau de la production, la maîtrise des coûts de production et des niveaux de stock pour augmenter la productivité et accroître ainsi la réactivité.
6. Au niveau de la communication interservices de l'entreprise, on soulignera une meilleure circulation des informations grâce à l'existence d'une base de données unique.

## **I.8. Des Inconvénients des ERP**

Malgré tous ces avantages des ERP que nous venons de voir, ces ERP possèdent quelques inconvénients qui sont difficiles à mettre en application car ils dépendent de la participation de nombreux acteurs comme : l'éditeur d'ERP, l'intégrateur de la solution et l'équipe de projet. A cet effet, nous citerons à titre d'exemple les inconvénients liés aux ERP [24] et [25]

1. La lourdeur et la rigidité d'implantation de la solution, mais ces inconvénients n'apparaissent que si le projet ERP n'est pas bien piloté.
2. Le coût s'avérera élevé sauf dans le cas des progiciels ERP libres qui nécessitent des formations aux utilisateurs de l'entreprise.
3. Les personnels utilisateurs étant souvent mal préparés aux nouvelles tâches, ils développeront une résistance aux changements.
4. Il y aura sous-utilisations du logiciel parce qu'il est très souvent surdimensionné par rapport aux besoins des entreprises.
5. Un certain nombre de module de l'ERP peuvent être moins performants qu'un développement spécifique interne ou un logiciel spécialisé.
6. Une maintenance continue devient ultra nécessaire, car elle est une forme d'intérêt et de captivité vis-à-vis de l'éditeur.
7. Il y a nécessité absolue à bien connaître les processus de l'entreprise dans le but de bien paramétrer le fonctionnement de l'ERP propre aux besoins de l'entreprise.

## **I.9. Topologies des ERP**

Globalement, on peut classer les ERP en trois grandes catégories [19], [20], [21], [22] et [23], soit : ERP généralistes, ERP spécialisés (métiers) et ERP verticaux, donc trois catégories principales dont nous tenterons d'expliquer le rôle de chacune :

### **I.9.1. ERP généralistes**

Tous les ERP qui font partie de cette catégorie, et comme leur nom l'indique, couvrent la globalité des processus d'une entreprise. Ces ERPs généralistes sont conçus de façon à répondre aux besoins de gestion standard, donc besoins communs à la plupart des entreprises économiques, soit : la gestion comptable et financière, la gestion commerciale, la gestion des ressources humaines, sans oublier la gestion de production et la gestion logistique le cas échéant. En plus, les ERP généralistes peuvent faire l'objet d'une implémentation sur une large majorité des secteurs d'activité peu complexes, mais n'arrivent malheureusement pas à couvrir toutes les spécificités des métiers.

### **I.9.2. Les ERP spécialisés (ERP métiers)**

Jusqu' à ce jour, ces ERP métiers ne possèdent pas encore l'agilité nécessaire pour suivre efficacement le rythme des changements inéluctables et rapides, puisque les métiers évoluent sans cesse et réclament des outils de plus en plus flexibles. A cet effet, les ERP spécialisés vont nous permettre de répondre parfaitement aux problématiques métiers complexes ou particulière mais qui reposent sur des règles métiers relatives aux domaines d'activités de l'entreprise, les rendant ainsi plus adaptés aux besoins de cette dernière et surtout plus souple à utiliser, tout en fournissant de meilleurs résultat.

Cependant, ces ERP, métiers coutent moins chers que ceux des ERP généralistes, et donc ils conviennent beaucoup mieux aux petites et moyennes entreprises dont les ressources financières sont en général plus limitées que celles des grandes entreprises. En outre, les ERPs métiers s'avèrent parfois, voire même souvent plus adéquats à ce type de besoin : un projet plus court et bien délimité mais capable de générer de la plus-value, donc des bénéfices mesurables. On citera entre autres les secteurs de la santé, de la pharmacie, de l'agroalimentaire de la chimie, de la biologie, du BTP, le commerce et la logistique sont recouverts par cette catégorie de progiciel.

### **I.9.3. Les ERPs verticaux**

Les ERP verticaux appelés aussi « best of breed », ces ERPs sont apparus pour pallier aux insuffisances fonctionnelles des ERPs généralistes ces ERP, verticaux proposent donc d'autres alternatives pour répondre parfaitement à un métier ou à une spécificité soit par l'ajout des couches logiciels plus ou moins intégrées, soit par des développements sur mesure, soit encore par des applications spécifique dédiées à telle ou telle fonction métier. Cependant, ils ont besoin des interfaces avec le reste des composants du système de l'entreprise, ce qui rend son intégration compliquée et fort coûteuse. On considère comme ERP ce type de logiciel du fait qu'il se comporte de façon à répondre à une logique semblable à celle d'un ERP. Pour l'analyste senior au CXP, M. Patrick Rahali en l'occurrence, la verticalisation des ERP comprend plusieurs volets dont :

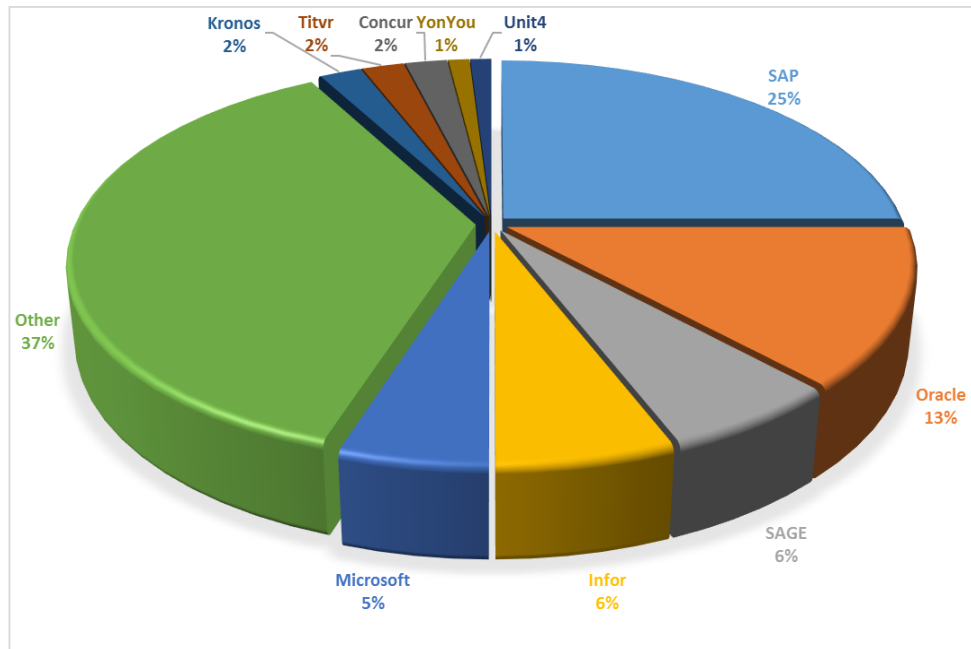
- un pré-paramétrage ou bien une pré-configuration du progiciel ;
- l'ajout de modules ou de fonctions spécifiques ;
- l'implication de consultants qui connaissent le métier de l'entreprise ;
- une méthodologie de mise en œuvre adaptée.

### **I.10. Marché des ERP**

Le marché de l'ERP représente en 2013 près de 3.700 milliards de dollars au niveau mondial, d'après la mise à jour de Monsieur Gartner en date du 9 juillet 2013. Il a ajouté que la croissance moyenne de ce marché est de 4.5 % annuellement. En outre, une étude récente menée en 2014 par des experts de cabinets de conseils et d'études a permis d'évaluer les parts de marchés des grands éditeurs d'ERP propriétaires au niveau mondial. Ainsi, le marché mondial des ERP était dominé en 2014 par SAP, le leader mondial, qui représente à lui seul 25 % de la couverture en ERP.

Oracle qui avait racheté Peoplesoft, représente 13 % du marché mondial. Cependant Sage reste bien placé avec une clientèle de PME en particulier.

La figure I.4 ci-dessous illustre à elle seule la répartition des parts de marché des principaux ERP en 2014 au niveau mondial.



**Figure I-4 : Marché mondial des ERP en 2014**

En général, on distingue deux types de marchés des ERP ; les ERP propriétaires et les ERP open sources :

### **I.10.1. Les ERP propriétaires**

Ce sont des logiciels créés par une société spécialisée dans la conception, la production de ces logiciels, l'achat d'une licence est nécessaire. Ainsi SAP, Peoplesoft, Oracle, Sage et Microsoft avec sa gamme DYNAMICS sont parmi les principaux détenteurs au niveau du marché mondial. Cependant SAP, par ordre de grandeur et de mérite est la société qui a permis aux ERP de voir le jour. Elle reste à ce jour encore leader sur le marché mondial des ERP. En effet, en proposant un progiciel multi langue et multidevises, SAP a obtenu de facto un succès important sur les autres entreprises. SAP s'est aussi intéressée au marché des PME qui demeure en pleine croissance en proposant sa suite Business One [17].

Ses modules couvrent la totalité des fonctions de gestion de l'entreprise. SAP étant une application client-Serveur, chacun de ses modules proposés couvre des besoins complets de gestion. Certaines entreprises implémentent tous les modules fonctionnels de SAP, ou se contentent d'implémenter seulement quelques-uns. SAP R/3, par exemple est entièrement paramétrable ; grâce à son environnement de développement, SAP R/3 peut être adapté à des besoins spécifiques [17].

### I.10.2. Les ERP Open source

Ces ERP Open source peuvent être produits gratuitement puisqu'il n'y a pas de coût de licence, mais on est obligé d'inclure dans le calcul du coût d'acquisition les frais de maintenance et d'assistance techniques. Parmi les principaux progiciel Open source, l'on citera : l'Aria, l'Open Bravo, Odoo, Compiere, ERP5, OFBiz, Tiny ERP. Dolibarr, SQL Ledger.

Les PME utilisent beaucoup les ERP open source du fait qu'ils sont plus faciles à intégrer et à personnaliser même si l'intégration et la personnalisation impliquent la nécessité des connaissances informatiques excellentes. En outre, par rapport aux ERP propriétaires, le choix d'un ERP open source donne divers avantages dans nous citerons quelques-uns :

- Un ERP open source ne nécessite pas l'acquisition d'une licence ;
- il reste 20 % à 50 % moins cher qu'un ERP propriétaire ;
- l'absence ou la non-nécessité d'acquérir une licence sur les ERP open source, donne une certaine indépendance aux entreprises qui ne prennent de ce fait aucun engagement préalable [19].

Actuellement, les ERP profitent du web. Les utilisateurs accèdent à ces systèmes au moyen d'un navigateur web. Ces progiciels sont de plus en plus destinés vers l'extérieur et sont surtout capables de communiquer avec les clients, les fournisseurs et d'autres organisations.

## I.11. De l'Implantation de l'ERP

Il existe, comme déjà signalé et présenté dans la section précédente, une forte évolution des produits ERP sur le marché. En choisissant tel ou tel progiciel ERP, une entreprise doit penser à investir sur différentes ressources à savoir les ressources humaines, matérielles et financières qui sont d'une nécessité absolue pour l'implémentation de l'ERP tels que : créer un réseau, recruter des employés, acquérir un serveur ERP et de nouveaux matériels informatiques...etc. On ne doit surtout pas oublier que l'implantation d'un ERP est à elle seule un projet d'entreprise qui signifie donc la refonte du système d'information mais surtout la remise à plat radicale des procédures de gestion au sein de l'organisation.

Dans la section qui suit, nous présentons le processus de l'implantation de l'ERP ainsi que les facteurs clés pour faire de l'implantation de la solution ERP un véritable succès.

### I.11.1. Processus d'implantation de l'ERP

Durant les nombreux problèmes rencontrés au moment de l'implantation d'un ERP, les auteurs de [26] ont préféré introduire un autre modèle d'implantation en quatre étapes pour la mise en place du progiciel ERP :

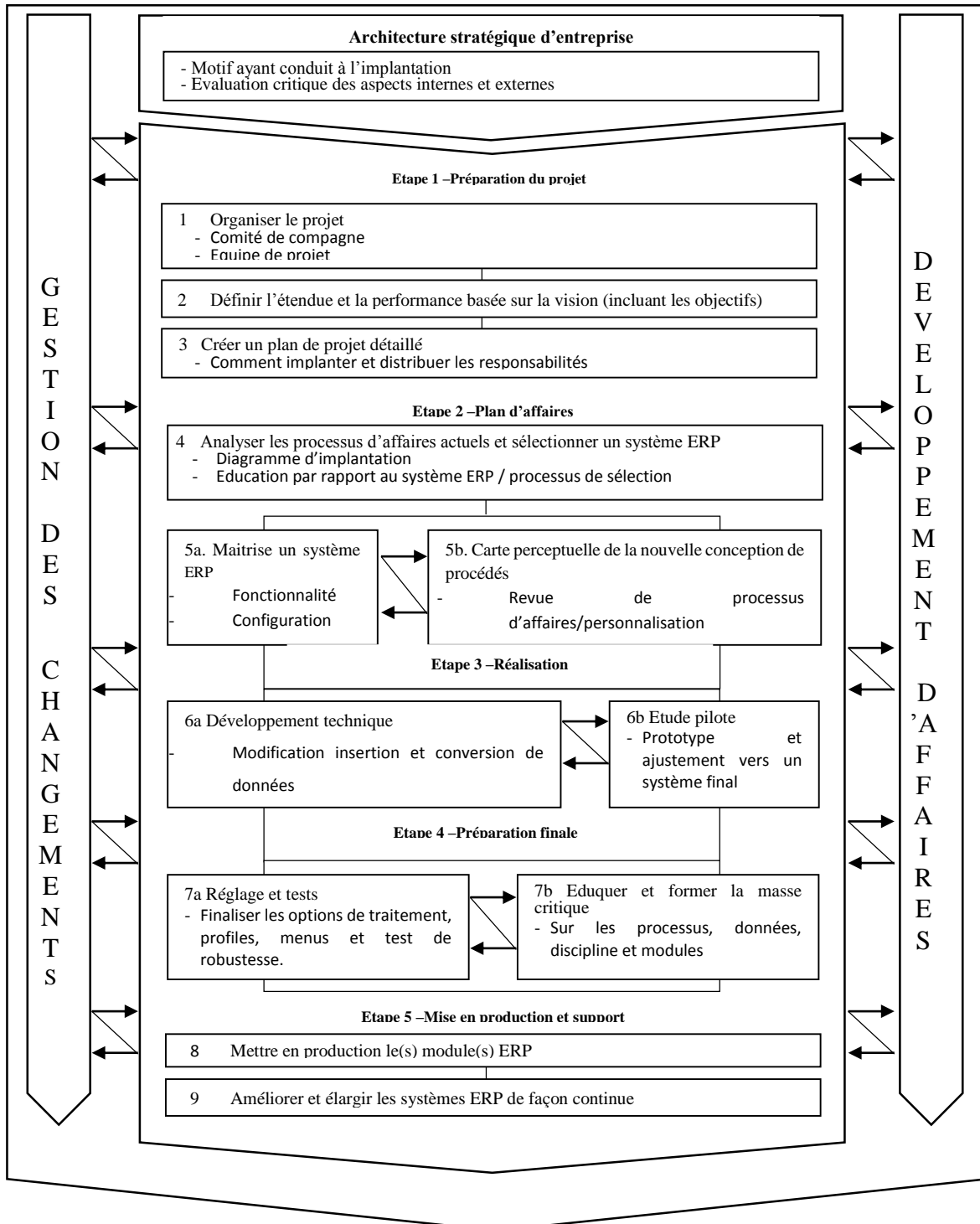
**1. Etape de conception** : c'est en quelque sorte la 1 ère partie de planification dans laquelle sont prises les principales décisions et qui sont : la sélection du progiciel, l'identification du chef de projet, l'approbation du plan de projet et du budget.

**2. Etape d'implantation** : dans cette deuxième phase, on procède à l'analyse de la situation (actuelle) de l'entreprise, et à la mise en place du nouveau système ERP et sa configuration aux systèmes existants avant de procéder aux tests.

**3. Etape de stabilisation** : Cette étape intervient juste après avoir rompu avec l'ancien système se stabilise pour permettre de situer et où détecter les probables anomalies et d'intervenir pour leurs correction éventuelles.

**4. Etape d'amélioration continue** : il s'agit là de la maintenance sans arrêt du système, de l'introduction de nouvelles fonctionnalités aux systèmes, du support des utilisateurs, de la mise à jour de nouveau support etc...

Cependant, Ehie et Madsen ont proposé en 2005 [27] un autre processus en cinq étapes cette fois-ci pour la mise en œuvre d'un ERP (Figure I-5) : la préparation du projet, le plan d'affaires, la réalisation, la préparation finale, et la mise en production et support.



**Figure 0I-5 : Processus d'implantation d'un ERP [27]**

Ce processus passe inévitablement par une première phase de préparation afin de définir les objectifs et le plan de projet pour établir l'équipe de projet avec les rôles de leadership et enfin pour fixer des objectifs budgétaires. Une deux phases s'intéressera à l'analyse des processus d'affaires actuels afin de choisir un système ERP approprié. L'équipe de projet sera de suite formée sur les fonctionnalités et les configurations du système ERP sélectionné afin de mieux s'adapter aux processus d'affaires de l'entreprise. Dans l'étape de réalisation, l'équipe de projet est de facto concentrée sur la mise en œuvre du nouveau système ERP y compris sur la modification, le développement d'interfaces et sur la conversion des données. Parallèlement, une autre équipe met en essai le nouveau progiciel sur un site pilote de façon à tester chaque module de l'ERP avec son processus métier correspondant, cela dans le but de déceler des problèmes des problèmes éventuels. Enfin, concernant la préparation finale, l'ensemble du processus métiers est entièrement intégré et testé dans toute l'organisation avec non seulement des données complètes mais avec différents scénarès.

Lors de cette phase, les utilisateurs finaux seront aussi formés pour l'utilisation du progiciel. Le but principal de cette phase de mise en production et support est surtout de garantir que les modules du système ERP sont stables et mis en production. Les managers de l'entreprise peuvent faire des extensions futures sur le nouveau progiciel, par des raisons concurrentielle ou autres.

**I.11.2. Facteurs clés de succès de l'implantation de l'ERP**

Afin d'identifier les facteurs clés de succès (FCS) de la mise en œuvre de l'ERP. Rabaa'i et al. [28] se basent en 2009 sur plusieurs revues de la littérature. Il considère les douze (12) premiers FCS des plus importants et les plus souvent cités sur les études principales :

- Implication et soutien de la direction générale (gestion du changement, gestion des projets ERP, Business Process Reengineering, et personnalisation du système ERP, Formation des utilisateurs, composition de l'équipe d'ERP, vision et planification, sélection des consultants et des relations, plan de communication , sélection du système ERP, l'intégration du systèmes ERP et les mesures d'évaluations post-mise en œuvre. Voici ci-après les huit principaux facteurs clés de succès pour la mise en œuvre d'un ERP :

- **L'Implication et soutien de la direction générale**

Plusieurs chercheurs sont unanimes pour dire que deux conditions sinéquanone de la réussite d'un facteur clé de succès dans la mise en œuvre d'un ERP, sont l'implication et le soutien de la Direction Générale. L'implication d'un ERP doit, à juste titre, avoir donc l'approbation et le soutien de la direction générale. De ce fait, le chef de l'entreprise doit donner une priorité absolue au projet de l'implantation de l'ERP. Il doit, en effet, s'engager non seulement avec sa bonne volonté mais aussi assurer sa propre participation et fournir les ressources nécessaires pour la réussite de la mise en œuvre de l'ERP [29].

- **Gestion du changement**

Ehie et Madsen, auteurs émérites ont déclaré en 2005 [27] que la mise en œuvre d'un ERP nécessite plus l'implication dans l'évolution des systèmes logiciel ou matériel. La réingénierie des processus d'affaires dans la mise en œuvre de l'ERP, peut aider l'organisation à bénéficier idéalement de niveau plus élevés d'efficacité et d'amélioration de la performance . Malgré tout, la mise en œuvre de l'ERP entraîne inéluctablement des changements qui pourraient conduire à la résistance chez une partie des employés qui sont réfractaires à tout ce qui est nouveau [30]. De ce fait, le juste milieu à trouver entre le personnel et la technologie, autrement dit l'équilibre des conflits entre le personnel et la technologie, et la gestion efficace des employés dans le processus de changement sont des éléments clés pour la réussite de la mise en œuvre d'ERP [31].

- **Gestion de projet ERP**

Cette gestion de projet ERP doit être d'une efficacité probante car elle demeure un facteur fondamental pour la réussite de l'implantation de l'ERP. Un manque de compréhension des besoins du projet, l'incapacité d'attribuer le leadership et l'absence de l'orientation du projet (SIC) ont été constatés par les auteurs Bingi [32] ; cependant, ces constats objectifs sont des facteurs essentiels pour faire échouer la mise en œuvre de l'ERP. Ainsi, une gestion efficace de projet ERP doit tenir compte de ces constats et donc définir au préalable des objectifs clairs, établir un plan de travail et de ressources et enfin suivre attentivement et pas à pas les progrès du projet.

- **Réingénierie des processus d'affaires et la personnalisation de système ERP**

La réingénierie des processus d'affaires et la personnalisation de système ERP [33] sont les deux approches logiques pour la mise en œuvre des systèmes ERP dans une organisation. Des changements profonds dans les processus organisationnels sont créés par la réingénierie des processus d'affaires, ceci dans le but de les adapter aux fonctions de l'ERP.

D'autre part, si une organisation souhaite garder ses processus existants à l'aide d'un système ERP. Elle peut personnaliser les fonctions de l'ERP. Cependant les résultats de nombreuses recherches déconseillent la personnalisation de l'ERP qui doit être à tout prix évitée ou du moins réduite au minimum afin de garantir le maximum d'avantages offerts par les systèmes ERP [34] et [35].

- **Formation des utilisateurs**

Cette formation concernant l'utilisateur final est un facteur crucial et une condition sine qua non pour la réussite de la mise en œuvre de l'ERP. Le système intégré de l'ERP étant complexe, cette formation reste essentielle pour une parfaite et solide compréhension de la manière dont le système ERP fonctionne ainsi que la façon de l'utiliser. De ce fait, une formation et une éducation adéquates de l'utilisateur final maximisera les avantages de l'ERP et accroîtra la satisfaction des utilisateurs.

- **Composition de l'équipe de projet ERP**



Cette composition est d'autant plus importante qu'elle devra être composée de représentants de toutes les unités fonctionnelles en rapport avec l'ERP.

- **Sélection des consultants et des relations**

On notera que dans la mise en œuvre de l'ERP, les consultants jouent un rôle prépondérant. Ces consultants peuvent être en effet de véritables sources de connaissance pour les matériels, les logiciels et les personnels de l'ERP. Ils viendraient éventuellement en aide à l'équipe de projet ERP et participer à la vérification du projet ERP dans sa phase finale. D'un autre côté et pour réussir la maintenance du système et son suivi après part mise en œuvre, il y aura facto le transfert de la technologie et de la connaissance expérimentale par les consultants vers l'équipe de projet ERP.

- **Plan de communication**

Une communication saine au sein de l'organisation au cours du processus forte au sein de l'organisation au cours du processus de la mise en œuvre augmente le pourcentage de réussite de l'implantation l'ERP. Il permet aux utilisateurs de l'organisation à comprendre le but et les avantages attendus du projet, ainsi que de partager les progrès du projet.

## **I.12. Impact de l'implantation de l'ERP sur la performance dans l'entreprise**

Marciniak fait remarquer, à juste titre, en 2001 [36], que la raison principale derrière le choix de la solution ERP est l'augmentation de la performance. En effet, l'implantation des ERP dans les entreprises est synonyme d'une attente de l'amélioration de leur performance.

Cette amélioration et/ou augmentation de la performance s'explique par une optimisation des coûts et un accroissement de la réactivité et de la flexibilité de l'entreprise. Mais certains auteurs [29] font une révélation pour le moins problématique car ils affirment que ce n'étant pas aussi facile pour mesurer cette performance. Pour éviter cette problématique, ces auteurs ont retenu une vision sur quatre dimensions inhérentes à l'implantation des ERP dans l'entreprise :

- Une dimension de projet ;
- Une dimension de technique ;
- Une dimension comportementale ;
- Une dimension organisationnelle.

A cela, les auteurs dans [37] ont ajouté la dimension sociétale. Cependant, l'étude de CHAABOUNI en 2006 [29] nous parle de l'impact de l'implantation de l'ERP sur la performance de l'entreprise, mesuré sur un ensemble de critères d'évaluation sur trois grandes dimensions :

- 1- Dimension économique-financière ;
- 2- Dimension organisationnelle ;
- 3- Dimension humaines.

## 1- L'impact d'ERP sur la performance économique et financière

D'après les travaux faits par les auteurs de [39] et lors de l'analyse des impacts financières de la mise en œuvre de l'ERP, on a constaté une amélioration sensible de la performance de l'entreprise, et cela seulement trois ans après l'implémentation d'un système ERP. On a aussi constaté une baisse conséquente dans certains ratios tel que le ratio de coût des marchandises vendues par revenus et le ratio de l'employé par revenus et cette fois-ci ; pour chacune de trois années examinées juste après la mise en œuvre du système ERP.

L'implantation de l'ERP dans l'entreprise peut générer beaucoup d'avantages concurrentiels en offrant des produits à faibles coûts tout en améliorant les relations avec ses clients et les différentes parties prenantes [38].

De ce fait, on peut affirmer sans mal doute, que l'impact de l'implantation de l'ERP sur la performance économique-financière de l'entreprise se trouve dans l'optimisation des ressources et des coûts, dans la réduction des délais, dans l'accroissement de la productivité, la diminution des prix, dans une amélioration de la qualité des prestations fournies aux clients et dans l'augmentation logique des ventes de l'entreprise. Tous ces atouts feront d'elle une vraie entreprise compétitive.

## 2)- L'impact d'ERP sur la performance organisationnelle

En nous basant sur les travaux de Kalika [41] et ceux de Chaabouni [40], les critères pour évaluer la performance organisationnelle résident dans la qualité de la circulation de l'information entre les services, la coordination.

La coopération, le degré de contrôle, la communication, la décentralisation, la flexibilité et l'intégration. A tout cela, l'auteur Reix [11] ajoute le processus de décision.

En résumé, on peut définir donc la performance organisationnelle comme « la manière dont l'entreprise est organisée pour atteindre ses objectifs et la façon avec laquelle elle parviendra à les atteindre » (SIC).

Il faut cependant se rendre à l'évidence que l'impact de l'implantation de l'ERP sur la performance organisationnelle de l'entreprise s'accomplira sur plusieurs points et qui sont :

- La modification de la structure de l'organisation par la création de nouveaux services ;
- la réorganisation des services informatiques
- La modification de la nature, la circulation et la création de l'information.
- L'affectation de processus de décision plus rapide par l'implantation de l'ERP.
- La facilitation du processus de contrôle et la culture de l'organisation [42].

En conclusion, on peut dire que l'ERP assure à l'entreprise l'amélioration de la qualité des informations communiquées, tout en favorisant la coordination et tout en permettant la décentralisation des décisions et en facilitant le contrôle.

### 3)- L'impact des ERPs sur la performance humaine

L'implantation de l'ERP a toujours été une bonne occasion pour améliorer les connaissances du personnel. De fait, grâce à l'introduction de l'ERP, les utilisateurs du progiciel acquerront de facto de nouvelles connaissances, de nouvelles compétences pour mieux la manipuler – la manipulation et son utilisation leur permettra d'améliorer leurs productivités [38] et [43] tout en leur permettant l'amélioration des connaissances du personnel. En effet, par l'introduction de l'ERP, les utilisateurs de ce progiciel doivent acquérir de nouvelles compétences pour mieux la manipuler. Ils arrivent ainsi un certain niveau de confiance et d'efficacité lors de l'utilisation de l'ERP leur permettant d'améliorer leurs productivités [38] et [43] tout en leur inadéquats un certain niveau de confiance en eux-mêmes, et les rendre ainsi plus efficace.

En plus, l'ERP supporte bien l'intégration avec les outils décisionnels comme le Data Mining pour assister les dirigeants de l'entreprise dans leurs tâches de planification et de prise de décision. N'oublions pas que l'ERP est conçu surtout et principalement sur une base de données centralisée, ce qui nous donne la possibilité de l'intégrer rapidement avec l'outil traditionnel «Data Mining» dont l'objectif est d'analyser intuitivement la richesse des données ERP métiers et ainsi de répondre aux différentes problématiques de plusieurs fonctions simultanément la production, la logistique, les finances, les ventes...etc.

#### I.13. Méthodes d'intégration des ERPs

Il y'a exactement trois méthodes qui permettent d'intégrer un ERP au sein d'une entreprise et qui assurent le passage définitif de l'ancien système d'informations vers le nouveau progiciel et qui sont [44] :

**La méthode du Bigbang :** C'est la méthode d'intégration d'ERP la plus difficile mais aussi la plus ambitieuse, car elle fait basculer en une fois tout le système vers l'ERP qui sera installé sur toute l'organisation. Cependant l'avantage de cette approche qui réside dans le gain de temps d'implantation réduit est contre balance le risque couru en cas d'un blocage de l'entreprise, ce qui nécessitant donc de nombreuses corrections.

**La méthode de Franchise :** Cette méthode s'adresse aux grandes entreprises constituées de plusieurs établissements mais qui ne partagent pas toujours de nombreux processus d'affaires. Donc, cette méthode doit choisir l'un d'eux pour y installer d'ERP et ensuite, une fois qu'on est rassuré de la stabilité du système, on passera à un autre établissement et ainsi de suite jusqu'à ce que toute l'entreprise migre vers l'ERP. Tout en liant les processus communs des différents établissements.

Ainsi l'avant de cette approche se situe dans la stabilité du système avant de mettre en place l'ERP dans toute l'entreprise ce qui minimise donc le taux ou le nombre de corrections par rapport à la méthode que nous venons de voir précédemment.

Mais l'inconvénient de cette approche réside dans la durée de temps plus longue (et le fait de l'existence de plusieurs systèmes au sein de l'entreprise) jusqu'à ce que la bascule soit complète.

**La méthode de mise en place par module :** Cette méthode concerne surtout et généralement les petites entreprises car dans ces cas précis, on intègre les modules un par un en commençant par le module prioritaire, donc relatif aux fonctions nécessaires pour lesquelles l'entreprise a fait appel à l'ERP.

Comme, toute méthode, cette dernière possède dans ses approches des avantages et des inconvénients. L'avantage de cette approche réside dans le peu de correction à effectuer. Quant aux inconvénients qui présentent cette approche, ils résident dans le fait que cette approche nécessite de multiples interfaces avec les logiciels encore en cours.

## I.14. Implantation multi sites de l'ERP

Différentes formes peuvent être prises quand il s'agit de l'implémentation multi-site d'un ERP. Parmi ces formes, nous pouvons citer :

- **L'autonomie locale totale :** Elle est basée sur le principe de dotation des sous-unités de l'entreprise d'une véritable autonomie de décision presque totale. Cette forme donne l'avantage de piloter les différents systèmes ERP pour les différents plans des unités de la même entreprise. Des avantages qu'offre cette forme, nous évoquerons le fait :
  - ✓ de pouvoir éluder les conflits inhérents aux changements établis au niveau des sous-unités de l'entreprise.
  - ✓ de permettre aux entreprises le suivi de leurs futures acquisitions et cessions bien loin de complications des systèmes ;
  - ✓ de minimiser le risque de défaillance d'implantation d'un projet ERP.

Cependant, il y a aussi des inconvénients liés à cette forme d'implémentation qu'on peut résumer pour l'échéance de capter le potentiel des systèmes ERP pour l'intégration des données, des systèmes et des processus à travers lieux et les unités d'affaires.

Le contrôle des sites se fait uniquement au niveau financier : en effet la totalité des sous-unités de l'entreprise possède une véritable autonomie dans tous les domaines à l'exception de celui de la comptabilité financière, d'où la nécessité de contrôle à ce niveau névralgique. Cependant, l'efficacité de cette méthode reste majoritairement importante lorsque les sous-unités de l'entreprise effectuent des tâches différentes.

- **Coordination des opérations entre les sites :** Cette forme se caractérise surtout par un degré très élevé d'autonomie locale dans les opérations, alors que les différents sites conservent leur capacité de gérer la chaîne d'approvisionnement globale grâce à l'accès aux informations locales qui concernent : les achats, les inventaires et les échéanciers de production. Toutefois, cette forme reste plus efficace car il y a des avantages à obtenir à partir d'achats communs, ou bien si l'entreprise a des clients aussi bien locaux que régionaux.
- **Coordination réseaux des opérations :** A travers un degré élevé de centralisation, les opérations locales peuvent accéder aux informations des autres sites.

- **Centralisation totale** : dans cette méthode, il y'a un degré élevé de centralisation puisque le tout est implémenté localement [45].

## I.15. ERP étendu vers une nouvelle génération

L'évolution de l'ERP a commencé au début des années 2000, et s'est étendue depuis pour inclure de nouveaux modules inhérents à deux nouvelles approches différentes : L'approche Customer Relation Management ou CRM, et l'approche Supply Chain Management ou SCM.

### I.15.1. L'Approche CRM

La Gestion de la Relation Client, autrement dit le Customer Relationship Management, est tout simplement un module tourné vers le client et qui vise à l'amélioration de la communication entre l'entreprise et ses clients, tout en automatisant les différentes composantes de la relation client. Le CRM s'occupe de l'analyse des informations collectées sur le partenaire client afin de permettre à l'entreprise de revoir sa gamme de produits et répondre ainsi à mieux à ses attentes. Il assure aussi la fourniture en plus des outils de pilotage permettant ainsi une assistance dans la gestion des prises de contact, l'élaboration des propositions commerciales aux clients.

Cependant, et comme inconvénient majeur, le CRM présente une difficulté d'intégration fonctionnelle avec les autres modules, sans omettre la difficulté d'obtenir une coopération entre les différents services de l'entreprise, dans la mesure où tous ces services sont concernés par le client.

### I.15.2. L'Approche SCM

La Gestion globale de la chaîne logistique ou la Supply Chain Management est un mode d'organisation permettant à l'entreprise de :

- ✓ d'analyser et de comprendre la demande du marché et surtout de sélectionner les produits et services de distributions ;
- ✓ de gérer le juste milieu, autrement dit l'équilibre entre la demande que l'entreprise peut et veut satisfaire et sa capacité d'y répondre ;
- ✓ de dominer et/ou maîtriser la logistique fournisseur, les achats, la production, la vente, le stockage, la logistique client, la distribution, le service après-vente et afin de maîtriser l'ensemble des processus.

La SCM repose donc sur une série de technologies comme l'échange de données informatisées ou EDI, et l'Advanced Planning and Scheduling ou APS que nous essayerons d'expliquer ci-dessous :

- ✓ L'EDI : il s'agit en fait de la transmission informatisée de documents commerciaux entre des systèmes d'application de partenaires d'affaires indépendants via les réseaux de communication sous un format conventionnel standardisés.
- ✓ L'APS : c'est en fait un système informatique permettant de planifier à l'avance l'ensemble des flux physiques et financiers de l'entreprise [46].

Mais avec l'apparition de la notion de processus métier mais aussi avec l'évolution des nouvelles technologies, l'ERP doit impérativement se transformer. Outre le fait qu'il doit être

cohérent, il doit nécessairement être réactif. Et pour cela, L'ERP doit adopter soit une structure de type EAI organisée autour d'un middleware assurant la circulation entre les différents modules, soit se doter d'outils de contrôle et de supervision et d'optimisation horizontaux, de type workflows.

D'un autre côté, les différents modes d'interconnexion et l'utilisation d'internet, tout cela dans un contexte de commerce collaboratif offre d'avantages de services à l'ERP.

Aujourd'hui, les applications d'ERP trouvent une nouvelle valeur ajoutée dans leur capacité de gérer l'ensemble des processus de gestion entre l'entreprise, ses partenaires et ses clients. De ce fait et dans ce but, les ERP ont évolué vers des suites e-Business intégrées en tout en s'enrichissant de nouvelles fonctionnalités complémentaires.

Comme la Business Intelligence, ou plus récemment le E-Procurement, le E-Commerce et le E-Market Place.

Le **Marketplace**, ou littéralement la place de marché d'Internet : il d'agit d'un site où les entreprises ont la possibilité d'échanger des produits et des services. On distinguera donc à cet effet les places de marché verticales consacrées et propres à un secteur d'activités ou à une filière, et les places de marché horizontales spécifiquement consacrées à une fonction tel par exemple le transport et les sites d'appel d'offres généralistes traitant et s'occupant surtout des achats généraux [50].

**E-PROCUREMENT** : il s'agit-là d'un terme purement générique qualifiant différents aspects de la gestion administrative des achats de l'entreprise quand elle recourt de diverses façon au commerce électronique sur Internet, aux services d'appels d'offres en ligne, et aux places de marchés virtuelles, etc [47].

Les systèmes ERP ont été transformés par cette évolution fonctionnelle et le développement technologique, alors qu'initialement ils furent bâtis sur le mode client-serveur. Ces système ERP évoluent pratiquement de nos jours vers une logique de portail d'entreprise, tout en intégrant des technologies complexes de l'internet et en reliant entre elles des bases de données, applications transactionnelles et collaboratives très sophistiquées, ce qui mène les ERP vers une nouvelle génération d'ERP, ou « les ERP II ».

### I.15.3. ERP et WFMS

Bien que chacune d'entre elles adopté une approche différente, les ERP et les WFMS Management systèmes sont deux classes de systèmes qui se focalisent sur le traitement des processus d'affaires. Le WFMS est basé sur le paradigme de la spécification et l'exécution des processus, alors que le modèle WorkFlow est élaboré, en premier lieu, dans le but de préciser l'organisation des processus de l'entreprise, ensuite des instances de ce Workflow sont créés pour effectuer les étapes décrite dans le modèle de l'instance, pendant l'exécution du Workflow, peut accéder aux bases de données ainsi qu'aux application et peut interagir avec les utilisateurs.

Toutefois, le modèle de Workflow dans les systèmes ERP n'est pas explicitement spécifié parce qu'il est intégré dans les applications. Cependant, la différence essentielle entre le WFMS et l'ERP réside, dans les systèmes ERP, dans le fait que la logique de flux et la logique de

fonction sont à la fois intégrés dans des applications, des données et du modèle que les personnes invoquent.

De ce fait, un WFMS donne la possibilité aux développeurs de séparer les flux entre les composants d'un système (les applications, les données, les personnes) à partir du modèle de Workflow.

Les systèmes WFMS se centralisent sur les processus tout en mettant l'accent sur la gestion de flux logiques. Cependant, les systèmes ERP se focalisent sur les données et ils sont centrés sur la gestion d'une logique de fonction à l'aide d'une infrastructure de données homogène à travers l'organisation pour supporter de multiples applications. Ce qui nous amène à présenter la différence entre une logique de flux et une logique de fonction.

- ✓ **La logique de fonction** : Cette logique s'occupe du traitement d'une tâche spécifique tels la mise à jour d'un dossier client et /ou le calcul de l'ordre.
- ✓ **La logique de flux** : Afin de résoudre des problèmes complexes comme le traitement d'une commande, cette logique de flux combine une multitude de fonctions dans une séquence.

Ainsi, pour améliorer un système ERP, les nouveau ERP tendent vers l'intégration d'un WFMS au sein du même système. Cette approche a pour but de tirer profit des points forts des deux technologies tout en permettant l'intégration des modules responsables du pilotage des processus d'affaires (nous parlons ici de moteur Workflow) [48].

#### **I.15.4. ERP et l'Internet**

Tout un chacun sait que l'Internet offre un potentiel inestimable de services en tant que plateforme d'intégration. En outre son accès est largement disponible et ses technologies ont pour bases des standards ouverts permettant ainsi aux systèmes ERP de s'y adapter et de tirer profit des innombrables et différents bénéfices qu'elle peut offrir. Ainsi les ERP, basées sur le protocole http, deviennent des serveurs web, des serveurs d'applications, des langages JAVA et XML, et utiliserons des interfaces basées sur un navigateur pour que les utilisateurs puissent accéder au système à partir de n'importe quel point du globe.

#### **I.15.5. Deuxième (2<sup>ème</sup>) Génération de l'ERP**

Connue sous le nom de deuxième génération des ERPs ou encore ERP II, cette nouvelle tendance est en train d'émerger. On la considère comme une remise en œuvre des ERPs, donc une nouvelle implémentation qui a tendance à provoquer un éloignement des architectures client-serveur vers une réoarchitecture basée Internet et/ou de nouvelles technologies comme l'e-commerce (électronique commerce), le m-commerce (mobile & wireless Technologies) y sont intégrées. L'intégration concerne également d'autre technologies comme le c-commerce

(collaborative commerce), middleware, Enterprise Portal Technologies, Web services, Analytical capabilities (Data Warehousing & Data Mining), CRM, SCM, SRM, Knowledge Management et Business Intelligence.

Le passage de l'ERP à l'ERP a été une nécessité due à un changement stratégique d'un système qui était totalement focalisé sur l'entreprise, préoccupé seulement par l'entreprise, préoccupé seulement par l'optimisation des ressources internes et de traitement transactionnel, à un système basé sur l'intégration des processus et la collaboration externes. L'objectif donc des ERP nouvelle génération est la couverture de l'ensemble des grands processus de l'entreprise de bout en bout- les ERP nouvelle génération se distinguent

La nouvelle tendance est en train d'émerger. Elle est connu sous le nom de la deuxième génération des ERPs ou encore l'ERP II. Elle est considérée comme une remise en œuvre des ERPs ; une nouvelle implémentation, qui a provoqué un éloignement des architectures client-serveur vers une nouvelle architecture basée Internet, où de nouvelles technologies y sont intégrées comme l'e-commerce (Electronic Commerce), m-commerce (Mobile & Wireless Technologies), c-commerce (Collaborative Commerce), Middleware, Enterprise Portal Technologies, Web Services, Analytical Capabilities (Data Warehousing & Data Mining), CRM, SCM, SRM, Knowledge Management et Business Intelligence.

Le passage de l'ERP à l'ERP II est due à un changement stratégique d'un système qui est totalement centrée sur l'entreprise et préoccupé par l'optimisation des ressources internes et de traitement transactionnel à un système reposant sur l'intégration des processus et la collaboration externe. Les ERP de deuxième génération ont pour objectif de couvrir l'ensemble des grands processus de l'entreprise de bout en bout, ils se distinguent de leurs prédécesseurs par deux évolutions importants :

- ✓ Une couverture fonctionnelle élargie, et une nouvelle architecture technique orientée services qui reposant sur des technologies EAI (Enterprise Application Intégration) qui représente un vaste ensemble d'outils informatiques donnant la possibilité de faire communiquer les applications de l'entreprise entre elles, et aussi avec l'extérieur.
- ✓ Les ERP II ont pris désormais en charge de nombreuses nouvelles fonctions inhérentes aux Web services.

Ainsi, du point de vue purement techniques un saut qualitatif s'est produit mettant ainsi fin aux architectures monolithiques. Les ERP II se basent donc sur une architecture orientée services composée de trois couches distinctes et qui sont :

- ✓ Les modules qui composent le système ERP comme le module des finances, ou celui de ressources humaines ...etc, regroupent des services auxquels on peut accéder à travers une interface de programmation ou API standard telle que JCA ou SOAP. Et tout cela en incluant la possibilité pour l'entreprise de rajouter ses propres applications grâce aux connecteurs EAI.
- ✓ La synchronisation des échanges entre les services repose en fait sur un « bus EAI » traditionnel composé d'un courtier chargé des messages et/ou d'un middleware orienté message.
- ✓ Un moteur d'orchestration chargé de relier l'ensemble des services selon des processus métier (achat, commande, etc.) [49].

### **I.15.6. Comparaison entre la première et la deuxième génération**



On exposera dans cette section un tableau récapitulatif pour démontrer les différences principales entre l'ERP traditionnel et celui de la deuxième génération.

	ERP	ERP II
Rôle	ERP traditionnel a pour rôle l'optimisation interne de l'entreprise	permet d'optimiser la chaîne d'approvisionnement grâce à la collaboration avec les partenaires commerciaux.
Domaine	Elle est concentrée sur la fabrication et la distribution	traverse tous les secteurs d'affaire en relation avec l'entreprise.
processus	Dans les systèmes ERP, les processus ont été axés sur les quatre murs de l'entreprise	Systèmes ERP II se connecter avec les partenaires commerciaux, pour prendre ces processus au-delà des frontières de l'entreprise.
Architecture	Les anciens systèmes ERP ont été monolithique et fermée	Systèmes ERP II sera basé sur le Web, ouvert à intégrer et à interagir avec d'autres systèmes qui permettent aux utilisateurs de choisir seulement les fonctionnalités dont ils ont besoin.
données	L'information dans les systèmes ERP est produite et consommée au sein de l'entreprise	Dans un système ERP II, cette même information sera disponible dans toute la chaîne d'approvisionnement pour les participants autorisés

**Tableau I-1:** Comparaison entre ERP1 et ERP2

On remarquera de prime abord que cette nouvelle génération d'ERP dite ERP II rend facile l'intégration des applications existantes, et apporte surtout plus de réactivité à l'entreprise en permettant à cette dernière de mieux faire évoluer ses processus.

## I.16. Coopération et interopérabilité des systèmes d'information de type ERP

On tient donc à le définir et à présenter la notion d'interopérabilité qui le caractérise.

### I.16.1. Définition de l'interopérabilité

De nombreuses définitions de l'interopérabilité des systèmes d'information données dans la littérature. On en citera quelques-unes plus appropriées.

- ✓ «L'interopérabilité est la capacité des systèmes informatiques et des processus qu'ils supportent d'échanger des données tout en permettant le partage d'information et de connaissance», d'après EIF [51].
- ✓ Selon IEEE 1990 [52], «L'interopérabilité est la capacité que possèdent deux ou plusieurs systèmes ou composants à échanger des informations, puis à les exploiter.
- ✓ D'après IEEE 2000 [53], «L'interopérabilité est la capacité, approuvée mais non nécessairement garantie par la conformité à un ensemble de standards, d'équipements

hétérogènes, généralement développés par différents fabricants, de travailler ensemble dans un environnement en réseau ».

- ✓ Cependant H. Panetto et al. [54] « L'interopérabilité des applications d'entreprise est définie comme la capacité pour un système d'échanger de l'information et des services dans un environnement technologique et organisationnel hétérogène ».

## I.16.2. Niveau d'interopérabilité

L'European Interoperability Framework (ou l'EIF) préconise trois niveaux d'interopérabilité [55] :

**I.16.2.1. Interopérabilité technique**, qui est relative à la mise en œuvre des technologies de l'information et de la communication relative aux normes pour présenter, stocker, échanger, traiter et communiquer les données au moyen de matériels informatiques.

**I.16.2.1. Interopérabilité sémantique**, possèdent un niveau plus conceptuel pour garantir que les informations échangées soient compréhensibles du point de vue de leur signification et de leur interprétation par les applications qui les utilisent, mais ayant été aussi développées pour des objectifs différents. Grace donc à l'interopérabilité sémantique des systèmes sont à même de combiner l'information reçue de la part d'autres sources d'information et de la traiter tout en conservant le sens.

**I.16.2.1. Interopérabilité organisationnelle** : elle définit les responsabilités, les autorisations, les confiances, les aspects légaux, les propriétés intellectuelles et les structures organisationnelles qui sont nécessaires à l'acceptation des échanges d'information entre applications, par les différents acteurs. Cette interopérabilité de haut niveau est particulièrement mise en avant et utilisée dans le cadre de l'interopérabilité de l'administration électronique et des gouvernements [54] [56].

## I.17. Interopérabilité et intégration

D'après LAROUSSE, «L'intégration des systèmes consiste à assembler les différentes parties d'un système et à assurer leur compatibilité ainsi que le bon fonctionnement du système complet». Il s'agit donc de faire tomber les barrières fonctionnelles et organisationnelles au sein même de l'entreprise pour que l'ensemble soit perçu comme un tout cohérent [57].

On doit comprendre que l'intégration est un moyen et non pas un but en soi. En effet, elle est un moyen d'assurer mieux de garantir la cohérence fonctionnelle et informationnelle de l'entreprise. L'interopérabilité des systèmes n'est aussi qu'un moyen parmi tant d'autres pour faciliter l'intégration. Le vocable d'interopérabilité de système est utilisé dans la littérature, quand le système concernés sont capable de s'échanger des informations et d'agir ensemble. Ainsi, on réserve l vocabulaire d'intégration des systèmes pour le cas où les systèmes coopèrent au sein d'un système unique et homogène [58]. Ces deux termes deviennent synonymes uniquement dans le cas où l'on s'intéresse à l'intégration faiblement couplée, autrement dit quand, on cherche à maintenir l'hétérogénéité et l'autonomie des systèmes constituants [59]. L'objectif de notre travail de recherches est d'assurer l'interopérabilité sémantique des systèmes d'information à base d'ERP tout en proposant une architecture adéquate pour cela. Le deuxième chapitre sera consacré à l'étude des outils techniques et conceptuels qui permettent la mise en œuvre de l'interopérabilité sémantique.

## **I.18. Conclusion**

Dans ce premier chapitre, on a présenté et de façon générale mais précise la notion de système d'information à base d'ERP et d'applications d'entreprise. Nous nous sommes surtout focalisés sur la dimension d'hétérogénéité qui représente un problème pour l'interopérabilité des systèmes d'information en inter et intra-entreprise. Dans ce même chapitre, on a présenté aussi la notion de coopération et d'interopérabilité qui a été définie à plusieurs niveaux : technique (et syntaxique), sémantique et organisationnel.

La mise en place un nouvel ERP est devenu un enjeu stratégique pour une entreprise, non seulement sur le plan financier, mais aussi sur le plan qualitatif de son fonctionnement. Il est donc essentiel de se donner tous les moyens pour réussir, et à commencer par les moyens humains. En effet, la qualité du projet, c'est d'abord la qualité des hommes qui s'en occuperont. Il faut aussi prendre conscience que même quand l'ERP a été mis en place, il est fondamentalement nécessaire de s'assurer que la connaissance et la maîtrise du progiciel continueront à être assurées interminablement et pendant toute sa durée de vie fonctionnelle.

## Chapitre II : Des outils conceptuels et technologiques pour la mise en œuvre des mécanismes de médiation sémantique

---

### II.1. Avant-propos

Avant de passer outre, il nous paraît important d'expliquer tout au long de ce chapitre II la notion de sémantique, sa typologie, sa classification ainsi que les méthodes qui permettent sa représentation en insistant notamment sur les ontologies que nous essayerons de voir de près. Nous présenterons donc la classification des hétérogénéités sémantiques des services Web ainsi que les méthodes qui facilitent leur description au niveau sémantique ; sans pour cela oublier la classification des architectures de médiation. Nous soulignerons également la nécessité de ces architectures de médiation pour solutionner et/ou résoudre des conflits entre les services Web en interactions afin de permettre l'interopérabilité sémantique des systèmes d'information de type ERP.

Nous aborderons donc dans ce chapitre II ce nouveau concept et comprendre comment cette technologie peut arriver à combler les lacunes du paradigme agent dans notre solution en proposant des principes suffisamment génériques et souples de façon à permettre un développement efficace avec le moins d'effort possible.

Nous clôturerons enfin ce chapitre II par une classification des approches d'interopérabilité et d'intégration des services Web et une étude appropriée des travaux existants.

### II.2. La notion de sémantique

#### II.2.1. Définition

Outre la définition universellement admise que la sémantique est l'étude du sens des mots (Larousse, le robert,...), elle représente une branche de la linguistique. Cependant en informatique, Woods [60] ira plus loin en la définissant «comme une forme formelle de représentation des connaissances humaines».

Cependant dans le domaine des systèmes d'information [59], la sémantique s'intéresse plus précisément au sens des différents éléments d'un système d'information pouvant être soit des données, des fonctions, voire même des processus. La définition de la sémantique peut donc différer d'un système à un autre, en fonction du contexte d'utilisation. Ces hétérogénéités de type sémantique sont à l'origine des conflits entre applications, ce qui contraint leur interopération.





#### II.2.2. De la typologie des hétérogénéités sémantiques

Pour Kavouras [61], une typologie d'hétérogénéités sémantiques doit prendre en considération les termes (T1, T2), et leurs définitions (D1, D2), qu'il faut associer à des entités de la réalité, comme proposé aux tableaux II-1 qui illustrent les notions suivantes :

- **La notion Equivalence** : deux entités sont dites équivalentes si elles ont le même terme et la même définition.
- **La notion Disjonction** : deux entités sont dites disjointes si elles ont des termes et des définitions différents.
- **La notion de Synonymie** : il s'agit de cas où les deux entités sont représentées par deux termes différents et partagent la même définition.
- **La notion Homonymie** : c'est le cas où les deux entités sont représentées par le même terme et deux définitions différentes.
- **La notion Complétude** : c'est le cas où les deux entités sont représentées par le même terme, mais la définition de l'une est incluse dans l'autre.
  - ✓ *Spécialisation* : cas où les deux entités sont représentées par deux termes différents, et où la définition de l'une est incluse dans l'autre.
  - ✓ *Chevauchement* : c'est ici le cas où les définitions des deux entités se recouvrent.

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

**Légende**

-  Pas de conflit
-  Faible conflit
-  Moyen conflit
-  Grand conflit

**Tableau II-1:** Typologie des conflits sémantiques [61]

Nous soulignons donc qu'une représentation de la sémantique est impérative pour la réussite de l'interopérabilité des systèmes d'information à base d'ERP tout en faisant appel à des mécanismes de médiation pour trouver des solutions aux conflits inhérents à des différentes hétérogénéités sémantiques.

### II.2.3. De la classification de la sémantique

Deux célèbres auteurs-chercheurs, en l'occurrence Uschold et Gruninger [62] ont proposé une classification de la sémantique permettant de faire la distinction entre les différents types de sémantiques pouvant exister dans le contexte du Web :

- ✓ *La sémantique implicite* : elle existe seulement dans le mental des gens.
- ✓ *La sémantique semi-informelle (explicite et informelle)* : il s'agit en fait d'une sémantique explicite souvent représentée de façon informelle avec l'utilisation en général des langages naturels comme le français ou l'anglais.
- ✓ *La sémantique semi-formelle* : ce concept désigne une sémantique explicite ; elle est relativement formelle et est destinée surtout aux humains en utilisant des

formalismes, le plus souvent graphiques comme les modèles sémantiques, ou les diagrammes UML.

- ✓ *La sémantique formelle* : elle est en fait une sémantique qui se base sur des formalismes mathématiques très rigoureux comme la logique de description, la logique de premier ordre, etc..., ce qui permettra à ce concept d'être traité de façon automatique.

#### **II.2.4. De la représentation de la sémantique**

Pour représenter l'interopérabilité des systèmes d'information au niveau sémantique, il est nécessaire voire impératif de faire appel à l'aide des techniques de représentation de la connaissance. Cette dernière, soit la représentation des connaissances étudie la façon de transformer l'expression du sens en une représentation formelle manipulable par une machine [59]. Ainsi pour présenter sous forme de modèle la sémantique, on peut utiliser l'une des deux approches, à savoir :

- ✓ L'approche procédurale utilisant des procédures ou des règles pour la représentation de la sémantique.
- ✓ L'approche déclarative qui, elle, se base sur la modélisation des faits.

Cette approche déclarative est plus judicieuse que l'approche procédurale puisqu'elle présente beaucoup plus d'avantages tels la standardisation, la capture, la réutilisation, l'inférence et la flexibilité [62]. Ainsi, dans ce contexte, elle reste l'un des outils les plus utilisés dans le domaine de la représentation sémantique et dans la notion d'ontologies que nous vous présenterons plus en détail dans ce qui suit.

### **II.3. Ingénierie Ontologique**

#### **II.3.1. Définition**

Avant de passer outre, il est utile de rappeler que l'ontologie est une branche de la philosophie qui traite de l'être indépendamment de ses déterminations particulières (Larousse). Ces travaux de Grüber et son équipe à Stanford [63, 64] introduisent ce terme dans l'univers du traitement automatique de l'information.

Selon Grüber, "une ontologie est une spécification formelle et explicite d'une conceptualisation partagée". Cette définition propre à Gruber est produite dans le contexte du partage et de la réutilisation des connaissances formule une ontologie via les concepts clés suivants :

- ✓ La conceptualisation qui se réfère à un modèle abstrait de certains phénomènes du monde en essayant d'identifier les concepts judicieux, qui dénotent du bon sens de ces phénomènes. Cette conceptualisation est en fait une vue abstraite et particulière concernant les entités réelles, les événements et leurs relations réciproques.
- ✓ Le type des concepts utilisés et les contraintes liées à leur utilisation sont explicitement définis.
- ✓ Une conceptualisation formelle se réfère au fait qu'elle doit être compréhensible pour les machines.

- ✓ La conceptualisation partagée comprend la notion de connaissance consensuelle décrite par l'ontologie, autrement dit, elle ne se limite pas au point de vue de certains individus seulement mais concerne un point de vue plus général accepté et partagé par tout un groupe.

Une ontologie contient donc Selon [65], une description de plusieurs types de concepts ainsi que la spécification de leurs attributs et des relations qui les relie (axiome ou relation). Le langage utilisé pour décrire les concepts ou attributs de concepts (les termes) d'une ontologie aura un impact direct sur le niveau de formalisme de cette même ontologie.

On distinguera donc les ontologies informelles utilisant le langage naturel et pouvant même coïncider avec les terminologies, les ontologies semi-formelles fournissant une faible axiomatisation tout comme les taxonomies et les ontologies formelles, qui définissent clairement la sémantique des termes par une axiomatisation rigoureuse et complète. De ce fait, et en fonction de leur utilisation, on distingue généralement quatre catégories d'ontologies [66] :

1. Les ontologies de domaine qui spécifient un point de vue sur un domaine particulier, ici la spécificité du vocabulaire, est liée à un domaine de connaissances génériques (médecine, justice..), ces concepts d'ontologie de domaine sont souvent définis comme une spécialisation des concepts des ontologies génériques.
2. Les ontologies génériques qui décrivent des concepts généraux, indépendants d'un domaine ou d'un problème particulier. Ces ontologies génériques sont aussi appelées modèles de haut niveau ou ontologies top (p.ex. Sowa [67], CYC [68]). Il s'agit par exemple des concepts de temps, d'espace, d'évènements....
3. Une ontologie légère qui comprend des concepts de types atomiques avec une hiérarchie IS-A entre concepts et des relations entre concepts.
4. Une ontologie riche qui comprend des contraintes de cardinalité, une taxonomie de relations, des axiomes et qui suppose donc l'existence d'un système d'inférence.

### II.3.2. Rôle et apport des ontologies

Les ontologies permettent essentiellement de résoudre les problèmes d'hétérogénéité sémantique entre les applications. Leur utilisation permet aussi d'exprimer de manière suffisamment claire et précise la sémantique inhérente aux applications de façon à permettre aux systèmes de coopérer sans faire de confusion avec les objets, les concepts, les appels, les méthodes,...etc. qui proviennent de systèmes différents [69].

Les ontologies ont naturellement été employées dans différents domaines et pour divers objectifs. Selon Uschold, leurs utilisations les plus répandues sont classées en trois catégories : **la communication, l'interopérabilité et l'ingénierie des systèmes** [86]. Les utilisations des ontologies ont donné entière satisfaction au sein des systèmes à bases de connaissances. Elles ont également porté leurs fruits au sein du Web sémantique. Nous résumerons les apports des ontologies comme suit :

*La communication* : elle peut avoir lieu entre les hommes et/ou les systèmes. Dans ces cas précis, les ontologies assurent alors le partage de la compréhension et la communication dans des contextes donnés, et selon les besoins. A cet effet, on peut donc utiliser l'ontologie pour créer un vaste réseau de relations qui définira les connexions entre les différents composants du système. Cette caractéristique de communication est rendue possible grâce à la non-ambiguïté des termes utilisés et définis par l'ontologie dans les systèmes.

*L'interopérabilité* : elle a lieu lorsque différentes organisations ont un besoin crucial de communiquer et d'échanger de l'information pour atteindre un objectif donné. Les ontologies contribuent donc à faciliter la compréhension et l'interprétation des informations échangées tout en se présentant comme un format d'échange. Cependant, ce point sera plus détaillé dans le chapitre suivant.

*L'ingénierie des systèmes* : dans ce cas de figure les ontologies jouent un rôle important sur trois aspects : la spécification, la fiabilité et la réutilisation.

- Une ontologie aide à définir les spécifications d'un système et à analyser ses besoins. Son rôle est dépendant du degré de la formalisation et de l'automatisation de la méthode de spécification. Cependant, dans le domaine de l'aspect informel, elles facilitent l'identification des besoins du système logiciel. Elles contribuent aussi à la compréhension des liens et des relations entre les composants de ce système. Quant à l'aspect formel, elles définissent la spécification déclarative du système.
- Les ontologies informelles : elles assurent et améliorent en même temps la fiabilité des systèmes logiciels en servant de base essentielle pour la vérification manuelle de la conception. Elles permettent aussi un rôle important dans la vérification semi-automatique du système en respectant, bien entendu, la spécification déclarative des différents composants du système.
- Cependant, devant les problèmes rencontrés quant à la réutilisation des ontologies lorsque les systèmes logiciels sont appliqués dans de nouveaux domaines, la parade ou parmi les solutions proposées dans [15], on relève la construction d'une librairie d'ontologies réutilisables et adaptables aux différents environnements. On rappelle pour cela que les ontologies satisfont seulement les applications d'origine, mais non les nouvelles, d'où la création de cette librairie d'ontologies réutilisables.

Dans un domaine de système à base de connaissances, et pour bien décrire les objets, leurs propriétés et leurs relations, on utilise nécessairement les ontologies qui aident à la représentation de ces connaissances. Cependant, l'un des objectifs principaux du *Web sémantique* reste l'échange des ressources entre machines, dans le but de permettre l'exploitation de grands volumes de services et d'informations. Ce sont donc les ontologies qui jouent un rôle prépondérant puisqu'elles permettent la réalisation du Web sémantique. Elles fournissent des vues structurées exploitables et partageables des ressources et permettent de définir une sémantique formelle pour l'information et le domaine. En exemple Le "E-commerce" est une des applications les plus répandues des ontologies. Enfin, elles sont utilisées en général pour essayer de déterminer les index conceptuels qui décrivent les ressources sur le Web.



### II.3.3. Méthodologies de construction d'ontologies

Ce sont habituellement les ingénieurs ontologiques (également appelés les ontologistes) qui se chargent du développement des ontologies qui est **somme toute** une activité de modélisation. Ces ontologistes possèdent la compréhension suffisante du domaine. Outre, qu'ils dominent les langues de représentation de la connaissance, l'aide des experts du domaine est non seulement nécessaire mais exigée pour la construction et la validation des ontologies.

Il faut reconnaître néanmoins que construire une ontologie est une activité longue mais surtout complexe. De ce fait, les méthodologies qui ont émergé, se sont basées sur les expériences accumulées dans la construction de grandes ontologies. Elles visent à faire du développement des ontologies beaucoup plus un processus de technologie plutôt qu'un art. Un bref aperçu très récent des méthodologies de développement d'ontologies est présenté dans [70]. Cependant des méthodologies plus complètes pour le développement d'ontologie sont récapitulées dans le Tableau 2-1, à l'exception de la méthode de capture d'ontologie IDEF5 [71] qui est quand même une partie de la méthodologie de modélisation d'entreprise complète IDEF (Integrated Computer Aided Manufacturing DEFinition), les autres méthodologies résultent des projets importants dans le but de construire de grandes ontologies :

- Projet TOVE (Toronto Virtual Enterprise) project [72] ;
- L'ontologie Enterprise [72] ;
- La méthode Methontology [73].

Les étapes essentielles qui peuvent être dérivées de ces méthodologies comprennent ce qui suit :

- Identification du travail et/ou de la tâche pour laquelle l'ontologie est développée ;
- Définition des objectifs et des portées (soit les besoins de l'ontologie) ;
- La construction d'une spécification informelle des concepts ou spécification informelle ;
- Codage ou représentation formelle des concepts et des axiomes dans un langage ;
- Evaluation de l'ontologie.

Malgré leur extensibilité et leur maintenance, les ontologies ne sont considérées par la plupart des auteurs une partie des méthodologies.

TOVE	Enterprise	Methontology	IDEF5
Scenarios de motivation	Objectif: niveau de formalité	Objectif et portée	Objectif et besoins
Besoins (questions informelles de compétence)	Objectif: masse d'information i.e. liste des concepts appropriés	acquisition de connaissance par l'analyse des textes et des interviews d'expert	Objectif: limites de l'ontologie collection des données basée sur les techniques d'acquisition de
specification formelle des objects, attributs and relations	definition formelle des termes	Conceptualisation : spécification des concepts, instances, relations basée sur une représentation informelle	Analyse de données : produire une liste des objets.

Besoins formels (questions formelles de compétence)		Intégration à partir des autres ontologies implémentation formelle dans un langage de représentation	Ontologie initiale Développement: définition des "proto-concepts" dans un langage schématique
specification des axiomes	Formal specification of axioms		raffinement de l'ontologie dans un langage plus structuré basé sur KIF.
Evaluation	Evaluation	Evaluation + documentation	Teste and validation utilisant les données actuelles.

**Tableau II-2:** Comparaison des méthodologies de construction d'ontologies

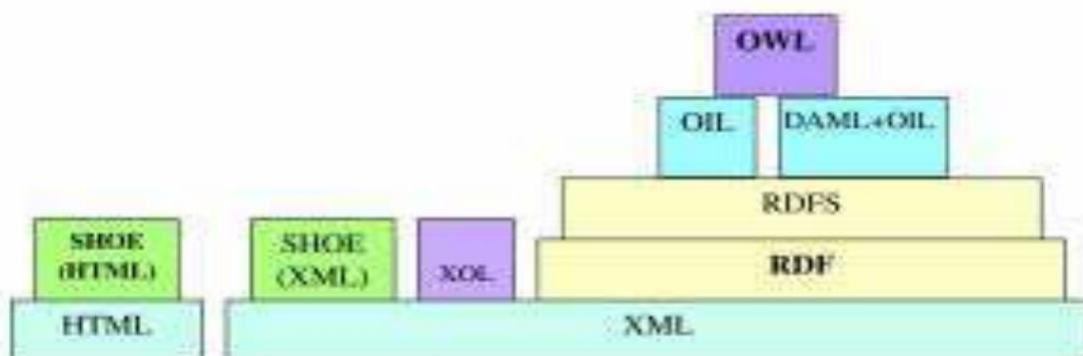
### II.3.4. Langages de représentation et outils de développement des ontologies

Vu la prolifération des ontologies dans de nombreux domaines et leur importance avérée, l'on assiste jusqu'à nos jours et cela depuis les années 90, à la naissance d'une série de langages de descriptions d'ontologies et peu même temps à de nombreux outils pour leur construction. Dans ce qui suit, on vous présentera quelques langages et outils.

Dans [74], [75], [76] et [77], se trouve une description dans la littérature abondante des listes exhaustives ainsi que des comparaisons de ces langages et outils.

#### II.3.4.1. Des langages de représentation des ontologies

Les langages de représentation des ontologies basés Web sont résumés par la figure II-1 [74]. Nous présenterons succinctement, dans ce qui suit, une revue de ces langages.



**Figure II-1:** Pile des langages de représentation des ontologies

**RDF et RDFS :** Resource Description Framework (RDF) est en fait une proposition et/ou recommandation du W3C pour mettre en relief les métadonnées [78]. Il reste un langage formel permettant d'affirmer des relations entre des "ressources". RDF est utilisé pour annoter des textes ou des documents écrits dans des langages non structurés, ou bien on utilise aussi comme une interface pour des documents écrits dans des langages qui ont une sémantique équivalente (p.ex. bases de données) [79]. RDF décrit les ressources en exprimant des propriétés tout en leur attribuant des valeurs. Ainsi et pour cela, il utilise le vocabulaire défini par RDFS [80]. RDF Schéma [88] a donc pour but **d'étendre le langage RDF** tout en fournissant un ensemble

de primitives simples. Lesdites primitives concernent la structuration de la connaissance d'un domaine en propriétés et sous-propriétés, en classes et sous-classes, tout en laissant la possibilité de limiter leur domaine d'origine (rdf:domain), et leur domaine d'arrivée (rdf:range). Enfin, tout en définissant la sémantique des nouveaux mots-clés, RDFS s'écrit toujours à l'aide de triplets RDF.

**SHOE** : SHOE (Simple HTML Ontology Extensions) : comme son nom l'indique, c'est une extension du langage permettant aux auteurs de pages Web de générer une annotation de leurs documents, compréhensible par la machine 9. Il n'en demeure pas moins que ce langage peut être utilisé par d'autres agents pour la gestion des pages Web [57]. En effet, autant ce langage HTML est inadapté pour permettre une lisibilité à un système informatique, autant il est utilisé pour rendre la connaissance facilement lisible pour un humain. De ce fait, un agent chargé d'extraire la sémantique d'un document donné, aura beaucoup de difficulté à le faire, car les données et leur présentation sont entremêlées. Ainsi SHOE évitera ce problème grâce à sa propriété d'inclusion des données qui sont directement lisibles et facilement exploitables dans les pages Web.

**XOL** : Le XOL ou XML Ontology Language : c'est le centre AI de SRI qui l'a développé en 1999. Le **XOL** permet en effet de spécifier des taxonomies, des concepts, ainsi que des relations binaires. Cependant ce langage ne permet pas d'effectuer des raisonnements.

**OIL** : Dans la perspective d'une utilisation d'ontologies sur le Web, le langage RDF(S) a été très enrichi par l'apport du langage OIL (Ontology Interchange Language) qui permet donc d'exprimer une sémantique à travers le modèle des frames et tout en utilisant la syntaxe de RDF(S). OIL offre donc de nouvelles primitives qui donnent la possibilité de définir des classes à l'aide de mécanismes ensemblistes tirés des logiques de description (intersection de classes, union de classes, complémentaire d'une classe). Il permet aussi d'affiner les propriétés de RDF(S) en contraignant la cardinalité ou en limitant la portée [81]. Ainsi a-t-on fusionné le langage OIL avec DAML pour former le DAML+OIL. DAML (Darpa Agent Markup Language5) a été conçu pour permettre l'expression d'ontologies dans une possible extension du langage RDF. Il offre les primitives usuelles d'une représentation à base de frames et utilise la syntaxe RDF [82]. Donc l'intégration de OIL a rendu possible les inférences compatibles avec les logiques de description, et essentiellement les calculs des liens de subsomption.

**DAML+OIL** : d'innombrables ont été faits dans le domaine de la représentation des connaissances parmi ces travaux, on peut citer les plus importants tels : SHOE, OntoBroker [84], OIL [85], ou encore DAML+OIL [83] qui a entretemps remplacé DAML-ONT6. DAML+OIL reste un langage construit sur des normes précédentes du W3C, telles que RDF et RDF Schéma, et étend ces langages avec des primitives de modélisation plus riches. DAML+OIL a été créé à partir du langage d'ontologie DAML-ONT (DARPA Agent Modelling Language-Ontology, Octobre2000), dans l'optique de combiner plusieurs composants du langage OIL [85]. OIL7 «*Ontology Inference Language*» est une représentation basée uniquement sur le Web, avec une couche d'inférence pour des ontologies. OIL7 combine donc les primitives de modélisation des langages à base de cadres

(frames) avec la sémantique formelle et le raisonnement fournis par la logique de description.

**OWL** : Alors que RDFS ne permet pas d'exprimer certaines notions que l'on voudrait décrire avec les ontologies, et dans le but d'étendre son expressivité de RDFS, le groupe de travail sur les ontologies du W3C, ont proposé le Ontology Web Language (OWL), basé sur le langage DAML+OIL [90] et [8]. L'OWL définit un vocabulaire qui permet de décrire des ontologies, tout en donnant des possibilités de description plus riches. Nous citons parmi les notions importantes apportées par OWL, les propriétés d'équivalence, d'identité de deux ressources, de différences entre deux ressources, de contraire, de symétrie, de transitivité et de cardinalité. Ces propriétés permettent l'utilisation de raisonneurs dans le but de déterminer des relations d'équivalence ou de subsumption entre des concepts de domaines de connaissances et d'évaluer automatiquement la compatibilité entre différentes représentations sémantiques. Il existe trois sortes d'OWL [88] :

- OWL Lite : Il s'agit-là d'une version d'OWL aux fonctionnalités réduites, mais suffisantes pour bien des usages, telles la constitution de taxonomies ou de thesaurus ;
- OWL DL : Cette version inclut OWL lite et correspond exactement aux logiques de description ;
- OWL Full : tout en incluant OWL DL, cette version donne à l'utilisateur une expressivité maximale [86] et [87].

#### II.3.4.2. Environnements et outils de modélisation des ontologies

Comme il a été déjà signalé, il existe un grand nombre d'outils de construction d'ontologies. Cependant, dans le cadre de cette thèse, nous vous présenterons seulement quelques-uns tels l'Ontolingua, l'Ontosaurus, le Tadzebao, le WebOnto, et Protégé 2000 [74], [75], [89] et [90].

**Ontolingua** : Il s'agit d'un mécanisme permettant aux utilisateurs de créer et de manipuler des ontologies. Ce mécanisme supporte les ontologies portables pour pouvoir être traduites dans différents systèmes. Ontolingua est basé sur le langage le Knowledge Interchange Format (KIF), Autrement dit sur le langage des connaissances. Il est conçu pour l'échange de connaissances entre des systèmes informatiques répartis. La syntaxe et sémantique utilisées dans KIF dans [43], ont été introduites par Thomas Gruber. En outre, l'ontologie permet aussi de traduire des ontologies génériques en LOOM, KIF, etc...

**Ontosaurus** : Créé et développé au début des années 90 par Information Science Institute de l'Université de Southern California. Ontosaurus consiste en un serveur qui utilise LOOM comme langage de représentation des connaissances, et en un autre serveur dit de navigation créant dynamiquement des pages HTML qui affichent la hiérarchie de l'ontologie. Ainsi, pour permettre à l'utilisateur d'éditer l'ontologie, le serveur utilise des formulaires HTML. Enfin des traducteurs du LOOM en Ontolingua, KIF, KRSS et C++, ont été développés.

**KAON** Basé à Java, et destiné à la conception, au développement et à la gestion d'ontologie, le Karlsruhe Ontology and Semantic Web est un environnement open source modulaire. Ici l'environnement intégré les modules suivants : API, Query, Serveurs (d'ontologie et

d'application), Générateur de portails web (basés sur les ontologies), Éditeur d'ontologie (construction et maintenance).

**WebOnto** : WebOnto et TADZEBAO sont des outils complémentaires développés en 1997 par le Knowledge Media Institute de l'Open University ; Alors que le TADZEBAO permet aux ingénieurs des connaissances de tenir des discussions sur les ontologies, en mode synchrone et asynchrone. Le WebOnto supporte quant à lui la navigation collaborative, la création et l'édition d'ontologies sur le Web.

**Protégé 2000** : Appartenant au département d'Informatique Médicale de l'Université Stanford, le Protégé-2000 successeur de PROTEGEWIN, est un outil, une plateforme et une librairie d'ontologies, qui permettent :

- de construire une ontologie du domaine ;
- de personnaliser des formulaires d'acquisition de connaissances ;
- de transférer la connaissance du domaine ;

Cependant Protégé 2000 est aussi une librairie Java qui peut être étendue et élargie afin de créer de véritables et réelles applications à bases de connaissances en utilisant un moteur d'inférence pour raisonner et déduire de nouveaux faits en appliquant des règles d'inférence aux instances de l'ontologie et à l'ontologie elle-même (méta-raisonnement) [91]. Il est à noter que des «plugin» pour les langages RDF, DAML+OIL et OWL ont été développés pour Protégé 2000. Des outils peuvent être intégrés à Protégé, exemple : un moteur d'inférence tel que JESS [92], ou des outils d'inférence spécifiques au Web sémantique basés sur des logiques de description [93] tel que RACER [94].

### **II.3.5. Approches d'utilisation des ontologies :**

On admet en général l'existence de trois approches d'utilisation des ontologies [95] qui sont : l'approche mono-ontologie, l'approche multi-ontologie et l'approche hybride.

**II.3.5.1. Des Approches mono-ontologie (mono-domaine)** : ces approches n'utilisent qu'une seule ontologie globale qui fournit un vocabulaire commun pour la spécification de la sémantique. Toutes les sources de données sont directement reliées à cette ontologie, permettant ainsi d'identifier et reconnaître les règles de correspondances entre les objets des différentes sources de données. Cette approche est très intéressante quand les différents SI donnent presque la même vue sur le domaine d'application. Mais si parmi les SI participants, il y a certains qui possèdent une vue plus précise ou plus détaillée sur le domaine, la construction d'une ontologie devient alors très difficile.

**II.3.5.2. Approches multi-ontologies (multi-domaines)** : Chaque SI, dans cette approche, est décrit par une ontologie locale pour traduire et exprimer le contexte d'utilisation de ses informations (système OBSERVER) [96]. Les avantages essentiels de ces approches sont non seulement l'autonomie dans la construction de l'ontologie du site local mais aussi l'extensibilité de la solution face à l'ajout et le retrait des SI. Cependant et vu l'absence d'un vocabulaire commun, les ontologies locales élaborées de manière autonome et indépendante peuvent présenter beaucoup de divergences dans la représentation de la sémantique ; ce qui compliquera

l'identification des correspondances entre les concepts des ontologies locales : donc l'identification des correspondances entre les objets des différentes sources de données sera aussi difficile. Cependant et pour définir les correspondances entre les concepts des ontologies locales et ainsi résoudre ce problème, ces approches ont prévu des relations inter-ontologies (mappings) sachant que les ontologies locales correspondantes ont été construites sans vocabulaire commun, et dans le cas où les SI ont des vues différentes sur le domaine d'application, la construction des relations inter-ontologies (mapping) revient donc à résoudre les problèmes d'hétérogénéité sémantique entre les différentes ontologies.

### II.3.5.3. Approches hybrides

Pour pallier aux insuffisances et aux inconvénients des approches mono-ontologie et multi-ontologies, des approches hybrides ont été créées et développées. Ces approches hybrides ressemblent aux approches multi-ontologies, sauf que la sémantique de chaque source de données est décrite par une ontologie locale. Ces approches hybrides ont cette particularité qui consiste à rendre les ontologies locales facilement comparables, et ce en les construisant sur un vocabulaire global commun [97] et [98]. Ce vocabulaire commun contient les limites de base (les primitifs) d'un domaine. Il peut être lui aussi une ontologie.

### II.3.5.4. Evaluation des approches ontologiques

	<b>Mono-ontologie</b>	<b>Multi-ontologies</b>	<b>Hybride</b>
<b>Hétérogénéité sémantique</b>	Vue similaire sur le domaine	Existence de l'hétérogénéité sémantique	Existence de l'hétérogénéité sémantique
<b>Ajout/retrait des sources de données</b>	Changements au niveau de l'ontologie globale	- Ajout d'une nouvelle ontologie - Création de relations avec les autres ontologies	Ajout d'une nouvelle ontologie
<b>Identification des correspondances inter-ontologies</b>	-	Très difficile en l'absence d'un vocabulaire commun	Simple grâce à un vocabulaire commun

**Tableau II-03:**Evaluation des approches ontologiques [95]

Ce tableau fait ressortir clairement, que l'approche hybride est en fait plus avantageuse grâce à l'ajout simple de nouvelles sources, sans nécessité de modifications. Aussi, supporte-t-elle l'acquisition et l'évolution des ontologies, et l'utilisation d'un vocabulaire partagé qui rend donc les ontologies sources comparables [95]. Les approches à ontologies multiples, et les approches hybrides peuvent cependant être confrontées au problème d'hétérogénéité ontologique [99]. Pour cela, il existe plusieurs autres approches telles que : le mapping, l'alignement, la transformation et la fusion d'ontologies [100]. L'interopérabilité revient donc à obtenir une médiation entre celles-ci, de ce fait nous opterons pour la technique de mapping ontologique, qui reste une solution faiblement couplée et apporte une certaine flexibilité quant à l'interopérabilité des systèmes d'information.

### II.3.6. Approches de mapping des ontologies

On peut définir le mapping des ontologies comme étant un ensemble de correspondances, au sens mathématique entre un certain nombre d'ontologies [101]. Le mapping repose sur la définition des relations de correspondance entre les entités de deux ontologies qui présentent des similitudes.

La principale spécificité 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, en général, les différentes étapes suivantes [102] :

- L'import des ontologies consistant à charger les ontologies dans l'outil de mapping en effectuant éventuellement des translations de format ;
- La recherche de similarité qui consiste à chercher et à trouver et de façon semi-automatique les similarités qui peuvent exister entre les entités des deux ontologies ;
- La spécification des mappings, consistant à spécifier les mappings et se faisant aussi de façon semi-automatique en utilisant un outil tel que PROMPT [103].

## II.4. De l'apport des SMA pour la médiation sémantique

Les SMA ou systèmes multi-agents sont des systèmes constitués d'un ensemble d'entités libres ou autonomes appelées agents. Ces SMA peuvent en l'occurrence apporter des réponses consistant à maîtriser la complexité des systèmes distribués. Ce domaine des SMA est né à l'intersection de domaines tels que l'intelligence artificielle distribuée, la représentation des connaissances, la théorie des actes de langages ou encore la vie artificielle. Dans cette petite partie, nous allons présenter les systèmes multi-agents et ses contributions dans la médiation sémantique des systèmes d'information.

### II.4.1. De la généralité sur les SMA

Les systèmes multi-agents ont connu durant cette quinzaine d'années un grand essor et sont utilisés dans des domaines très variés tels que le domaine de la simulation et de la vie artificielle, de la robotique, le traitement d'images, les bases de données et les bases de connaissances distribuées coopératives, applications distribuées comme le web, etc...

Les systèmes multi-agents appartient à l'intelligence artificielle distribuée (IAD) qui elle-même une branche de l'intelligence artificielle, ces systèmes multi-agents offrent donc une approche pour s'intéresser aux systèmes composés de plusieurs éléments qui interagissent fortement entre eux et aussi sur leur environnement. Décentralisée, leur démarche consiste à les rendre surtout adaptés pour ce type de systèmes. De fait, la programmation orientée objet n'est pas toujours adaptée aux besoins des applications d'aujourd'hui. Ce qui fait que les systèmes récents sont fréquemment distribués sur plusieurs machines, ils communiquent entre eux et interagissent tout en s'exécutant indépendamment les uns des autres. Ces systèmes récents sont souvent divisés en sous-systèmes indépendants qui font chacun une partie de travail mais dans un but commun. La programmation agent permet une façon bien plus naturelle d'analyser, de concevoir et de programmer ce type de système [104]. L'ensemble des thèmes SMA (ou la thématique) se focalise sur l'étude des comportements collectifs mais aussi sur la répartition de

l'intelligence entre des agents plus ou moins autonomes et capables de s'organiser. Nous expliquerons plus en détails dans ce qui suit la notion d'agent.

#### II.4.2. Différents Types d'agent

Il existe plusieurs classifications des agents dans le vocabulaire des SMAs. Selon les références [105], [106], la taxonomie ou classification scientifique et systématique la plus complète des agents comprend :

**Des agents collaboratifs :** Ils sont utilisés comme agents exécutants pour diverses tâches et interagissent l'un sur l'autre pour répondre l'information et manipuler des conflits dans des environnements multi-agents ouverts. Ils accomplissent donc les tâches pour la résolution des problèmes. Ils peuvent aussi apprendre, bien que cet aspect ne soit pas typiquement leur rôle principal ou opération principale. Les agents collaboratifs, pour coordonner leurs activités, négocient dans le but d'arriver à des accords mutuellement acceptables. Ce type d'agents permet de [107] :

- ✓ La résolution de problèmes qui sont trop vaste et compliqués pour un agent simple centralisé, à cause des limitations des ressources ou en raison du risque d'avoir un système centralisé ;
- ✓ D'agir par l'intercommunication et par l'interopération des systèmes anciens existants (legacy system) ;
- ✓ De fournir des solutions fiables, aux problèmes de nature distribués.

**Les agents Interfaces :** Ils proviennent des domaines de l'interface homme-machine et de l'intelligence artificielle. Ils ont pour but de simplifier la vie de l'utilisateur en automatisant des tâches réalisées couramment par le passé à partir de l'observation des comportements répétitifs ou en surveillant des ressources tel le courrier électronique.

**Les agents Mobiles :** Ils proviennent directement du domaine des réseaux et dont les objectifs sont la réalisation de tâches réparties sur un ensemble de machines qui sont interconnectées pour le compte d'un utilisateur ou d'une application donnée. Ces agents mobiles sont des agents capables de se déplacer à l'intérieur de son environnement, et ils possèdent donc des dispositifs qui assurant leurs mobilités.

**Des agents d'Information/Internet :** Ils sont surtout conçus pour supporter des situations de surcharge d'information. Ces agents sont responsables non seulement de la gestion, de la manipulation de l'information mais aussi de sa collecte (ou rassemblement) à partir de plusieurs sites distribués. Ils sont capables de filtrer de grandes quantités d'informations et de choisir des données appropriées [107].

**Les agents hybrides :** Ces architectures hybrides mélangent des composants réactifs et délibératifs dans le but de produire un modèle plus puissant.

**Les agents hétérogènes :** Ce sont des systèmes d'agents hétérogènes faisant référence à une organisation qui contient au moins deux et/ou plus d'agents appartenant à deux et/ou plus classes différentes. Un système d'agents hétérogènes peut lui aussi contenir aussi des agents hybrides. Ces différents agents peuvent donc intercommuniquer en utilisant un langage de communication d'agent (ACL).



**Agents réellement intelligents** : Ils représentent la forme idéale des agents intelligents et autonomes avec un modèle interne spécifique capable de raisonnement. Des agents réellement intelligents sont des agents, qui incluent toutes les formes d'agents mentionnées au-dessus [107].

#### II.4.2.1. Du système multi-agents

Il s'agit d'un système distribué composé d'une multitude d'agents interagissant. Ces SMA sont conçus et implémentés comme un seul ensemble d'agents, le plus souvent, selon des modes de *coopération*, de *concurrence* ou de *coexistence* [108]. Enfin, selon [109], un SMA est généralement caractérisé par le fait :

- 1- Que chaque agent possède des informations ou des capacités de résolution de problèmes limitées, et ainsi chaque agent a un point de vue partiel ;
- 2- Qu'il n'y a aucun contrôle global a priori du système multi agents ;
- 3- Que les données sont décentralisées ;
- 4- Que les calculs sont asynchrones.

Les SMA restent quand même des systèmes idéaux pour représenter des problèmes aux multiples méthodes de résolution, aux multiples perspectives. Enfin ces SMA sont aussi des systèmes idéaux pour représenter des problèmes possédant de multiples solveurs. Ces SMA ont les avantages traditionnels de la résolution distribuée et concurrente de problèmes telles la modularité, la vitesse, et la fiabilité [108]. Ils recueillent également des bénéfices envisageables de l'Intelligence Artificielle, comme par exemple le traitement symbolique, la facilité de maintenance, la réutilisation et la portabilité, mais ils sont surtout l'avantage de faire appel à des schémas d'interaction sophistiqués [109]. Les types courants d'interaction incluent [108] :

- ✓ La coopération : œuvrer ensemble à la résolution d'un but commun ;
- ✓ La coordination : afin d'éviter les interactions nuisibles, il faut organiser la résolution d'un problème : c'est aussi bien exploiter les interactions bénéfiques ;
- ✓ La négociation : passer par la négociation pour arriver à un accord qui soit accepté par toutes les parties concernées.

#### II.3.2.2. De coopération entre agents

La coopération entre agents est une caractéristique très importante dans les SMA. Le résultat de l'interaction coopérative entre les différents agents est une résolution distribuée d'un problème. Voici les méthodes de coopérations telles que citées dans la littérature [110] et [111] :

- **Le regroupement et la multiplication** : Les agents représentent un bloc commun composé de l'ensemble des individus, ce qui leur permet de faire des actions impossibles par un agent seul.
- **La communication** : Les agents définissant d'abord les protocoles de communication avant de coopérer ; ainsi les agents pourront se comprendre et échanger des informations.
- **La spécialisation** : Pour cela, elle repose sur le fait que chaque agent se spécialise à un seul et unique type de tâche spécifique.

- **La collaboration par partage des tâches et des ressources** : Il existe plusieurs méthodes pour distribuer les tâches. On cite pour cela les mécanismes d'élection où les tâches sont données à des agents après un accord ou un vote. On cite aussi les réseaux contractuels où des tâches sont attribuées aux agents après bien sûr des cycles d'appels d'offres ou de propositions. Cependant la planification multi-agents attribuée aux agents responsables de cette planification, la responsabilité de la distribution des tâches et la structure organisationnelle où les agents assument des responsabilités pour des tâches particulières.
- **La coordination d'action** : Pour assurer un fonctionnement cohérent du système, Il faut coordonner les actions.
- **La résolution de conflit par arbitrage et négociation** : les agents sont tenus de coopérer efficacement s'ils sont capables de résoudre le conflit et empêcher que des désaccords entre personnes ne se transforment en lutte ouvertes et entraîner ainsi la dégradation du système entier ainsi que de ses performances.

### II.4.3. Communication entre agents

Comme nous l'avons déjà montré les systèmes multi-agents constituent une approche de résolution distribuée des problèmes. Ainsi dans un SMA, l'expertise nécessaire pour le traitement des problèmes dans le domaine d'application est distribuée entre les agents du système. Ces agents communiquent et interagissent pour coordonner non seulement leurs actions, mais aussi pour résoudre des conflits (de ressources, de buts ou d'intérêts) mais aussi pour s'entraider mutuellement. Ces SMA utilisent à cet effet le langage ACL (Agent Communication Language) pour la communication (échange d'information, de connaissances ou encore de services) entre agents. C'est la communication inter agents qui fait qu'un groupe d'agents forme un SMA. Le format qui est utilisé pour l'échange des connaissances est fourni par un langage de contenu, tout à fait indépendant du langage ACL (par exemple : KIF, FIPA-SL, Prolog, Clips, .. etc.). Cependant le vocabulaire commun entre les agents concerne les définitions précisées dans une ontologie.

**II.3.3.1. Des modèles de communication** : Nous relevons essentiellement deux types de modèles de communication entre agents : la communication par envoi de messages et la communication par partage d'informations.

**1- La communication par envoi de messages** : Ce type de communication est spécifique aux agents possédant une représentation propre et locale de l'environnement qui l'entoure. Chaque agent va alors interroger directement les autres agents sur cet environnement ou leur envoyer des informations sur sa propre perception des choses. La transmission de l'information est faite sous deux modes, soit le mode du point à point et le mode par diffusion.

- **Dans le Mode point à point**, l'agent émetteur du message connaît avec précision l'adresse de ou des agent(s) destinataire(s). Ce genre de communication est généralement le plus utilisé dans les systèmes à agents cognitifs.

- **Dans le Mode par diffusion**, le message est envoyé à tous les agents du système. Ce type de transmission est très utilisé dans les systèmes ainsi que dans les systèmes d'agent réactif.

- 2- La communication par partage d'information :** Cette forme de communication se fait via le tableau noir (blackboard). Ce tableau noir fournit un environnement pour échanger les données, qui sont écrites dessus par les agents participants. Cet accès aux informations et aux messages signalés sur le tableau noir est ouvert à n'importe quel agent. Ce genre de communication est utilisé quand il y a recouvrement des domaines d'expertise de chaque agent. Cela suppose aussi que les agents ne possèdent qu'une connaissance limitée sur les domaines d'activité des autres agents. Ce type de communication est l'un des plus utilisés dans la conception des systèmes multi-agents.

L'architecture d'un système à base de tableau noir comprend 03 sous-systèmes.

- Les sources de connaissance (agents).
- La zone partagée « blackboard » cette zone contient les résultats partiels des problèmes en cours de résolution ainsi que les hypothèses qui spécifient à un moment donné quelles sont les sources de connaissances qui peuvent aider à l'avancement ou à la progression de la résolution du problème, et d'une manière générale toutes informations que s'échangent les agents.
- Un dispositif de contrôle qui gère les conflits d'accès entre agents.

**II.3.3.2. Du langage de communication :** on relève plusieurs et différents langages de communication entre agents :

**Les Langages de communication entre agents (ACL) :** Dans le but de faciliter la communication et l'interopérabilité entre les agents, de nombreux langages de communication ont été développés. Ces langages occupent une place dans la couche logiquement supérieure à celle des protocoles de transfert (TCP/IP, HTTP, IIOP) et adressent le niveau intentionnel et social des agents [112].

Les deux ACL les plus utilisés sont KQML (Knowledge Query and Manipulation Language) et FIPA-ACL (FIPA Agent Communication Language) qui est en fait une extension du langage KQML [113].

**Langage de communication KQML :** Le Knowledge Query and Manipulation Language (KQML) est né en fait qu'en 1992 et n'a atteint sa maturité qu'en 1993. Il est basé sur un ensemble d'actes de langage standards et surtout utiles pour que les agents cognitifs puissent coopérer. KQML repose principalement sur la séparation de la sémantique liée au protocole de communication (Indépendante du domaine d'application). Cependant et malgré les avantages offerts par ce langage KQML, sa sémantique n'est pas très formalisée, ce qui entraîne différentes interprétations [114] et [115]. Des efforts faits autour de ce langage, ont quand même abouti à de nouveaux langages plus fiables, tel en particulier le FIPA-ACL.

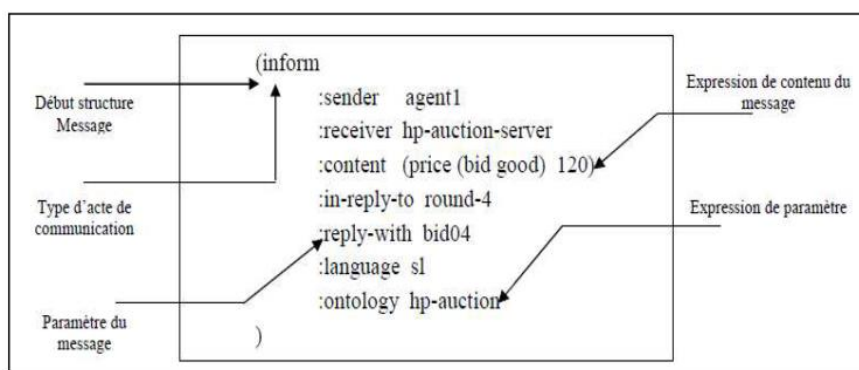
**Langage FIPA-ACL :** Développé par la Foundation for Intelligent Physical Agents en 1996, le langage FIPA-ACL est demeuré un langage standard de communication inter-agents qui repose essentiellement sur la théorie des actes de langages comme KQML. Cette fondation a défini des standards pour la structuration des messages, leur représentation et les mécanismes de leur transport. La caractéristique spécifique de ce langage repose sur un ensemble de types de messages d'une part, et de protocoles d'interaction de haut niveau d'autre part [116]. Le FIPA-ACL est constitué de quatre primitives principales : confirm, disconfirm, inform, request.

La totalité des autres actes de langage est construite sur la base de ces quatre primitives [117]. Ce langage dispose de 22 actes de langage exprimés par des performatives, et peuvent en l'occurrence être classés selon leur fonctionnalité [116] (tableau II.4).

	Transmission d'information	Demande d'information	Négociation	Action	Gestion d'erreurs
Accept-proposal			X		
Agree				X	
Cancel				X	
Cfp (call for proposal)			X		
Confirm	X				
Disconfirm	X				
Failure					X
Inform	X				
Inform-if	X				
Inform-ref	X				
Not-understood					X
Propagate				X	
Propose			X		
Proxy				X	
Query-if		X			
Query-ref		X			
Refuse				X	
Reject-proposal			X		
Request				X	
Request-when				X	
Request-whenever				X	
Subscribe		X			

**Tableau 0II-4:** Actes de communication FIPA-ACL, regroupés par catégories

Le langage FIPA-ACL qui repose sur une sémantique beaucoup plus formalisée que KQML, permet d'offrir un ensemble de protocoles de communication où chacun implique plusieurs actes de langage [124] tels que : contract net, demande d'action, ... etc. La structure d'un message FIPA-ACL comprend plusieurs éléments illustrés dans le Figure II.2.



**Figure II-2 :** Structure d'un message FIPA-ACL

#### II.4.4. De l'interopérabilité des SMA

Dans les SMA, peuvent être distingués trois niveaux d'interopérabilité : soit le niveau technologique, le niveau opérationnel et enfin le niveau sémantique [118].

### **II.4.4. 1. L'interopérabilité technique**

Elle a été longuement et largement abordée dans le cadre de l'informatique distribuée. Mais à ce niveau, le problème de la possibilité d'interagir entre les entités implantées dans des systèmes hétérogènes utilisant des langages hétérogènes et surtout des systèmes de typages hétérogènes, se pose avec acuité. Donc l'intégration des applications qui utilisent des langages et des modèles, constitue le premier vrai obstacle pour faire interagir des applications. A ce niveau, et pour résoudre le problème de l'hétérogénéité, la standardisation sera l'approche adoptée. L'utilisation d'XML et Java jouent actuellement un rôle important dans les standards émis par FIPA pour assurer l'interopérabilité technique.

### **II.4.4.2. L'interopérabilité opérationnelle**

En général, chaque entité logicielle ou chaque agent est implémenté selon un modèle opérationnel, que ce soit un objet ou un processus, une approche RPC ou orientée messages. A ce niveau donc, l'interopérabilité concerne seulement le mode de fonctionnement des entités logicielles. Deux entités sont interopérables au niveau opérationnel quand elles sont interopérables au niveau technique et qu'elles peuvent communiquer selon le mode opérationnel mutuel. L'interopérabilité au niveau opérationnel permet donc aux entités de gérer et d'exécuter toutes les interactions possibles des deux entités.

### **II.4.4.3. L'interopérabilité sémantique**

L'interopérabilité sémantique est propre aux systèmes multi-agent. En effet, au niveau opérationnel deux entités interopérables peuvent interagir sans pour autant comprendre le sens et mettre à profit leurs interactions. Ces entités interagissent, au niveau sémantique, en échangeant le sens et peuvent donc alors se rendre service mutuellement. Deux systèmes sont dits sémantiquement interopérables lorsqu'ils sont interopérables au niveau opérationnel et lorsqu'ils partagent la même référence sémantique sur l'objet de l'interaction. Au niveau opérationnel, les entités sont à même de capter le fonctionnement d'un protocole cependant, sans interopérabilité sémantique, les entités deviennent incapables d'assurer son bon déroulement en vérifiant les contraintes fonctionnelles et en construisant les messages avec le contenu conforme.

## **II.5. De l'apport de services web dans la médiation sémantique**

La technologie des Web services se présente aujourd'hui comme étant incontournable et est devenue un nouveau paradigme des architectures logicielles distribuées. Cette technologie, tout en englobant de nombreux concepts, tend à s'imposer comme le nouveau standard en terme d'intégration et d'échanges des applications au sein et entre les entreprises.

Cependant la solution orientée services reste une réponse très efficace à toutes les problématiques rencontrées par les entreprises en termes de réutilisabilité, d'interopérabilité et de réduction de couplage entre les différents systèmes qui implémentent leurs systèmes d'information. Les Web services sont devenus la forme de middleware que l'informatique

professionnelle attendant depuis des années. Cette forme est comme un middleware universel pour les applications distribuées de l'entreprise.

Nous allons donc présenter dans cette section ce nouveau concept et voir comment cette technologie peut emplir et combler les lacunes du paradigme agent dans notre solution tout en offrant des principes suffisamment génériques et souples en permettant un développement efficace et avec moins d'efforts.

### II.5.1. Des éléments de définition

Les Web services sont définis par plusieurs intervenants par des caractéristiques technologiques distinctives. Cependant la majorité des acteurs considèrent les Web services comme des composants logiciels réutilisables et faiblement couplés qui renferment sémantiquement des fonctionnalités discrètes, distribués et accessibles à tout un chacun via les protocoles Internet standards (Stencil, 2006). Ce qui veut dire que les Web services sont en fait des applications qui relient des programmes, des objets, des bases de données ou des processus d'affaires à l'aide de XML et de protocoles Internet standard. D'après Gillbert (Gillbert & al, 2003), on peut définir les Web services comme étant des compléments aux programmes et applications existantes, et de ce fait servent de pont pour que ces programmes communiquent entre eux. Nous pouvons donc considérer les Web services comme une application logicielle identifiée par un identificateur de ressource uniforme et servent de passerelle pour que ces programmes puissent inter communiquer. Nous pouvons également considérer un web service comme une application logicielle identifiée par un identificateur de ressource uniforme URI (uniforme ressource identifier). Un Web service reste une application autosuffisante en ce sens qu'elle s'occupe d'une seule tâche et que ses composantes décrivent leurs propres entrées et sorties de manière à ce que d'autres logiciels qui invoquent le Web service peuvent interpréter ce qu'il fait, comment invoquer sa fonctionnalité et à quel résultat cet autre service peut s'attendre ou espérer. Les Web services sont, entre autres, le prolongement de la programmation objet. Un Web service est donc une sorte d'objet avec une seule fonctionnalité qui permet avec d'autres services Web, la composition d'une application plus large avec la possibilité d'avoir plusieurs fonctionnalités. En fait, c'est comparable à l'une des briques d'un mur qui en comporte sur plusieurs. Un URI reste en quelque sorte la façon d'identifier un point de contenu sur le Web, que ce soit un fichier texte, audio ou vidéo. L'URI le plus connu est sis à l'adresse d'une page Web, par exemple : <http://www.cirano.qc.ca>. Donc, en spécifiant son URI, un Web service est facilement accessible.

Les Web services se caractérisent par la capacité des interfaces et liaisons (*bindings*) d'être publiées, localisées et invoquées via XML. Un Web service peut être donc publié dans un registre situé à l'intérieur et/ou à l'extérieur d'un système d'information. Il peut être localisé en interrogeant juste le registre où il est hébergé. Une fois localisé, un Web service peut être invoqué et cela même par un autre Web service pourvu, qu'on envoie une requête appropriée.

Un autre avantage des Web service est leur capacité d'interagir avec d'autres composantes logicielles à l'aide des éléments XML en utilisant des protocoles de l'Internet. L'une des bases des Web services est l'utilisation de protocoles standards de l'Internet tels que l'*Hypertext Transfer Protocol*, le protocole du Web ou le HTTP, le *Simple Mail Transfer Protocol*, le

protocole du courriel électronique, et XML. Les Web services peuvent donc traverser les pare-feu classiques (conventionnels) sans problème. Notons cependant les Web services ne sont pas destinés à une interaction humaine directe, contrairement à une page web ou à une application de bureautique. Ils sont plutôt conçus pour être utilisés par d'autres logiciels. En plus, les Web services peuvent être assimilés à des composantes logicielles couplées à interaction dynamique. Autrement dit le Web service et le programme (ou le consommateur du Web service) qui l'invoque peuvent être modifiés indépendamment l'un de l'autre. Cela signifie qu'il n'est pas nécessaire, pour qu'une communication puisse avoir lieu, de connaître la machine, le langage, le système d'exploitation ou tout autre détail. Cela permet donc une flexibilité aux entreprises pour éviter les coûts engendrés par l'intégration via des communications fortement couplées telles qu'exigées par l'EDI, par exemple. Par interaction dynamique, cela signifie que le consommateur de Web services peut facilement localiser et invoquer ce dernier au moment de l'exécution du programme sans pour autant avoir à programmer cette habileté à l'avance. Donc, on peut affirmer que l'on peut modifier le Web service sans pour autant modifier le logiciel de chacun des utilisateurs potentiels.

### **II.5.2. De l'architecture orientée service**

Un modèle de référence pour Architectures Orientées Services (AOS) [119] a été proposé par le consortium OASIS. Dans ce modèle proposé, il définit l'AOS comme un paradigme pour organiser et utiliser des capacités ou fonctionnalités distribuées qui peuvent être contrôlées par différents propriétaires. On entend par capacités une fonctionnalité métier automatique invocable à distance. Ce paradigme est basé sur trois concepts essentiels, à savoir, la visibilité, les interactions et les effets. Le premier concept ou visibilité permet ainsi aux fonctionnalités et aux besoins de se rencontrer. Cependant, les interactions consistent en l'utilisation des fonctionnalités à travers les échanges de messages. Quant à un effet est considéré comme le résultat d'une interaction. L'effet consiste donc en un changement d'état des entités informationnelles impliquées dans l'interaction en question. Ces concepts clés se retrouvent délégués au niveau d'un autre concept qui est le service considéré comme lui-même comme la brique de base de l'AOS.

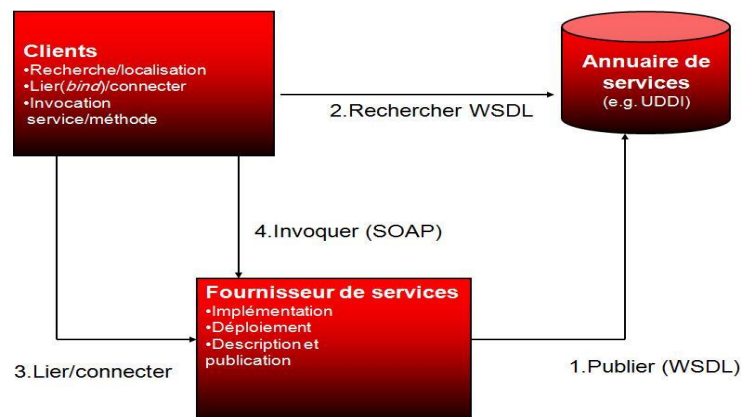
Selon [120], le service reste un moyen d'accès à des fonctionnalités offertes par un composant applicatif. Le service donne une description de son interface indépendante de toutes plateformes, il est appelé aussi service auto-descriptif. Encapsulant pratiquement toute la logique d'exécution (self-contained). Le service est autonome, complet et cohérent possédant une adresse, le service permet de l'invoquer dynamiquement donc localisable. On peut enfin dire que le service Web est l'une des déclinaisons technologiques de l'AOS.

### **II.5.3. De l'architecture de Service Web**

Il s'agit d'une instance d'architecture orientée service SOA (Cf. figure II.3). Sa définition d'architecture pour les services Web (WSA : Web Services Architecture) consiste à montrer les concepts, les relations entre ces concepts ainsi qu'à mettre en évidence un ensemble de contraintes qui assurent l'objectif premier des services Web et qui est l'interopérabilité. Ces éléments bien étendus structurés via un modèle de déploiement et de fonctionnement des

services Web. Les concepts de l'architecture ne sont pas supposés avoir une existence dans la réalisation informatique, ils désignent des personnes, des organisations ou encore une entité logicielle ou matérielle. La définition et/ou la description des concepts de l'architecture est présentée sous forme d'une identification nominale du concept, ainsi que les relations qu'il entretient avec les autres concepts. Les principaux concepts qui interviennent dans l'architecture des services Web sont :

- ✓ Le fournisseur du service qui, d'un point de vue conceptuel, désigne la personne ou l'organisme responsable juridiquement de l'agent logiciel (i.e. le service). Du point de vue opérationnel, les fournisseurs de services peuvent également sous-entendre le serveur qui héberge les services déployés.
- ✓ Le client du service représente, comme pour le client de service, une personne ou une organisation, soit un client potentiel des services. Alors que d'un point de vue opérationnel, il désigne l'application cliente qui invoque le service.
- ✓ L'annuaire des services qui lui représente la personne ou l'organisation responsable de l'hébergement ou de la publication des services. D'un point de vue purement fonctionnel, il représente l'entité logicielle qui joue le rôle de l'intermédiaire entre les clients et les fournisseurs de services. Le concept registre ou annuaire de service est très important, voire essentiel, dans l'architecture services Web.



**Figure II-03** : Architecture des services web

#### II.5.4. Technologies des Web Services

Pour assurer, voire garantir l'interopérabilité des trois concepts que nous venons d'expliquer (Fournisseurs de services, Client du service, Annuaire de publication), Plusieurs propositions de standards (technologie pour les services Web) ont été élaborées : XML ou Extensible Markup Language, le SOAP ou Simple Object Access Protocol, le WSDL Web Services Description Language ou enfin l'UDDI ou l'Universal Description, Discovery and Integration, que nous allons tenter d'expliquer ci-dessous :

**XML** : Il s'agit d'un standard permettant de décrire des documents structurés transportables sur les protocoles d'Internet, autrement dit sur toutes les règles et conventions qui régissent l'échange de données via internet. Autrement dit XML représente la technologie de base des architectures services web, puisqu'il apporte à l'architecture l'extensibilité et la neutralité par



rapport aux plates-formes et aux langages de développement. En outre et grâce à la structuration, XML fait la distinction entre les données des applications et les données des protocoles, ce qui facilite ainsi une correspondance entre les différents protocoles. L'interopérabilité entre des systèmes différents nécessite des mécanismes puissants de correspondance et de gestion des types de données des messages entre les fournisseurs et les clients. Ceci est donc une tâche où les schémas de type de données XML s'avèrent incontournables et bien adaptés.

**SOAP**, Il s'agit là d'un protocole de transport réseaux dépendant d'un format de message XML qui permet à un client d'invoquer un service distant. Le protocole SOAP donne aussi la possibilité d'invoquer un objet distant en communiquant les informations nécessaires à l'appel soit le nom de la méthode et sa signature dans un message au format XML. La réponse à la requête est aussi renvoyée encapsulée au format XML. Un message SOAP est donc une structure qui comporte des parties obligatoires, l'enveloppe (envelop) et le corps du message ainsi qu'une partie optionnelle comme l'entête (header).

**WSDL**, Il s'agit là d'un langage de description basé sur XML et qui utilise pour d'écrire l'interface d'utilisation d'un service Web. Le fournisseur de service utilisera donc un document WSDL pour indiquer les opérations données par un service Web ainsi que les paramètres et les types de données associés à ces opérations. Le service web est spécifié par le WSDL sur deux niveaux, le premier abstrait (réutilisable) et le deuxième concret que nous essayerons de développer :

1. Le niveau abstrait : Ce niveau définit et décide les opérations que le service pourra exécuter. Chaque opération représente un ensemble de messages d'entrée et de sortie, et chaque message contient une ou plusieurs données types. Un binding ou une liaison établit une correspondance entre le protocole utilisé pour communiquer, le port et l'opération. Nous rappelons que le niveau abstrait est réutilisable
2. Le niveau concret : Le document WSDL définit le service comme des collections de points finaux (ou endpoints) de réseaux appelés aussi "ports". le service communique via ces ports par échange de messages. Cependant, chaque port est relié à une liaison ou binding, laquelle détermine comment on peut accéder à l'opération en utilisant un protocole de communication particulier (SOAP, HTTP,...).

**UDDI**, Egalement basé sur XML, l'UDDI, ce protocole d'annuaire permet de stocker et/ou rechercher des informations sur des services Web. UDDI nous permet aussi d'automatiser la découverte des services Web. Autrement dit, ce protocole d'annuaire ou UDDI est dédié à la publication des services et aux informations qui concernant le fournisseur de services. UDDI assure donc le stockage, le regroupement et la diffusion des descriptions de services Web.

### **II.5.5. De l'avantage des services Web**

Derrière les Web service, la finalité principale est d'arriver à morceler les applications et les processus d'affaires mais en morceaux réutilisables appelés services, de telle manière que chacun de ces segments ou morceaux effectuent une tâche distincte. Ces services faciliteront

l'interopérabilité entre tous ces services et pourraient alors servir à l'intérieur comme à l'extérieur de l'entreprise. De par leur nature, les Web services [121] :

- Permettraient une communication facile et à moindre frais entre les différentes portions de logiciels écrites dans différents langages en évoluant sur différents systèmes d'exploitation.
- Permettraient des applications supportant différents processus d'une ou de plusieurs et différentes organisations, d'inter communiquer et/ou d'échanger des données facilement et à peu de frais. Enfin, plus particulièrement, les Web services devraient permettre aux entreprises [122].
- donner aux clients un accès direct à l'information, aux données et aux fonctionnalités dont ils auront besoin pour interagir ;
- donner aux partenaires d'une entreprise un accès direct à la fonctionnalité dont ils ont besoin afin de mieux servir les clients qu'ils ont en commun avec cette entreprise ;
- donner aux fournisseurs d'une entreprise un accès direct à l'information et à la fonctionnalité dont ils ont besoin pour leur permettre d'ajuster et/ou réajuster les inventaires ;
- de se servir et d'intégrer des applications de bout en bout, de manière abordable tout en rendant facile l'implantation autant hors des frontières de l'entreprise qu'à l'intérieur de celle-ci ;
- d'avoir des équipes de développement travaillant indépendamment et plus efficacement sur des systèmes qui interagissent, parce que ces équipes chercheront à développer des interfaces communes plutôt que d'avoir à synchroniser les processus ;

Nous pouvons donc conclure que les Web services offrent aux entreprises la flexibilité de réponse tout en permettant d'anticiper les besoins changeant des clients, Les web services offrent aussi aux entreprises la possibilité de rationalisation des infrastructures logiciels et la flexibilité d'interaction et de configuration des alliances externes avec les partenaires et fournisseurs.

### **II.5.6. Les services Web et les SMA**

De nos jours, on assiste à un rapprochement entre les recherches en SMA et les services Web, un rapprochement qui se manifeste sous différents aspects. Et il y a essentiellement et principalement deux aspects. D'un côté l'utilisation des agents en tant que cadre conceptuel pour mettre en place des services Web sophistiqués, ou encore des outils de planification, découverte et composition des services Web, ce qui est le cas de notre cadre d'étude, i. e, utiliser les SMAs comme environnement pour la découverte des services Web. D'autre part, l'utilisation des services Web comme un cadre architectural et/ou technologique pour mettre en place des SMAs ou des agents accessibles à travers Internet.

Le but ou l'objectif essentiel est de rendre les capacités des agents accessibles à travers des services Web. Enfin, le tableau suivant représente une comparaison entre les différentes technologies (services Web et SMA) :

Environnement	SMA	Web Services
Description de service	Description d'agent	WSDL
Registration	DF/AMS (détaillé en bas)	UDDI
Communication	ACL	SOAP
Langage sémantique	SL	-
Interaction	IP	WS-BPEL WS-CDL
Autre capacité	Négociation	-

**Tableau II-5:** Comparaison entre les services web et les SMAs

Cependant, la technologie des services web actuelle présente quelques limites, dont nous citerons quelques-unes :

- Les services web restent passifs jusqu'à ce qu'ils soient invoqués ;
- Le service web ne connaît que lui-même mais ignore tout de ses applications ou de ses utilisateurs clients ;
- Le service web est incapable de bénéficier des nouvelles capacités de l'environnement afin d'apporter des services améliorés. Donc il n'est pas adaptable ;
- Les points faibles des services web, sont des domaines qui ont largement été explorés par la recherche dans les systèmes multi-agents. Leurs points faibles demeurent donc : l'autonomie, l'adaptation et la coopération.

Bien que les SMAs puissent être considérés comme solution aux limites actuelles que présente des services web, ces derniers soit les services web, restent souvent des interfaces de programmes orientés objet (développés via J2EE ou .NET) traités pour générer des descriptions WSDL afin de pouvoir communiquer avec des messages SOAP. A ce moment alors, Ils bénéficient des caractéristiques d'abstraction du paradigme orienté objet (encapsulation, héritage, passage de messages, etc...). L'une des évolutions majeure a consisté à passer d'une approche orientée objet des services vers une approche orientée agents. Les services seraient alors dans l'obligation de tenir compte du contexte d'ouverture et de distribution de l'environnement, et ainsi bénéficier des caractéristiques de proactivité, de flexibilité et d'adaptative des agents, et devenir en plus réellement interactifs. En plus, puisque les agents peuvent potentiellement participer à des interactions complexes tout en maintenant la coordination avec les autres agents, les services pourraient être dotés des prérequis pour la modélisation d'une collaboration dynamique entre services [123].

### II.5.7. De description sémantique de Services Web

Comme déjà souligné, la pile des standards des services Web ne prend pas en considération la sémantique de ces derniers. Donc, pour atteindre l'interopérabilité sémantique, les services Web doivent être capables de [124] :

- de lire et d'interpréter correctement la sémantique des données à envoyer ou à recevoir ;
- de décrire les fonctionnalités qu'ils fournissent en s'appuyant sur l'utilisation d'une sémantique explicite et compréhensible pour les machines.

C'est pour cela que nous distinguons essentiellement deux approches qui permettent de décrire sémantiquement les services Web et qui sont [124] : Les approches exploitant les langages de description sémantique et les approches d'annotations de langages existants.

### II.5.7.1. Langages de description sémantique

Ces services Web capables de description sémantique sont en fait issus du web sémantique et sont :

**Langage OWL-S** : ce OWL-S ou Ontology Language for Services Web est un sous-ensemble du langage OWL propre à la description sémantique de services Web [125]. Du point de vue purement Technique OWL-S est une ontologie OWL composée de trois sous-ontologies où chacune décrit respectivement ce qui suit :

- «Que fait le service» : désigné par le "service profile", il décrit les fonctionnalités des services Web ; il est utile lors de la découverte et la sélection des services. Le "service profile" consiste en les informations liées au fournisseur du service et à la description fonctionnelle du service (IOPE : entrée, sortie, précondition, effet).
- «Comment le service fonctionne» : appelé "service model", il permet de détailler la sémantique des données impliquées dans les messages échangés entre services. Le "service model" décrit le processus d'exécution du service à travers le "ProcessModel" qui fournit une description détaillée sur la manière dont le service opère. Le "ProcessModel" spécifie la transformation de données et les transitions d'état.
- «Comment accéder au service» ou le "service grounding", il désigne et spécifie l'encodage des données échangées, les protocoles de communication, en incluant tous les détails concrets nécessaires à l'invocation du service.

**Langage Web Service Modeling Ontology (WSMO)** : proposée par le laboratoire DERI, le WSMO est en fait une architecture conceptuelle pour expliciter la sémantique des services Web [126]. Quant au méta-modèle du WSMO. Il définit quatre éléments principaux qui sont : les services Web, les objectifs, les ontologies et les médiateurs, développés ci-dessous :

- Les services web : Le WSMO donne une description pour chaque service Web, et qui comprend quatre parties, à savoir, 1) les propriétés non fonctionnelles telles que le fournisseur, le coût, la performance, la sécurité, etc. Ces propriétés sont utilisées pour la découverte et la sélection automatique des Web services ; 2) Les used mediators médiateurs utilisés par le service Web. 3) La capability ou fonctionnalité décrite par les préconditions, les post-conditions et les effets ; 4) L'interface est décrite selon deux points de vue, à savoir, le premier point de vue

utilisateur ou la chorégraphie, et le deuxième point de vue qui est l'orchestration appelé le point de vue d'autres fournisseurs de services.

- Les objectifs : il s'agit là des fonctionnalités requises par l'utilisateur et que le service Web se doit de satisfaire un objectif, dans ce cas précis, décrit la fonctionnalité, les entrées-sorties, les préconditions et postconditions d'un service Web ainsi que les flux de données et de contrôles.
- Les ontologies : elles fournissent la terminologie de référence aux autres éléments de WSMO afin dans le but de spécifier clairement le vocabulaire du domaine de façon à ce qu'il puisse être interprétable par les machines.
- Les médiateurs : sont importants et très utiles pour résoudre de nombreux problèmes d'incompatibilité, à savoir, par exemple, les incompatibilités de données dans le cas où les services Web utilisent différentes terminologies, les incompatibilités de processus lors de la combinaison de services Web et les incompatibilités de protocoles lors de l'établissement des communications.
- Les objectifs : il s'agit là des fonctionnalités requises par l'utilisateur que le service web, de satisfaire un objectif, pour être plus précis, décrit la fonctionnalité, les entrées-sorties, les préconditions.

**II.5.7.2. Des approches d'annotation des langages existants :** Ces approches consiste à annoter ou si l'on veut à mieux expliquer les langages de description de services existants avec de l'information sémantique. Dans cette optique, des références vers des ontologies sont ajoutées aux fichiers de descriptions des services (p.ex. WSDL). L'avantage essentiel de ce genre de solutions se trouve dans la facilitation mise à la disposition des fournisseurs de services dans le but d'adapter la description de leurs services aux annotations et/ou remarques proposées. A ce titre, le "Semantic Annotations for WSDL and XML Schema" (SAWSDL) [127] et le "Web service semantics" (WSDL-S) [128] figurent parmi les langages d'extension supportant ces approches.

**SAWSDL :** Une recommandation technique a été proposée par le world wide web désignée sous le vocable SAWSDL afin d'augmenter l'expressivité des descriptions WSDL par de l'information sémantique [129]. Ainsi, et pour annoter ses opérations, ses messages d'entrée et de sortie, le SAWSDL fournit un ensemble d'attributs d'extensibilité applicable sur les éléments du WSDL. Cependant, trois attributs d'extensibilité sont définis par défaut : l'attribut <sawSDL:modelRefer> qui nous permet d'associer n'importe quel élément du WSDL à un ou plusieurs concepts d'ontologies ; Quant aux deux autres attributs <sawSDL:liftingSchemaMapping > et <sawSDL:loweringSchemaMapping>, ils sont ajoutés aux définitions des types (schemamapping) avec pour but de spécifier, respectivement, les correspondances entre les éléments du message du service Web (schéma des données en XML) et l'information sémantique de l'ontologie (lifting) et vice-versa (lowering).

**WSDL-S :** Cette approche qui augmente le langage WSDL, est basée sur l'exploitation des éléments d'extensibilité du WSDL [124]. Ainsi, le langage WSDL augmente avec plusieurs extensions liées aux opérations et aux messages d'entrée et de sortie des services Web par l'information sémantique [130]. Ces extensions qui consistent en des références aux concepts,

sont décrits dans les ontologies du domaine pour spécifier et faciliter la sémantique de messages, des préconditions et des effets. En conclusion, on peut affirmer que le WSDL-S vise surtout à donner une approche «allégée» d'annotation sémantique des services Web.

### II.5.8. Des médiations des Services Web

Des hétérogénéités peuvent souvent apparaître entre les services Web et provoquer des conflits. La résolution de ces conflits est donc prise en compte par une médiation nécessaire pour assurer l'interopérabilité des systèmes d'information de type ERP. Cette médiation est considéré à juste titre comme le noyau des architectures orientées services. Cette médiation consiste généralement à résoudre les conflits entre deux acteurs. Ce travail de résolution de conflits est effectué par un élément spécifique appelé «médiateur» [124]. C'est le professeur-chercheur G. Weiderhold [131] qui a conçu la première architecture de médiation en 1992, et qui introduit la notion de médiateur. Il donne aussi la définition de la notion de médiateur : «Un médiateur est un module logiciel qui exploite la connaissance codée relative à des ensembles ou sous-ensembles de données afin de créer des informations pour une couche supérieure d'application». Ainsi la notion de médiation a été utilisée initialement dans le domaine des bases de données, puis adaptée aux services Web. On peut classifier les différentes tâches de médiation selon diverses perspectives en conformité avec la classification des hétérogénéités [132], [133] et [134]. Cependant nous retiendrons celles de Mrissa [124] puisqu'elle tient compte des propriétés fonctionnelles et non fonctionnelles des services Web. En outre Mrissa propose trois niveaux de médiation comme suit :

***Le niveau intégration des services Web*** : Il a pour but de résoudre toutes les hétérogénéités entre les services Web à savoir, les hétérogénéités des données et les hétérogénéités des propriétés fonctionnelles et non fonctionnelles.

- ***Le niveau adaptation d'interfaces*** : Il a trait à la résolution des hétérogénéités des propriétés fonctionnelles.

- ***Le niveau médiation des données*** : Il concerne seulement la résolution des hétérogénéités des données.

### II.5.9. De l'interopérabilité et de l'intégration des Services web

Pour faire face à une concurrence impitoyable, et suivre économiquement, les entreprises sont tenues d'actualiser et d'améliorer sans cesse leurs moyens de gestion. Ainsi les parmi les nouvelles orientations des entreprises, il y a l'utilisation des services Web pour développer des systèmes d'information interopérables. Mais cette interopérabilité est freinée par les éventuelles hétérogénéités sémantiques. Cependant, et dans le but de surmonter ce problème, il s'avère nécessaire d'intégrer et faire inter-opérer des services Web par des mécanismes qui permettront leur publication, sélection et médiation au niveau sémantique. En sous-section dans ce qui suit, nous essayerons de décrire les approches d'intégration des services Web [124].

- **Approches à base d'adaptateurs** : Il s'agit ici d'encapsuler les services web par l'utilisation d'une couche d'adaptateurs. Cette couche est incorporée et/ou intégrée à une architecture homogène qui supporte leur découverte, leur composition et leur administration. Il est à noter que les hétérogénéités sont résolues manuellement au moment de l'intégration des services dans l'architecture par les fournisseurs, en établissant les correspondances entre leurs services et les adaptateurs [136].
- **Approches à base de communautés** : Elles regroupent plusieurs services tout en fournissant une fonctionnalité équivalente derrière une interface unique. A cet instant, on dit que 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 [137].
- **Approches à base de descriptions étendues** : Elles sont utiles pour ajouter des informations supplémentaires dans les descriptions des services Web. Ces informations concernent les propriétés fonctionnelles et non fonctionnelles des services Web. Les solutions de médiations utilisent ces descriptions améliorées pour réconcilier les services Web [135].
- **Approches à base de services Web** : Elles reposent sur les services web essentiellement pour la publication, l'invocation, la médiation et l'exécution. Ces approches sont beaucoup plus prometteuses par rapport aux premières, car elles bénéficient des avantages des architectures orientées services.

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

## II.6. Conclusion

Dans ce chapitre, nous avons présenté la notion de sémantique et sa représentation notamment par les ontologies, qui constituent le meilleur outil conceptuel de capture et de description des connaissances. Vu leur importance et leur large utilisation dans plusieurs domaines, une panoplie de définitions ont été dressées par beaucoup d'auteurs depuis leur apparition dans le domaine de la philosophie jusqu'à leur adoption en IA (Intelligence Artificielle), aussi, plusieurs méthodes de conception, outils et langages de description des ontologies ont été présentés. Les services Web, les systèmes multi-agents ainsi que les ontologies offrent un cadre de travail pour assurer l'interopérabilité entre les applications et les ERPs. Nous avons présenté dans ce chapitre la technologie des agents, sa définition, ses

caractéristiques ainsi que les SMA et le principe d'interopérabilité dans les SMA et les services Web afin de donner une idée sur la réalisation d'une approche hybride entre SMA et WS (Web Service). En d'autres termes, cela consiste à intégrer les agents en tant que cadre conceptuel pour mettre en place des services Web sophistiqués (outils de découverte des services Web).

Nous avons présenté aussi des solutions et quelques travaux pour dresser l'interopérabilité technique et syntaxique des systèmes d'information. Vu les avantages offerts par chacune de ces solutions, nous sommes convenus à l'utilisation dans notre approche une combinaison des outils et des techniques, à savoir les ontologies, les Services web et les systèmes multi-agents. Ceci nous permet de profiter des points forts de chaque paradigme. Ces techniques favorisent la flexibilité, la résolution du problème d'hétérogénéités des plates-formes tout en encapsulant la complexité des systèmes hétérogènes. Mais, elles présentent un manque en ce qui concerne la présentation de l'aspect sémantique. Ceci freine l'interopérabilité des systèmes d'information en raison de multiples conflits qui sont liés aux problèmes de compréhension.

Dans le chapitre suivant, nous présenterons les différents approches d'interopérabilité des systèmes d'information proposées dans la littérature et de positionner notre approche de ce qui existe.



## Chapitre III : Etat de l'art sur les approches d'interopérabilité sémantiques

---

### III.1. Introduction

Comme nous avons déjà présenté les différents outils ainsi que les techniques utilisées dans la modélisation de la sémantique, il nous paraît utile de présenter dans ce chapitre les approches proposées dans la littérature relative à l'interopérabilité sémantique des systèmes d'information distribués et hétérogènes.

Une approche est en effet un ensemble d'idées et d'architecture et d'architectures générales, une sorte de survol de quelques projets et prototypes réalisés dans ce domaine qui permette de connaître et/ou d'entrevoir différentes démarches de mise en œuvre.

C'est pour cela qu'un certain nombre de critères doit être arrêté pour pouvoir établir une comparaison des approches et de présenter un bilan le domaine d'interopérabilité et ainsi de positionner notre approche. Dans ce chapitre donc, nous présentons notre première et humble contribution concernant la modélisation de l'ERP en utilisant les systèmes multi-agents et les ontologies.

### III.2. Définition des critères de comparaisons

L'étude des innombrables solutions pour l'interopérabilité de SI se base sur différentes critères discriminants qui nous donne la possibilité de comparer et de catégoriser chaque solution. Les buts et objectifs de toutes les approches sont identiques soit localiser les données pertinentes représentant un modèle et une présentation homogène. Dans ces cas d'approches, cinq critères ont été retenus pour comprendre comment l'interopérabilité est obtenue et comment la sémantique intervient dans cette problématique :

- Le modèle de représentation et d'interrogation des données ;
- Le modèle et les outils de représentation de la sémantique ;
- La méthode de résolution dynamique ou statistique les requêtes ;
- L'architecture d'interopération ;
- Le respect des propriétés pour l'interopérabilité.

### III.3 de l'Approche fédérée

Des logiciels appelés systèmes de gestion de bases de données (SGBD) avec un ensemble de bases de données auxquelles revient la responsabilité de gestion constituent le système de bases de données. Quant au SBDF ou Système de Base de Données Fédérées est en fait un ensemble de systèmes de bases de données coopératifs, autonomes et éventuellement hétérogènes.

Par analogie avec le SGBD ou système de gestion de bases de données, le logiciel responsable du contrôle et coordination des accès à la fédération s'appelle Système de Gestion

de Bases de données Fédérées (SGBDF). Cependant, en général, un système fédéré est élaboré à partir de systèmes préexistants, ce qui explique la distribution, l'hétérogénéité des données ainsi que le besoin d'autonomie des systèmes composants.

- **Distribution** : Les données sont distribuées sur plusieurs bases de données qui peuvent être stockées sur un ou plusieurs systèmes informatiques ; ces derniers peuvent être regroupés sur un seul site mais reliés par tout un réseau de communication bien que dispersés géographiquement. La distribution des données sur les différentes façons : Partitionnement vertical et horizontal (en termes des techniques du modèle relationnel) et duplication globale ou partielles des données mais comme signalé précédemment, un système fédéré est bâti généralement sur des systèmes préexistants, autrement dit que la distribution des données sur les différents sites existe déjà avant le déploiement du système fédéré.
- **Hétérogénéité** : Tous les systèmes composants peuvent présenter plusieurs types d'hétérogénéités : Une hétérogénéité matérielle due aux différences technologiques des systèmes informatiques qui supportent les données, une hétérogénéité logicielle due à la différence des environnements logiciels, une hétérogénéité des SGBD, hétérogénéité due à la différence des modèles de données, une hétérogénéité structurelle due aux choix de conception différents, et une hétérogénéité sémantique. On trouvera les problèmes d'hétérogénéité présentées en détails dans le chapitre II.
- **Autonomie** : Il est utile de souligner qu'un système composant doit préserver son indépendance et son contrôle local pour continuer à répondre aux besoins du site local. Cette autonomie se présente sous plusieurs aspects, dont :
  - ✓ L'autonomie de modélisation qui suppose que le site est capable de faire le choix de ses propres modèles et langages d'interrogation de données.
  - ✓ L'autonomie de communication qui implique que le système composant est capable de décider quand et comment répondre à une requête externe.
  - ✓ L'autonomie d'exécution qui suppose qu'un système composant doit être capable d'exécuter des requêtes locales sans pourtant interférer avec des requêtes qui proviennent d'un utilisateur externe appartenant à la fédération.
  - ✓ L'autonomie d'association qui implique que le site local peut décider de s'associer ou de se dissocier de la fédération et d'appartenir à une ou plusieurs fédérations.

### III.3.1 Des composants d'un système fédéré

C'est un ensemble de systèmes de bases de données qui peuvent d'ailleurs de différents types, qui constitue le système fédéré. Cet ensemble peut engendrer une structure très complexe pour la fédération. Les systèmes qui composent la fédération peuvent être un des types suivants :

- Un système de base de données centralisé : Il s'agit là d'un système composant qui ne présente ni distribution, ni hétérogénéité. Il s'agit simplement d'un système de gestion

de bases de données qui centralise les données en une seule base implantée sur une seule machine.

- Un système de base de données distribué : Il s'agit là d'un système de gestion de bases de données unique qui gère des bases de données implantées sur une ou plusieurs machines éventuellement hétérogènes sur la plateforme matérielle et environnement logiciel.
- Un système multi-bases : Ici, il est possible qu'il s'agisse d'un système Multi-bases contrôlant plusieurs SGBD qui fonctionnent sur une ou plusieurs machines et où chacun gère une ou plusieurs bases de données. On peut donc parler d'un système multi-bases homogène lorsque tous les SGBD composants sont identiques, sinon le système multi-bases est dit hétérogène.
- Un système de bases de données fédérées : Il s'agit d'un système composant qui peut être lui-même et à lui seul un système fédéré qui gère plusieurs systèmes de bases de données autonomes coopératifs et éventuellement hétérogènes, ce qui implique qu'un système composant peut appartenir à la fois à plusieurs fédérations en même temps.

### III.3.2. De l'Architecture

Trois niveaux de représentation sont décrits par l'architecture des systèmes de gestion des bases de données classiques.

- 1- Le niveau interne qui correspond au schéma physique des données ;
- 2- Le niveau conceptuel qui correspond au schéma conceptuel des données ;
- 3- Le niveau externe qui correspond aux vues des différentes catégories d'utilisateurs.

Pour les systèmes fédérés MM. Sheth et Larson [138] nous proposent une architecture, mais cette fois-ci à cinq niveaux : schéma local, schéma composant, schéma export, schéma fédéré, et enfin schéma externe :

- Le schéma local : C'est un schéma conceptuel exprimé dans le modèle natif du SGBD local.
- Le schéma composant : Il est obtenu par traduction du schéma local dans le modèle commun ou pivot présente un double intérêt :
  - ✓ détecter les divergences des schémas locaux exprimés dans le même modèle de représentation.
  - ✓ il est possible que les sémantiques oubliées dans les schémas locaux peuvent être rajoutées dans les schémas composants correspondants.
- Le schéma export : Ce schéma représente une petite partie d'un schéma composant qui est disponible au niveau du système fédéré. Il peut donc contenir aussi des informations portant sur le contrôle d'accès aux données de la fédération.
- Le schéma fédéré : il s'agit d'une intégration multiple des schémas d'export correspondant à un domaine d'application. Il inclut aussi des informations sur la distribution des données. Ce schéma est appelé également schéma d'importation ou schéma conceptuel global.

- Le schéma externe : c'est enfin une partie du schéma fédéré qui représente la vue d'un ou plusieurs utilisateurs. Ce schéma externe permet de renforcer les contrôles de contraintes d'intégrité et d'accès.

### III.3.3. Des Types de fédération

On rencontre deux types de fédération, l'une appelée fédération faiblement couplée, et l'autre fédération fortement couplée.

#### III.3.3.1. De la fédération faiblement couplée

L'intégration des informations des différents systèmes au coup par coup, en fonction des besoins, est l'objectif de la fédération faiblement couplée. Cependant, cette approche repose sur l'utilisation d'un langage d'interrogation et d'un modèle de représentation commun. Donc chaque SI exporte ses informations sous la forme d'un schéma décrit dans le modèle commun (généralement un modèle orienté objet). Donc chaque SI exporte ses informations sous la forme d'un schéma (généralement un modèle orienté objet), le langage multi-bases (extension de SQL ou OQL) permettant une interrogation multi-sites. L'interopérabilité des BD hétérogènes s'opère autour d'un langage de requêtes puissant. Donc, la résolution des conflits schématiques et sémantiques demeure entièrement à la charge de l'utilisateur. Cependant, le langage des requêtes possède la particularité d'être doté de constructeurs pour la réconciliation sémantique (résolution des conflits) des données hétérogènes.

Mais la fédération faiblement couplée a des avantages et des inconvénients parmi lesquels nous citerons : L'absence d'un schéma préétabli (schéma fédéré) pour l'intégration des différentes sources de données n'exige aucune tâche à l'administrateur pour ajouter ou retirer un système de la fédération, ce qui fait que la fédération faiblement couplée reste une approche extensible ; à l'encontre, la résolution des conflits ainsi que la localisation des différentes sources sont à la charge de l'utilisateur.

#### III.3.3.2. Fédération fortement couplée

Son objectif est la construction d'un schéma fédéré qui intègre les différents schémas d'exports des différents et innombrables systèmes de la coopération. Ce schéma fédéré traitera les mises à jour et requête lancées par les utilisateurs. Les cinq niveaux proposés par Sheth (Cf. plus haut) sont ainsi respectés par cette approche. En outre, et grâce à l'utilisation d'un modèle pivot et aux assertions de correspondance inter-schémas, le schéma fédéré résout aussi les conflits syntaxiques et schématiques. Quant aux conflits sémantiques, ils sont résolus implicitement puisque le schéma fédéré correspond à un domaine d'application particulier.

Cependant, la fédération fortement couplée possède aussi des avantages et des inconvénients : - Ainsi un schéma fédéré permet donc ce type d'accès aux données partagées de manière uniforme et globale ainsi que la transparence qui est assurée. Pour cela, il suffit de poser une requête sur le schéma fédéré. En outre les conflits schématiques et sémantiques sont résolus par le processus d'intégration. De même, L'extensibilité (ajout et retrait d'un SI) et la scalabilité (évolution du nombre de SI participants) ne s'adaptent pas bien à l'intégration. La

fédération fortement couplée est donc une approche intéressante dans le cadre d'une coopération de quelques SI mais peu évolutive.

### **III.4. Des Approches de type médiation**

Parmi les solutions multi-bases, les approches fédérées ont essayé et même réussi à faire partager des informations appartenant à plusieurs systèmes autonomes, distribués et hétérogènes.

Néanmoins l'extensibilité des solutions ainsi que la transparence de localisation restent faibles. Cependant les approches de type médiation ont permis d'étendre les approches fédérées en apportant davantage de flexibilité et transparence. Il faut noter que le principal apport de l'approche fédérée reste le fonctionnement de la coopération dans un environnement dynamique caractérisé par un nombre assez important de système participant à la coopération, à la modification fréquente des systèmes de la coopération, et à l'ajout et ou retrait dynamique des systèmes.

En termes plus explicative, l'approche de médiation cherche à donner l'interopérabilité au mécanisme des SI, la transparence de l'approche fédérée fortement couplée et la flexibilité de la fédération faiblement couplée.

#### **III.4.1. De la Caractéristique**

On appelle médiateur un composant essentiel sur lequel repose la médiation ; sa mission est de répondre à des besoins spécifiques à partir de connaissances mises à sa disposition. Le médiateur est capable de localiser l'information et de résoudre les conflits schématiques et sémantiques. Le Wrapper, un composant secondaire, sert d'interface avec les SI composants. Ce wrapper résout les conflits syntaxiques en présentant les données dans le modèle de médiation.

Le Médiateur : sa mission est la localisation des SI et des données pertinentes. Le médiateur résout et d'une manière transparente les conflits de données. Il simplifie, résume, réduit, combine et décrit les données distribuées [139]. Ses fonctionnalités principales sont énoncées dans [139]. Il suffit d'un ensemble de connaissances pour permettre au médiateur de générer un plan d'exécution pour traiter une requête à l'aide d'un langage de médiation. Le médiateur sert donc d'interface entre les utilisateurs d'une coopération et les informations partagées.

On peut donc affirmer que Wrapper est un outil qui permet à un ou plusieurs médiateurs d'accéder au contenu d'une source d'information. Son véritable rôle est centré sur la résolution des conflits syntaxiques. Le Wrapper entretient le lien entre la représentation locale des informations et leur représentation dans le monde de la médiation.

Le Langage de médiation : MM. Buneman et al. dans [140] nous proposent un standard de langage de médiation d'où il ressort de la nécessité de représenter les données complexes et semi-structurées, d'échanger des informations à l'aide de différents types de règles (contrainte, requête, traduction, mise en correspondance, intégration, etc.) et d'échanger des méta-informations pour améliorer à chaque fois le processus de médiation.

Alors que l'approche fédérée qui distingue cinq niveaux, celle de type médiation se présente en trois seulement et qui sont : le Niveau système d'information qui dispose des sources de données locales, le niveau médiation composé du couple Médiateur/Wrapper, ainsi que les connaissances nécessaires à l'interopérabilité des sources de données, et enfin le Niveau application qui correspond aux applications.

### III.4.2. Du type de médiation

On peut distinguer deux types de médiation selon la façon avec laquelle sont traitées les requêtes et utilisées la sémantique des sources de données par le médiateur.

La médiation de schéma associe au médiateur un ensemble de connaissances qui localisent et intègrent à priori les informations pertinentes à un contexte d'utilisation. Des règles qui définissent où se trouvent les données pertinentes, leurs corrélations autrement dit leurs transformations et leurs combinaisons pour fournir un résultat ou des résultats exploitables sont établis par un plan d'exécution que la résolution d'une requête et l'accès aux données distribuées doivent prendre en compte. La notion du contexte est ici implicite : En effet le schéma de médiation est actif au niveau de la structure des données, et des outils sont là pour simplifier la résolution à l'avance des nombreux conflits de données. Le côté dynamique du médiateur se trouve donc principalement dans sa capacité à gérer la disparition de sources et ou information incomplète, et de s'adapter aux capacités d'interrogation différente de chaque site.

La médiation de contexte est dépendante de l'intégration dynamique des informations en fonction du contexte de l'application ou de l'utilisateur. Chaque application et source d'informations décrivent le contexte de ses informations. Donc le médiateur identifie, localise, transforme et intègre les informations pertinentes vis-à-vis du contexte associé à une requête. La résolution des conflits de données est de ce fait dynamique et n'a nul besoin de la définition d'un schéma de médiation. La médiation repose donc principalement sur la comparaison de la sémantique des informations et sur le rapprochement contextuel des sources. Cependant les outils du type métadonnées et ontologie sont usités pour décrire les contextes.

### III.4.3. Des prototypes réalisés dans l'approche de médiation

Nous citerons parmi les innombrables médiateurs qui ont été développés dans la médiation de schéma, les prototypes suivants :

Le projet DIOM [141] qui vise comme objectif la Scalabilité, l'évolutivité, l'autonomie et la flexibilité. Une extension de l'ODMG'93 appelé aussi DIM-IDL a été choisie comme modèle pour représenter une interface.

Le projet DISCO [142] propose un langage de définition et d'interrogation déclaratif afin de spécifier une source de données, un wrapper, une interface et une extension.

Le projet Tsimmis [143] schématise les données à l'aide d'un modèle auto descriptif OEM (Object Exchange Model) et intégrées dans des médiateurs à l'aide d'un langage logique MSL (Mediator Specification Language). Le modèle OEM permet la représentation des objets semi-structurés et peut s'adapte à des représentations multiples d'une même information.

Le projet Hermes [144] : ce projet est basé sur un modèle de définition de domaines pour rendre plus simple la construction de règles d'intégration où un domaine est associé à un type de représentation des informations et qui généralise un modèle de données pour représenter des données structurées (BD relationnels, objets, etc.) et données complexes (image, vidéo, etc.).

Le système YAT [145] : il définit un langage basé sur des règles pour combiner et restructurer des informations hétérogènes en provenance de plusieurs systèmes d'informations. Le modèle de YAT utilise une représentation sous forme de graphes pour expliciter la sémantique d'un schéma. Une des caractéristiques du modèle est la notion de motif (pattern dans le texte) qui représente la structure possible d'une information. Des motifs de base sont prédéfinis dans le système YAT et par la suite d'autres motifs peuvent être ajoutés pour représenter des informations particulières.

Le système MIX [146] utilise quant à lui XML comme modèle commun pour l'échange des données. Les wrappers fournissent un schéma d'export XML des sources d'information et permettent l'accès à des sites web ainsi qu'à des SGBD de différents modèles : HTML-XML wrapper, BDR-XML wrapper, BDOO-XML wrapper, etc.

Quant à l'approche médiation de contexte, nous citerons les prototypes suivants :

Le système SIMS [147] : ce système définit le domaine de référence comme le moyen d'accès aux informations avec un schéma terminologique. Ainsi pour décrire le contexte, on utilise le langage LOOM qui permet de représenter les classes sémantiques formant les concepts du domaine, les relations entre ces classes, les rôles définissent la signification d'un concept par rapport à un autre. Les informations des SI sont également modélisées dans des classes sémantiques du domaine. Un schéma regroupant les classes et des rôles d'une source d'information représente son contexte local.

Le projet On2Broker [148] est considéré pour extraire, interroger et raisonner sur des informations qui proviennent de pages WEB html. Cette solution consiste en une extension du langage html mais en rajoutant des balises pour associer des informations sémantiques aux données d'une page web à la façon XML. Ces informations sémantiques sont donc extraites d'une ontologie et jouent le rôle de métadonnées dans la description des informations.

Le système COIN [149] propose une interopération reposant sur la définition d'un domaine d'application, de contextes et de schémas d'export enrichis. Ce système utilise un langage provenant de la F-Logique pour décrire les différents niveaux sémantiques. Le domaine d'application est un schéma logique définissant clairement le vocabulaire commun aux différents SI de la coopération pour décrire les contextes, les relations entre les différents concepts, les propriétés qui caractérisent les valeurs associées aux concepts et les fonctions de transformation pour changer de contexte de valeur. Chaque source d'information et chaque application (utilisateurs) spécifient l'interprétation de ce domaine d'application sous la forme de contextes. Chaque SI de la coopération présente un schéma d'export enrichi par les informations contextuelles.

Le système Observer [150] : Ce système définit un large treillis de domaines mis en relation à l'aide d'un dictionnaire qui détermine spécifiquement les relations terminologiques entre les termes décrivant chaque domaine. Or un domaine est formé d'une ontologie sous la forme d'une hiérarchie terminologique dont les concepts peuvent avoir des attributs, un ensemble de sources d'informations relatives à ce domaine et un dictionnaire de mises en correspondance pour relier les données des sources aux concepts d'une l'ontologie. Les informations de mise en correspondances entre la description locale des données et les concepts d'une ontologie utilisent une algèbre relationnelle étendue.

Le projet InfoSleuth [151] : ce projet propose une approche agents de l'interopération. Des agents utilisateurs définissent donc le contexte d'un groupe utilisateurs et servent d'interface avec les autres agents intelligents pour poser une requête et prendre les réponses. Des agents ontologies seront chargés de gérer des serveurs d'ontologies, autrement dit, la création, la mise à jour et l'interrogation d'une collection d'ontologies. Cependant des agents d'exécution gèrent des requêtes basées sur les concepts d'une ontologie en la décomposant sur différents agents - ressources-. Les agents ressources vont à leur tour identifier les informations d'une source de données par rapport à une ontologie et gèrent la traduction d'une requête exprimée sur une ontologie dans le langage d'interrogation local d'un SI. Ensuite ils renvoient les résultats exprimés dans le modèle de l'ontologie. Des agents d'analyse fournissent des fonctionnalités avancées d'analyses et de déductions sur les données retournées par une requête (Datamining). Les agents de courtage (broker agents) répertorient les fonctionnalités et les services des différents agents de l'architecture. Ils peuvent aussi enregistrer les services d'autres agents de courtage.

Le projet DILEMMA [152], [153] : est une solution d'interopération qui permet de définir une large coopération de systèmes d'informations et d'assurer ainsi un accès clair et transparent aux informations partagées par la coopération. Cette approche proposée est donc de type médiation sémantique. Son rôle permet de réconcilier des systèmes d'informations autour d'une coopération tout en exploitant l'aspect sémantique des informations. L'objectif principal de DILEMMA est de permettre à un système qui fournit son contexte d'application (contexte de consommateur ou contexte de référence), d'aboutir aux SI et aux informations partagées par ceux-ci. Pour représenter les informations et leur sémantique, DILEMMA se sert d'un modèle de médiation qui permet de décrire en même temps les informations d'un système par rapport à son contexte local d'application et la sémantique de ses informations par rapport à son contexte.



### III.5. Comparaison entre les solutions d'interopérabilité :

L'objectif de cette section est d'établir un bilan de la progression et de l'évolution de l'interopérabilité via les différentes approches proposées. Nous avons cru utile de retenir les critères suivants pour faire la comparaison entre les différentes approches :

- Le Modèle d'interrogation et de représentation des données ;
- La Modélisation de la sémantique ;
- Le Processus d'intégration (statique/dynamique) ;
- L'Extensibilité de l'architecture pour maîtriser l'ajout et le retrait de SI ;
- La Transparence à la localisation et au format de données ;
- L'Architecture.

#### III.5.1. Le modèle d'interrogation et de représentation des données

Pour réussir l'interopération, un paramètre important est nécessaire : il s'agit du modèle des données. En effet, la représentation de la sémantique des informations, partagée par une coopération, nécessite une richesse au niveau du modèle et des concepts ; ces derniers permettent d'explicitier cette sémantique.

Les systèmes préexistant sur lesquels se construit la coopération, sont généralement conçus avec des modèles sémantiquement pauvres (fichiers classiques, BD hiérarchiques, BD Codasyl, BD relationnelles). Cependant la solution d'interopérabilité peut y remédier en adoptant un modèle commun qui soit sémantiquement riche.

Des solutions d'interopérabilité sont basées sur un modèle commun orienté objet issu du standard de l'ODMG (ODL, DIOM-IDL, GDL, OGM, DISCO-ODL). Malgré la richesse sémantique du modèle objet par rapport au modèle relationnel, il permet toutefois de représenter une partie de la sémantique d'un objet contenu dans les SI : dans ses attributs, ses fonctionnalités (opérations), relations d'héritage et d'agrégation.

C'est pour cela que d'autres solutions ont adopté des modèles logiques qui nous permettent en général de représenter les informations avec des concepts objets mais en ajoutant un langage à base de règles pour expliquer et définir des contraintes et des relations complexes entre les objets. Ces modèles sont donc basés sur la logique terminologique comme KIF et LOOM ou bien sur la logique déductive comme la logique de frames (F-logique). Le projet DILEMMA, comme nous l'avons vu précédemment, est un bon exemple puisque il utilise les deux types de modèles : est un bon exemple puisque il utilise les deux types de modèles : la logique déductive pour la représentation des données sous forme d'I-objets et la logique terminologique pour la représentation du contexte sous forme de classes conceptuelles et rôles. D'autres solutions ont choisi des modèles propriétaires pour prendre en charge des données spécifiques. Pour cette raison le modèle doit comporter des concepts permettant de représenter les particularités du système comme les données géographiques pour les SIG. En fait, il s'agit, en général, d'une extension de modèles objets ou logiques avec des concepts particuliers autrement dit spécifiques au domaine. OEM et YAT sont autant d'exemples de modèles propriétaires adoptés respectivement par les systèmes TSIMMIS et YAT.

Concernant les langages utilisés pour l'interrogation des données partagées par la coopération, nous distinguerons donc plusieurs tendances, entre autre :

- Les langages de programmation : généralement C++ et Java.
- Les langages déclaratifs : généralement, il s'agit d'une extension objet du SQL tels que : GQL de GARLIC, DIOM-IQL de DIOM et LOREL de TSIMMIS.
- Les Langages logiques : il s'agit en fait des dérivés de Prolog, F-logique, KIF et LOOM.
- Une interface graphique : il s'agit là d'un moyen utilisé également par certaines solutions pour aider l'utilisateur à construire sa requête. Ces outils existent et sont disponible par exemple sur les systèmes INFOMASTER et HERMES.

### III.5.2. De la représentation de la sémantique

Cette représentation sémantique permettra, dans un système d'interopération, de comprendre le contexte d'utilisation et surtout la signification exacte des informations partagées par les différents SI d'une coopération. Nous distinguerons donc des solutions de représentation de la sémantique selon trois niveaux :

- Un niveau coopération : il s'agit là d'un domaine commun aux différents SI participants. La sémantique permettra à ce niveau de présenter le domaine global sur lequel se traitent les différents SI de la coopération.
- Le Niveau SI appelé également contexte source des informations. Ainsi, et pour écarter toute ambiguïté, la sémantique permet d'exprimer, à ce deuxième niveau, le contexte où les informations et les mises à la disposition des applications sont offertes. En outre la même information peut apparaître dans différents systèmes mais cependant avec des objectifs et des contextes différents.
- Le Niveau application (ou consommateur) : ou contexte d'utilisation des informations. L'application proprement dite ou le système consommateur doit posséder lui aussi un moyen qui exprime la sémantique d'utilisation des informations recherchées.

Concernant le premier niveau, nous distinguerons quatre types de représentations du domaine traité par la coopération :

- Le Schéma : c'est la description du domaine sous forme d'un schéma conceptuel exprimé, dans le modèle commun en général. La majorité des solutions de type médiation de schéma utilisent cette technique.
- L'Interface objet : il s'agit en fait de la description d'une source de données générique où chaque objet (ou instance) correspond à un groupe de données locale. Cependant la description du domaine se limite à un certain nombre d'attributs de l'objet.
- L'Annuaire : On parle ici, comme son nom l'indique d'un annuaire qui contient un résumé sur chaque SI rattaché à un domaine particulier qui simplifie la localisation. Le Système DILEMMA utilise un annuaire pour accéder aux schémas de médiation des SI participants.
- L'Ontologie : il s'agit d'un outil qui sert à décrire un domaine de manière explicite et formelle. Les solutions de type médiation de contexte utilisent, en général une ontologie

pour la description du vocabulaire commun aux différents SI de la coopération (SIMS, OBSERVER, ON2BROKER).

Quant à la représentation du contexte source, il s'agit de décrire la sémantique des données du SI par rapport au domaine de la coopération. Ainsi, plusieurs manières sont utilisées telles :

- Le schéma export : il s'agit d'un schéma de données partagées par un SI sur la coopération mais exprimé dans le modèle commun. Le schéma export peut donc être exploité comme moyen de représentation de la sémantique des informations du SI. C'est aussi ce schéma qui est utilisé par les approches de fédération fortement couplée pour représenter la sémantique des informations d'un SI.
- Les règles d'interprétation : c'est une technique utilisée par les approches médiation de contexte. Cette technique consiste à interpréter les données d'un système local sur le contexte local du SI ou sur le domaine global de médiation pour expliciter la sémantique des informations et donc pouvoir en quelque sorte identifier les correspondances entre les données hétérogènes (COIN, ONTOBROKER, SIMS, OBSERVER, DILEMMA).
- Les vues : pour extraire une partie d'un ou plusieurs schémas fédérés pour une application particulière. On utilise cette technique par les approches fédérées fortement couplées. Ces schémas peuvent être considérés comme des contextes d'utilisation des informations par un groupe d'utilisateurs d'une fédération.
- Le schéma terminologique : il s'agit d'un schéma qui représente les termes et les relations entre eux pour pouvoir décrire le contexte local d'un SI ou le contexte d'utilisation d'une catégorie d'utilisateurs. (OBSERVER, KRAFT, SCOPE, DILEMMA)
- Le médiateur : il représente en fait la sémantique globale d'un domaine d'application pour une coopération de SI. Des systèmes comme DISCO et TSIMMIS définissent un grand nombre de médiateurs hiérarchisés qui communiquent entre eux. A cet effet, un médiateur donné dans la hiérarchie peut être considéré aussi comme contexte d'utilisation d'une catégorie d'utilisateurs.
- L'Agent utilisateur : il s'agit de l'agent qui avec une requête, décrit les besoins d'une application particulière et qui communique avec d'autres agents spécialisés et pas n'importe lesquels. En effet, il communique avec des agents spécialisés dans la recherche d'informations pertinentes tout en utilisant la requête comme contexte d'utilisation (INFOSLEUTH).

### III.5.3. De la transparence

Cette caractéristique principale des approches fédérées fortement couplées est assurée par le schéma fédéré qui renferme toute l'information afin de localiser les sources de données distribuées et d'assurer ainsi la transparence des modèles de données. On rappellera que dans les approches de médiation, la transparence est assurée à plusieurs niveaux (hiérarchie de médiateurs/Wrapper). Le médiateur assure la transparence de localisation et de conflits schématiques et sémantiques (transparence aux modèles de données par le Wrapper).

#### **III.5.4. De l'extensibilité :**

Il s'agit d'une caractéristique des approches de médiation de contexte. Les SI participants, autrement dit la découverte des SI participants se fait de manière dynamique et n'a nul besoin de connaissances préétablies au niveau de la médiation. Ainsi, l'ajout ou le retrait ne provoque pas de changements au niveau de la médiation. C'est pour cette raison que les modifications dans le schéma des SI participants ne sont suivies d'aucune incidence sur la médiation.

#### **III.5.5. De l'architecture**

Les architectures essentielles adoptées dans les systèmes d'interopération peuvent être résumées comme suit :

- **Le Wrapper** : Pour accéder directement aux sources de données locales, les applications utilisent des interfaces (Wrapper) conçues toutes dans un modèle unique (modèle commun). Ainsi, seuls les conflits syntaxiques sont pris en charge dans ce type d'architecture.
- **Le Médiateur** : il représente la sémantique du domaine d'une coopération de SI. Le médiateur prend cependant en charge les conflits structurels et sémantiques tout en pouvant communiquer avec des Wrappers qui eux, prennent en charge les conflits syntaxiques. Quant aux applications, elles communiquent leurs requêtes au médiateur pour avoir ainsi accès aux différentes sources de données. On peut cependant trouver une hiérarchie de médiateurs qui traitent sur différents domaines tels TSIMMIS, DISCO, dans ce type d'architecture.
- **SMA** : Les systèmes multi-agents sont mieux adaptés aux besoins des applications du fait que les agents sont spécialisés, chacun en ce qui le concerne, pour une tâche particulière et coopèrent pour livrer le résultat final aux utilisateurs (INFOSLEUTH).

Critères	Modèle de données		Sémantique		Interopération			
	Solutions	Données	Requête	Contexte	Langage	Intégration	Extensibilité	Transparence
<b>Fédération fortement couplée</b>								
Larab	terminologique	terminologique	schéma	Terminologique étendu	Statique	Non	-	-
Mind	ODL	SQL	schéma, mapping	propriétaire	Statique	Faible	Oui	Corba
<b>Fédération faiblement couplée</b>								
Garlic	GDL	GQL, C++	schéma	propriétaire	requête	Faible	faible	wrapper
InfoMaster	relationnel	Logique graphique	schéma, mapping	KIF	Dynamique	Oui	Oui	Agent, wrapper
Florescu	ODL	OQL	Schéma, règles	propriétaire	Statique	Non	Oui	-
<b>Médiation de schéma</b>								
Diom	DIOM-IDL	DIOM-IQL	Schéma, mapping	Propriétaire	Dynamique	Faible	Oui	3-tiers
Disco	ODMG	OQL	schéma, mapping	propriétaire,	Statique	Faible	Faible	-
Tsimmis	OEM	logique, LOREL	wrapper	MSL	Statique	Oui	Oui	Médiateur
Hermes	Non	Graphique	Catalogue	Propriétaire, logique	Statique	Non	Oui	3-tiers
Yat	Yat	Non	Schéma	yatl	Statique	-	Faible	Wrapper
Mix	XML	Vue	schéma	XMAS	Statique	Non	Faible	propriétaire
<b>Médiation de contexte</b>								
Sims	LOOM	LOOM	Classe terminologie	LOOM	Dynamique	Oui	Oui	-
On2Broker	F-logique étendue	F-logique, Graphique	Schéma HTML	propriétaire	Dynamique	Oui	Oui	Web, warehouse
COIN	Relationnel	SQL	Schéma, axiomes	F-logique	Dynamique	Oui	Oui	web
Observer	Logique terminologique	Logique terminologique	Schéma, mapping	propriétaire	Dynamique	Oui	Oui	-
InfoSleuth	KIF, LDL	SQL	Agent ressource	KIF	Dynamique	Oui	Oui	Agents
DILEMMA	F-logique, librairie de fonctions	F-logique	schéma	Logique terminologique	Dynamique	Oui	Oui	web

**Tableau III-1:** Comparaison de solutions d'interopérabilité

### III.6. De la Classification des Approches d'interopérabilité par rapport aux techniques et outils utilisés

Dans cette partie, nous présentons et expliquons comment un certain nombre d'approches classées par catégories peuvent résoudre les problèmes liés à l'interopérabilité, mais en particulier les conflits de données et la prise en compte de la sémantique des SI et de données distribuées. L'état de l'art des multiples et différentes solutions retenues, est basé sur l'étude de critères qui permettant de comparer chacune de ces solutions. Ces solutions sont classées en deux types d'approches : Pour les présenter, il est nécessaire de diviser ces approches en deux catégories selon les techniques utilisées, de les évaluer sur des critères inhérents à la modélisation de la sémantique, et d'établir un bilan et enfin de proposer une nouvelle architecture qui prendrait en compte la modélisation de la sémantique pour permettre l'interopérabilité des ERPs. Nous présenterons, en premier lieu, les travaux qui s'articulent autour des systèmes multi-agents et ontologies ainsi que ceux inhérents aux services Web et ontologies.

Donc une première catégorie se basera sur les systèmes multi-agents et ontologies. Cependant, l'on rappellera que le paradigme agent donne une nouvelle vue pour le développement des systèmes de nature complexe, hétérogène, distribué et/ou autonome [154].

Plusieurs projets d'interopérabilité sémantique utilisent celui-ci, à savoir [155], [156], [157] et [158].

Le projet Infosleuth [159] a pour objectif d'implémenter un ensemble d'agents coopératifs qui découvrent, intègrent et présentent l'information en fonction des besoins d'un utilisateur ou d'une application pour lesquels ils produisent une interface simple et cohérente. Un ensemble d'agents collaboratifs, qui communiquent l'ACL KQML, constitue l'architecture du projet Infosleuth. Quant aux utilisateurs, ils expriment leurs requêtes sur une ontologie spécifique. KIF et SQL sont utilisés pour représenter les requêtes. Ces dernières sont expédiées aux agents spécialisés (agent broker, ontologique, planner,...) pour la recherche des données sur des sources distribuées et pour l'intégration. Résoudre donc de nombreux conflits sémantiques reste du ressort de l'utilisateur [3]. Ils emploient des agents spécialisés qui peuvent considérer comme des threads, sont loin de la définition de l'agent cognitif comme défini en intelligence artificielle distribuée.

Les auteurs [156] proposent un système multi-agents pour assurer l'interopérabilité sémantique et résoudre ainsi les problèmes et conflits sémantiques liés à l'évolution des ontologies de domaine. Cependant, dans cette approche, le traitement des requêtes et la validation des mappings sont complètement liés aux utilisateurs.

Les auteurs [157] proposent des agents pour la méta recherche intelligente des produits de multiples attributs, en utilisant des mappings d'ontologies et les services Web. Ce travail est dédié au champ du commerce électronique.

Leclercq et al. [160] présentent eux une solution de médiation qui repose sur une architecture multi-agents adaptée cependant aux spécificités des SIG (Systèmes d'information

géographiques) ; ils utilisent une représentation explicite de la sémantique au travers d'une ontologie commune. Mais la solution est une approche sans schéma de fédération et l'architecture permet d'établir une interconnexion dynamique aussi bien des fournisseurs que des consommateurs, c'est-à-dire que les liens sont fait dès l'exécution de la requête et ne sont pas préalablement fixés ; en outre elle permet une résolution dynamique des conflits sémantiques à l'exécution de la requête. Basée sur une ontologie commune, elle est de type médiation de contextes et mono-domaine ; celle-ci décrit un vocabulaire commun pour un domaine d'application et intègre une ontologie spatiale. Cinq types d'agents composent l'architecture générale : l'agent ontologie, l'agent interface, les agents wrapper, les agents de coopération, l'agent routeur et/ou l'agent processeur de requêtes.

Ainsi, les différents agents pour la représentation des schémas importés et pour la représentation partielle des ontologies utilisent un modèle objet étendu nommé AMUN (An Object Oriented Model For Cooperative Spatial Information). Ce modèle objet étendu est proche du modèle objet de l'ODMG (Object Data Management Group) et regroupe les notions de classes, les attributs simples, les attributs complexes, les relations et opérations. Les communications entre les différents agents se font par des envois de messages en KQML.

Le projet Kraft (Knowledge Re-use And Fusion / Transformation) [161] propose quant à lui une approche orientée agent de la médiation de contexte multi-domaines. Les interactions entre les agents sont réalisées grâce à KQML. La requête utilisateur est envoyée vers des agents médiateurs et des agents facilitateurs par l'intermédiaire via un agent wrapper qui, lui, exécute les traductions de langages, de protocoles et d'ontologies des systèmes locaux en standard de l'application Kraft (comme SQL/ODBC en SQL). Ainsi, pour résoudre les problèmes d'hétérogénéité sémantique, une ontologie commune contient la terminologie du domaine et, pour chaque source de connaissances, une ontologie locale est spécifiée. Pour aplanir ou éliminer les conflits entre ces deux ontologies, des règles de mapping sont définies pour chaque ontologie locale. Les problèmes de relations inter-ontologies sont donc schématisés sous la forme de hiérarchies de spécialisation/généralisation. Chaque nouveau domaine qui apparaît est ainsi défini sous la forme d'une ontologie qui spécifie ces concepts par rapport à ceux d'une autre ontologie. Kraft opère avec le langage Ontolingua pour représenter les définitions ontologiques des Ontolingua qui est un langage riche mais aussi relativement complexe qui rend difficile la comparaison de concepts [153], [162].

Une architecture GAACSM ou Generic Architecture based Agent for Context and Schema Mediation, est présentée par les auteurs [163]. Cette architecture est basée sur une approche multiagents pour la médiation sémantique des systèmes d'information dans un contexte multi-domaines. Elle couple donc l'approche de la médiation de schéma avec celle basée sur le contexte. Elle assure en quelque sorte une meilleure transparence, un partage sémantique fiable de données et une meilleure scalabilité.

Les approches proposées ci-dessus ont donc pour points communs la volonté d'apporter des outils, des architectures, des méthodologies au problème de l'interopérabilité sémantique des systèmes d'information. Cependant, et malgré le nombre important des solutions déjà proposées, certains aspects restent imprécis. Chacune des solutions qui existent donne des

avantages au détriment d'autres, et chacune d'elles ne traite qu'une partie des conflits de données sans tenir compte du partage de services (Applications et fonctions). En ce qui se rapporte au niveau des conflits sémantiques, la plupart des projets ne résoudront que les conflits de nommage. Aussi, c'est l'utilisateur [159] qui guide la résolution des conflits des valeurs. Cette dernière possède également une faible scalabilité face à l'évolution du nombre de participants. Dans tous ces cas sus-évoqués, et dans le cadre de la coopération de systèmes d'information hétérogènes, la presque totalité des architectures d'interopérabilité a recours au paradigme objet [160] sans tenir compte du paradigme agent.

Cependant une deuxième catégorie se focalise sur les web services et les ontologies.

**L'architecture WSMX :** Il s'agit là d'un environnement d'exécution des services Web basée sur le Web Services Modelling Ontology (WSMO)[164]. Cette architecture orientée services sémantiques WSMX de Haller et al. permet la découverte dynamique, l'invocation et la composition des services Web sémantiques, elle est surtout utilisée par les fournisseurs pour enregistrer et offrir leurs services, mais aussi par les clients pour découvrir dynamiquement et invoquer les services sélectionnés. WSMX assure également l'interopérabilité des systèmes B2B (Business to Business), qui se présente elle-même en tant que service Web. Ainsi, la médiation entre les propriétés non fonctionnelles est faite grâce à un travail manuel effectué durant la phase conception du processus métier, par la définition de règles qui permet d'identifier les équivalences entre les différentes chorégraphies. Ces mêmes règles sont stockées puis utilisées par le moteur de chorégraphie dans le but de réconcilier les gestions des séquences de messages. De ce fait, La médiation d'interfaces est effectuée en suivant le format de description de services du modèle conceptuel WSMO. Les hétérogénéités entre les concepts seront résolues par des mécanismes de raisonnement, qui établissent des correspondances entre les concepts des différentes ontologies dont se servent les fournisseurs. Le médiateur des données est constitué d'un premier composant qui identifie les similarités entre les concepts des ontologies et génère des correspondances entre ces derniers. Quant au deuxième composant, il effectue la transformation des données durant l'exécution de la composition en se servant des correspondances stockées du premier composant.

**L'architecture IRS-III :** L'architecture IRS-III [134] est un Framework permet la publication, la configuration et l'exécution des services Web hétérogènes. Cette architecture est basée également sur le Web Services Modelling Ontology ou WSMO, et comprend les différents composants suivants : Server, Publisher et client qui établissent des communications via le protocole SOAP. L'IRS-III s'occupe donc de l'implémentation de la médiation des données, la médiation des objectifs, de la médiation des processus. Ainsi, pour résoudre les conflits de communication en exploitant un ensemble de règles logiques, Cabral et al. [134] utilisent la médiation de processus. Ce médiateur lui-même utilise un interpréteur de chorégraphie, qui permet d'exécuter un service Web en créant les messages selon la requête de ce dernier et sur la base de sa description. Il mémorise également les appels exécutés par un composant appelé invocateur.

Le service Web médiateur qui utilise un composant appelé interpréteur d'orchestration prend en charge la médiation d'interfaces. Cet interpréteur d'orchestration fait aussi 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 puis il les stocke dans une ontologie temporaire qui est en fait une fusion des ontologies de départ. Grâce à laquelle il est possible de convertir les données d'une ontologie à l'autre en passant par une représentation intermédiaire. Tous ces médiateurs appartiennent à un composant appelé « broker » ou courtier. L'architecture IRS-III est utilisée par [165] comme une sorte d'infrastructure de base pour l'interopérabilité sémantique au sein d'une architecture d'un portail e-gouvernemental. Cette architecture est appelée, à juste titre « *Architecture basée services Web sémantiques pour l'interopérabilité des services E-government* », et comprend trois couches qui sont : Interaction utilisateur, Middleware et services Web. Le Middleware en quelque sorte le cœur de cette architecture et comprend l'infrastructure pour l'interopérabilité sémantique (utilisant IRSIII) et le modèle conceptuel formé de l'ontologie de services, l'ontologie de domaine Egovernment et l'ontologie de domaine des événements de la vie (Life Event). Ce portail est utilisé pour la publication de nouveaux services et l'invocation des objectifs.

L'architecture ODSOIA : M. Izza propose quant à lui [59] une architecture orientée services pour l'intégration des applications des systèmes d'information industriels appelée ODSOIA (Ontology Driven Service Oriented Architecture). Il s'agit là d'une solution intra-entreprise qui augmente le bus ESB de deux couches architecturales en plus de la couche orientée services et qui sont : la couche sémantique et la couche d'intégration. Quant à l'architecture de services, M. Izza propose une toute autre démarche de construction de services fondamentaux : services IT (Informatique) et services métier, et d'urbanisation des services obtenus avant leur publication syntaxique grâce à l'aide d'un service de publication (syntaxique). L'architecture sémantique est formée d'un ensemble d'ontologies globales, de domaines et locales, des ontologies de services et de mappings concernant les processus, les fonctions et les données, ensuite vient une publication sémantique des descriptions des services par l'utilisation un service de publication (sémantique). Et, enfin, pour décrire sémantiquement les services Web, il utilise l'OWL-S.

L'architecture d'intégration utilise les deux premières architectures. Ces architectures sont composées de services d'intégration qui sont : service broker qui permet la gestion du processus d'intégration des services fondamentaux, service de description sémantique, service de publication sémantique, service de découverte, service de médiation qui prend en charge la résolution des hétérogénéités sémantiques liées aux services fondamentaux, et service d'exécution qui permet la supervision de l'exécution des services fondamentaux. Cette architecture utilise aussi des référentiels de services et des référentiels d'ontologies. Cependant, Pour la médiation, il existe trois catégories qui sont : le service de médiation de données, le service de médiation fonctionnelle et le service de médiation de processus [59]. Dans ces différentes médiations on utilise plusieurs types d'ontologies et de mappings.

**Architecture de médiation sémantique orientée contexte :** Mrissa [124] propose pour sa part une architecture de médiation sémantique orientée contexte pour la composition de services

Web. Cette dernière comprend les couches suivantes : une couche fournisseur, une couche composition et une couche description. Dans architecture, la médiation de données est 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 des ontologies contextuelles sont associées. Ainsi, Mrissa définit l'objet sémantique grâce à la notion de contexte lui-même définit comme un objet sémantique, il obtient de cette façon une définition quoique récurrente de la notion de contexte qui constitue tout élément nécessaire à l'interprétation correcte de la donnée. Nous ferons le point de tous les travaux présentés par le tableau comparatif suivant (Tableau III.2) :

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

**Tableau III-2:**Tableau comparatif des travaux existants basés Service Web

Les approches sus-indiquées font appel à une description sémantique des services, soit par langage de description sémantique [164, 59] ou par annotation des langages existants par l'exploitation de leurs éléments d'extensibilité [124]. Cependant ces annotations ont plus d'avantages que les langages de description car elles sont relativement simples et utilisent tout en les exploitant les normes des services Web. Nous noterons aussi que, à part les travaux de [124], tous les autres supposent l'adaptation de la sémantique locale des services Web à la sémantique de l'ontologie partagée, pour rendre effective la médiation.

Pour notre part et afin de permettre à une entreprise de coopérer faiblement avec les autres organisations, et sans échec notable, nous proposons d'opter pour les annotations des descriptions des services Web à l'aide du langage récent SAWSDL [166] qui constitue une recommandation du W3C. Donc, l'approche que nous suggérons et l'architecture qui en découle sont inspirées en premier lieu des travaux de Mrissa [124]. Ensuite, nous étalons nos travaux [9] et [163] pour intégrer la médiation sémantique de service dans le contexte de l'interopérabilité inter-ERP.

### III.7. Modélisation de l'ERP par les SMA et ontologies

Les SMA permettent la distribution des entités communicantes et autonomes, dotées de compétences, et cela représente leur principal intérêt. Ainsi et, pour réaliser un SMA selon ces critères, il faut d'abord définir l'architecture de ses agents et la structuration des connaissances qui nécessite en leurs différentes activités. Le modèle que nous proposons pour l'ERP est un système multi agent du type cognitif, les systèmes d'agents cognitifs sont composés d'un petit nombre d'agents coopérants dans le but d'arriver à un objectif global via l'exécution d'opérations complexes. Ces agents ont à leur disposition une capacité de raisonnement, et ils traitent, grâce à une base de connaissances, les informations relatives au domaine d'application et à leur environnement. Cependant, chaque agent représente un système expert, qui, en fait, est un logiciel a pour objectif la modélisation du comportement d'un expert humain. La base de connaissance et le moteur d'inférence sont les deux éléments essentiels qui composent le modèle de base d'un système expert. En outre, un agent cognitif fonctionne en générale comme expliqué suit ; il acquerra des informations sur son environnement par la perception d'abord puis par la communication avec d'autres agents, et de ce fait, mettre à jour sa base de connaissances, à partir de laquelle il décide quel but va réaliser et donc suggérer et proposer les plans à suivre, ce n'est qu'après que les tâches vont être exécutées.

L'architecture présentée est constituée des agents cognitifs, chacun étant expert dans un domaine précis et nécessaire au fonctionnement de l'entreprise, ces agents font partie de cinq classes distinctes qui sont :

- L'agent du domaine Stocks : il est l'agent responsable du stock.
- L'agent du domaine Ressources Humaines : il est responsable de la gestion des ressources humaines.
- L'agent du domaine planification : il prend sous sa responsabilité la fonction de planification de l'entreprise.
- L'agent du domaine vente : il s'agit de l'agent responsable du processus de vente de produits.
- L'Agent accueil : il contrôle l'accès aux différents agents.

Cette architecture assure une pleine indépendance entre les modules de l'entreprise, et en même-temps elle les garde en liaison puisque les fonctions d'une entreprise donnée ne doivent en aucune sorte être isolées mais au contraire complémentaires. Cette synergie est donc garantie à travers la communication inter-agents par envoi de messages.

Le flux de contrôle indiqué dans le schéma, est en fait le contrôle d'accès effectué par l'agent administrateur sur les autres agents tout en garantissant la sécurité du système. En plus, l'unicité et l'intégrité de l'information sont assurées par la base de données unique et centralisé. Cette dernière est gérée par un système de gestion de base de données permet le transfert des informations depuis et vers les agents de domaines.

### III.7.1 Présentation générale de l'architecture

De la présentation générale de l'architecture, les différents agents ou composants de notre système et leurs interactions seront décrits avec précision tout en soulignant les relations établies entre les différents agents [167], [168], [169] et [170]. La figure III.1 ci-dessous illustre bien l'approche Multi-Agent de l'ERP en trois niveaux :

- Le premier représente donc l'agent interface utilisateur qui joue le rôle d'interface entre le système proprement dit et l'extérieur (que sont les fournisseurs, les clients, les banques ...etc.). Il récupère les messages envoyés par les différents agents du système et les transmet vers les agents externes. Il est aussi responsable de l'opération inverse, qui permet de récupérer les messages envoyés par les différents agents externes et les transmettre aux agents du système.
- Le deuxième niveau est composé de l'agent de domaine Achat/Appro, de l'agent de domaine Vente/Commercial, de l'agent de domaine Comptabilité/Finance ...etc, qui assurent la communication et liaison entre les différents domaines ainsi qu'avec l'agent interface utilisateur, ce niveau garantit l'établissement d'une interaction directe avec les autres agents de ces domaines pour réaliser une tâche commune.
- Le troisième niveau : il représente quant à lui, tous les agents de services pour chaque domaine fonctionnel tout en permettant la réalisation des fonctions bien déterminées tels que par exemple le stock, l'achat, la comptabilité ... etc.

Comme modèle de communication, nous avons opté pour la communication par l'envoi de messages qui assure aux agents le partage de la charge d'un problème en le subdivisant en sous problèmes. Dans les systèmes multi-agents fondés sur la communication par l'envoi de messages, chaque agent possède une représentation propre et locale de l'environnement qui l'entoure. Chaque agent interrogera alors les autres agents sur cet environnement ou leur enverra des informations sur sa propre perception des choses. La communication se fait soit en mode point à point ou en diffusion [172].

Dans notre proposition nous préférons choisir le mode point à point pour la transmission de l'information puisque le type d'agent employé est l'agent cognitif. L'agent émetteur du message doit connaître en précision l'adresse de(s) agent(s) destinataire(s).

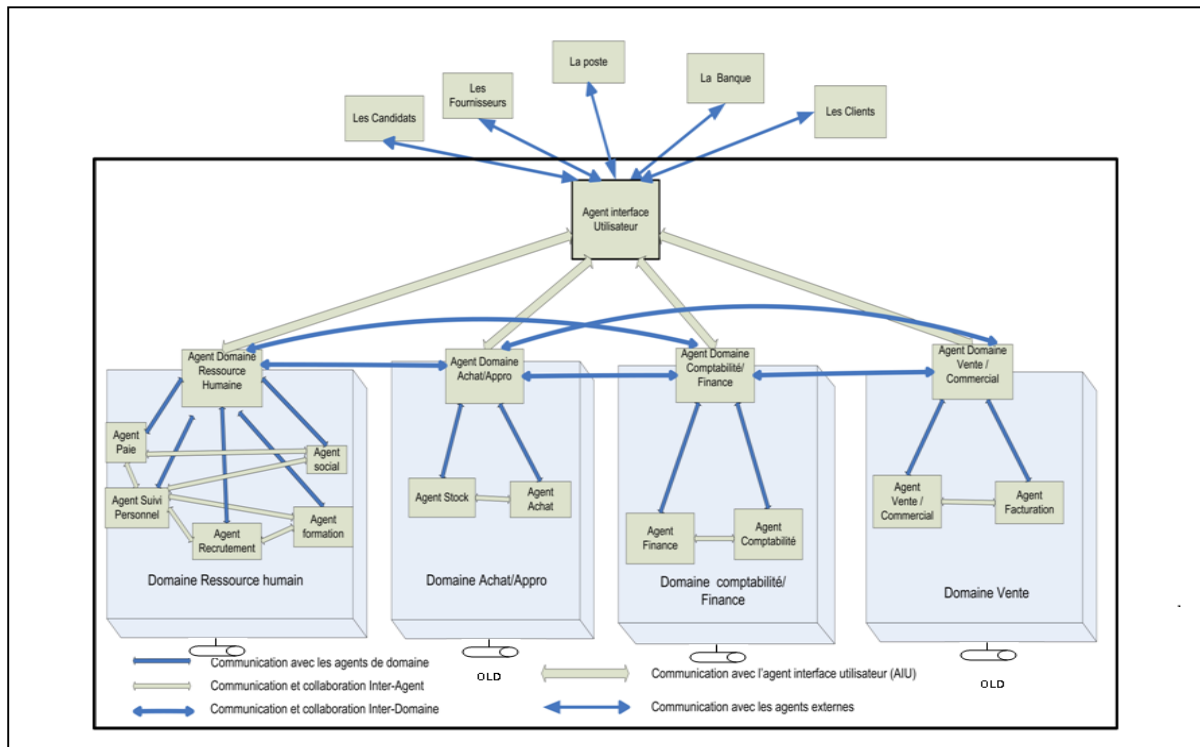


Figure III-01: Modélisation à l'ERP basée sur les agents et ontologies

## III.7.2 Agents et leurs rôles

### III.7.2.1. Agent interface utilisateur

Cet agent garantit et assure l'interface complète entre le système et l'extérieur. Il prend en charge la requête introduite par les agents externes (Client, fournisseur, Banque ... etc.). L'agent interface utilisateur n'est donc pas seulement chargé de représenter l'utilisateur dans le système, mais il est également le point d'entrée des requêtes extérieures vers l'ensemble des agents de domaines. Son rôle consiste à récupérer les messages envoyés par les différents agents du système et les transmettre aux agents externes.

### III.7.2.2. Agents de domaine Achat/Appro

Ces agents fournissent des services et servent en premier lieu de rationaliser la gestion des articles et de matériels grâce à une surveillance permanente et en ligne du stock. Ensuite, ils attribuent les matériels aux ordres de travail et génèrent automatiquement les listes à servir pour les matériels. Enfin, ils identifient les articles pour lesquels on établit les demandes d'achat en se basant sur des niveaux de stock existants, sur les prévisions et les réservations en cours. Ils calculent le coût des services en utilisant des prix fixes ou en se basant sur le système pièce et main d'œuvre. Ils contrôlent aussi les processus de rapprochement Commande/facture pour les matériels en achat direct, la main d'œuvre externe et les services. Ils enregistrent les contrats standards et les commandes ouvertes passées aux fournisseurs en même temps que les accords

et condition financières associées. Enfin, ils surveillent l'avancement des devis concernant les matériels et services.

**Agent Achat :** la mission de cet agent est de gérer les achats de l'entreprise, et cela de la demande d'achat jusqu'à la réception et la comptabilisation. Il est responsable des matériels gérés en stock, des matériels hors stocks ou consommations directes, de l'investissement (Actifs) et enfin des prestations de services.

**Agent Stock :** Comme son nom l'indique, il est responsable du stockage et du management de l'ensemble des marchandises ou des articles accumulés en attendant leur utilisation ultérieure plus ou moins proche et qui permet donc l'alimentation les utilisateurs au fur et à mesure de leurs besoins. Il gère tous types de stocks comme les matières premières, Les produits fabriqués et/ou réparés, les pièces de rechange et les produits consommables.

### III.7.2.3. Agents de domaine Comptabilité/Finance

Ces agents calculent et déterminent à tous les moments l'état financier présent de l'entreprise afin d'aider les décideurs à la prise de décisions pertinentes et opportunes quant aux prévisions du moment et futures. Ils sont responsables de l'automatisation du processus de mise en place des budgets et de la saisie, du suivi, du contrôle et de l'analyse ultérieurs des dépenses inhérentes aux travaux de maintenance. Ils établissent un lien entre les structures financières de l'entreprise, celles de son organisation et les arborescences d'actifs physiques. Ces agents calculent donc une suite d'indicateurs financiers et de performance. Tout cela permet de communiquer et de collaborer avec les autres agents du système. Ils sont chargés aussi de récupérer et de répondre aux requêtes des agents externes par l'intermédiaire de l'agent utilisateur.

**Agent Comptabilité :** Cet agent permet d'enregistrer dans des comptes comptables les flux de l'entreprise (les opérations réalisées par les autres agents dans l'entreprise : Agent Achat, Vente, Stock ...etc.) afin de déterminer le résultat de l'exercice (bénéfices ou pertes) et de présenter la situation patrimoniale de l'entreprise.

**Agent Finance :** Cet agent détermine l'état financier actuel de l'entreprise pour aider les décideurs de la prise de décisions importantes concernant la prévision actuelle et future.

### III.7.2.4. Agents de domaine Vente/Commercial

C'est un agent principal qui permet aux agents de domaine Vente/Commercial de communiquer et de collaborer avec les autres agents du système. Il est chargé aussi de récupérer et de répondre aux requêtes des agents externes par l'intermédiaire de l'agent utilisateur.

**Agent Facturation :** Le rôle de cet agent est l'établissement des factures pour indiquer aux clients les détails du prix à payer.

**Agent Vente/Commercial :** Cet agent présente les volets commerciaux de l'entreprise auprès ses clients et la marché. Il assure toutes les opérations commerciales important à l'entreprise.

### III.7.2.5. Agents de domaine Ressources humaines

C'est un agent principal qui permet aux agents de domaine Ressource humaine de communiquer et de collaborer avec les autres agents du système. Il est chargé aussi de récupérer et de répondre aux requêtes des agents externes par l'intermédiaire de l'agent utilisateur.

**Agent social :** Cet agent produit les principaux indicateurs nécessaires à l'établissement du bilan social, tels que la pyramide des âges de l'ancienneté, les tableaux mensuels des effectifs, l'historique des mouvements (départs, embauches, mutations).

**Agent recrutement :** Cet agent permet de gérer de façon souple et efficace toutes les phases opérationnelles de prévision, de prospection, de sélection et d'engagements, liés à l'administration du recrutement interne et externe.

**Agent formation :** Cet agent permet de gérer les processus d'aide à la planification, la réalisation et l'administration de la formation, complètement intégrés et mis en relation avec les autres agents de domaine ressource humaine.

**Agent suivi personnel :** l'agent suivi personnel a comme rôle de suivre tous les mouvements du personnel, tels que l'évolution des effectifs, les absences du mois, Effectif permanent, temporaire, et global, prévisions des départs en retraites, prévisions des départs en congé.

**Agent paie :** Cet agent permet de traiter et suivre la paie tels que : l'historique de paie, l'historique des congés, le détail des congés et absences, la situation des prêts et avances et les statistiques de paie (Taux et indicateurs).

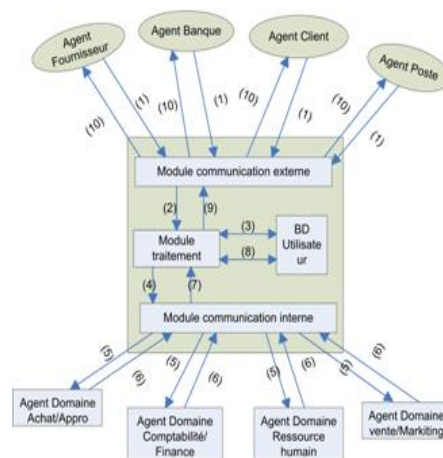
### III.7.3. Architecture fonctionnelle des agents du système

Parmi les innombrables et différents agents, nous nous limitons à l'agent interface utilisateur et à deux agents de domaines Achat/Appros pour représenter leurs architectures fonctionnelles.

#### III.7.3.1. Architecture Fonctionnelle : Agent interface utilisateur

Cet agent sert en quelque sorte de complémentarité entre les agents du système et l'extérieur. Cet agent d'interface comprend un module de communication externe, une base de données utilisateur, un module de traitement et un module de communication interne dont nous détaillons brièvement les rôles ci-dessous :

- **Module communication externe** : il sert à récupérer et rassembler les requêtes des agents externes pour les envoyer vers les agents du système.
- **Base de données utilisateur** : elle contient toutes les croyances de l'agent interface utilisateur sur tous les autres agents et sur les agents externes (Fournisseur, Client, Banque, ...etc.).
- **Module traitement** : il permet l'acheminement des messages vers les agents concernés.
- **Module communication interne** : garantit et assure l'interaction avec les différents agents de domaine.



**Figure III-2:** Architecture Fonctionnelle : Agent interface utilisateur

Nous donnons ci-dessous les détails du fonctionnement :

1. L'agent externe s'occupe de l'envoi de la requête au système qui
2. est récupérée par le module communication externe qui la renvoie au
3. module traitement qui à son tour analyse et vérifie la requête avec la base de données utilisateur pour l'envoyer vers les agents de domaine concerné
4. Le module traitement passe à son tour la requête au
5. module communication interne qui renvoie la requête vers

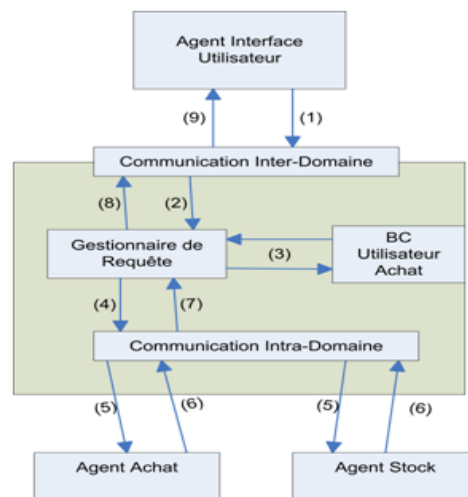


6. l'agent de domaine concerné (Ex : Agent de domaine Achat/Appro) qui récupère la requête et l'envoie vers
7. le module communication interne, à son tour, passe le message au
8. module traitement qui analyse et vérifie la réponse avec la base de données utilisateur pour déterminer l'agent externe concerné. Il passe ensuite le message au
9. module traitement qui à son tour passe le message au module de communication externe qui au final renvoie le message à
10. l'agent externe concerné.

### III.7.3.2. Architecture Fonctionnelle : Agents de domaines Achat/Appro

**Architecture Fonctionnelle de l'agent de domaines Achat/Appro :** Cet agent présente une interface qui permet aux agents de domaine Achat/Appro de collaborer et de communiquer avec les autres agents du système. Il comprend les modules suivants :

- **Module communication inter-domaine :** il s'agit de l'interface de l'agent qui interagira avec l'agent interface utilisateur et les autres agents de domaines.
- **BC Utilisateur Achat :** il contient toutes les croyances de l'agent de domaine Achat/Appro sur les autres agents du système et sur les agents externes.
- **Gestionnaire de requête :** il procède à l'analyse de la requête et l'envoie vers les agents concernés.
- **Module communication intra-domaine :** il est en interaction directe avec les différents agents de domaine Achat/Appro (Ex : Agent stock).



**Figure III-3:** Architecture Fonctionnelle : Agent domaine Achat/Appro

Les détails du fonctionnement sont expliqués brièvement comme suit :

1. L'agent interface utilisateur renvoie la requête (Facture, BL) à l'agent de domaine Achat/Appro via le module communication inter-domaine ;
2. Ce dernier récupère à son tour la requête et l'envoie au gestionnaire de requête ;
3. Le gestionnaire de requête analyse la requête et la vérifie avec la base Utilisateur Achat avant de l'envoyer à l'agent concerné ;

4. Le gestionnaire de requête envoie la requête au module communication intra-domaines ;
5. Le module communication intra-domaine reçoit la requête qu'il transmet à son tour à l'agent concerné (Ex : La facture à l'agent Achat ou bien la BL à l'agent Stock) ;
6. L'agent d'achat ou agent de stock selon que l'un ou l'autre soit concerné traite la requête et donne la réponse au module communication intra-domaines ;
7. Le module communication intra-domaines reçoit la réponse et la communique, à son tour, au gestionnaire de la requête ;
8. Ce dernier analyse la réponse et la transmet au module communication inter-domaine ;
9. En fin, l'agent utilisateur reçoit la réponse de la requête et l'envoie vers l'agent externe concerné.

### 7.2.2. Architecture Fonctionnelle : Agent Achat

Cet agent est responsable de la gestion des achats de l'entreprise, de la demande d'achat à sa réception et sa comptabilisation. Son rôle comporte quatre modules : 1- Un module de communication, 2- Un moteur de raisonnement, 3- Une Base de Connaissance achat, 4- Une BD achat. Leurs rôles sont définis brièvement comme suit :

- **Module de communication** : il assure en fait les interactions des messages avec l'agent de domaine Achat\Appro et avec les autres agents des services.
- **Moteur de raisonnement** : il contrôle et supervise l'exécution de ses actions.
- **Base de Connaissance Achat** : comprend un ensemble de critères définissant les conditions des achats, autrement dit les critères aux choix des produits (prix, les marques et leurs caractéristiques), et aussi les critères liés aux choix des offres de fournisseurs (prix du produit, la garantie, la qualité, la gamme de produits, délai de livraison, la disponibilité).
- **BD achat** : contient et englobe toutes les données inhérentes aux achats reparties comme suit :
  - ✓ Toutes les données concernant les fournisseurs (nom, localisation, tel, fonction, code comptable, etc.) ;
  - ✓ Toutes les informations relatives à la demande d'achat (N°, Demandeur, Article, Quantité, Prix unitaire, Date, Prix total, etc.) ;
  - ✓ Toutes les informations concernant les contrats établis (Article, Prix, délai de livraison, Durée du contrat, etc.) ;
  - ✓ Toutes les données relatives aux articles et produits à acheter (Nom, famille, sous famille, tarif de vente, tarif d'achat, etc.) ;
  - ✓ Toutes les informations relatives au bon de commande (N°, date, fournisseur, Article, Quantité, marque, Prix, etc.).

Dans ce qui suit, on essaiera d'apporter plus de clarification au fonctionnement de l'agent achat dans le cas d'un établissement de contrat ou processus d'appel d'offre. Ce cas précis illustre le fonctionnement de l'agent achat lors d'un appel d'offre lancé par une entreprise pour acquérir des produits ou des articles manquants, On peut donc souligner que le résultat de ce processus est l'établissement d'un contrat entre d'un côté l'entreprise et de l'autre côté le fournisseur. Nous décrivons son fonctionnement à travers les étapes suivantes :

**1. Identification d'un besoin :** l'agent demandeur qu'il soit l'agent Stock ou l'agent de domaine Achat/Appro, signale les produits ou les articles manquants par l'envoi d'un message explicite (DA) à l'agent achat, cela dans ce cas où le produit demandé ne figure pas dans un contrat entre l'acheteur et le fournisseur).

**2. Recherche d'un produit :** l'agent achat analyse tout en vérifiant si la demande d'achat est en conformité avec les critères de base de connaissance concernant les produits à acheter, qu'il trouve que le résultat de l'ensemble des produits est conforme et satisfaisant les critères de choix (prix, les marques et leurs caractéristiques), puis il les stocke dans la BD d'achat.

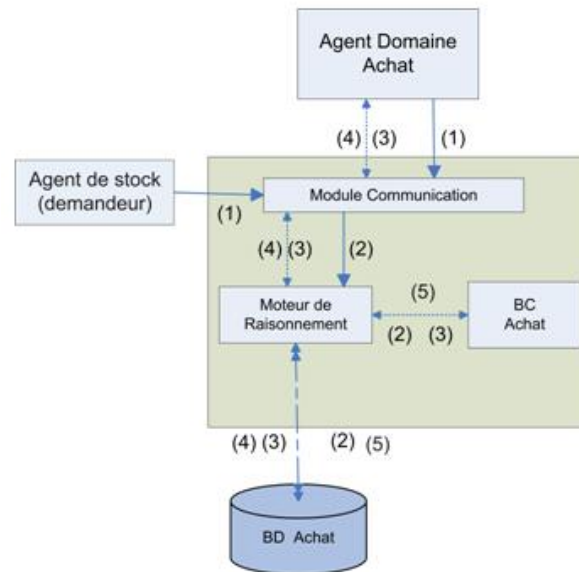
**3. Recherche d'un fournisseur (Comparaison des offres de fournisseurs) :** les choix des produits et leurs caractéristiques étant établis, l'agent achat lance la procédure d'appel d'offre. Il reçoit en retour les offres des fournisseurs via le module de communication et les stocke dans la BD d'achat. L'agent achat procède à une comparaison entre les différentes offres proposées par les fournisseurs en se basant sur les critères définis dans la base de connaissance (prix du produit, la garantie, la qualité, la gamme de produits, le délai de livraison, la disponibilité) tout cela dans le but de chercher le meilleur fournisseur qui fournira les meilleurs produits à l'entreprise.

**4. Négociation Acheteur-fournisseur :** Cette quatrième étape a trait à la négociation entre l'agent achat et le fournisseur en se mettant d'accord sur le prix du produit et les autres termes et condition de transaction (les contraintes de temps, les contraintes géographiques, la modalité de paiement). Le module de communication a ici un rôle essentiel dans cette étape qui permet d'échanger les messages entre l'agent achat et le fournisseur en passant par l'agent de domaine Achat/Appro et l'agent utilisateur. Les résultats de cette quatrième étape sont :

- 1- L'établissement des contrats entre l'acheteur et le fournisseur,
- 2- L'enregistrement des contrats dans la BD d'achat et en fin,
- 3- l'achat et la livraison du produit.

**5. L'évaluation après-achat :** la mission et la responsabilité de l'agent achat se prolongent après l'achat effectif du produit : Service après-vente et des évaluations relatives aux achats en se basant bien entendu sur l'historique des achats établis.

Le schéma ci-dessous présente le fonctionnement de l'Agent Achat et/ou le acheminement de la mission de l'agent achat dans le cas d'un appel d'offre.



**Figure III-4:**Architecture Fonctionnelle : Agent Achat

### III.3. Conclusion

Dans ce chapitre, nous sommes efforcés de présenter les différentes classifications des hétérogénéités sémantiques des services Web, les approches de description de la sémantique de ces derniers, les médiations qui résolvent les conflits inhérents, et une classification des approches d'interopérabilité et d'intégration des services Web pour dresser l'interopérabilité sémantique des systèmes d'information à base d'ERP. A l'issue de cette classification, nous avons retenu surtout l'utilisation des approches hybrides basées sur les services web, ontologie et système multi-agents et au final nous avons présenté une étude comparative des travaux existants.

Le chapitre suivant sera consacré à notre contribution qui concernera le niveau d'interopérabilité sémantique, tout en proposant une approche et une architecture flexible et orientées services Web pour l'interopérabilité des ERPs.

# Chapitre IV : Une approche sémantique pour l'interopérabilité des ERPs

---

## IV.1. Introduction

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

Nous allons consacrer ce chapitre pour décrire l'approche et l'architecture proposées pour mettre en œuvre les mécanismes de l'interopérabilité sémantique des ERPs. Pour assurer l'interopérabilité au niveau sémantique, nous avons retenu l'utilisation d'une approche hybride basée sur les ontologies, les systèmes Multi-agents et les services Web. Aussi, nous proposons l'enrichissement des descriptions des services Web par l'information sémantique décrite par une architecture conceptuelle basée sur la notion d'ontologies, et la notion de contexte. Les ontologies obtenues et les annotations des descriptions, seront utilisées pour la découverte sémantique et la médiation des services Web pour assurer leur interopérabilité. L'architecture qui en découle est structurée en couches dont le noyau est une infrastructure de médiation. Dans le cadre de cette thèse nous détaillons le sous modèle sémantique des données et les échanges entre les services Web.

## IV.2. Motivations et objectifs de l'approche proposée

Dans tous les travaux présentés (Voir les sections III.5 et III.6) et afin de remédier à ces lacunes, notre objectif est d'intégrer et de combiner les ontologies, les systèmes multi-agents pour mettre en œuvre les mécanismes de l'interopérabilité sémantique entre les données échangées entre les services web et ayant les caractéristiques suivantes :

- ✓ Permettre aux différents systèmes participant à la coopération de ne pas modifier leurs fonctionnalités habituelles en respectant l'autonomie de l'ERP ;
- ✓ Permettre à un utilisateur, client et fournisseur d'accéder de manière transparente à des ressources fournies par les ERPs ;
- ✓ Construire un système de médiation sémantique flexible, permettant l'ajout de fonctionnalités et de services ;
- ✓ Résoudre la majorité des conflits sémantiques par le recours aux nouvelles technologies, à savoir les ontologies ;
- ✓ Permettre à un nouveau module fonctionnel -non couvert par l'ERP- ou bien un autre ERP qui fournit son contexte, de trouver l'ERP, les données et les services partagés par le module fonctionnel, de les intégrer dynamiquement à l'ERP et de les utiliser de

manière transparente.

### **IV.3. Architecture multicouche de l'approche proposée**

L'architecture générale proposée (Cf. Figure IV.2) [170], [171] est en multi couches, dont chaque couche possède ses propres fonctionnalités avec des responsabilités précises. Elle a pour objet essentiel de résoudre le problème d'échange d'informations dans des organisations possédant des ERPs hétérogènes. Voici une description plus précise des différentes couches et de leurs relations.

#### **IV.3.1. Couche ERP**

C'est la couche de base, composée d'un ensemble de métiers et une base de données ERP unique et centrale, ayant une variété des données collectées à partir de n'importe quel domaine de l'ERP. Elle assure la communication et la collaboration entre les différents métiers tels que le métier de l'entreprise Achats/Appros, le métier Vente/Commercial, le métier Comptabilité/Finance...etc. Cette couche joue aussi un rôle fondamental pour la communication par échange d'information avec les autres couches des différents ERP via la couche d'interopérabilité à partir de la base de données de l'ERP. Cette couche comporte un ensemble de composants illustré comme suit :

- ✓ Base de données unique et commune.
- ✓ Ensemble d'agents situés : dans notre architecture comporte une liste d'agents cognitives. Ces agents se communiquent entre eux à travers l'envoi des messages.
- ✓ Ontologie propre à l'ERP : regroupe une collection de termes et de définitions du métier de l'entreprise. Elle regroupe un nombre de sections principales, à savoir, les activités et processus, l'organisation, la stratégie et le marketing. Elle est construite par fusion des ontologies de domaines fonctionnels de l'ERP.
- ✓ Une interface inhérente aux services web. L'intégration des Web services pour offrir tout ce qui est nécessaire et accessible à travers le Web.

#### **IV.3.2 Couche de l'interopérabilité**

Cette couche modélise le processus d'interopérabilité entre les ERP, qui passe par un ensemble des étapes successives. Elle comporte un ensemble de composants, à savoir :

- ✓ Un agent de médiation qui englobe l'ontologie de domaines propres aux ERPs, et celle des aspects conflictuels.
- ✓ Un registre contient les services web. Il regroupe aussi l'ensemble de services de médiation publiés pour chaque aspect conflictuel. La création de service peut être intuitive pendant la publication de service web. Elle peut être basée sur les relations existantes entre les classes instanceable d'un aspect conflictuel dans l'OAC (Ontologie des Aspects Conflictuels).

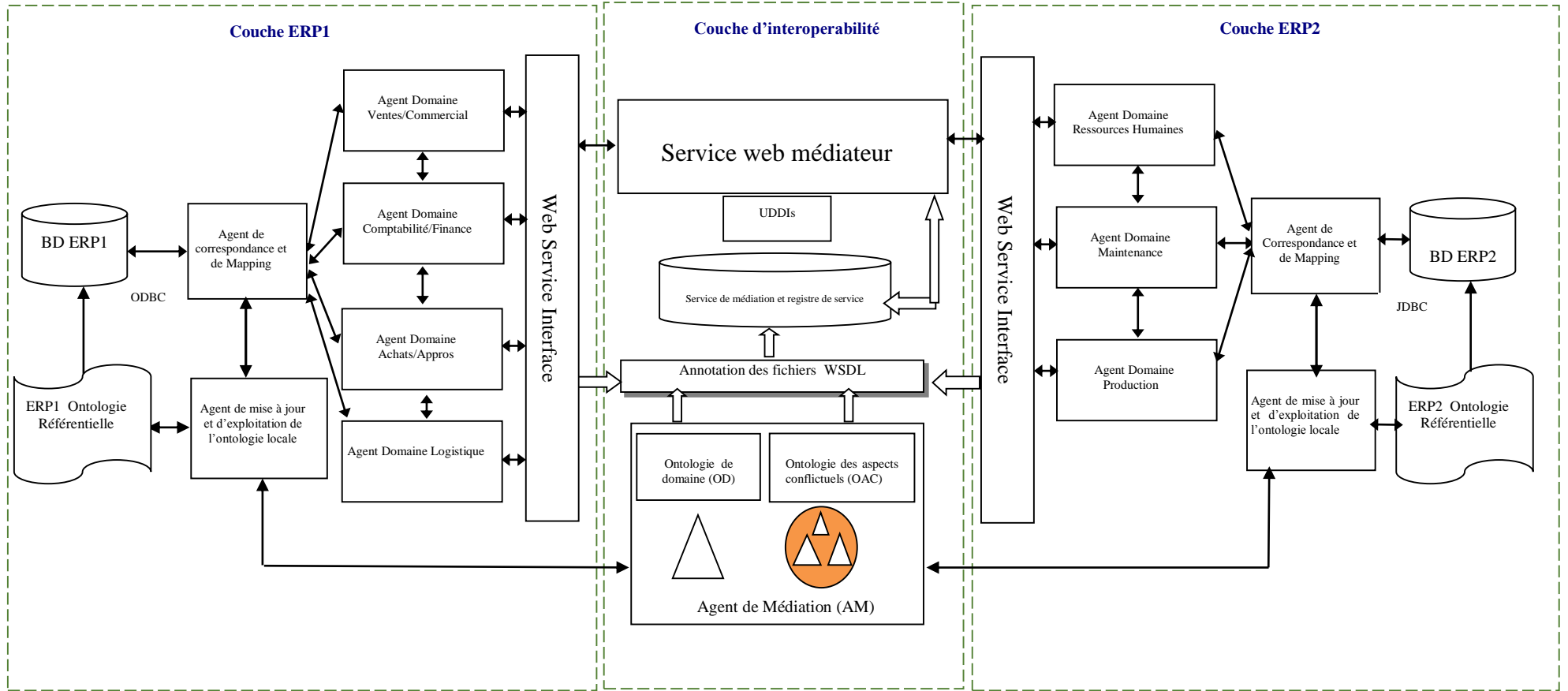


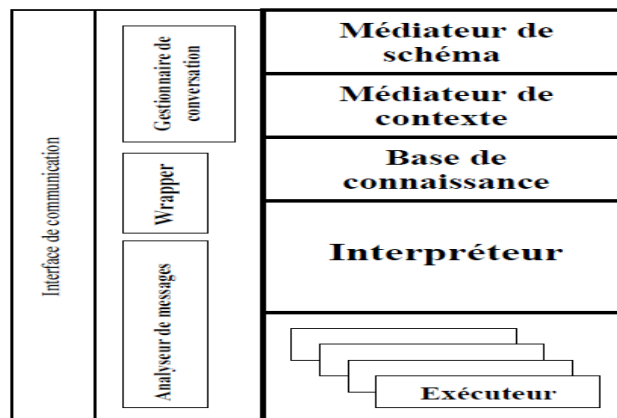
Figure IV-1: Architecture d'interopérabilité sémantique

#### IV.4. Architecture fonctionnelle des agents :

Parmi les différents agents, nous allons seulement choisir l'agent de mise à jour et de correspondance et l'agent de mapping pour représenter leurs architectures fonctionnelles.

##### IV.4.1. Agent de mise à jour et exploitation de l'ontologie locale de l'ERP

Cet agent exploite et met à jour l'ontologie locale de l'ERP en utilisant les ontologies locales de chaque domaine fonctionnel. Il gère tout accès aux ontologies des domaines fonctionnels. Ces dernières construisent par fusion l'ontologie de l'ERP et celles des applications externes de l'ERP. Elle regroupe une collection de termes et de définitions du métier de l'entreprise. Il envoie les mises à jour vers l'ontologie globale de l'agent médiation.



**Figure IV-2:** La structure interne de l'agent de mise à jour et exploitation

Les composants de l'agent de mise à jour et de correspondance intelligent sont comme suit :

- ✓ L'interface de communication : permet à un agent de communiquer avec l'environnement en utilisant l'interface de communication, l'agent ME envoie des requêtes aux agents de domaines fonctionnels qui lui sont associés et reçoit des réponses, il répond à des requêtes des utilisateurs. Il coopère aussi avec les autres agents du système.
- ✓ L'analyseur de messages : analyse les messages reçus, et détermine leurs types.
- ✓ Le gestionnaire de conversations : s'assure de la bonne compréhension du message en vérifiant que le contexte du message est connu. Il identifie le protocole de coopération utilisé pour le dialogue entre l'émetteur et le récepteur et il contrôle l'état d'avancement de la conversation.
- ✓ Le wrapper : joue principalement trois rôles :
  - La traduction des sous requêtes écrites en utilisant le langage adopté vers le langage propriétaire à la source locale ;
  - La traduction du schéma de la source locale vers un schéma écrit en utilisant le modèle de données commun (OWL) ;
  - La traduction des résultats obtenus de la source locale vers le format commun XML.
  - La base de connaissance : contient les connaissances de l'agent sur l'environnement, cette base est toujours mise à jour par l'agent durant sa communication avec les autres agents. Elle contient principalement le contexte du système d'information local, le nom du domaine, l'ontologie qui décrit le nom du domaine, l'ontologie des conflits

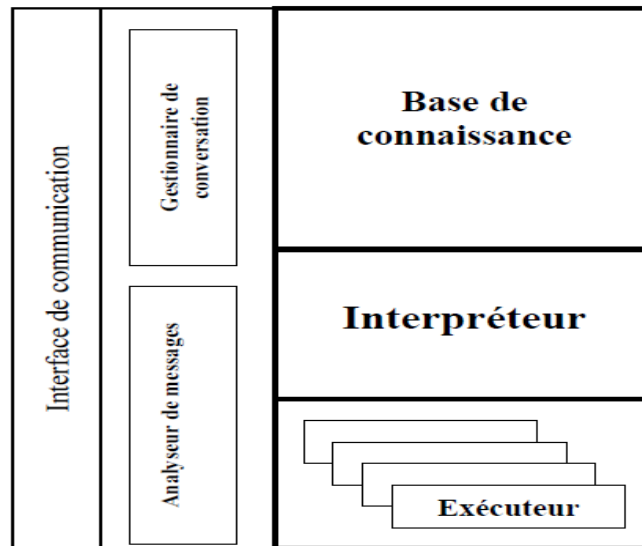


sémantiques de valeurs, ses croyances et des règles de correspondances entre des schémas d'autres agents qui sont découverts par l'agent.

- ✓ Le médiateur de schéma : réalise la médiation de schéma.
- ✓ Le médiateur de contexte : réalise la médiation de contexte
- ✓ L'interpréteur : fait la gestion de synchronisation entre les différents threads. Ainsi selon le type du message reçu, il active et appelle le composant nécessaire pour le traitement des requêtes.
- ✓ L'exécuteur : le rôle d'un exécuteur est multiple selon la mission pour laquelle il est créé. Par exemple : un Exécuteur est créé par le médiateur de schéma pour exécuter une requête de recombinaison des résultats.

#### IV.4.2. Agent de correspondance et de mapping (Agent de routage)

Cet agent permet de connecter entre les agents et la base de données ERP. Il est apte à établir une conversation avec les différents agents via l'agent de mise à jour et exploitation de l'ontologie locale de l'ERP. Il peut faire toute recherche des agents pertinents pour une coopération.



**Figure IV-3:** La structure interne de l'agent de correspondance et de mapping

Les composants de l'agent de correspondance et de mapping sont comme suit :

- ✓ L'interface de communication : permet à un agent de communiquer avec l'environnement. En utilisant l'interface de communication, un agent de mapping coopère aussi avec les autres agents du système multi agent.
- ✓ L'analyseur de messages : analyse les messages reçus, et détermine leurs types.
- ✓ Le gestionnaire de conversations : s'assure de la bonne compréhension du message en vérifiant que le contexte du message est connu. Il identifie le protocole de coopération utilisé pour le dialogue entre l'émetteur et le récepteur et il contrôle l'état d'avancement de la conversation.
- ✓ La base de connaissance : contient les connaissances de l'agent de mapping sur l'environnement, cette base est toujours mise à jour par l'agent de mapping durant sa

communication avec les autres agents. En particulier, cette base contient pour chaque agent intelligent, une liste ordonnée de ses agents intelligents fournisseurs qui ne sont pas encore découverts par lui et qui sont proches sémantiquement à lui (figure IV.3). Si un agent de domaine a obtenu le plus grands nombre de réponses aux ses requêtes de agents fonctionnels relatifs à l'ERP2, alors l'agent de mapping place les agents de domaine de l'ERP2 à l tête de la liste ordonnée des agents de domaine de l'ERP1. De ce fait les agents de domaine de l'ERP2 sont le fournisseur sollicité pour la prochaine intégration dynamique effectuée par les agents de domaine de l'ERP1.

- ✓ L'interpréteur : fait la gestion de la synchronisation entre les différents threads. Ainsi selon le type du message reçu, il active et appel le composant nécessaire pour le traitement des requêtes.
- ✓ L'exécuteur : le rôle d'un exécuteur est multiple selon la mission pour laquelle il est créé. Par exemple, un exécuteur est créé par le médiateur de schéma pour exécuter une requête de recombinaison des résultats.

### IV.4.3 Construction du Modèle Sémantique des Données

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

#### IV.4.3.1 Notion de contexte et d'objet sémantique

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

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

- ✓ Un concept (c), qui définit la famille ontologique à laquelle cette donnée appartient, c'est-à-dire la classe abstraite dont cette donnée est une instance.
- ✓ Un type (t), qui définit le genre de contenu de la donnée (schéma XML simple / complexe).
- ✓ Une valeur (v), qui est la donnée elle-même, c'est une instance du type de la donnée.
- ✓ Un contexte (C), qui apporte des précisions sur l'interprétation de la donnée.

Donc, d'après [3] un objet sémantique S est défini comme suit :  $S = (c, v, t, C)$

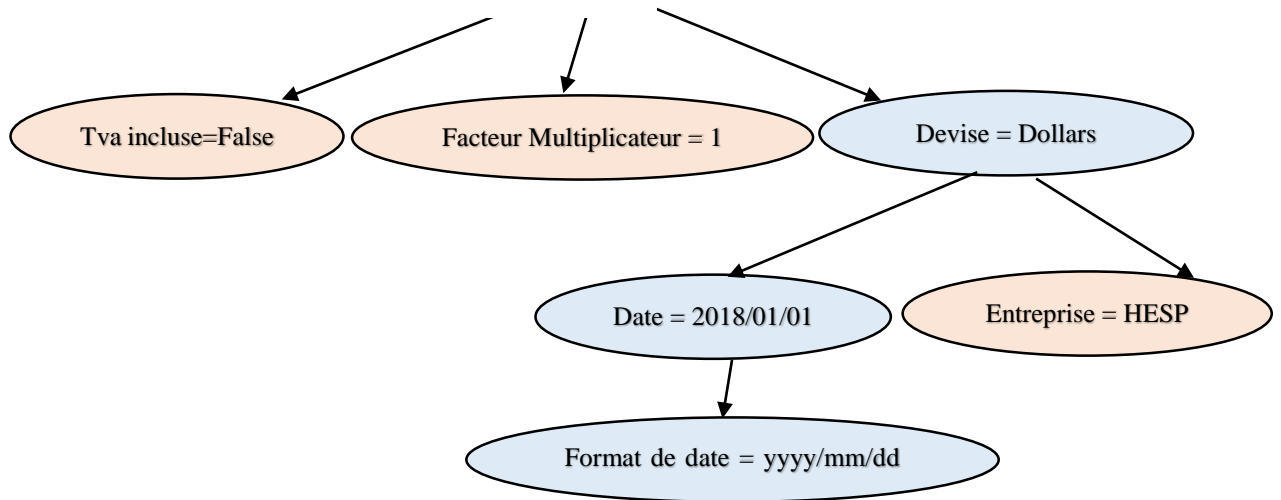
Le contexte (C) est lui-même composé d'objets sémantiques appelés **modifieurs**. Ces derniers ont la capacité de modifier la signification de l'unique objet sémantique auquel ils sont associés.

Une description formelle d'un contexte C est :  $C = \{(c_1, v_1, t_1, C_1), \dots, (c_n, v_n, t_n, C_n)\}$ ,  $n \in \mathbb{IN}$  Où  $(c_i, v_i, t_i, C_i)$ ,  $1 \leq i \leq n$ , sont *les modifieurs* qui décrivent les différentes propriétés sémantiques de S. Ceci permet de former des descriptions récursives et de représenter le contexte des objets sémantiques avec des structures arborescentes composées de **modifieurs**. Ces derniers peuvent être statiques ou dynamiques.

- ✓ **Modifieur statique** : est indépendant des autres modifieurs. Il possède une valeur explicite qui apporte une information quant à l'interprétation de l'objet sémantique.
- ✓ **Modifieur dynamique** : est dépendant d'un ou de plusieurs autres modifieurs. Il possède une valeur qui peut être déduite, par une fonction ou un ensemble de règles logiques des valeurs prises par d'autres modifieurs appartenant au même contexte.

**Exemple** : Prenant l'exemple de l'objet sémantique S concernant le concept « prix » avec son contexte (figure IV.4).

S= (ns : Montant, 10 000, xsd : double, **Contexte**)



**Figure IV-4:**Exemple d'un objet sémantique et son contexte

Les modifieurs en bleu sont dynamiques. La valeur du **modifieur** *Format de date* est déduite à partir de la valeur du **modifieur** *Entreprise* appartenant au même contexte. La règle logique permettant cette déduction peut être décrite comme suit : «Si *Entreprise = HESP*, alors *Format de date= yyyy/mm/jj*». Cependant les autres modifieurs sont statiques.

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

#### IV.4.4. Mécanisme de coopération dans l'approche

Dans cette section, nous rappelons les différentes méthodes utilisées pour la coopération entre les agents : le regroupement et la multiplication, la communication, la spécialisation, la collaboration par partage de tâches et de ressources, la coordination d'action, la résolution des conflits par arbitrage et négociation. La méthode de coopération entre les agents est considérée comme une caractéristique primordiale dans un système multi-agents. En effet, une résolution distribuée d'un problème est le résultat de l'interaction coopérative entre les agents.

Dans notre approche, nous avons adopté « la communication par l'envoi des messages » comme une méthode de coopération entre les agents afin de réaliser les tâches 'échange de

données entre les services web. Chaque agent décide son comportement pour satisfaire le but global du système.

#### IV.4.5. Communication des agents dans l'approche

Puisque les agents de notre approche sont de type cognitif, la communication entre les agents de notre système s'effectue par l'envoi de messages. Elle permet aux agents de partager la charge d'un problème en le subdivisant en sous problèmes.

Les systèmes multi-agents fondés sur la communication par l'envoi des messages se caractérisent par le fait que chaque agent possède une représentation propre et locale de l'environnement qui l'entoure. Chaque agent va alors interroger les autres agents sur cet environnement ou leur envoyer des informations sur sa propre perception des choses.

Le langage de communication que nous avons adopté pour l'envoi de messages entre les agents de l'approche est le langage ACL (Agent Communication Language) dont la spécification de FIPA (Foundation for Intelligent Physical Agents) permet une interopérabilité maximale entre les agents. D'ailleurs, la communication par l'envoi de messages se fait en deux modes : soit en mode point à point, ou en mode diffusion. Dans notre proposition nous utilisons le mode « point à point » pour la transmission des messages entre les agents car le type d'agent utilisé dans notre approche est de type cognitif, l'agent émetteur de message doit connaître avec précision l'adresse de(s) agent(s) destinataire(s). La syntaxe d'un message FIPA-ACL est illustrée comme suit :

```
(request
:sender Agent_A
:receiver Agent_B
:content
(... )
:in-reply-to action
:reply-with
reponse
:language FIPA-SL0)
```

Figure IV-5 : Syntaxe d'un message FIPA-ACL

#### IV.4.6. Scenario de fonctionnement des agents du système

Pour comprendre le fonctionnement de l'architecture, il est utile de présenter un cas d'école réel. Dans notre contexte, le Groupe ENSP est une société de services pétroliers et parapétroliers. Il regroupe plusieurs filiales et participations éloignées en partenariats avec des entreprises étrangères, à savoir HESP, BJSP, MESP, WESP et BASP. Chacune d'eux possède un ERP et un réseau informatique autonome. Afin de présenter un reporting fiable à la Direction Générale du groupe, il est indispensable de rendre ces progiciels interopérables entre eux et résoudre les différents conflits sémantiques tels que les concepts d'une écriture comptable dans deux ERPs.

Format de date : dd/mm/yyyy ou yyyy/mm/dd.

Nature : Investissement / Charge (montant <30000 DA).

Classe : 2/6.

Compte : 2xxxx/6zzzzzzz.

Montant : DA/Dollars.

TVA Incluse : True/False

Entreprise : ENSP/HESP

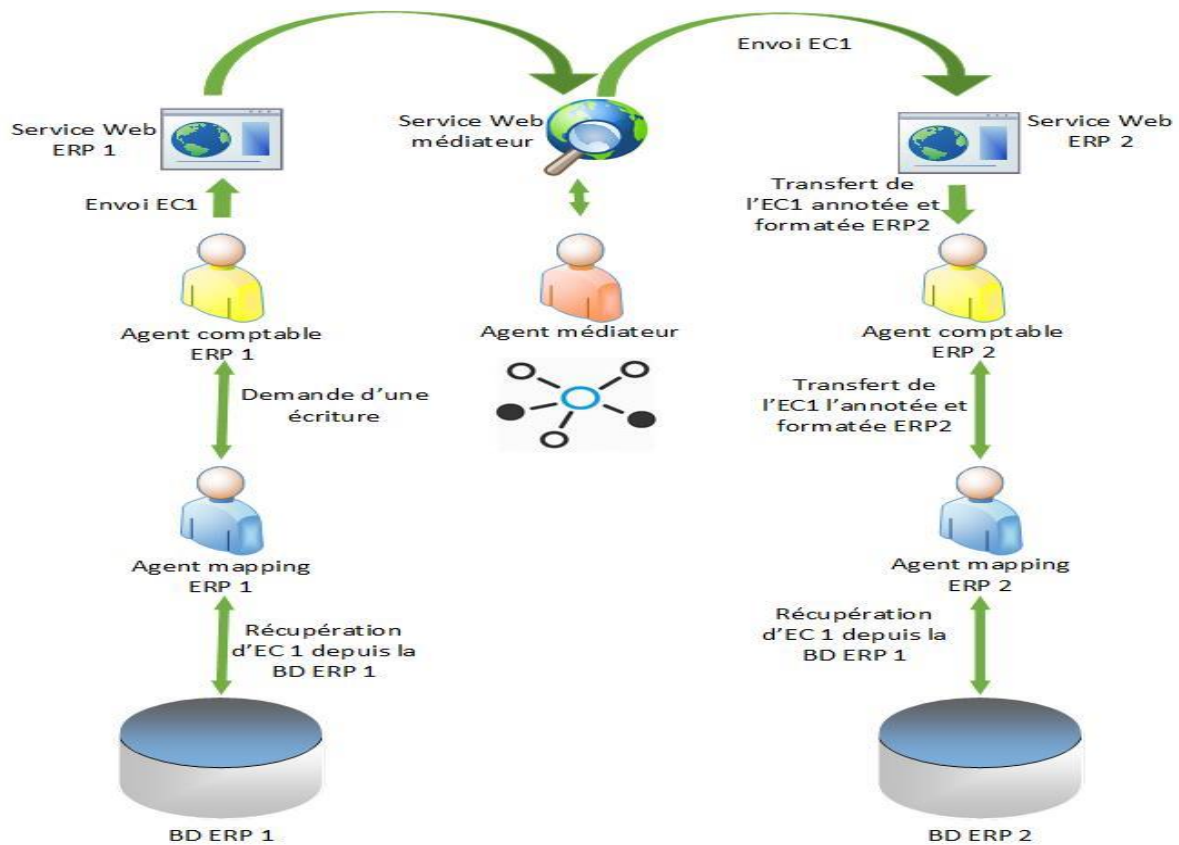


Figure IV-6: Articulation de système d'information du Groupe ENSP (Scénario)

Dans la figure précédente, l'agent comptable ERP 2 lance une requête via un service web vers un agent comptable 1, après la réception de la requête par l'agent cible. Ce dernier interroge la base de données de l'ERP1 pour récupérer les informations nécessaires via l'agent de mapping et de correspondance. Ensuite, il envoie ses informations vers l'agent de médiation qui lance le service web médiateur pour une médiation sémantique fiable. Puis, il achemine ses informations vers l'agent cible, qui à son tour procède à une vérification selon son domaine de compétences. En cas d'échec, il relance une autre requête. Dans l'autre cas, il transmet les informations à son agent de mapping et de correspondance pour mettre à jour la base de données.

L'interaction entre les agents sources et cibles est faite par le biais des services web, un service médiateur est utilisé pour résoudre les conflits entre ces services par les annotations écrites en langage WSDL proposées par l'agent de médiation. Celui-ci utilise une ontologie de domaine et une autre de gestion des conflits. L'approche SMA permet de lancer plusieurs agents au même temps en cas de plusieurs conflits.

Afin de résumer les rôles de chaque agent mentionné dans les paragraphes précédents, on a opté pour une présentation AUML (Cf. Figure IV.7) pour clarifier les différentes interactions entre les agents du système.

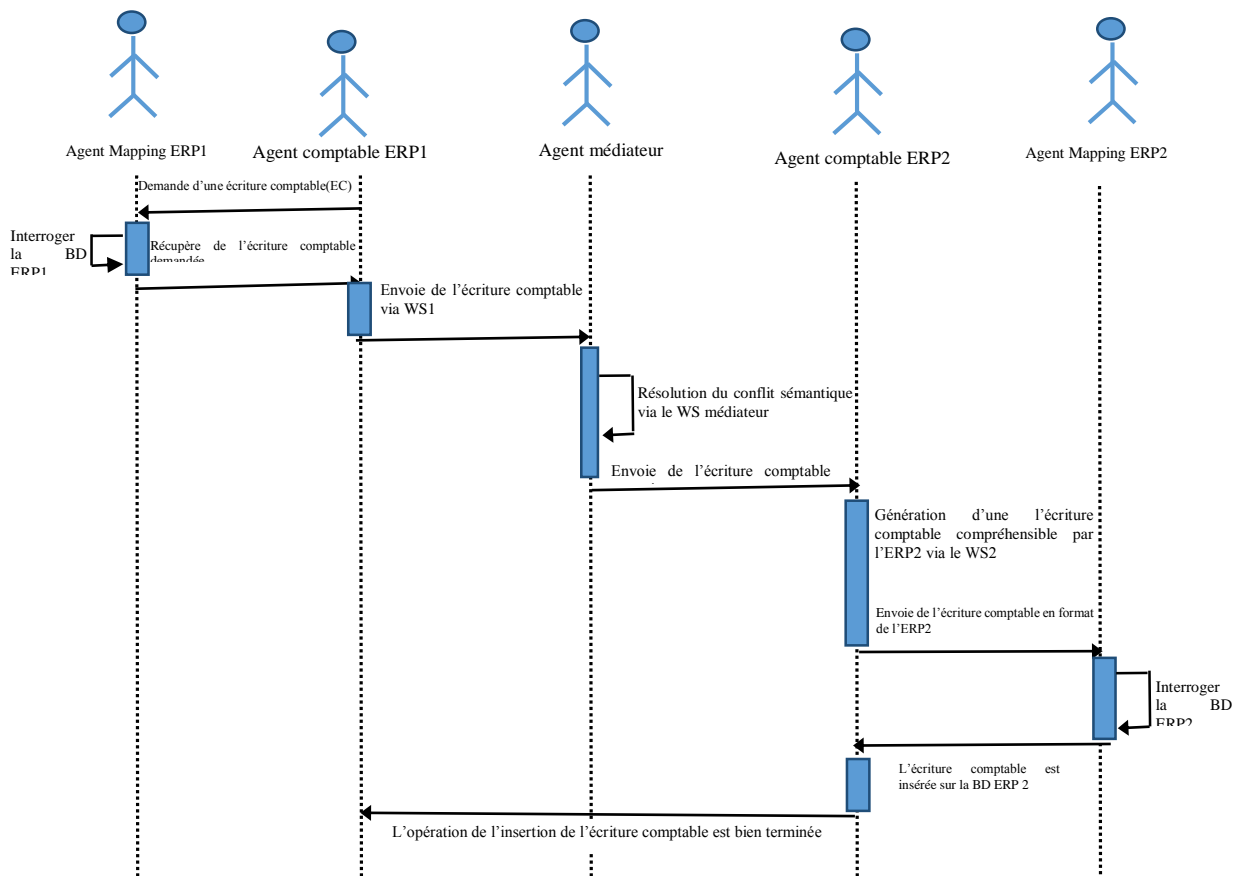
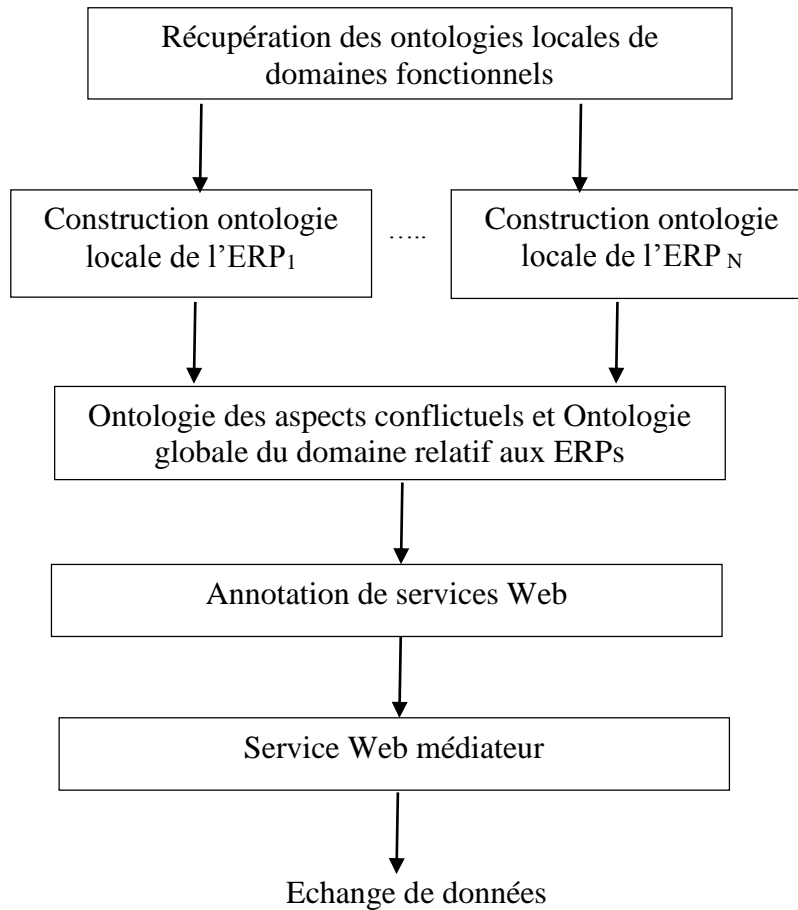


Figure IV-7: Diagramme de séquence AUML

### IV.5. Processus d'interopérabilité sémantique

Afin de décrire les différents rôles de chaque composant de l'architecture précédente, il est utile de présenter la démarche préconisée et illustrée dans la figure IV.8.



**Figure IV-8:** Processus d'interopérabilité sémantique

#### IV.5.1. Récupération des ontologies locales de domaines fonctionnels

Dans cette étape, nous reprenons en premier lieu notre travail [6] où nous avons modélisé l'ERP par un ensemble d'agents. Avant de créer un agent intelligent, il faut tout d'abord créer sa base de connaissance ; qui contient des informations nécessaires pour que ce dernier puisse effectuer ses buts. Un agent intelligent est un intermédiaire entre l'ERP et l'environnement de médiation via un service WEB. Principalement, la BC d'un agent de domaine contient le contexte du module fonctionnel dédié. La BC contient l'ontologie locale qui décrit le domaine fonctionnel. L'ontologie locale décrit la sémantique locale des données 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.

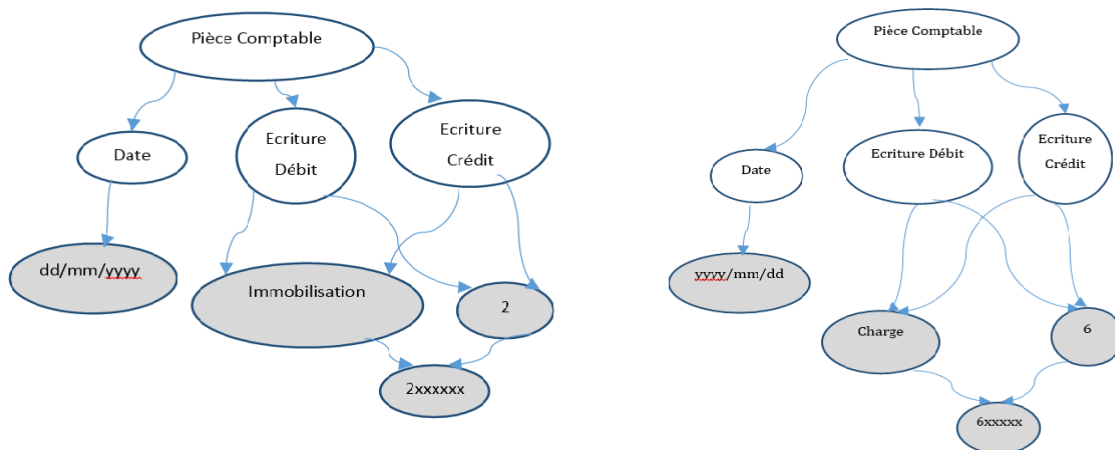
### IV.5.2. Construction de l'ontologie locale de l'ERP

Après avoir créé des ontologies de domaines fonctionnels, nous avons utilisé deux agents situés pour concrétiser cette phase.

Agent de mise à jour et exploitation de l'ontologie locale de l'ERP : Il exploite et met à jour l'ontologie locale de l'ERP en utilisant les ontologies locales de chaque domaine fonctionnel. Il gère tout accès aux ontologies des domaines fonctionnels. Ces dernières construisent par fusion l'ontologie de l'ERP et celles des applications externes de l'ERP. Elle regroupe une collection de termes et de définitions du métier de l'entreprise. Il envoie les mises à jour vers l'ontologie globale de l'agent médiation.

Agent de correspondance et de mapping (Agent de routage) : Il permet de connecter entre les agents et la base de données ERP. Il est apte à établir une conversation avec les différents agents via l'agent de mise à jour et exploitation de l'ontologie locale de l'ERP. Il peut faire toute recherche des agents pertinents pour une coopération.

**Exemple 1 :** Pour illustrer l'architecture que nous avons proposé, nous commençons en premier lieu par la construction d'une partie des ontologies du module fonctionnel de la comptabilité relative à l'ERP1 et ERP2 Figure IV-9.



**Figure IV-9:**Extrait de l'ontologie de l'ERP1 & l'ERP2 (les concepts à l'intérieur en zone grisées représentant le contexte du concept

### IV.5.3. Construction de l'ontologie globale du domaine relatif aux ERPs et celle des aspects conflictuels

La construction d'une ontologie de médiation a pour objectif essentiel d'explicitier tous les concepts des organisations. Elle sera utilisée pour annoter les différents services Web et constitue la base sur laquelle les conflits sont résolus. Nous proposons de scinder l'espace conceptuel en deux niveaux, à savoir, l'ontologie de domaine (OD) et l'ontologie de contexte relatif aux aspects conflictuels (OAC).

**Ontologie du Domaine :** L'OD définit les concepts génériques et leurs relations couvrant les ERPs. Les concepts définis, à ce niveau, ne doivent pas faire l'objet de conflits entre les différents services web. D'une autre manière, elle constitue un accord concernant les noms des



concepts sémantiques utilisés pour permettre l'interopérabilité. Donc, chaque ERP (Fournisseur) doit adhérer à cette ontologie.

**Ontologie des aspects conflictuels :** Elle est créée pour enrichir la description de concepts génériques décrits au niveau de l'ontologie des ERPs. Cet enrichissement a pour but de mettre en évidence la connaissance nécessaire à la détection d'éventuels conflits sémantiques survenant lors de toute interaction entre les services WEB. En effet, les OAC permettent de décrire les différents aspects conflictuels caractérisant le contexte de chaque paramètre de service Web.

**Définition 1 :** Ontologie du Domaine est un sextuple  $\langle C;D;OP;DP; SC; SP \rangle$  où :

- C'est un ensemble de classes ;
- D est un ensemble de types de données ;
- OP est un ensemble de propriétés d'objet. Chaque propriété objet a un ensemble d'origine (domaine) et un ensemble d'arrivée (range) dans C ; les classes sont reliées par des relations non taxonomiques.
- DP est un ensemble de propriétés de données. Chaque propriété de données a un ensemble d'origine (domain) dans C et un ensemble d'arrivée (range) dans D ;
- SC est une relation dans  $C \times C$ , représentant une relation de sous-classe ;
- SP est une relation dans  $(OP \times OP) \cup (DP \times DP)$  représentant une relation de sous-propriété entre des propriétés homogènes.

**Exemple 2 :** Dans la figure IV.10, nous utilisons en premier lieu OWL comme modèle de représentation des ontologies. Puis, nous illustrons (Cf. figure IV.11) une partie de l'ontologie commune qui constitue un accord concernant les noms de concepts sémantiques utilisés pour permettre l'interopérabilité. Chaque ERP doit adhérer à cette ontologie du domaine. Elle contient aussi l'ontologie des aspects conflictuels. Elle permet de décrire le contexte de concepts de l'ontologie de domaine. Ceci permet de décrire clairement les hétérogénéités sémantiques. Après avoir adhéré à l'ontologie de domaine, l'expert métier doit mettre à jour les ontologies des aspects conflictuels associées à sa sémantique locale. Ceci permet d'établir les correspondances avec les représentations des autres fournisseurs. L'ontologie de conflits fournit les vocabulaires qui permettent de spécifier les différentes représentations structurelles et sémantiques des modificateurs.

**Définition 2 :** Ontologie des Aspects Conflictuels est un 3-tuple  $\langle ACg, ACi, \mathcal{F} \rangle$ , où :

- "ACg" est un ensemble de classes représentant différents aspects conflictuels liés aux concepts de l'OD. Chaque classe "acg", appartenant à "ACg", a une superclasse et un ensemble de sous-classes. Chaque classe "acg" porte une désignation ou un nom représentant un aspect conflictuel ;

- "ACi" est un ensemble distinct de classes instanceables ayant une seule superclasse dans "ACg". Par définition, "aci" n'a pas de sous-classes.

- "£" représente les relations entre concepts similaires ou de fratrie (sibling relationship) entre concepts "ACi" ou "ACg". Les relations entre les classes appartenant à "ACg" sont de type

disjoint. Par contre b, les relations entre les classes de "ACi", d'un "acg" donné, sont des relations entre pairs (Peer relationship) qui indiquent des conflits entre des données ayant la même sémantique.

```

<?xml version="1.0"?>
  <rdf:RDF xmlns="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#"
  xml:base="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <owl:Ontology rdf:about="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP"/>
    <owl:ObjectProperty
  rdf:about="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Bénéficiaire">
      <rdfs:domain
  rdf:resource="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Pièce_Comptable"/>
      <rdfs:range rdf:resource="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Tiers"/>
    </owl:ObjectProperty>
    .
    <owl:ObjectProperty
  rdf:about="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Concerne">
      <rdfs:domain
  rdf:resource="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Ecriture"/>
      <rdfs:range rdf:resource="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Classe"/>
      <rdfs:range rdf:resource="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Nature"/>
    </owl:ObjectProperty>
  
```

**Figure IV-10:**Format OWL de l'ontologie de médiation

**Définition 3 :** Le contexte "Ct" est un ensemble de couples dimension-valeur identifié de manière unique ayant la forme suivante :  $\{(D_i, V_i) | i \in [1; m] \text{ où } D_i \in AC_g \text{ et } V_i \in AC_i\}$ .

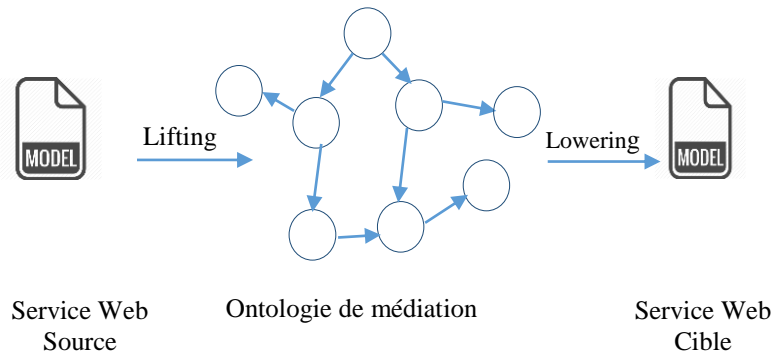
**Définition 4 :** La relation entre deux contextes "Ct" et "Ct'" où  $Ct = \{(D_i, V_i)\}$  et  $Ct' = \{(D_i, V_i')\}$ , existe si et seulement si pour chaque  $i \in \{1..n\}$  il y'a  $v_i R_{D_i} v_i'$  où  $R_{D_i} \in \mathcal{L}_{AC_i}$ .



Figure IV-11: Extrait de l'Ontologie de médiation

#### IV.5.4. Annotation de services Web :

L'enrichissement des données échangées via l'information sémantique est nécessaire pour le bon déroulement du processus de médiation. Cela se fait à partir d'une annotation des descriptions WSDL des services. Le modèle d'annotation des services est exprimé en termes d'une OD. Néanmoins, une OD est censée conceptualiser les connaissances du domaine des ERPs. On note également la présence de trois attributs d'extensibilité qui sont définis par défaut : l'attribut <Montant > permet l'association entre un composant WSDL et un concept d'une ontologie, et les attributs <liftingSchemaMapping> et <loweringSchemaMapping> sont ajoutés aux définitions de types pour spécifier les correspondances entre les éléments du schéma des données et l'information sémantique de l'ontologie (Cf. Figure IV-8).



**Figure IV-12: Annotation des descriptions relatives aux services web de l'exemple**

#### XML Schéma du service Web source

...

`<xs : element name = «Montant» type=double`

`Sawsdl : liftingSchemaMapping='http://MK/AOSS/  
mapping/Montant2OLD.xslt'>`

...

`/xs :sequences>`

#### XML Schéma du service Web cible

...

`<xs : elementname = «Montant» type=double`

`Sawsdl : liftingSchemaMapping='http://MK/AOSS/  
mapping/Montant2OLD.xslt'>`

`Sawsdl : loweringSchemaMapping='http://MK/AOSS/  
mapping/OLD2Montant.xslt'>`

...

`/xs : sequences>`

Comme le montre la figure IV-7, des attributs SAWSDL sont ajoutés aux descriptions des services Web source et cible, il s'agit des schémas XML. Ces annotations font référence aux définitions de mapping des données XML de et vers les ontologies locales de l'ERP dans laquelle nous trouvons les modificateurs et leurs valeurs liés au concept concerné Montant.

#### IV.5.5. Service Web Médiateur

Notre architecture comprend plusieurs types de services. Les services Web broker, permettant de superviser et orchestrer les autres services d'interopérabilité qui sont les services Web : de publication, de découverte, de médiation et d'exécution. La médiation sémantique concerne les services sélectionnés, elle est prise en charge par un service Web de médiation, qui s'active une fois la requête suivante soit reçue, émanant du service Web broker :

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

Où : DescriptionWS<sub>i</sub> désigne les URLs des descriptions WSDL. Il télécharge les fichiers WSDL des SW source et SW cible, et appelle les définitions de mapping relatives aux annotations `liftingSchemaMapping` pour exécuter les transformations sémantiques, des schémas via l'ontologie de médiation. Chaque service Web médiateur généré propose une opération appelée `mediateX2Y`, où X et Y sont remplacés par les noms des messages d'entrée et sortie des opérations. Ces dernières sont spécifiées dans les fichiers WSDL des services mis en relation via le service Web médiateur. Les entrées de l'opération de médiation sont les valeurs qui doivent être transformées dans la représentation requise, lesquelles sont spécifiées par les annotations des fichiers WSDL. Les sorties de l'opération sont les valeurs transformées, qui ont la signification désirée par l'opération cible.

**Définition 5** : Le service de médiation a la forme suivante  $MS_j(\$I_j ; ?O_j)$  :

$G_I \rightarrow G_O$ , où :

- "\$I<sub>j</sub>" et "?O<sub>j</sub>" sont respectivement un ensemble de variables d'entrée et de sortie de  $MS_j$  ;
- $G_I$  et  $G_O$  sont un ensemble de triplets RDF représentant les paramètres d'entrée et de sortie contextualisés.

#### IV.6. Conclusion

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

- ✓ La couche ERPs.
- ✓ La couche sémantique et enrichissement des services.
- ✓ La couche d'infrastructure pour l'interopérabilité des services Web.

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

Nous avons présenté le processus de coopération qui est pris en charge par le service Web broker. Ce dernier orchestre les autres services Web d'interopérabilité sémantique : de publication, de découverte, de médiation et d'exécution. Nous avons aussi évoqué la médiation des données qui est une partie des autres médiations à savoir la médiation des propriétés fonctionnelles et celle des propriétés non fonctionnelles.

---

## Chapitre V : Etude de cas et implémentation

---

### V.1. Introduction

Parmi les aspects clés de toute architecture c'est de réussir à mettre en œuvre différents aspects d'interopérabilité (syntaxique, sémantique, etc.) en matière de communication, connaissance et plateforme d'implémentation. L'interopérabilité apparaît, auprès de notre architecture, comme un élément indispensable à sa réussite, plutôt une nécessité et par conséquent une maîtrise de l'hétérogénéité.

Pour cela, il convient d'adopter des normes et des standards communs à la disposition du système. Des standards qui permettent d'assurer l'interopérabilité des agents partenaires à échanger, ou transmettre des informations et des connaissances totalement transparentes.

Parmi les objectifs de notre contribution, encourager et pousser l'évolution des standards ouverts, normalisés et compatibles pour supporter la réutilisation et l'interopérabilité dans le domaine des progiciels de gestion intégrée. En effet, il s'agit de résoudre les conflits syntaxiques, sémantiques et organisationnels des informations partagées pour coopérer.

Dans le chapitre précédent de cette thèse, nous avons proposé une architecture pour l'interopérabilité sémantique des ERPs. Pour mettre plus en clair les idées et les concepts inclus dans l'architecture proposée, nous allons utiliser cette architecture comme base d'une étude de cas dans un environnement réel. Cette étude de cas va nous permettre de simuler le fonctionnement des systèmes d'information à base d'ERP modélisés comme étant un SMA.

### V.2. Etude de cas

Nous allons aborder les systèmes d'information d'un groupe industriel, dont l'ENSP est une entreprise publique économique et ses filiales et participations éloignés en partenariats avec des entreprises étrangères, à savoir HESP, BJSP, MESP, WESP et BASP.

Pour rappel, l'ENSP (Entreprise Nationale de Services aux Puits), filiale de SONATRACH, est un groupe représentant un important capital de savoir-faire et d'expérience accumulée depuis plus de 30 années dans les services techniques relatifs à l'exploration et l'exploitation pétrolière. Elle intervient pour tester et entretenir les puits de pétrole, ainsi que le tubage et les bacs de stockage des hydrocarbures.

L'ENSP a été créée le 1er août 1981, issue de la restructuration du secteur de l'énergie et des industries pétrochimiques. Elle regroupe tous les services se rapportant à la fonction puits.

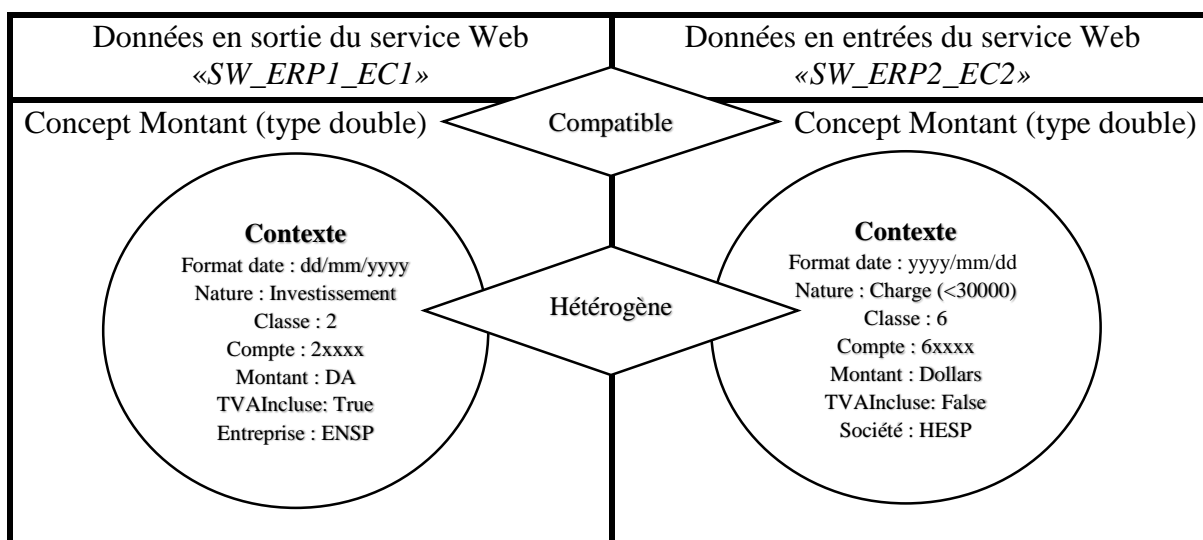
L'ENSP est un groupe de services parapétroliers dont la mission est de satisfaire durablement, par l'innovation et l'action, les besoins de Sonatarch et de ses associés étrangers. L'ENSP a été récemment réorganisé afin de répondre mieux aux besoins des clients et améliorer sa productivité et sa compétitivité.

Le groupe ENSP est organisé comme suit :

- ✓ La direction générale et le corporate management
- ✓ Les unités opérationnelles : Snubbing, Wire line & Well testing, Protection de l'Environnement, Fabrication, well services et mud logging.
- ✓ Les Directions Fonctionnelles regroupent les fonctions suivantes : Direction Centrale Finances, Direction Ressources Humaines, Direction Pilotage Stratégique, Direction Business Development, Direction HSE, Direction Audit, Direction juridique et la Direction Approvisionnements et Logistique.
- ✓ Les Filiales : BJSP, HESP, BASP, MESP, WESP.

### V.2.1. Problématique

Chacune d'eux possède un ERP et un réseau informatique autonome. Afin de présenter un reporting fiable à la société mère c.-à-d. la Direction Générale du groupe, il est utile de rendre ces progiciels interopérables entre eux et résoudre les différents conflits sémantiques tels que les concepts d'une écriture comptable dans deux ERPs (Cf. Tableau V.1). Nous remarquons que ces hétérogénéités touchent le niveau structurel et sémantique des données échangées.



**Tableau 0V-1:**Hétérogénéités sémantiques des données des deux services Web exemples

Dans la figure V.1, nous présentons la communication entre deux ERPs qui ont (02) deux différents plans comptables. Dans ce contexte, l'agent comptabilité ERP1 veut envoyer vers l'agent comptable ERP2.

1. l'Agent ERP1 lance une demande à l'agent mapping pour récupérer l'écriture,
2. l'agent mapping à son tour interroge la base ERP1 et achemine l'écriture vers l'agent comptable ERP1,
3. l'agent comptable 1'envoi l'écriture comptable vers l'agent médiateur via un WS1,
4. l'Agent médiateur ajoute des annotations nécessaires à écriture via le service web médiateur,
5. et envoi l'écriture avec les annotations vers le service web concerné de l'ERP2,
6. le WS2 génère une écriture sous format dans système comptable 2,

7. ensuite l'agent médiateur envoie cette écriture à l'agent comptable ERP2,
8. l'agent comptable ERP2 achemine l'écriture à son rôle vers l'agent mapping ERP2,
9. qui mis à jour la base ERP2.

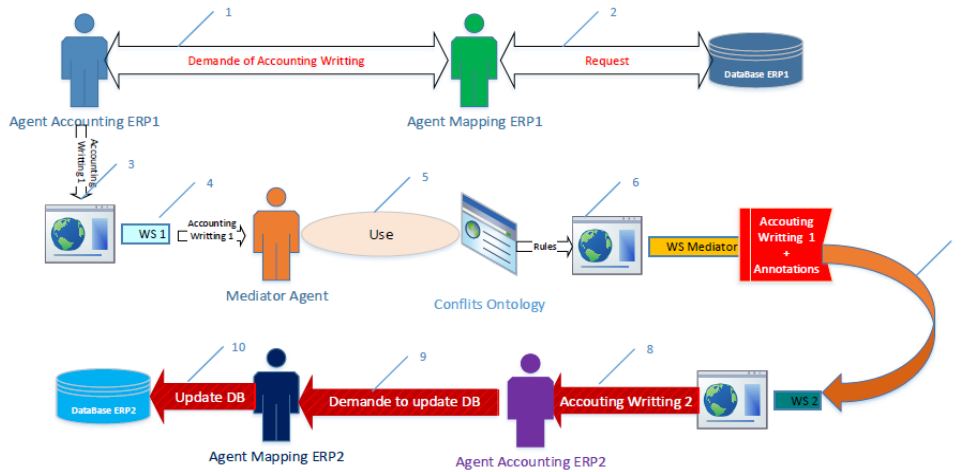


Figure V-1: Scénario d'interopérabilité.

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

### V.2.2. Construction du modèle sémantique

Dans le cadre de cette thèse, nous avons détaillé la construction du modèle sémantique de données (MSD), pour traiter la médiation des données.

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

- ✓ Des ontologies locales de données.
- ✓ Une ontologie de domaine de données.
- ✓ Une ontologie des aspects conflictuels.

Dans ce qui suit nous allons présenter d'une manière exhaustive ces ontologies que nous avons créé à l'aide de l'outil Protégé. Les ontologies locales de données véhiculent la sémantique locale (de l'ERP1 et l'ERP2) et spécifient les valeurs des modifieurs statiques. Chaque ERP après avoir adhérer l'ontologie de domaine doit mettre à jour l'ontologie des aspects conflictuels grâce à sa sémantique locale, il va établir par exemple les différentes relations d'équivalence, de subsomption ... etc. (ex. relation d'équivalence entre les concepts 'Entreprise' et 'société'). Rappelons que l'ontologie contextuelle spécifie la structure et la sémantique des modifieurs.

Concernant la publication du modèle sémantique des données, cette étape est primordiale pour les services de médiation sémantique des données. Une fois que les ontologies sont construites, nous devons alors les publier. Pour cela un référentiel commun est établi sous forme d'une base de données relationnelle en utilisant ORACLE. Sur la base du modèle sémantique obtenu, et



grâce aux mécanismes d'annotation décrits au chapitre précédent, nous procédons à l'enrichissement des schémas XML de la description des services Web.

Aussi, des attributs SAWSDL sont ajoutés aux descriptions des services Web source et service Web cible, il s'agit des schémas XML. Ces annotations font référence aux définitions de mappings des données XML de et vers les ontologies locales de données dans lesquelles nous trouvons les modifieurs statiques et leurs valeurs reliés au concept concerné (Montant). Suite à cette annotation vient l'étape de publication des descriptions annotées des services Web.

### V.2.3. Description de la médiation sémantique de données

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

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

## V.3. Outils et aspects techniques de l'implémentation

Afin d'implémenter un prototype d'une manière rapide et structurelle, il faut faire recours aux outils, plateformes et bibliothèques près existantes qui facilite la tâche d'implémentation.

### V.3.1. Développement des agents

Les plateformes sont un environnement permettant de gérer le cycle de vie des agents et dans lequel les agents ont accès à certains services. Elle constitue un réceptacle au sein duquel les agents peuvent évoluer. De ce fait l'utilisation d'une plate-forme facilite le développement de notre système SMA.

Il existe un grand nombre de systèmes multi-agents dédiés à différents modèles d'agent. De ce fait, plusieurs outils ont été développés pour répondre aux besoins liés à la programmation

orientée agent [178]. Ces outils peuvent prendre différentes formes, allant d'outils d'ordre méthodologique, à des outils de développement, ou des supports d'exécution.

Nous en avons examiné plusieurs, parmi les plus récents, en fonction de leur popularité et de leur pertinence : Madkit [176], AgentBuilder [177], Zeus [174], Jack [173] et Jade [175]. En fonction des propriétés de notre architecture, nous avons opté pour l'utilisation de la plate-forme Jade version 3.5. Ce choix a été motivé par les raisons suivantes :

- ✓ Jade est fondée sur les spécifications FIPA. Ces spécifications ont été élaborées afin de faciliter la communication entre différentes plateformes multi agents hétérogènes.
- ✓ Jade peut gérer un système multi agents dont les agents sont répartis sur un réseau informatique. Installée sur différentes machines, elle permet en effet à des agents répartis de communiquer d'une manière transparente à travers le réseau.
- ✓ Jade possède des outils simplifiant la gestion des erreurs.
- ✓ Jade est facilement intégrable avec les environnements du développement comme JBuilder 7.
- ✓ Jade permet de suivre le cycle de vie des agents et leurs interactions et le code généré par Jade est facilement adapté aux utilisations spécifiques aux agents.

### V.3.1.1. Description de la plate-forme Jade

Dans ce qui suit nous décrivons les composantes de la plate-forme Jade et l'agent Jade.

#### V.3.1.1.1. Composantes de la plate-forme

Jade (Java Agent Development Framework) [179], [185] est une plate-forme multi-agents développée en entier en Java. Elle a comme but la construction des systèmes multi-agents et la réalisation d'applications conformes aux spécifications FIPA.

La plate-forme d'agent peut être répartie sur plusieurs serveurs. Une seule application Java, et donc une seule machine virtuelle de Java (JVM), est exécutée sur chaque serveur. Chaque JVM est un conteneur d'agents qui fournit un environnement complet pour l'exécution d'agent et permet à plusieurs agents de s'exécuter en parallèle sur le même serveur.

La plate-forme d'agent de Jade inclut tous les composants obligatoires qui contrôlent un SMA. Ces composants sont l'AMS (Agent Management System), l'ACC (Agent Communication Chanel) et le DF (Director facilitatory). Toute la communication entre agents est exécutée par messages FIPA ACL. L'AMS fournit un service pages blanches et un service de gestion du cycle de vie (supervise l'enregistrement des agents, leur authentification, leur accès et leur utilisation dans le système). Le DF est l'agent qui fournit le service de pages jaunes par défaut. L'ACC gère la communication entre les agents. La plate-forme d'agent fournit une interface graphique utilisateur (GUI) pour la gestion à distance des agents RMA (Remote Management Agent), surveillant le statut des agents, permettant par exemple d'arrêter et de remettre en marche des agents. Le GUI permet également de créer et de lancer un agent sur un serveur à distance, à condition qu'un conteneur d'agent fonctionne déjà. Le GUI permet de commander d'autres plates-formes d'agent à distance.

Jade fournit quelques outils pour simplifier l'administration de la plate-forme : *L'agent Dummy* est un outil simple et très utile pour visualiser des échanges de messages entre agents,

et donne une interface graphique pour l'édition, l'écriture et l'envoi des messages ACL vers les agents, et de recevoir et lire les messages des autres agents, et éventuellement sauvegarder ces messages. *L'agent Sniffer* permet de suivre des messages échangés pendant qu'ils sont traités et les afficher sous forme de diagrammes de séquence. *L'agent Introspector* permet de surveiller et de commander le cycle de vie d'un agent courant ainsi que ses messages échangés (provenant de la file d'attente des messages envoyés et reçus).

Dans le but de simplifier la transmission, les messages internes (sur la même plate-forme) sont transférés et codés comme des objets Java et non comme des chaînes de caractères. Quand l'expéditeur ou le récepteur n'appartient pas à la même plate-forme, le message est automatiquement converti au format de chaîne de caractères spécifiés par la FIPA. De cette façon, la conversion est cachée au programmeur d'agents, qui a seulement besoin de traiter la classe d'objets Java. Concernant l'implémentation des protocoles d'interaction, Jade facilite la tâche et offre une bibliothèque de protocoles d'interaction FIPA prêts à l'utilisation.

#### V.3.1.1.1. L'agent Jade

Un agent selon Jade est conforme au standard FIPA, possède un cycle de vie, possède toujours un état (AP\_INITIATED, AP\_ACTIVE, AP\_DELETE, etc.), possède un ou plusieurs comportements (Behaviours) et un comportement peut être de plusieurs types simples (oneShot, cyclic) ou composites (Finite State Machine, parallel, sequentiel).

Un agent est identifié de manière globale par un nom unique (l'AgentIdentifier ou AID). Chaque agent peut joindre ou quitter librement la plateforme et rentrer en contact avec chacun des autres agents. Il communique avec des messages de type ACL et rend des services.

La création d'un agent (Cf. figure V.1) Jade est aussi simple que la définition d'une classe qui hérite la classe `Jade.core.agent`. Cela permet à l'agent d'hériter un comportement fondamental caché (qui traite toutes les tâches liées à la plate-forme, telles que l'enregistrement, la configuration, la gestion à distance, etc.), et un ensemble de méthodes qui peuvent être appelées pour implémenter les tâches spécifiques à l'agent. Il faut implémenter la méthode `setup()`. Cette dernière est destinée d'inclure les initialisations d'un agent. Le comportement réel d'un agent est représenté comme des comportements (behaviours). Le squelette du code d'un agent Jade est le suivant :

```
import jade.core.*;
public class MonAgent extends Agent {
// declaration des variables si nécessaire
protected void setup {
// to do: add necessary start-up code
    addBehaviour(new Comportement(this));
}
}
```

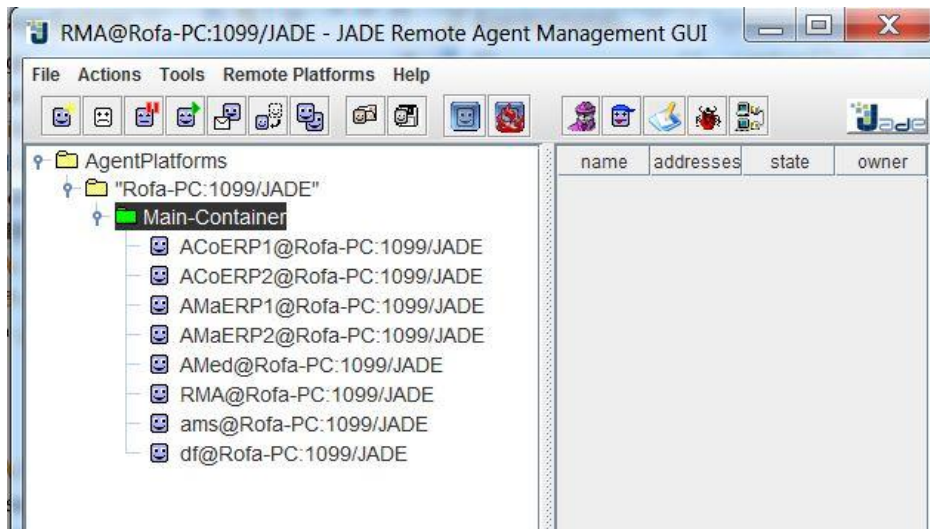


Figure V-2:Création des agents du système

### V.3.2. Simulation de l’interaction entre les agents

La Figure V.2 présente l’interaction entre les agents du système. l’agent comptable ERP 2 lance une requête via un service web vers un agent comptable 1, après la réception de la requête par l’agent cible. Ce dernier interroge la base de données de l’ERP1 pour récupérer les informations nécessaires via l’agent de mapping et de correspondance. Ensuite, il envoie ses informations vers l’agent de médiation qui lance le service web médiateur pour une médiation sémantique fiable. Puis, il achemine ses informations vers l’agent cible, qui à son tour procède à une vérification selon son domaine de compétences. En cas d’échec, il relance une autre requête. Dans l’autre cas, il transmet les informations à son agent de mapping et de correspondance pour mettre à jour la base de données.

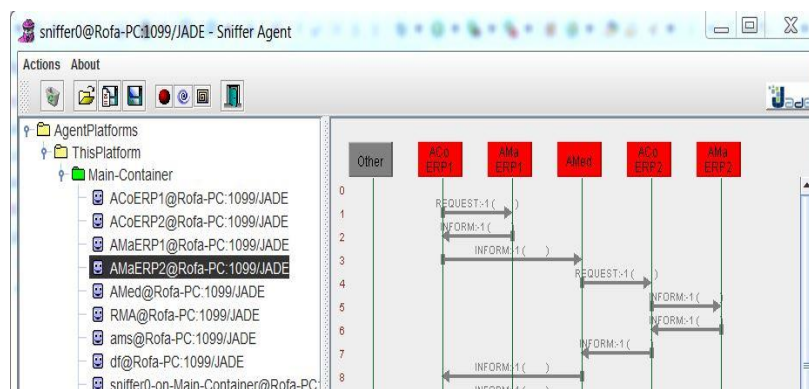


Figure V-3:Simulation du protocole d’interaction entre les agents entre agents

### V.3.3. Développement lié à la médiation sémantique

Comme nous l'avons dit, l'utilisation des bibliothèques de classes près existantes pour implémenter la comparaison des ontologies OWL facilite la tâche d'implémentation. Nous décrivons dans ce qui suit les deux bibliothèques OntoSim et Alignment API.

#### V.3.3.1. OntoSim version 2.0

OntoSim est une bibliothèque de classes Java [180] destinée au développement de mesures de similarité entre ontologies. Cette bibliothèque fournit de nombreuses mesures de similarités entre chaînes de caractères. Ces mesures sont utiles pour le développement d'outils d'alignement d'ontologies, en particulier cette bibliothèque contient les mesures de base de similarité adoptées par notre approche (distance JaroWinkler, distance ...).

#### V.3.3.2. Alignment API version 4.0

Alignment API est une bibliothèque de classes Java [181] principalement basée sur la bibliothèque OWL API [182] et d'autres bibliothèques bien connues dans la littérature. La figure V.1 montre les différentes dépendances.

Selon les créateurs de l'Alignment API, la bibliothèque est utilisée pour le développement d'applications d'alignement d'ontologies, selon quatre manières [183], [184], [181] :

- ✓ Développer l'API : en modifiant le code de l'API pour ajouter d'autres fonctions ou fixé des bugs ;
- ✓ Etendre ou faire extension l'API : créer de nouvelles instances de l'interface API (nouvelles méthodes d'alignement etc...) ;
- ✓ Utiliser l'API : créer une application en utilisant directement les classes et les méthodes d'alignement fournies par l'API.
- ✓ Utiliser l'API comme un serveur : créer une application en utilisant l'API qui communique avec une instance d'un serveur d'alignement.

Nous utilisons la bibliothèque Alignment API dans le sens "étendre ou faire extension l'API" afin de faciliter le développement de notre distance sémantique. Alignment API permet d'intégrer des nouvelles méthodes de mesure de similarités (entre deux ontologies OWL) en implémentant une interface Java. Il y a plusieurs méthodes pour ajouter une nouvelle méthode de similarité entre les entités de deux ontologies OWL. Dans notre contexte la plus efficace est de développer, pour chaque mesure de similarité, une classe hérite de la classe *ObjectAlignment* et en implémentant l'interface *AlignmentProcess*. Le squelette du code source est le suivant :

```

import org.semanticweb.owl.align.Alignment;
import org.semanticweb.owl.align.AlignmentProcess;
import org.semanticweb.owl.align.AlignmentException;

// Alignment API implementation classes
import fr.inrialpes.exmo.align.impl.ObjectAlignment;

import fr.inrialpes.exmo.ontowrap.OntowrapException;

import java.net.URI;
import java.util.Properties;
/** The Skeleton of code for extending the alignment API */
public class NewMatcher extends ObjectAlignment implements AlignmentProcess {
    public NewMatcher() {
    }
    /**
     * The only method to implement is align. All the resources for reading the ontologies and rendering the
     alignment are from ObjectAlignment and its superclasses: - ontology1() and ontology2() returns objects
     LoadedOntology - addAlignCell adds a new mapping in the alignment object */
    public void align( Alignment alignment, Properties param ) throws AlignmentException {
        try {
            // Match classes
            // Match dataProperties
            // Match objectProperties
        } catch (Exception e) { e.printStackTrace(); }
    }
    public double match(Object o1, Object o2) throws AlignmentException {
        try {
            String s1 = ontology1().getEntityName(o1);
            String s2 = ontology2().getEntityName(o2);
            /* ... */
        } catch (OntowrapException owex) {
            throw new AlignmentException( "Error getting entity name", owex );
        }
    }
}

```

### V.3.3.3. Connexion aux bases de données locales

La deuxième étape de réalisation consiste à connecter la plate-forme à des sources d'information locales. La base de données locale de l'ERP1 et la base de données sont implantées sous le SGBDR ORACLE et le système d'exploitation Windows 7. L'API Java JDBC est l'interface java pour accéder aux bases de données relationnelles. Chaque base doit proposer son driver JDBC qui permet d'y accéder. Quel que soit la base, le driver JDBC se manipule de la même manière, cette interface fournit un ensemble de classes permettant l'utilisation d'un ou plusieurs SGBD relationnels à partir des programmes Java [186]. La base de données de l'ERP2 est implémentée sous le SGBD SQLSERVER et sous le système d'exploitation Windows 7.

## V.4. Conclusion

Nous avons présenté dans ce chapitre une étude de cas qui a illustré les différents concepts et étapes de notre approche. Nous avons aussi, construit le modèle sémantique de données à l'aide de l'outil Protégé en transformant les modèles de données en ontologies locales, puis sont construites l'ontologie de domaine et l'ontologie des aspects conflictuels. A l'aide de ce sous modèle sémantique, et les mécanismes d'annotation du standard SAWSDL, nous avons enrichi

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

Nous avons exposé aussi quelques aspects d'implémentation de notre architecture. L'objectif du prototype est de montrer que l'architecture fonctionne au niveau de l'implémentation. Nous avons choisi des bibliothèques de classes très puissantes (OntoSim, Alignment API et Jade). De ce fait l'implémentation de notre architecture est très flexible et extensible.

Pour que nos agents soient capables de maîtriser toutes l'hétérogénéités des environnements, nous proposons que les agents communiquent entre eux des informations sur la nature des ontologies à comparer (par exemple on définit des niveaux de détail d'une ontologie,...) et suivant ces informations et les croyances des agents sur l'environnement, on définit une méthode (de comparaison entre ontologies) adaptative varie suivant les ontologies à comparer.

## Conclusion générale

Nous rappelons que notre travail de recherche s'inscrit dans le cadre des systèmes d'information hétérogènes et distribués à base d'ERP, et dont l'objectif principal est la proposition d'une approche sémantique basée sur les ontologies et les systèmes multi-agents pour la mise en œuvre des mécanismes de l'interopérabilité des systèmes d'information des entreprises de type ERP. Notre ambition à travers cette architecture est de répondre aux problématiques d'interopérabilité technique par le passage à une architecture orientée services, et d'interopérabilité sémantique, car la nouvelle génération des ERPs apparaît et consiste à fournir une interface pour l'accès aux services et sources de données distribuées et hétérogènes.

Une étude de l'état de l'art nous a permis de constater que les technologies permettant l'interopérabilité des systèmes d'information ont pris une énorme importance de part leur création, normalisation et standardisation par de grands constructeurs et organisations (exemples : W3C, OASIS ...). Dans cette étude, nous avons présenté une panoplie d'outils techniques et conceptuels permettant l'interopérabilité au niveau technique et syntaxique ainsi qu'au niveau sémantique, ce qui nous a permis de retenir les paradigmes des services Web et les systèmes multi-agents pour dresser les mécanismes de l'interopérabilité technique. Comme nous l'avons mentionné, la principale limite de ces paradigmes est leur manque en ce qui concerne la présentation de l'aspect sémantique, principale cause de l'échec de l'interopérabilité des systèmes d'information de type ERP. Pour remédier à ce problème, nous avons retenu l'approche déclarative de représentation de la sémantique, notamment les ontologies.

Dans ce contexte, beaucoup de travaux ont été réalisés dans le domaine de l'interopérabilité sémantique et ont donné naissance à des approches et prototypes dont certains ont été présentés dans cette thèse. Nous avons commencé par présenter le domaine : la coopération des systèmes d'information, l'interopérabilité, l'hétérogénéité sous ses différentes formes et particulièrement l'hétérogénéité sémantique. Nous avons mis l'accent tout au long de cette thèse sur l'importance de la sémantique pour la réussite d'un système d'interopération.

L'approche que nous avons proposée a conduit à une architecture en couche. Nos principales contributions sont résumées comme suit :

- ✓ L'adoption d'une méthodologie basée sur les ontologies et les systèmes multi-agents qui permet d'une part une migration progressive et adéquate des systèmes d'information hétérogènes et distribués existants vers des nouveaux systèmes d'information orientés services, qui tirent profit des avantages de SOA et des standards des services Web (SOAP, WSDL, UDDI).
  
- ✓ La proposition d'un modèle sémantique basé sur la notion d'ontologies pour capturer la sémantique des données échangées entre les services Web. Nous avons déjà détaillé ce modèle qui comprend trois types d'ontologies à savoir, des ontologies locales qui véhiculent la sémantique locale de chaque ERP, une ontologie de domaine qui présente un accord des ERPs sur des concepts communs, et des ontologies des aspects



conflictuels. Notre modèle exploite la notion du contexte que nous avons dissociée de l'ontologie de domaine sous forme d'une ontologie des aspects conflictuels, liée aux concepts de l'ontologie de domaine, dans le but de décrire explicitement le contexte des données échangées entre services Web, et résoudre les conflits lors de la mise à jour de cette ontologie.

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

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

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

## Bibliographie

- [1] S. Baudry, "Développement sur l'ERP OfbizNéogia," Rapport de stage, Polytechnique de l'Université de Tours, 2005.
- [2] Durgansh, S., Naveen, K., (2009). Interoperability Maneuver of the Entreprise's Dimensions with ERP system. Global Journal of Entreprise Information System, Vol. 01, N°01, pp. 42-52.
- [3] MARKESS International (2013). Attentes des entreprises pour les solutions de gestion intégrée ERP/PGI face aux nouveaux enjeux, Référentiel de pratiques (2011-2013), <http://www.markess.fr>.
- [4] A. Hammami, P. Burlat, J. Pierre, Contribution à la conception et au pilotage d'une entreprise réseau, 3e Conférence Francophone de MODélisation et SIMulation "Conception, Analyse et Gestion des Systèmes Industriels" MOSIM'01, Apr 2001, Troyes, France. 2001.
- [5] S. Ourari, B. Bouzouia, Approches et Outils d'Aide à la Décision pour le Pilotage des Systèmes de Production", Laboratoire de Robotique et d'Intelligence Artificielle, Centre de Développement des Technologies Avancées, France.
- [6] J. François, Planification des chaînes logistiques : Modélisation du système décisionnel et performance, Thèse de Doctorat à l'Université Bordeaux I, Volume 17 décembre 2007.
- [7] M. Abbas, ERP Systems in HEI Context from a Multiple Perspective View: A Case Study, 2011.
- [8] M. E. Shacklett, L'essentiel sur SAP Business Suite L'essentiel sur SAP Business Suite», <http://www.lemagit.fr/conseil/Lessentiel-sur-SAP-Business-Suite>, consulter le 21-02-2016.
- [9] N. Mesbahi, O. Kazar, S. Benharzallah, M. Zoubeidi, D. Rezki. A clustering approach based on cooperative agents to improve decision support in ERP. Technological Innovations in Knowledge Management and Decision Support. IGI Publishing Hershey, PA, USA, 2018.
- [10] J-L. Lequeux, Manager avec les ERP, Editions d'organisation, 2008.
- [11] R. Reix, et F. Rowe. La recherche en Systèmes d'Information: de l'histoire au concept. In ROWE, F. Faire de la recherche en Systèmes d'Information. Paris, Ed. Vuibert, FNEGE, Septembre 2002, p.355
- [12] T. Willis-Brown, et A. Mcmillan. Stratégies de maîtrise des coûts lors de l'implantation de systèmes ERP, Revue Française de Gestion Industrielle, Issue 1, Volume 22, 2003.
- [13] A. Chaabouni, Implantation d'un ERP (Enterprise Resource Planning) : antécédents et conséquences, AIMS, XVème Conférence Internationale de Management Stratégique, Annecy / Genève 13-16 Juin 2006.
- [14] K. JONES. L'ABC de la gestion intégrée : Guide d'introduction pour les dirigeants. 2006.
- [15] F-A. Blain, Présentation générale des ERP et leur architecture modulaire, <http://fablain.developpez.com/tutoriel/presenterp/>, 20/03/2006.

- [16] F. Darras, Proposition d'un cadre de référence pour la conception et l'exploitation d'un progiciel de gestion intégré", Thèse Présentée en vue de l'obtention du grade de docteur, Institut National Polytechnique De Toulouse, 2004.
- [17] C. Lassiette, « Les progiciels de gestion intégrée, Article, Centre de Recherche en Economie et Gestion (CREG) Paris», [http://www.creg.ac-versailles.fr/article.php3?id\\_article=209](http://www.creg.ac-versailles.fr/article.php3?id_article=209).
- [18] PGPP, Progiciel de Gestion pour la PME Principaux ERP propriétaires et Open Source, version 1.0, 2006.
- [19] A. Iarionova, gestion des processus de la création d'une maison erp, Université nationale d'économie de Kharkiv Simon Kuznets, Ukraine, Université Lumière Lyon 2, France, 2014.
- [20] C.LEROY, EDITO - La fin de l'ERP généraliste "tout en un" ?, Chief Editor Groupe CXP, Conseil et analyse en solutions logicielles pour l'entreprise et ses métiers, France, <http://www.cxp.fr/content/news/edito-la-fin-de-lerp-generaliste-tout-en-un>, Consulté le 05/11/2014.
- [21] P.Rahali, Senior Analyst ,Groupe CXP,EDITO – L'ERP métier, la solution pour un ROI rapide ?, <http://www.cxp.fr/content/news/edito-lerp-metier-la-solution-pour-un-roi-rapide>, Consulté le 16/10/2015.
- [22] R. Beretz, « Les ERP verticalités, gages d'efficacité ? », [http://www.erp-infos.com/info\\_article/m/2031/les-erp-verticalises-gages-defficacite%20.html](http://www.erp-infos.com/info_article/m/2031/les-erp-verticalises-gages-defficacite%20.html), publié le 03/06/2013.
- [23] La couverture fonctionnelle des ERP pour les PME, <http://www.voxime.com/quel-erp-pour-quelmetier.html?PHPSESSID=7eb43c5814e02d4e2aeecd566fb80f17>, Consulté le 01/02/2015.
- [24] Inconvénients ERP, <http://www.guidepme.com/article514/Inconvenients-ERP>, Consulté le 14/06/2014.
- [25] LES PROGICIELS DE GESTION INTEGREE, [http://www.creg.ac-versailles.fr/IMG/pdf/progiciels\\_gestion\\_integree.pdf](http://www.creg.ac-versailles.fr/IMG/pdf/progiciels_gestion_integree.pdf).
- [26] L.M. MARKUS, et C. TANIS. The Entreprise System Experience\_from adoption to succes », in Framing the domains of IT management, R.W.Zmud Editor, Pinnaflex, Cincinatti. 2000.
- [27] I. Ehie, et M. Madsen, Identifying critical issues in enterprise resource planning (ERP) implementation, Computers in Industry, Volume 56, issue 6, p. 545 557, 2005.
- [28] A. Rabaa'i, W. Bandara and G. Gable. ERP systems in the higher education sector: a descriptive study. In Proceedings of the 20th Australasian Conference on Information Systems, Monash University: Caulfield Campus, Melbourne, pp. 456-470. 2009.
- [29] A. Chaabouni, Implantation d'un ERP (Enterprise Resource Planning) : antécédents et conséquences, AIMS, XVème Conférence Internationale de Management Stratégique, Annecy / Genève 13-16 Juin 2006.
- [30] S. M. Glover, D. F. Prawitt, et M. B. Romney, Implementing ERP: Internal auditing can help eliminate mistakes that commonly derail organizations' ERP initiatives. Internal Auditor, Volume 56, p. 40-47, 1999.
- [31] C. Ash, et J. Burn., A strategic framework for the management of ERP enabled e-business change. European Journal of Operational Research, Volume 146 Issue 2, p. 374-387, 2003

- [32] P.,Bingi, M. K. Sharma, et Godla, Critical issues affecting an ERP implementation. *Information Systems Management*, Volume 16 Issue 3, p. 7-14. 1999.
- [33] E. Shehab, M. Sharp, L. Supramaniam, et T. A. Spedding, Enterprise resource planning: An integrative review. *Business Process Management Journal*, Volume 10 Issue 4, p. 359-386, 2004.
- [34] G. Shanks, A. Parr, B. Hu, B. Corbitt, T. Thanasankit, et P. Seddon, Differences in critical success factors in ERP systems implementation in australia and china: A cultural analysis. *Proceedings of the 8th European Conference on Information Systems*, p. 537-544, 2000.
- [35] D. S. Bajwa, J. E. Garcia, et T. Mooney, An integrative framework for the assimilation of enterprise resource planning systems: Phases, antecedents, and outcomes. *Journal of Computer Information Systems*, Volume 44 issue 3, 81-90, 2004.
- [36] R. Marciniak. *Piloter les technologiques de l'informatique et des télécoms – modèles et outils*, ouvrage collectif, éditions Weka, 2001.
- [37] B. Vincent, et S. Gharbi. Impact du déploiement de SAP R/3 sur la performance globale d'une entreprise et facteurs clés de succès : proposition d'un tableau de bord et application dans le secteur de l'industrie pharmaceutique, Journée de recherche à l'IAE de Montpellier, 1-27, 2004.
- [38] S. Shang, P. Seddon, Assessing and managing the benefits of enterprise systems: the business manager's perspective, *Journal of Information Systems*, Volume 12. 2002.
- [39] P. Grabski, Financial impact of entreprise ressource planning implementations, *international journal of accounting information systems* Volume 2, p. 271-294, 2001.
- [40] J. Chaabouni. « le concept de performance dans les théories du management », Actes de Colloque, FSEG Sfax, 1992.
- [41] M. Kalika. *Structure de l'entreprise, réalité, déterminants et performance*, Economica, Paris, 1988.
- [42] M-L. B. Gomez A. Frot, Duwer Quels effets organisationnels pour les ERP, Actes de conférence AIMS, p. 1-24, 2002.
- [43] F. FOURATI, *Veille stratégique : de l'évaluation de l'utilisation des agents intelligents à la prise de décision*, Thèse de doctorat, Université Paris Dauphine, 2006.
- [44] F. GARCIA, " Systèmes d'information integers: Système d'information ... Progiiciel de gestion integée ... ERP (Entreprise Ressource Planning)", IUP Tarbes – M1, 2006
- [45] S. SPIEKERMANN, "Introduction to Enterprise Resource Planning (ERP): System Analysis and Design - Lecture 11", Institute of Information Systems, Allemagne, 2007
- [46] L. GHARSALLAH, "Impact de l'ERP sur la performance : cas d'IGL", Université de Sfax - Mastère Professionnel, Rapporté de [http://www.memoireonline.com/05/07/463/impact-erp-performance-cas-igl.html#\\_Toc165367699](http://www.memoireonline.com/05/07/463/impact-erp-performance-cas-igl.html#_Toc165367699).
- [47] "Définition E-PROCUREMENT", Rapporté de <http://www.agrojob.com/dictionnaire/definition-E-procurement-2784.htm>.
- [48] Ministère de l'Économie, de l'industrie et de l'emploi, "Enquête sur les TIC et le commerce électronique en 2002: Définitions", [http://www.industrie.gouv.fr/sessi/enquetes/tic/tic2002\\_definition.htm](http://www.industrie.gouv.fr/sessi/enquetes/tic/tic2002_definition.htm)
- [49] J. CARDOSO, R. P. BOSTROM, A. SHETH, "Workflow Management Systems and ERP Systems: Differences, Commonalities, and Applications", Apporté de <http://dme.uma.pt/jcardoso/Research/Papers/ABS-ITM02.pdf>.

- [50] A. BAAZI, A. DAHANE, "Entreprises & Systèmes d'Information", Communication à Inforum'2005, Oran, Algérie, Mars 2005.
- [51] European Interoperability Framework for pan-European eGovernment Services, Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens (IDABC), November 2004, Luxembourg.
- [52] IEEE (Institute of Electrical and Electronics Engineers). Standard Computer Dictionary-A compilation of IEEE Standard Computer Glossaries NY. 610-1990. ISBN : 1559370793.
- [53] IEEE. Standards Information Network. IEEE 100, "The Authoritative Dictionary of IEEE Standards Terms", Seventh Edition. Newyork, NY, IEEE, 2000.
- [54] D. Chen, H. Panetto, et al. « D6.1 : Practices, principles and patterns for Interoperability », INTEROP-NOE, Interoperability Research for Network enterprises Applications and Software Network of Excellence, N° IST 508-011, Jun 2005.
- [55] "European Interoperability Framework, - ICT Industry Recommendation-". Brussels, 18 February 2004, WWW. Comptia.org.
- [56] R. Klischewski ; "Information integration or process integration: How to achieve interoperability in administration". EGOV04 at DEXA, 30 August - 3 September, 2004, Zaragoza, Spain.
- [57] H. Panetto ; "Meta-modèles et modèles pour l'intégration et l'interopérabilité des applications d'entreprises de production". Habilitation à diriger des recherches. Université Henri Poincaré – Nancy I, 2006.
- [58] J.P. Meinadier ; "Le métier d'intégration des systèmes". Lavoisier-Hermes, 2002.
- [59] S. Izza ; "Intégration des systèmes d'information industriels, une approche flexible basée sur les services sémantiques". Thèse de doctorat de l'Ecole Nationale Supérieure des Mines de Saint –Etienne, France, 20 Novembre 2006.
- [60] W.A.Woods ; "What's in a link: foundations of semantic networks". In D.G.Bobrow & A. M. Collins (Eds.), 1975.
- [61] M. Kavouras ; "A unified ontological framwork for semantic integration". International Workshop on Next Generation Geospatial Information, Cambridge, UK, 2003.
- [62] M. Uschold, and M. Gruninger ; "Creating Semantically Integrated Communities on the World Wide Web". In Semantic Web Workshop, Honolulu, Hawaii-Invited Tallk, 2002.
- [63] T. Gruber. What is an ontology. Encyclopedia of Database Systems, 1, 2008.
- [64] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In International Journal of Human-Computer Studies, pages 907\_928. Kluwer Academic Publishers, 1993.
- [65] Oscar Corcho and Asunción Gómez-Pérez. Evaluating knowledge representation and reasoning capabilities of ontology speci\_cation languages, 2000.
- [66] G. van Heijst, A. Th. Schreiber, and B. J. Wielinga. Using explicit ontologies in kbs development. Int. J. Hum.-Comput. Stud., 46:183\_292, March 1997.
- [67] John E Sowa and Stuart C. Shapiro. Knowledge representation: Logical, philosophical, and computational foundations computational foundations by john f. sowa (book review), 2006.
- [68] Douglas B. Lenat and R. V. Guha. Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

- [69] R.RIFAIEH, A.ARARA, A.N.BENHARKAT, Multi-representation Ontologies in the context of Enterprise Information Systems, Americas Conference on Information Systems 2004, August 6-8, New York City, NY, USA
- [70] Klass Objecten for quality in Object and Component Technology  
<http://www.klasse.nl/ocl/ocl-introduction.html>
- [71] IDEF5 Ontology Description Capture Method Overview, KBSI Report, 1994,  
<http://www.idef.com/idef5.html>.
- [72] Uschold, M.; Grüninger, M.: Ontologies Principles Methods and Applications, In Knowledge Engineering Review. 11(2), 1996.
- [73] Fernandez, M.; Gomez-Perez, A.; Juristo, N.: Methontology from Ontological Art Toward Ontological Engineering, In AAAI'97 Spring Symposium Series on Ontological Engineering, USA, 1997.
- [74] O. Corcho, M. Fernandez-Lopez, and A. Gomez-Pérez ; ‘‘Methodologies, tools and languages for building ontologies : Where is their meeting point ?’’. Data & Knowledge Engineering Vol. 46 (No. 1) pp. 41-64, Elsevier, 2003.
- [75] A. Gomez-Perez ; ‘‘Développements récents en matière de conception, de maintenance et d'utilisation d'ontologies’’. 3èmes rencontres Terminologie et Intelligence Artificielle TIA, Vol 19, pp. 9-20, 2000.
- [76] IDEAS ; ‘‘ IDEAS Project Deliverables (Wp1-Wp7)’’. Public reports,  
[www.ideasroadmap.net](http://www.ideasroadmap.net), 2003.
- [77] 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.
- [78] Ora Lassila, Ralph R. Swick, World Wide, and Web Consortium. Resource description framework (rdf) model and syntax specification, 1998.
- [79] J.F. Baget, É. Canaud, J. Euzenat, and M. Saïd-Hacid. Les langages du web sémantique. Revue I, 2004.
- [80] S. Decker, P. Mitra, and S. Melnik. Framework for the semantic web: an rdf tutorial. Internet Computing, IEEE, 4(6):68 \_73, nov/dec 2000.
- [81] D. Fensel, I. Horrocks, F. Van Harmelen, S Decker, M. Erdmann Et M. Klein. Oil in a nutshell. In Proceedings of European Knowledge Acquisition Workshop (EKAW'2000), volume 1937, pages 1–16. Springer-Verlag LNAI, 2000.
- [82] J. Hendler Et D.L. McGuinness. The Darpa Agent Markup Language.  
<http://www.daml.org>, 2001.
- [83] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. DAML+OIL Reference Description. W3C Note 18 December 2001. <http://www.w3.org/TR/daml+oil-reference>.
- [84] D. Fensel, S. Decker, M. Erdmann, and R. Studer. Ontobroker: The very high idea. In Proceedings of the 11th International Flairs Conference (FLAIRS-98), Sanibal Island, Florida, May 1998.
- [85] I. Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, Goble, F. van Harmelen, M. Klein, S. Staab, R. Studer, and E. Motta. OIL: The Ontology Inference Layer. Technical Report IR-479, Vrije Universiteit Amsterdam, Faculty of Sciences, Sept. 2000.  
<http://www.ontoknowledge.org/oil/>.
- [86] INTEROP, 2004. State of the art and state of the practice including initial possible research orientations. Deliverable D8.1. <http://www.interop-noe.org/>
- [87] ‘‘RDF Primer’’. Recommendation RDF du W3C. <http://www.w3.org/>
- [88] OWL Coalition ; ‘‘OWL - Web Ontology Language - (W3C Recommendation 10

- February 2004)”. W3C,<http://www.w3.org/TR/2004/REC-owl-guide-20040210/>, 2004.
- [89] 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.
- [90] La bibliothèque Jena2, <http://jena.sourceforge.net/>
- [91] Protégé website, <http://protege.stanford.edu/>
- [92] JESS website, <http://herzberg.ca.sandia.gov/jess/>
- [93] site Web sur « description logics », <http://dl.kr.org/>
- [94] Racer Semantic Web Inference Engine website, <http://www.sts.tuharburg.de/~r.f.moeller/racer/>
- [95] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner ; ‘‘Ontology-based Integration of Information - A Survey of Existing Approaches’’. In: Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, 2001, Vol. pp. 108-117.
- [96] MENA E. et al., An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. International Journal of Distributed and Parallel Databases (DAPD) 2000.
- [97] GOH C. H., Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources. PhD, MIT, 1997.
- [98] H. WACHE. Towards rule-based context transformation in mediators. In S. Conrad, W. Hasselbring, and G. Saake, editors, International Workshop on Engineering Federated Information Systems (EFIS 99), Kuhlungsborn, Germany, 1999. Infix-Verlag.
- [99] M. Klein ; ‘‘Combining and relating ontologies: an analysis of problems and solutions’’. International Joint Conferences on Artificial Intelligence (IJCAI), Workshop on Ontologies and Information Sharing, Seattle, USA, 2001.
- [100] INTEROP ; ‘‘Deliverables of INTEROP Project’’. [www.interop-ne.org](http://www.interop-ne.org), 2005.
- [101] KnowledgeWeb ; ‘‘Deliverables of KWEB Project’’. EU-IST-2004-507482, <http://knowledgeweb.semanticweb.org/>, 2004.
- [102] Sekt, ‘‘Deliverables of SEKT Project’’. <http://www.sektproject.com/>, 2004.
- [103] N. F. Noy, and M. A. Musen ; ‘‘The Prompt suite: Interactive tools for ontology merging and mapping’’. International Journal of Human Computer Studies, Vol.59(6): pp. 983-1024, 2003.
- [104] R. GIRARDI, "An analysis of the contributions of the agent paradigm for the development of complex systems", In (SCI 2001) and (ISAS 2001), Orlando, Florida. 2001.
- [105] H. S. Nwana, Software Agents: An Overview, Knowledge Engineering Review, Volume 11, Issue 3, p. 205-244, 1996.
- [106] M. Grabner, F. Gruber, L. Klug, W. Stockner ,EvalAgents, D2.1 - Agent technology: State of the Art,2000.
- [107] M. Grabner, F. Gruber, L. Klug, W. Stockner ,EvalAgents, D2.1 - Agent technology: State of the Art,2000.
- [108] Brahim chaib-draa, Agents et systèmes multiagents ( IFT 64881A ), Note de cours, département d’informatique faculté des sciences et de génie, université LAVAL QUEBEC, Novembre 1999.
- [109] Imed Jarras et Brahim Chaib-draa, Aperçu sur les systèmes Multiagents, Série Scientifique, Montréal Juillet 2002

- [110] J. Ferber. *Les systèmes multi-agents, vers une intelligence collective*. InterEditions, 1995.
- [111] Said Brahim, Thèse magister : « Une approche de planification distribuée dans un environnement multi agent », Université Mentouri de Constantine, 2003.
- [112] J. Denis, Thèse de Doctorat : « Délégation de Rôle et Architectures Dynamiques de Systèmes Multi-Agents Conversationnels », Université Lyon I – Claude Bernard, 2003.
- [113] H. Mazyad, Une Approche Multi-agents à Architecture P2P pour l'Apprentissage Collaboratif, Thèse Pour l'obtention du titre de Docteur de l'Université du Littoral Côte d'Opale, 2013.
- [114] D. Nessah., « un modèle de raisonnement pour un système de recherche sémantique d'informations sur le web basé agents », Thèse de Doctorat, l'Université de biskra, 2014.
- [115] J. Ferber, “Les systèmes multi-Agents – vers une intelligence collective”, inter éditions 1995.
- [116] FIPA Communicative Act Library Specification, 2000.
- [117] S. Benharzallah, Thèse de Doctorat, Coopération multi agents pour le traitement des requêtes sur des sources de données hétérogènes et distribuées, Université Mentouri de Constantine, 2005.
- [118] Tarek MELLITI, Interopérabilité des services Web complexes. Application aux systèmes multi-agents, Doctorat de l'Université Paris IX Dauphine le 8 décembre 2004.
- [119] P.F. Brown, R. Metz, and B.A. Hamilton. Reference model for service oriented architecture 1.0. Technical report, Tech. rep., 2006. Retrieved December 16, 2008. from <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>, 2005.
- [120] Thomas Erl. *Service-Oriented Architecture : A Field Guide to Integrating XML and Web Services*. Prentice Hall PTR, 2004.
- [121] H. VERJUS, « Conception et construction de fédérations de progiciels » Thèse de Doctorat, ESIA, 2001.
- [122] M. Uschold, M. King, S. Moralee & Y. Zorgios « The enterprise ontology », University of Edinburgh, 1997.
- [123] Yasmine CHARIF, Chorégraphie dynamique de services basée sur la coordination d'agents introspectifs. Thèse de doctorat. UNIVERSITE PIERRE ET MARIE CURIE, 2007.
- [124] M. Mrissa ; “ Médiation Sémantique Orientée Contexte pour la Composition de Services Web ”. Thèse de doctorat de l'université Claude Bernard Lyon I. 15 Novembre 2007.
- [125] David Martin, Mark Burstein, Drew Mcdermott, Sheila Mcilraith, Massimo Paolucci, Katia Sycara, Deborah L. Mcguinness, Evren Sirin, and Naveen Srinivasan. Bringing semantics to web services with owl-s. *World Wide Web*, 10:243\_277, September 2007.
- [126] S. Arroyo and M. Stollberg. Wsmo primer. wsmo deliverable d3. 1, deri working draft. Technical report, Tech. Rep., WSMO. <http://www.wsmo.org/2004/d3/d3.1>, 2004.
- [127] J. Farrell and H. Lausen. Semantic annotations for wsdl and xml schema. W3c recommendation, 28, 2007.
- [128] R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M.T. Schmidt, A. Sheth, and K. Verma. Web service semantics-wsdl-s. W3C member submission, 7, 2005.
- [129] J. Kopecky, T. Vitvar, C. Bournez, and J. Farrell. Sawsdl: Semantic annotations for wsdl and xml schema. *IEEE Internet Computing*, pages 60\_67, 2007.



- [130] J. Miller, K. Verma, P. Rajasekaran, A. Sheth, R. Aggarwal, and K. Sivashanmugam. Wsdl-s: Adding semantics to wsdl-white paper. Technical report, Tech. Rep., Large Scale Distributed Information Systems. <http://lstdis.cs.uga.edu/library/download/wsdl-s.pdf>, 2004.
- [131] G. Weiderhold ; ‘‘ Mediators in the Architecture of future Information Systems’’. IEEE Computer, Vol. 25 (N° 3), pp. 38-49, 1992.
- [132] G. Athanasopoulos, A. Tsalgatidou, and M. Pantazoglou. Interoperability among heterogeneous services. In Society [69], pp. 174–181.
- [133] C. Bussler ; ‘‘Semantic web services : Reflections on web service mediation and composition’’. In WISE, pp. 253–260. IEEE Computer Society, 2003.
- [134] L. Cabral and J. Domingue. Mediation of semantic web services in irs-iii. In First International Workshop on Mediation in Semantic Web Services (MEDIATE 2005) held in conjunction with the 3rd International Conference on Service Oriented Computing (ICSOC2005), Amsterdam, The Netherlands., December 12th 2005.
- [135] M. Dumas, M. Spork ; ‘‘Adapt or perish : Algebra and visual notation for service interface adaptation’’. In S. Dustdar, J. L. Fiadeiro, and A. P. Sheth, editors, Business Process Management, volume 4102 of Lecture Notes in Computer Science, pp. 65–80. Springer, 2006.
- [137] Y. Taher, D. Benslimane, M.-C. Fauvet, and Z. Maamar ; ‘‘ Towards an approach for web services substitution’’. In P. Ghodous, R. Dieng-Kuntz, and G. Loureiro, editors, IDEAS, pp. 166–173. IOS Press, 2006.
- [138] SHETH A., LARSON J. Federated database systems for managing distributed, heterogeneous, and autonomous databases, ACM Computer Surveys 22, 3 Sept. 1990
- [139] WIEDERHOLD G., Interoperation, mediation and ontology. In proceedings of the International Symposium on Fifth Generation Computer Systems (FGCS94). Workshop on heterogeneous Cooperative Knowledge-Bases, Tokyo, Japan, December 1994
- [140] BUNEMAN P., RASCHID L. et ULMAN J. D., Mediators Languages – a Proposal for a Standard. SIGMOD Record 1997.
- [141] LIU L. et al., Adaptive Query Mediation across Heterogeneous Information Sources. In proceedings of the First IFCIS International Conference on Cooperative Information Systems (CoopIS’96) Brussels, Belgique IEEE-CS Juin 96.
- [142] A.TOMASIC et al., Scaling Heterogeneous Databases and the Design of Disco, In Proceedings of the 16th International Conference on Distributed Computing Systems (ICDCS’96) Hong Kong, IEEE Computer Society Press, Mai 1996.
- [143] GARCIA-MOLINA H. et al. The Tsimmis approach to mediation: data models and languages. Journal of Intelligent Information System (JIIS) 1997.
- [144] SUBRAHMANIAN V. et al., HERMES : a heterogeneous reasoning and mediator system 1995
- [145] CLUET S. et al., Your mediators need data conversion! , in Haas L. M. et Tiwary A., Eds, proceedings ACM SIGMOD International Conference on Management of Data, Seattle, Washington, USA, ACM Press 98.
- [146] LUDASCHER E. et al. A framework for navigation-driven Lazy mediators. In Cluet S et Milo T., Eds, Proceedings of the ACM SIGMOD Workshop on the Web and Databases (WebDB’99), Philadelphia, Pennsylvania, USA, INRIA, Juin 1999.
- [147] AMBITE J. L. et KNOBLOCK C. A., Reconciling Distributed Information Sources. In Working Notes of the AAAI Spring Symposium on Information Gathering in Distributed Heterogeneous Environments. Palo Alto, CA, 1995.

- [148] FENSEL A. D. et al., Ontobroker: Or How to Enable Intelligent Access to the WWW. In proceedings of the 11th Banff Knowledge Acquisition for Knowledge-Based System Workshop (KAW98). Banff, Canada, Avril 1998.
- [149] BRESSAN S. et al. The COntext INterchange Mediator Prototype. In SIGMOD 1997, proceedings ACM SIGMOD International Conference on Management of Data. Tucson, Arizona, USA, ACM Press, May 1997.
- [150] MENA E. et al., An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. International Journal of Distributed and Parallel Databases (DAPD) 2000.
- [151] BAYARDO R. et al., Semantic Integration of Information in Open and Dynamic Environments. Rapport technique MCC-INSL-088-96, Microelectronics and Computer Technology Corporation (MCC), USA, October 1996.
- [152] Nadine CULLOT - Fabrice JOUANOT - Kokou YETONGNON 2003. Une méthode de réconciliation sémantique pour l'extraction de connaissances.
- [153] Fabrice JOUANOT, DILEMMA : vers une coopération de systèmes d'informations basée sur la médiation sémantique et la fusion d'objets. Thèse de doctorat, 21 Novembre 2001, Université de Bourgogne, UFR des Sciences et Techniques.
- [154] Carbonell, J., G., and Siekmann, J., (2006). Agent-Oriented Information Systems. 7th International Bi-Conference Workshop, AOIS 2005 Utrecht, ISBN-10 3-540-48291-1 Springer Berlin Heidelberg New York.
- [155] Purvis, M., Cranefield, S., Bush, G., Carter, D., McKinlay, B., Nowostawski, M., Ward, R., (2000). The NZDIS Project: an Agent-Based Distributed Information Systems Architecture.
- [156] Talens, G., Boulanger, D., and Séguran, M., (2007). Domain Ontologies Evolutions to Solve Semantic Conflicts. M. Collard (Ed.): ODBIS 2005/2006, LNCS 4623, pp. 51–67, 2007., Springer-Verlag Berlin Heidelberg 2007.
- [157] Kim, W. et al., (2008). Agent based intelligent search framework for product information using ontology mapping. *J Intell Inf Syst* (2008) 30:227–247 DOI 10.1007/s10844-006-0026-8, Springer Science + Business Media, LLC 2008.
- [158] Gal, A., Segev, A., (2005). Agent oriented data integration. In: Akoka J et al., editors. *R Workshops 2005*, LNCS 3770, Springer-Verlag, Berlin, Heidelberg, 2005. p. 98–108.
- [159] Bayardo, Jr., R., J., et al., (1997). *InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments*. Microelectronics and Computer Technology Corporation.
- [160] Leclercq, E., Benslimane, D., Yétongnon, K., (1999). ISIS: A Semantic Mediation Model and an Agent Based Architecture for GIS Interoperability, in *Proceedings of the International Database Engineering and Applications Symposium (IDEAS'99)*, Montreal, CANADA, August 2-4, 1999. Published by IEEE Computer Society Press, ISBN 0-7695-0265-2, pp. 81—92.
- [161] Fabio Bellifemine, Giovanni Caire, Dominic Greenwood, *Developing Multi-Agent Systems with JADE*, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, WestSussex PO19 8SQ, England, Copyright 2007.
- [162] Magali Séguran, *These: Résolution des conflits sémantiques dans les systèmes d'information coopératifs : proposition d'un modèle d'interaction entre agents*, Université Jean Moulin, Lyon3, 2003.

- [163] Benharzallah, S., Kazar, O., Caplat, G., (2011). Intelligent query processing for semantic mediation of information systems, *EIJ: Egyptian Informatics Journal*, Vol. 12, pp. 151-163.
- [164] Haller, A., Cimpian, E., Mocan, A., Oren, E., and Bussler, C., (2005). Wsmx - a Semantic service-oriented architecture. In I. C. Society, editor, *ICWS*, pages 321–328. IEEE Computer Society, 2005.
- [165] A. Gugliotta, L. Gabral, J. Domingue, V. Roberto, M. Rowlatt, and R. Davies; “ A Semantic Web Service-based Architecture for the Interoperabilité of E-government Services”. Italy 2006.
- [166] Hausberger, T., (2009). Semantic Annotations for WSDL. Departement of ComputerScience Leopold Franzens University Innsbruck. *Semantic Web Services PS*. 5 May 2009.
- [167] Nadjib MESBAHI, Okba KAZAR, Merouane ZOUBEIDI and Saber BENHARZALLAH. An agent-based modeling for an enterprise resource planning (ERP), *ACIIDS 2014, Bangkok, Thailand LNCS Volume 551*, 2014, pp 225-234, Springer.
- [168] Merouane ZOUBEIDI, Okba KAZAR, Saber BENHARZALLAH and Nadjib MESBAHI. ISPGI : Une architecture sémantique à base d’agents pour l’interopérabilité d’un PGI. *Proceeding of the International Conference on Artificial Intelligence and Information Technology (ICA2IT 2014)*. Ouargla, Algerie.
- [169] Merouane ZOUBEIDI, Okba KAZAR, Nadjib MESBAHI ET Saber BENHARZALLAH. Vers une architecture d’intégration de la sémantique via les Services Web : Etude de cas du Progiciel de Gestion Intégré. *La conférence francophone sur les Systèmes Collaboratifs (SysCo’2014)*. Hammamet, Tunisie, pp. 101-114.
- [170] Merouane ZOUBEIDI, Okba KAZAR, Saber BENHARZALLAH and Nadjib MESBAHI. Toward an architecture for the Semantic Integration in an Enterprise Resource Planning. *Proceeding of the 3rd IEEE International Workshop on Advanced Information Systems for Enterprises (IWAISE’14)*. Tunis, Tunisia. [Http://www.lifl.fr/iwaise14/program.html](http://www.lifl.fr/iwaise14/program.html).
- [171] Merouane Zoubeidi, Okba Kazar, Saber Benharzallah, Nadjib Mesbahi. Une approche d’interopérabilité sémantique via les services web dans les progiciels de gestion intégrée. *Revue courrier du savoir*, Volume 26, Mars 2018, Pages 381-392.
- [172] N. Dailly, « Utilisation des systèmes multiagents dans les EIAH », Université de Technologie de Compiègne (UTC), <http://www.dailly.info/>, Octobre 2007.
- [173] Jack Intelligent Agents : <http://www.agent-software.com.au/>
- [174] Zeus Agent Building Toolkit : <http://www.labs.bt.com/projects/agents/zeus/index.htm>
- [175] Jade : <http://sharon.csel.it/projects/jade/>
- [176] MadKit : <http://www.madkit.org/>
- [177] AgentBuilder : <http://www.agentbuilder.com/>
- [178] Tony Garneau & Sylvain Delisle, Programmation orientée-agent : évaluation comparative d’outils et environnements, *JFIADSMA 2002*, 28-30 octobre, Lille, France
- [179] <http://jade.tilab.com/>
- [180] <https://gforge.inria.fr/projects/ontosim/>
- [181] <http://alignapi.gforge.inria.fr/lib.html>
- [182] <http://owlapi.sourceforge.net/>
- [183] Jérôme Euzenat, Alignment API and server INRIA & LIG , Montbonnot, France, February 29, 2008
- [184] Jérôme Euzenat. An API for ontology alignment. In *Proc. 3rd ISWC*, pages 698–712, Hiroshima (JP), 2004.
- [185] Fabio Bellifemine, Giovanni Caire, Dominic Greenwood, *Developing Multi-Agent*

Systems with JADE, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, Copyright 2007

[186] Gregory D. Speegle, *JDBC: Practical Guide for Java Programmers*, The Morgan Kaufmann Practical Guides Series Series Editor Michael J. Donahoo, ISBN: 1-55860-736-6, ISBN: 1-55860-736-6.