

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Béjaïa

Faculté des Sciences Exactes
Département d'Informatique
Ecole doctorale Réseaux et Systèmes Distribués

Mémoire de Magistère

En Informatique

Option : Réseaux et Systèmes Distribués

Thème

**Gestion de la sémantique dans les systèmes
d'information coopératifs basés web**

Présenté par

Mounira ZEKIOUK

Devant le jury composé de :

Président	Abdelnacer DAHMANI	Professeur	Université de Béjaïa.
Rapporteur	Mahmoud BOUFAIDA	Professeur	Université de Constantine.
Examineur	Abdelkamel TARI	MCA	Université de Béjaïa.
Examineur	Abdessamed reda GHOMARI	MCA	E.S.I, Alger.

Promotion 2008-2009

Remerciements

J'exprime mes profonds remerciements à mon promoteur, *Mr Mahmoud Boufaïda*, Professeur de l'université de Constantine, pour l'aide et le temps qu'il a bien voulu me consacrer et sans qui, ce mémoire n'aurait jamais vu le jour.

Je remercie Mr *Abdelnacer DAHMANI*, Professeur à l'université de Bejaia pour m'avoir fait l'honneur d'accepter de présider le jury de ma thèse.

Des remerciements particuliers à Mr *Abdessamed reda Ghomari*, Maître de conférences à E.S.I, Alger, pour avoir accepté d'être membre du jury de ma thèse.

J'exprime ma gratitude à Mr *Abdelkamel TARI*, le chef de département d'informatique et responsable de l'école doctorale ReSyD, sans qui, il m'aurait sans doute été très difficile de soutenir ma thèse dans de bonnes conditions.

Finalement j'adresse un grand merci du fond du cœur à toute ma chère famille, à tous mes collègues et mes amis qui ont toujours été présents dans les moments les plus difficiles.

Sommaire

Introduction générale.....	1
a. Contexte.....	1
b. Problématique et Objectifs du travail.....	2
c. Organisation du document.....	3
Chapitre 1 : Caractéristiques des systèmes d'information coopératifs.....	5
1.1. Introduction.....	5
1.2. Présentation des SIC.....	6
1.2.1. Origine des SIC.....	6
1.2.2. Définition des SIC.....	7
1.2.3. Quelques concepts émergents du travail coopératif.....	8
1.2.3.1. Définition du Groupware.....	8
1.2.3.2. Différentes applications du Groupware.....	8
1.2.4. Propriétés des systèmes d'information coopératifs.....	9
1.2.4.1. Distribution.....	9
1.2.4.2. Autonomie.....	9
1.2.4.3. Hétérogénéité.....	10
1.3. Classification des SIC.....	11
1.3.1. SIC fortement couplés.....	12
1.3.2. SIC faiblement couplés.....	13
1.4. Apport des SMA dans la modélisation des SIC.....	14
1.4.1. Concept d'agent.....	14
1.4.2. Approches de conception d'agents.....	15
1.4.2.1. Agents cognitifs.....	15
1.4.2.2. Agents réactifs.....	15
1.4.2.3. Agents hybrides.....	16
1.4.3. Systèmes multi-agent (SMA).....	16
1.4.4. Interactions dans les SMA.....	17
1.4.4.1. Communication.....	17
1.4.4.2. Coopération.....	18
1.4.4.3. Coordination.....	19
1.4.4.4. Négociation.....	19
1.5. Quelques architectures de SIC basées agents.....	20
1.5.2. Le système DAARache.....	20
1.5.3. Le système Infosleuth.....	20
1.5.4. Le système Netsa.....	20
1.6. Conclusion.....	21
Chapitre 2 : Notion de sémantique et d'interopérabilité sémantique dans Les systèmes d'information coopératifs.....	22
2.1. Introduction.....	22
2.2. Notion de sémantique.....	23
2.2.1. Définition.....	23
2.2.2. Typologie des hétérogénéités sémantiques.....	23

2.2.3. Classification de la sémantique.....	23
2.2.4. Représentation de la sémantique.....	24
2.3. Notion d'ontologie	24
2.3.1. Définitions	24
2.3.2. Constituants d'une ontologie	25
2.3.3. Rôle des ontologies.....	26
2.3.3.1. La communication.....	26
2.3.3.2. L'aide à la spécification de systèmes.....	26
2.3.3.3. L'interopérabilité	26
2.3.3.4. L'indexation et la recherche d'information	26
2.3.4. Dimensions de classification d'ontologies.....	27
2.3.4.1. Classification selon l'objet de conceptualisation.....	27
2.3.4.2. Classification selon le niveau du formalisme de représentation.....	28
2.3.5. Quelques formalismes de représentation de connaissance.....	28
2.3.6. Langages de représentation des ontologies.....	29
2.3.7. Approches d'utilisation des ontologies.....	30
2.3.7.1. Approche mono-ontologie.....	31
2.3.7.2. Approche multi-ontologies.....	31
2.3.7.3. Approche hybride.....	32
2.4. Notion d'interopérabilité	32
2.4.1. Définitions.....	32
2.4.2. Techniques pour l'interopérabilité sémantique basées sur les ontologies.....	34
2.4.2.1. Alignement des ontologies.....	34
2.4.2.2. Fusion d'ontologies.....	34
2.4.2.3. Mapping d'ontologies	35
2.5. Description générale du processus et des outils de Mapping	35
2.5.1. Processus de Mapping.....	35
2.5.2. Méthodes de découverte de mappings.....	36
2.6. Quelques travaux portant sur la gestion de la sémantique dans les SIC	37
2.6.1. L'architecture OWSCIS.....	37
2.6.2. L'architecture ACSIS.....	40
2.6.3. L'architecture DECA.....	41
2.6.4. Une architecture multi-couches pour le traitement distribué de requêtes.....	42
2.6.5. Une architecture multi-agents pour le traitement distribué de requêtes.....	44
2.6.6. Discussion.....	46
2.7. Conclusion	47

Chapitre 3 : Une architecture Multi-Agents pour la mise en œuvre de L'interopérabilité Sémantique dans les SIC..... 48

3.1. Introduction	48
3.2. Vue globale de l'approche proposée	49
3.3. Une architecture générique pour l'approche proposée	50
3.3.1. Agents d'Information	51
3.3.1.1. Module de communication.....	53
3.3.1.2. Cache sémantique.....	53
3.3.1.3. Base de connaissances.....	53
3.3.1.4. Base de description des schémas locaux.....	56
3.3.1.5. Moteur de requêtes Globales.....	56
3.3.1.6. Planificateur local.....	56
3.3.1.7. Bibliothèque de fonctions.....	57
3.3.2. Agents Adaptateurs.....	57
3.3.2.1. Module de communication.....	57
3.3.2.2. Traduction syntaxique.....	57

3.3.3. Agent Annuaire.....	58
3.3.3.1. Registre d'expertise.....	59
3.3.3.2. Module de gestion.....	60
3.3.3.3. Module de Communication.....	60
3.3.4. Communication entre les agents de l'architecture.....	60
3.4. Formulation des requêtes	60
3.4.1. Grammaire utilisée	61
3.4.2. Sujet d'une requête	62
3.5. Spécification du protocole coopératif de traitement des requêtes globales	62
3.5.1. Traitement locale de la requête.....	63
3.5.2. Traitement coopératif de la requête.....	64
3.5.2.1. La phase d'initialisation.....	65
3.5.2.2. La phase des traitements locaux de la requête.....	65
3.5.2.3. La phase de recombinaison des résultats.....	68
3.6. Conclusion.....	68
Chapitre 4 : Etude de Cas et perspectives d'implémentation	70
4.1. Introduction	70
4.2. Etude de cas Coopération des sources d'information du domaine touristique et du domaine de la location des maisons.....	71
4.3. Outils d'implémentation utilisés.....	72
4.3.1. La plate forme JADE.....	72
4.3.2. Protégé.....	73
4.4. Simulation des agents avec JADE.....	74
4.4.1. Niveau de coopération.....	74
4.4.2. Niveau des adaptateurs.....	74
4.4.3. L'agent annuaire.....	75
4.5. Construction des ontologies.....	76
4.5.1. Ontologie locale 1.....	76
4.5.2. Ontologie locale 2.....	76
4.5.3. Ontologie globale.....	77
4.6. Construction des tables de mappings.....	77
4.7. Exemples de traitement de requêtes.....	78
4.7. Conclusion.....	84
Conclusion générale.....	85
Références bibliographiques.....	87

Liste des figures

Figure 1.1. Domaines d'héritages des SIC	6
Figure 1.2. Propriétés des SIC	9
Figure 1.3. Classification de l'hétérogénéité	10
Figure 1.4. Classification des approches de SIC	12
Figure 1.5. Un agent autonome dans son environnement	15
Figure 1.6. Communication par envoi de message.....	17
Figure 1.7. Communication par partage d'information	18
Figure 2.1. Pile des langages d'annotation d'ontologies.....	29
Figure 2.2. Approche mono-ontologie.....	31
Figure 2.3. Approche multi-ontologies.....	31
Figure 2.4. Approche hybride.....	32
Figure 2.5. Processus de mappings d'ontologies.....	35
Figure 2.6. Composants de l'architecture OWSCIS.....	38
Figure 2.7. Différents niveaux de l'architecture ACSIS.....	40
Figure 2.8. Différents composants de l'architecture DECA.....	42
Figure 2.9. Architecture multi-couches pour le traitement distribué de requêtes.....	43
Figure 2.10. Architecture multi-agents pour le traitement distribué de requêtes.....	45
Figure 3.1. Interaction des agents pour le traitement des requêtes.....	49
Figure 3.2. Approche ontologique hybride pour l'interopérabilité sémantique.....	50
Figure 3.3. Une architecture générique pour les SIC.....	51
Figure 3.4. Architecture interne d'un agent d'information.....	53
Figure 3.5. Les composants de la base de connaissances.....	54
Figure 3.6. Etapes de construction de l'ontologie globale.....	55
Figure 3.7. Architecture interne d'un agent adaptateur.....	58
Figure 3.8. Architecture interne de l'agent annuaire.....	59
Figure 3.9. Traitement des requêtes globales.....	63
Figure 3.10. Traitement local.....	64
Figure 3.11. Etape du protocole de coopération.....	64
Figure 4.1 lancement des agents dans des containers différents.....	74
Figure 4.2. Inscription des agents d'information AI1 et AI2 au prés de DF.....	75
Figure 4.3. Simulation des agents avec GUI de JADE.....	75
Figure 4.4. Classes et relations de l'ontologie1.....	76
Figure 4.5. Classes et relations de l'ontologie2.....	76
Figure 4.6. Classes et relations de l'ontologie globale.....	77
Figure 4.7. Requête ACL envoyée par AI1 à AA1.....	79
Figure 4.8. Résultats ACL envoyés par AA1 à AI1.....	80
Figure 4.9. Requête ACL envoyée à l'agent AI2.....	81
Figure 4.10. Requête ACL envoyé par AI1 à AA1.....	82
Figure 4.11. Résultats envoyés par l'agent AA1 à AI1.....	82

Liste des tableaux

Tableau 1.1. Agents cognitifs vs agents réactifs.....	16
Tableau 3.1. Description des composants de la requête.....	61
Tableau 3.2. Table de mappings de l'agent AI1.....	66
Tableau 3.3. Table de mappings de l'agent AI3.....	66
Tableau 4.1. Structure d'un message FIPA-ACL.....	73
Tableau 4.2. Extrait des tables de mappings des deux agents AI1 et AI2.....	78

a. Contexte

Les mutations économiques et l'évolution rapide des technologies ont bouleversé profondément le domaine de l'informatique et plus particulièrement celui des **systèmes d'information (SIs)**. En particulier, les besoins d'ouverture et de partage d'information entre SIs sont devenus importants surtout dans le cadre de l'évolution des performances des réseaux de communication. Les nouvelles applications telles que les applications de commerce électronique, de simulation et les systèmes d'aide à la décision ont besoin d'échanger et de partager différentes sources d'information afin de synthétiser les informations provenant de ces sources [43]. Ce partage nécessite une communication inter-systèmes afin de satisfaire les nouveaux besoins de ces applications qui exigent la cumulation des ressources et des expertises des différents SIs. Par conséquent, les SIs doivent désormais être coopératifs.

Cette nouvelle génération de SIs est appelée **Systèmes d'Information Coopératifs (SIC)**. Ces systèmes sont apparus pour supporter les relations de coopération qui lient un ensemble de SIs. Leur but est d'améliorer les échanges et le partage des informations entre les différents systèmes coopérants. Il s'agit de systèmes offrant les mêmes fonctionnalités qu'un système classique mais devant par ailleurs supporter le travail coopératif. C'est un domaine qui emprunte à des domaines variés comme les **Systèmes d'Information (SIs)**, **l'intelligence Artificielle Distribuée (IAD)**, **les Bases de données (BD)**, **les Sciences de gestion et la Sociologie**,...etc. [54]. En effet, plusieurs sens sont attribués au terme SIC. Ce terme peut désigner entre autre, le support de la coopération au travers des bases de données, l'intégration de la dimension sociale ou bien la dérivée de la structure de données à partir de plusieurs sources d'information qui n'est pas toujours connue [42]. Par conséquent, la coopération des SIs n'a pas toujours la même signification. Dans notre contexte de travail, nous abordons la problématique des SIC en s'intéressant à la coopération des sources d'information distribuées et hétérogènes.

Dans les SIC, les opérations de production (de bien ou de service) sont largement distribuées géographiquement et liées conceptuellement en termes de dépendances de matériels, des flux d'information et de connaissances. Les SIs qui supportent ces opérations sont autonomes et dans la plus part des cas sont hétérogènes car ils ne sont pas nécessairement conçus pour travailler ensemble. Chaque composant est mis en œuvre pour satisfaire des objectifs locaux

[54]. Ces propriétés de distribution, d'autonomie et d'hétérogénéité constituent les caractéristiques essentielles des SIC. Par conséquent, La mise en œuvre d'un SIC nécessite la prise en compte de ces propriétés en plus de quelques aspects tel que le partage d'information, le parallélisme des tâches, la prise de décision collective, l'ouverture de l'environnement, l'interopérabilité et la communication entre les différents SIs qui le composent. L'IAD et en proposant le paradigme de **systèmes multi-agents (SMA)**, définit des réponses aux différents besoins des SIC et peut représenter un bon support pour leur modélisation.

b. Problématique et Objectifs du travail

Dans le cadre de notre travail nous nous intéressons à la coopération de sources d'information distribuées et hétérogènes dont l'objectif est de permettre l'utilisation conjointe de ces sources d'une manière transparente.

La mise en œuvre d'un processus coopératif pour le traitement des requêtes globales requiert la résolution des conflits issus de la coopération des sources d'information hétérogènes. Cette hétérogénéité constitue un obstacle majeur à la coopération. L'interopérabilité fournit des solutions pour répondre à ces besoins. Elle a pour objectif de permettre à un utilisateur d'utiliser, de façon transparente, des données issues d'un ensemble de systèmes d'information autonomes, répartis et hétérogènes. Les conflits découlant de l'hétérogénéité des sources d'information peuvent se situer au niveau technique et font référence à la coexistence de systèmes d'exploitation et de plateformes disparates, les conflits syntaxiques concernent les différences au niveau des modèles de données. Dans le cadre du présent travail, nous nous intéressons à l'hétérogénéité sémantique. Cette hétérogénéité présente un défi majeur dans un processus coopératif. Elle est due aux différentes interprétations des objets du monde réel. En effet, les sources d'information sont conçues indépendamment par des concepteurs différents, ayant des objectifs différents. Chacun peut donc avoir un point de vue différent sur le même concept.

Pour assurer l'interopérabilité sémantique, l'information échangée entre systèmes doit d'abord être décrite dans une structure formelle permettant de préserver sa sémantique. Le domaine de l'ingénierie des connaissances propose des méthodologies et des techniques pour percevoir, analyser, organiser et partager des connaissances entre différents systèmes. Parmi ces techniques ; les *ontologies* sont jugées comme le moyen le plus efficace pour la

représentation des connaissances. L'interopérabilité au niveau sémantique requiert aussi l'utilisation de techniques et de méthodologies qui établissent sémantiquement des liens de correspondances entre les informations fournies par les entités communicantes du système coopératif. Dans la littérature, la recherche de ces liens s'appelle *le mapping d'ontologies*. Le mapping vise à trouver les correspondances entre les concepts appartenant à différentes ontologies. Dans cet esprit, plusieurs travaux se sont intéressés à la notion de sémantique et sa représentation. Ces travaux ont notamment montré l'efficacité de la représentation formelle de la sémantique par des ontologies dans le cadre des SIs. Cette notion d'interopérabilité sémantique entre SIs est au cœur de notre problématique.

Le travail rapporté dans ce mémoire vise deux objectifs. Le premier consiste à construire une architecture basée agent pour supporter la coopération des sources d'information distribuées et hétérogènes. Le deuxième permet la résolution des conflits sémantiques entre Ces sources. Afin d'atteindre ces deux objectifs, notre contribution se situe à l'intersection de plusieurs domaines de recherches tel que : les SIC, les BDD (base de données distribués) et les SMA. Nous empruntons également des notions de l'ingénierie ontologique pour assurer la gestion de la sémantique entre les déférentes sources de la coopération.

Nous avons donc développé une architecture multi-agents organisée en niveaux où le contrôle et les connaissances sont décentralisés. Les agents de cette architecture communiquent dans un cadre coopératif pour assurer un traitement transparent aux requêtes des utilisateurs. Notre deuxième contribution concerne la combinaison de l'approche ontologique hybride avec le paradigme multi-agents pour assurer une interopérabilité sémantique entre les différentes sources d'information. Cette interopérabilité est basée sur la description du contenu des sources d'information par des ontologies locales et la définition des correspondances entre ces ontologies pour faciliter la décomposition et la traduction des requêtes.

c. Organisation du document

Le document est structuré en quatre chapitres comme suit :

- A travers le premier chapitre, nous essayons de bien expliquer le contexte de notre travail. Nous présentons les SIC en indiquant leurs origines et leurs caractéristiques. Nous montrons ensuite l'apport des SMA dans la modélisation des SIC. Enfin, nous présentons quelques SIC basés sur ce paradigme.

- Le deuxième chapitre est dédié à l'étude de la notion de sémantique et de l'interopérabilité sémantique. Aussi la notion d'ontologie est présentée comme outil important pour assurer l'interopérabilité au niveau sémantique dans les SIC.
- Le troisième chapitre présente d'une manière détaillée le système que nous avons proposé. Nous commençons d'abord par présenter les fondements de ce dernier. Ensuite, nous exposons l'architecture multi-agents qui supporte la coopération des sources d'information. Enfin, nous terminons le chapitre par la présentation du modèle d'interaction entre les agents de l'architecture et en particulier, le protocole de coopération entre les agents d'information.
- Le dernier chapitre fait l'objet d'une étude de cas qui met en évidence les concepts et mécanismes de notre architecture.

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

Chapitre 1

Caractéristiques des Systèmes d'Information Coopératifs

1.1. Introduction

Le développement rapide des technologies dans ces dernières décennies a bouleversé profondément la vie des organisations. Les systèmes d'information supportant ces organisations n'opèrent plus seulement dans leur propre localisation. L'idée de faire travailler d'une manière synergique ces différents systèmes est devenue un besoin crucial afin de répondre aux nouvelles exigences des organisations.

Cette nouvelle tendance des systèmes d'information a donné naissance à une nouvelle génération de systèmes d'information appelés les systèmes d'information coopératifs (SIC). De nombreux travaux menés dans plusieurs domaines de recherche ont pour objectif de proposer des solutions pour la conception et le développement de tels systèmes d'information. Ces travaux de recherche ont abouti à l'émergence de différentes techniques supportant la coopération de systèmes d'information hétérogènes et autonomes. .

L'objet du présent chapitre est de présenter la notion de SIC et de discuter quelques outils et approches de conception de ces derniers.

1.2. Présentation des SIC

Nous nous focalisons dans cette section sur le domaine d'intérêt de notre travail, les SIC, nous essayons de présenter une description de ces derniers et leurs caractéristiques.

1.2.1. Origine des SIC

Le paradigme des SIC est un domaine relativement récent qui a comme objectif le développement des SIs distribués. Dans les premiers travaux de recherches, ce paradigme a été identifié comme la future génération des SIs. Les SIC posent une série de défis technologiques qui sont dûs selon le point de vue de certains scientifiques aux avancées technologiques (dans les télécommunications, matériels, logiciels, etc.), comme c'est le cas dans presque tous les systèmes coopératifs assistés par ordinateur. Du point de vue d'autres chercheurs, ces défis sont dûs principalement au changement dans les structures d'organisation du monde de travail [54]. Afin de délimiter les frontières de ce nouveau domaine, une identité lui a été attribuée. Cette identité permet de le situer par rapport aux autres disciplines de l'informatique. En effet, la recherche dans ce domaine intègre des idées qui sont issues des SIs, des bases de données, du génie logiciel et de l'intelligence artificielle...etc.

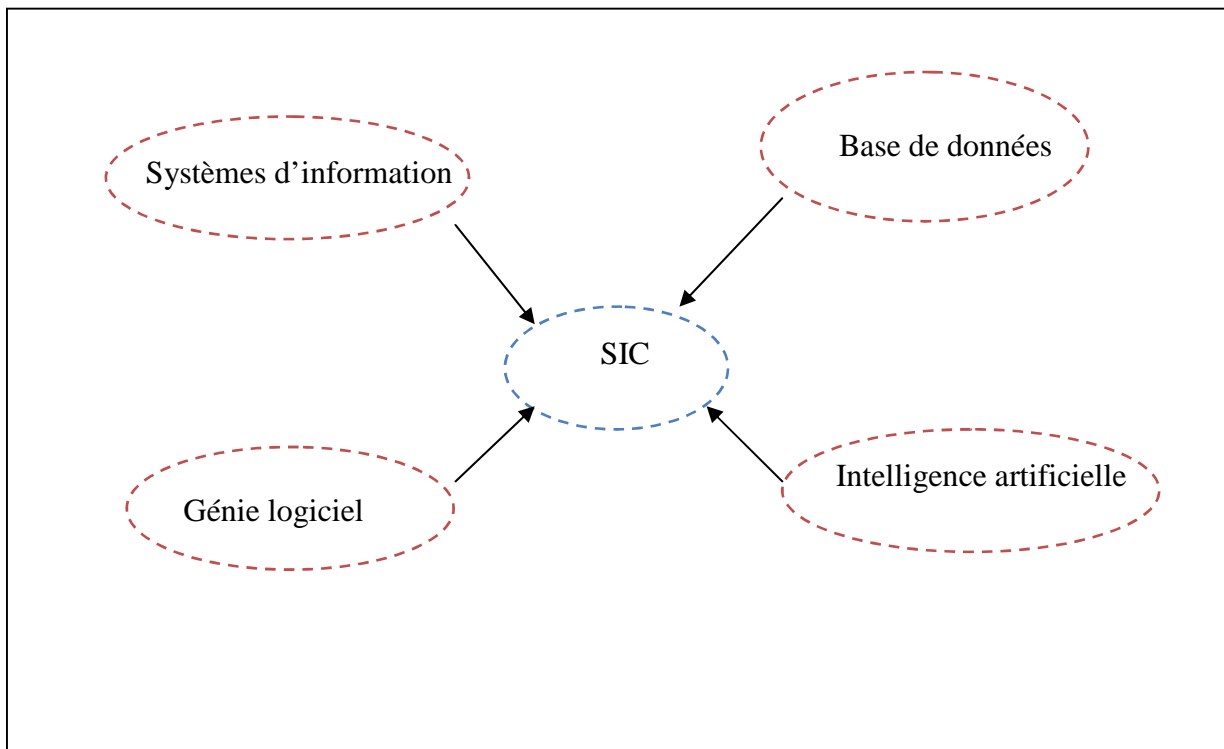


Figure 1.1. Domaines d'héritage des SIC [54].

1.2.2. Définition des SIC

Le terme de coopération dans le domaine des SIs peut se définir comme la capacité pour un ensemble de systèmes d'information autonomes, coexistant dans un environnement virtuel commun, à accomplir une tâche. Chaque système possède sa propre existence, indépendamment de la tâche globale à résoudre, dont l'accomplissement nécessite plus de ressources que n'en possède chaque système séparément [44].

Plusieurs définitions ont été attribuées aux SIC.

La communauté de l'Intelligence Artificielle Distribuée (IAD) définit un SIC comme :

« Un ensemble d'agent " computationnels" qui partagent continuellement des objectifs avec d'autre systèmes d'information, des agents humains aussi bien avec l'organisation au sein de leur environnement opérationnel » [27].

Dans le domaine du CSCW (Computer Supported Cooperative Work) un SIC est :

« Un ensemble de composants plus ou moins autonomes, souvent préexistants qui travaillent de manière synergique en échangeant information, expertise et en coordonnant leurs activités » [5].

Dans l'approche des bases de données distribuées, un SIC :

« Intègre des sources d'information distribuées, base de données ou systèmes à base de connaissance pouvant utiliser des représentations de connaissances et de données hétérogènes » [6].

En réalité, la définition des SIC englobe toutes ces définitions, comme les suggèrent les travaux de Zarour [54] qui définit un SIC comme *« un environnement qui rassemble un nombre important de SIs distribués au travers d'un grand nombre de réseaux d'ordinateurs et de communication complexe. Ces systèmes supportent toutes les catégories d'application, des plus simples au plus complexe. De tels SIs gèrent et accèdent à une grande quantité d'information et de services automatisés. Ils supportent également le travail humain individuel ou de groupe. L'exécution des tâches s'effectue de manière concurrente sur des réseaux à l'aide de logiciel spécifiques. Les informations et les services sont disponibles sous plusieurs formes grâce à des référentiels. La communication entre les systèmes compétents peut s'effectuer de façon centralisée ou distribuée grâce à des protocoles de communication ».*

Notre travail est lié à la coopération des sources d'information distribuées et hétérogènes. Alors, nous considérons un SIC comme un ensemble de composants autonomes qui interagissent pour assurer l'accès transparent et unifié à plusieurs sources d'information distribuées et hétérogènes.

1.2.3. Quelques concepts émergents du travail coopératif

Plusieurs technologies réseaux ont permis l'apparition des SIC. Parmi ces technologies nous citons les services de messageries, de conférences électroniques, du workflow et des agendas. Ces outils renvoient à la notion de Groupware.

1.2.3.1. Définition du Groupware

Le Groupware est défini comme l'ensemble des technologies et des méthodes de travail associées qui, par l'intermédiaire de la communication électronique, permettent le partage de l'information sur un support numérique à un groupe engagé dans un travail coopératif. C'est l'ensemble des processus et des procédures d'un groupe de travail devant atteindre un objectif particulier plus les logiciels conçus pour faciliter ce travail de groupe [62].

1.2.3.2. Différentes applications du Groupware

Dans cette section nous présentons quelques applications du Groupware.

- **Le Workflow :** La WFMC (WorkFlow Management Coalition), définit le workflow comme *«l'automatisation de tout ou partie d'un processus d'entreprise, au cours duquel l'information circule d'une tâche à l'autre, c'est-à-dire d'un participant à l'autre, pour une action en fonction d'un ensemble de règles de gestion»* [61]. La réalisation de tout projet sous-entend l'implication d'un certain nombre de personnes, voire d'un certain nombre d'équipes ou d'entreprises. Ces entreprises interagissant et échangeant des données, de plus en plus via Internet et le Web. Cependant, il ne suffit pas simplement d'échanger des données pour travailler ensemble, il faut aussi contrôler et gérer ces échanges dans le cadre d'une démarche. Chaque entreprise possède sa propre démarche et donc son propre procédé d'entreprise ou Workflow. La coopération entre ces entreprises signifie l'interconnexion de ces Workflow
- **Le système électronique de réunion « Electronic meeting support » :** L'objectif ici est d'utiliser les TIC (Technologie de l'information et de la communication) pour assister aux réunions de groupe en améliorant la productivité du groupe et en assurant la satisfaction des individus en leur donnant les moyens pour s'exprimer.

- **La messagerie électronique** : la messagerie est considérée comme un routage simple de l'information. L'émetteur détient l'information et la distribue aux destinataires qu'il désigne. elle présente plusieurs avantages comme : le gain de temps dû à l'asynchronisme, la certitude de joindre son correspondant, la fiabilité et la rapidité par rapport au courrier postale.

1.2.4. Propriétés des systèmes d'information coopératifs

Différentes propriétés des SIC ont été évoquées dans différents articles. Nous nous limiterons à trois propriétés essentielles: distribution, autonomie et hétérogénéité.

1.2.4.1. Distribution

Comme c'est le cas dans tous les systèmes distribués, les données et les traitements dans un SIC ne sont pas physiquement sur un même lieu ou machine mais, au contraire, répartis sur un réseau. La question de la distribution physique est orthogonale à l'autonomie et à l'hétérogénéité des systèmes.

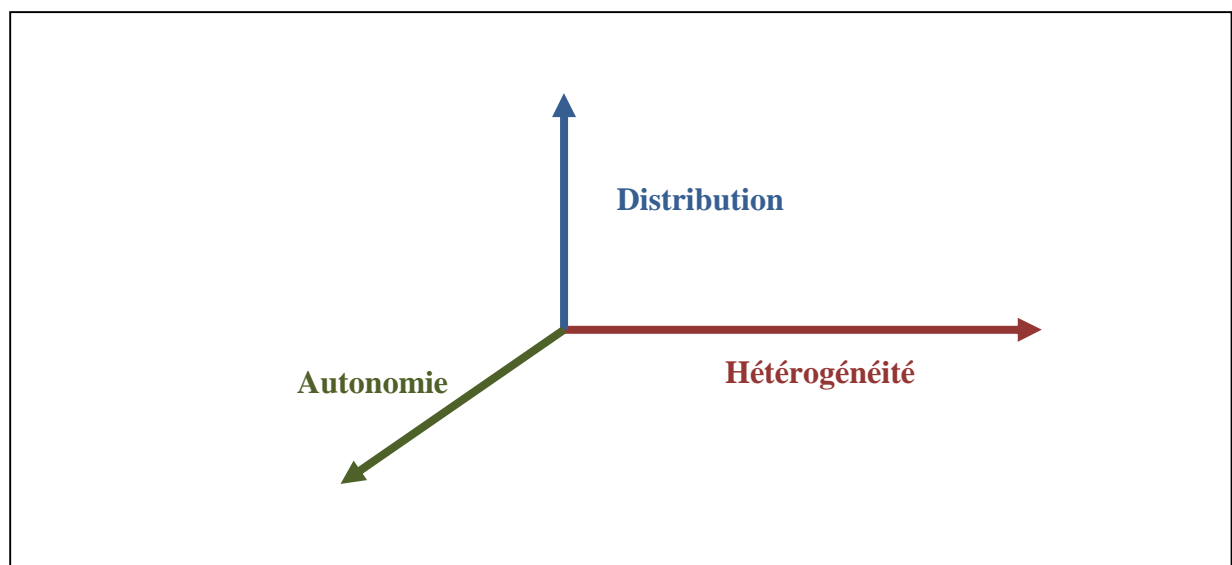


Figure 1.2. Propriétés des SIC [54].

1.2.4.2. Autonomie

L'autonomie est considérée comme la propriété la plus importante des SIC. Elle fait référence au contrôle indépendant et séparé des systèmes composants. Ainsi, l'intégration ou l'abandon d'un composant ne doit pas perturber le fonctionnement global du SIC. L'autonomie apparaît sous différentes formes [41] :

- *l'autonomie de conception*: concerne la liberté du SIC dans le choix de sa propre conception en respectant certaines contraintes comme le modèle de données, le langage de requêtes, la représentation des données et les contraintes d'intégrité et d'implémentation.
- *l'autonomie de communication*: concerne la capacité d'un système composant de décider de la façon de communiquer avec les autres systèmes.
- *L'autonomie d'exécution* : permet à un système d'exécuter ses opérations locales sans interférence avec les opérations soumises par d'autres systèmes composants.

1.2.4.3. Hétérogénéité

Dans un SIC l'hétérogénéité peut survenir à différents niveaux. Elle peut être provoquée par les différences technologiques au niveau des plates-formes matérielles et logicielles. Elle peut être aussi induite par les différences des systèmes d'information composants. Ces différences se situent en général au niveau des modèles de données pour l'expression des structures et des contraintes et au niveau des langages de requêtes [44]. Dans ce qui suit, nous présentons une classification proposée par Busse et al [9]. Dans cette classification, trois types d'hétérogénéités sont distinguées : hétérogénéité syntaxique, hétérogénéité du modèle de données et hétérogénéité logique.

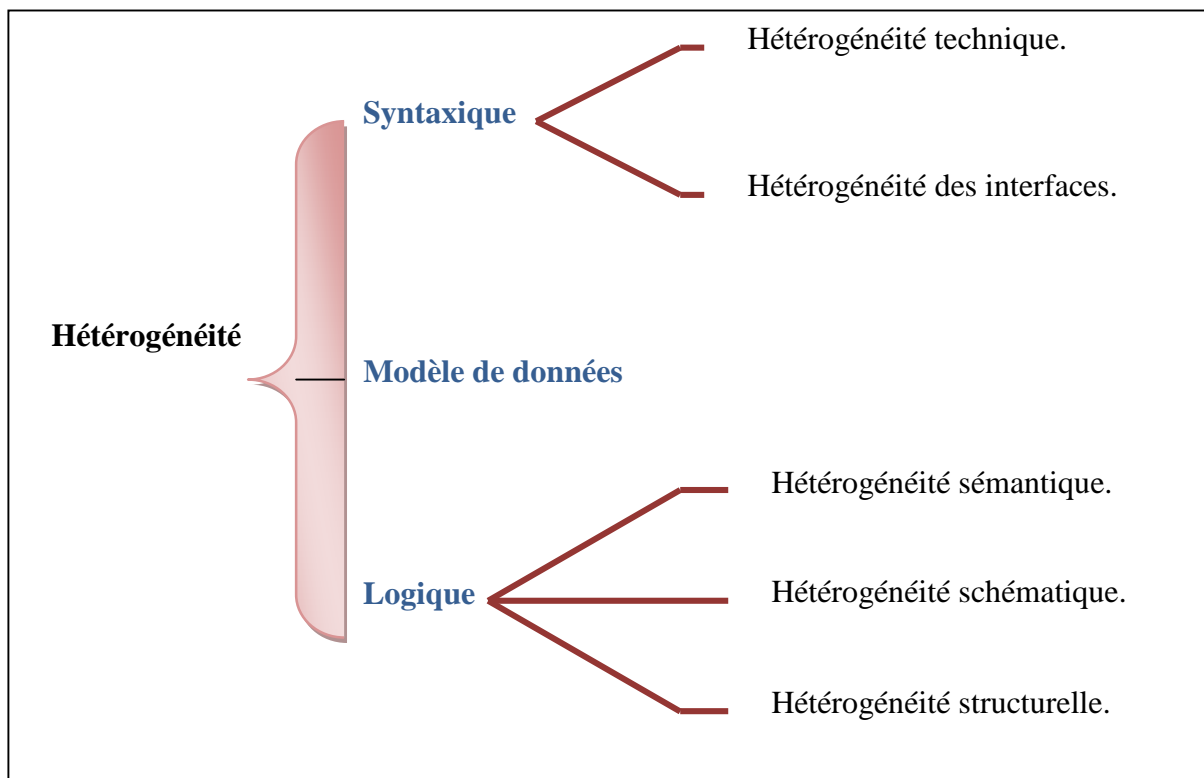


Figure 1.3. Classification de l'hétérogénéité [9]

1. Hétérogénéité syntaxique

L'hétérogénéité syntaxique peut être technique ou liée aux interfaces :

- *l'hétérogénéité technique* : couvre différents aspects techniques comme les plates-formes d'exécution et les systèmes d'exploitation, les méthodes d'accès (protocoles, méthodes de connexion, etc.).
- *l'hétérogénéité des interfaces* : existe si différents composants sont sollicités au travers des langages différents. Il ne s'agit pas dans ce cas des choix techniques mais plutôt des choix des méthodes d'accès.

2. Hétérogénéité des modèles de données

Cette hétérogénéité résulte de l'utilisation de modèles de données différents. La même information est représentée par des syntaxes et des concepts différents d'un modèle à un autre. (Ex : relation dans le modèle relationnel, classe dans le modèle objet, balise dans XML, etc.).

3. Hétérogénéité logique

L'hétérogénéité logique peut être sémantique, schématique ou structurelle :

- *L'hétérogénéité sémantique* : Elle est due aux différentes interprétations des objets du monde réel, même si les schémas qui leur correspondent sont identiques. Cela signifie que si l'on considère un objet dans le monde réel, on peut le représenter de diverses manières dans le monde conceptuel. Deux types de conflits sémantiques ont été mis en évidence. Nous avons l'hétérogénéité de nommage qui consiste en des relations de **synonymie** (un même concept est décrit par deux noms différents), d'**homonymie** (un même nom peut être utilisé pour dénoter deux concepts différents) et l'hétérogénéité d'échelle ou d'unité qui concerne l'utilisation d'unités différentes pour mesurer les valeurs des attributs [45].
- *L'hétérogénéité schématique* : concerne l'encodage des concepts par différents éléments du modèle de données.
- *l'hétérogénéité structurelle* : elle apparaît lorsque des éléments identiques ayant le même sens (contenu sémantique identique) et modélisés avec le même modèle de données sont schématiquement homogènes mais structurés de différentes façons.

1.3. Classification des SIC

De tous les travaux réalisés dans le domaine des SIC résulte une certaine confusion sur la nature et les mécanismes de la coopération. Ces travaux concernent des aspects assez différents. Par conséquent, il est difficile de déduire des critères de classification générale

pour les SIC. Plusieurs classifications sont proposées dans la littérature en se basant sur des critères différents comme par exemple l'autonomie des systèmes ou les architectures logicielles adoptées. Nous présentons dans ce qui suit, une classification basée sur les outils de résolution de conflits de données présentée dans [54]. Cette classification s'adapte bien à notre domaine d'intérêt (la coopération de sources d'information). Elle distingue deux catégories de SIC : les SIC fortement couplés et les SIC faiblement couplés.

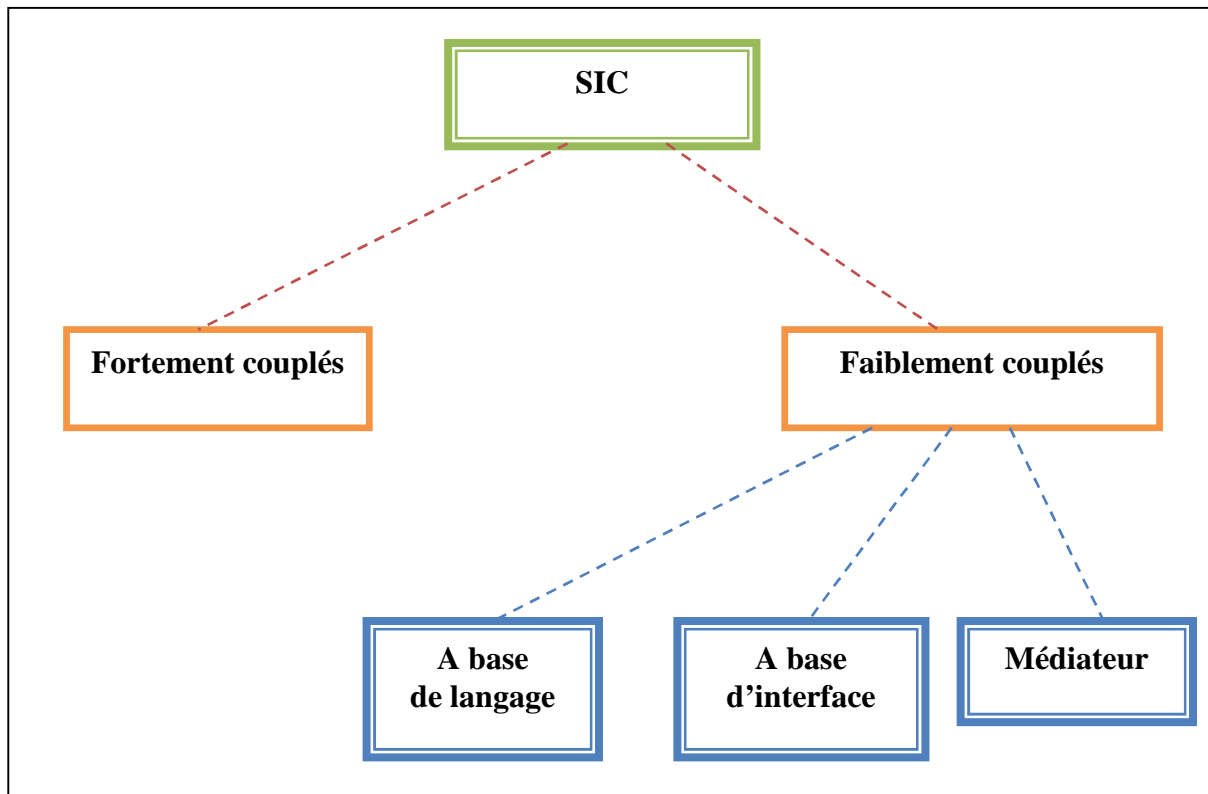


Figure 1.4. Classification des approches de SIC [54]

1.3.1. Les SIC fortement couplés

Dans ce type de systèmes, Un administrateur global qui s'occupe de l'ensemble de la fédération est responsable de la résolution de conflits de données. Ce dernier fournit à l'ensemble des utilisateurs un schéma global intégré qui combine les informations accessibles dans les différentes sources de données. L'utilisateur formule ses besoins par rapport à ce schéma global. Ainsi, ces systèmes offrent à l'utilisateur une transparence au niveau des conflits et de la distribution des données. En revanche, ils présentent deux inconvénients principaux : (i) l'extensibilité du système est couteuse et nécessite une réintégration des schémas à chaque évolution de la fédération, (ii) l'intégration de tous les schémas est une

tâche difficile et nécessite de la part de l'administrateur une connaissance de la sémantique des schémas locaux

1.3.2. Les SIC faiblement couplés

Ce type de SIC est caractérisé par l'absence de schéma global intégré. Par contre, il met à la disposition de l'utilisateur des outils qui facilitent la combinaison des sources de données. Nous distinguons trois solutions pour les systèmes faiblement couplés :

- **Les systèmes à base de langage** : reposant sur l'utilisation d'un langage de type SQL étendu aux capacités d'accès simultanés à plusieurs bases de données et sur l'absence totale de définition de schéma global. Deux inconvénients principaux peuvent être cités pour les systèmes à base de langage multi-base : la localisation des données et la résolution des conflits de données restent à la charge de l'utilisateur. En revanche, ces systèmes prennent mieux en compte l'évolution de la fédération que les systèmes fortement couplés.
- **Les systèmes à base d'interface** : ces systèmes reposent sur les bibliothèques de fonctions permettant la connexion et l'accès aux sources de données. Ils exploitent les architectures d'objets distribués construites sur le concept de bus logiciel de communication. Ce concept permet un échange de données transparent entre applications. Comme dans les systèmes à base de langage, le principal inconvénient de cette solution est la localisation des données qui reste à la charge du programmeur. Un des avantages des systèmes à base d'interface est que l'utilisation des interfaces font des SIs de vrai systèmes ouverts capables de communiquer avec les autres sources d'information (base de connaissances, base de données Web, SIs géographiques, etc.).
- **Les systèmes de médiation** : ces systèmes sont basés sur l'utilisation de composants évolués qui prennent en charge la résolution des conflits, la localisation des données et le traitement des requêtes globales. Les systèmes de médiation sont bâtis sur des architectures qui utilisent deux composants spécialisés : les wrappers et les médiateurs. Le wrapper est un composant logiciel fournissant un ensemble de services d'accès uniformes à différents types de données. Il joue le rôle d'interface entre les médiateurs et les sources de données en assurant la traduction des requêtes d'un langage commun vers un langage local et la mise en forme pour le médiateur des résultats de l'exécution locale des requêtes. Le médiateur est un composant logiciel fournissant un accès transparent et

homogène à des données représentées dans plusieurs wrappers. Il est chargé de cacher la distribution des données, de résoudre les conflits entre les données de différentes sources et de traiter les requêtes multi-sources.

1.4. Apport des SMA dans la modélisation des SIC

Les SIC sont naturellement distribués, manipulant un grand flux de données et font intervenir un grand nombre de sous systèmes autonomes qui interagissent pour accomplir un objectif global. La mise en oeuvre des SIC nécessite la prise en compte de beaucoup d'aspects. Nous citons entre autres : le partage d'information, le parallélisme des tâches, l'ouverture de l'environnement, l'interopérabilité et la communication entre les différents SIs qui le composent. Ces caractéristiques des SIC exigent l'utilisation des outils adéquats pour leur modélisation. Dans cette perspective, la technologie agent définit des réponses aux différents besoins des SIC et peut représenter un bon support pour leur modélisation.

1.4.1. Concept d'agent

Ferber [18] considère l'agent comme une entité (physique ou abstraite) capable d'agir sur elle même et son environnement, disposant d'une représentation partielle de celui-ci, pouvant communiquer avec d'autres agents et dont le comportement est la conséquence de ses observations, de sa connaissance et des interactions avec les autres agents.

Tel que présenté dans le schéma de la figure 1.5, un agent contient un mécanisme de raisonnement. Ce mécanisme prend en entrées les perceptions provenant de l'environnement, grâce aux capteurs de l'agent, et produit en sorties les actions de l'agent qui viennent agir sur l'environnement. Dans la littérature, plusieurs définitions mettent en évidence d'autres propriétés clés tel que [21] :

- les aptitudes sociales : désignent la capacité d'un agent à interagir avec les autres agents de façon coopérative ou compétitive pour atteindre ses objectifs ;
- la pro-activité : désigne l'aptitude d'un agent à se fixer des buts pour atteindre ses objectifs sur sa propre initiative.

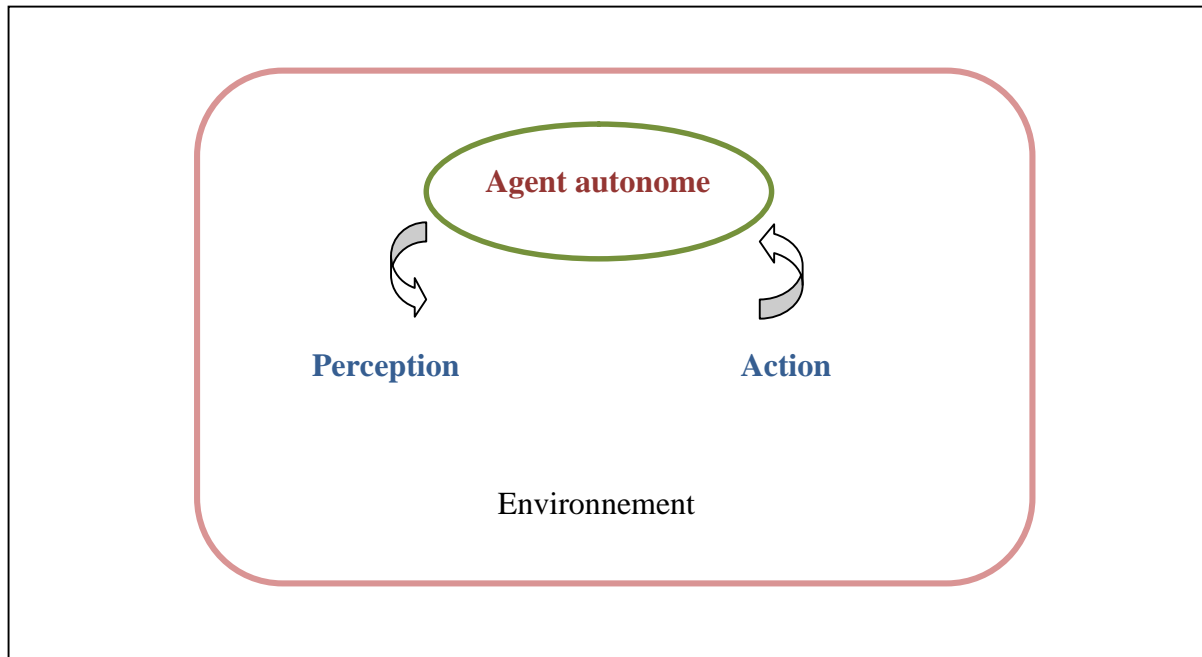


Figure 1.5. Un agent autonome dans son environnement [21].

1.4.2. Approches de conception d'agents

Plusieurs approches de conception d'agents sont apparues. Ces approches se distinguent les unes des autres du point de vue aspect interne de l'agent. Elles peuvent être regroupées en trois classes : les agents cognitifs, les agents réactifs [12] et les agents hybrides [37].

1.4.2.1. Agents réactif

Les agents réactifs sont des agents ayant la capacité de répondre uniquement à la loi Stimulus → réponse [21]. Ils n'ont pas une représentation symbolique de l'environnement ou des connaissances et ils ne possèdent pas de croyances. Un SMA constitué d'agents réactifs possède généralement un grand nombre d'agents et présente un comportement global intelligent. Ce comportement intelligent devrait émerger de l'interaction entre ces agents réactifs dans l'environnement [54]. Plusieurs chercheurs ont utilisé cette catégorie d'agents principalement dans le domaine de la robotique mobile, où une réaction instantanée de l'agent est préférée à une réponse tardive de qualité.

1.4.2.2. Agents cognitifs

Les agents cognitifs sont des agents à base de connaissance se caractérisant par leur capacité de raisonnement et par la complexité de leur structure. Chaque agent dispose d'une base de connaissances et diverses informations liées à son domaine d'expertise et à la gestion des

interactions avec son environnement [21]. Ils sont capables de planifier leur comportement, mémoriser leurs actions passées, communiquer par envoi de messages, négocier, etc. un SMA constitué d'agents cognitifs possède communément peu d'agents.

Une comparaison des deux catégories d'agents est synthétisée dans le tableau 1.1.

Systèmes d'agents cognitifs	Systèmes d'agents réactifs
Représentation explicite de l'environnement	Pas de représentation explicite
Possibilité de tenir compte de son passé	Pas de mémoire de son historique
Agents complexes	Fonctionnement stimulus/action
Petit nombre d'agents	Grand nombre d'agents

Tableau 1.1. Agents cognitifs vs agents réactifs [54]

1.4.2.3. Agents hybrides

Quelques travaux de recherche Mueller [37] ont essayé d'unifier les agents réactifs et les agents cognitifs afin de surmonter leurs limites respectives, i.e., d'une part, la difficulté de mise en œuvre de l'approche cognitive dans les environnements complexes et à forte évolution, et d'autre part, le manque de modèles formels pour l'approche réactive. L'idée est la suivante [54] : d'une façon simple, les agents hybrides réagissent à des stimulations simples en exécutant de simples routines. Cependant, le module cognitif contrôle celui qui est réactif quand ce dernier veut exécuter des actions à stimulation libre (comme le raisonnement) ou bien changer des buts à long terme.

1.4.3. Systèmes multi-agents (SMA)

« Un Système Multi-Agents est un macro-système composé d'agents autonomes qui interagissent dans un environnement commun pour réaliser une activité collective cohérente, bien qu'ils puissent chercher à atteindre des objectifs individuels parfois contradictoires » [22]. Le résultat de l'organisation des agents correspond au système multi-agent [21]. Chaque agent possède une information incomplète sur le système global car son point de vue est limité. Par conséquent, un agent ne peut pas adopter un contrôle global sur tout le système. Un agent possède son propre état qui n'est pas accessible par les autres participants du système. Le changement de cet état ne peut s'effectuer qu'après exécution de certaines règles décrivant le comportement de ce même agent. Ces règles sont elles-mêmes influencées par les données perçues de l'environnement [54]. Les SMA offrent plusieurs avantages par rapport aux

systèmes basés sur un seul agent car ces dernières facilitent l'implantation naturelle des problèmes concernant les données et le contrôle distribués et ils permettent également le passage à l'échelle puisque un SMA peut être facilement étendu en ajoutant de nouveaux agents. Ils sont aussi réutilisables grâce à la modularité des SMA qui facilite leur réutilisation, ce qui permet aux agents d'être facilement réintégrés dans de nouveaux SMA.

1.4.4. Interactions dans les SMA

Une des propriétés principales d'un agent dans un SMA est celle d'interagir avec les autres agents. Ces interactions sont généralement définies comme toute forme d'action exécutée au sein du SMA. Dans ce qui suit nous présentons quelques concepts liés à l'interaction dans les SMA.

1.4.4.1. Communication

La communication est la base de la résolution coopérative des problèmes. Une communication peut être définie comme une forme d'action locale d'un agent vers un autre [54]. Un agent qui ne communique pas est un agent isolé, muet et sourd aux informations potentiellement utiles en provenance des autres agents du système. La coopération efficace implique que les agents engagés dans une tâche commune communiquent entre eux. Dans la littérature, deux modes de communication sont connus [44]:

- La communication par envoi de messages où les agents envoient leurs messages directement et explicitement au destinataire. La seule contrainte est la connaissance de l'agent destinataire. Les systèmes fondés sur ce mode relèvent d'une distribution totale à la fois de la connaissance, des résultats et des méthodes utilisées pour la résolution du problème.

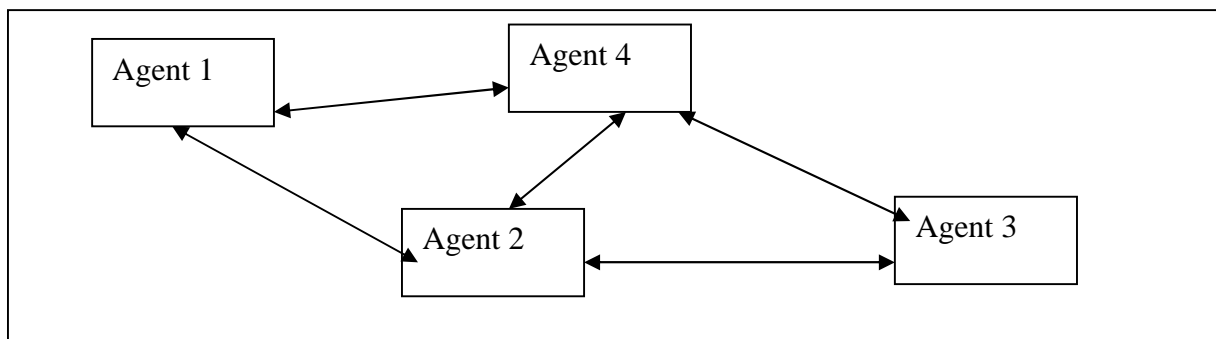


Figure 1.6. Communication par envoi de messages [44]

- La communication par partage d'information où les agents ne sont pas en liaison directe mais communiquent via une structure de données partagée dans laquelle se trouvent les connaissances relatives à la résolution qui évolue durant le processus d'exécution.

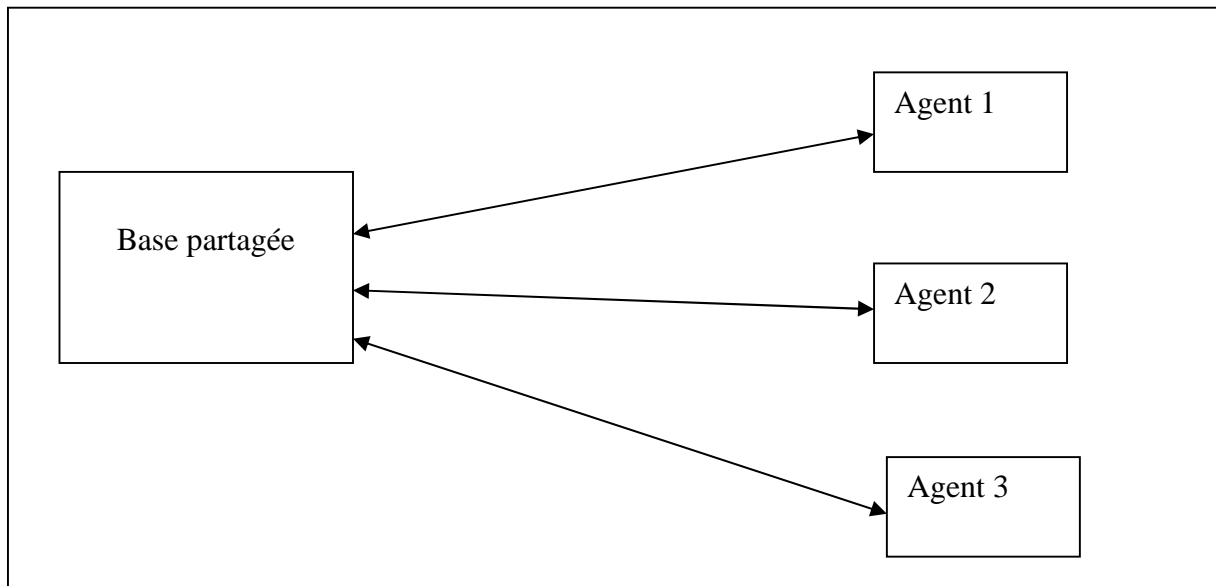


Figure 1.7. Communication par partage d'information [44]

1.4.4.2. Coopération

La coopération est une caractéristique très importante dans les SMA. Une résolution distribuée d'un problème est le résultat de l'interaction coopérative entre les différents agents. Un agent doit pouvoir mettre à jour le modèle du monde environnant, intégrer les informations émanant d'autres agents, interrompre pour aider d'autres agents et déléguer une tâche qu'il ne sait pas résoudre à un agent dont il connaît les compétences [59]. Ces caractéristiques constituent les qualités essentielles d'un agent coopératif. Dans une société d'agents un agent coopère selon les modes suivants [54] :

- *Le mode commande* : dans ce mode, un agent superviseur décompose un problème en sous-problèmes qu'il répartit entre les agents. Ces derniers les résolvent et envoient les résultats à l'agent superviseur.
- *Le mode appel d'offre* : dans ce mode, un agent superviseur décompose un problème en sous-problèmes dont il diffuse la liste aux agents. Chaque agent envoie une offre. L'agent superviseur choisit les meilleures offres et distribue les sous-problèmes, le système fonctionne ensuite en mode commande.

- *Le mode compétition* : dans ce mode, un agent superviseur décompose un problème en sous-problèmes, dont il diffuse la liste aux agents, Chaque agent résout un ou plusieurs sous problèmes et envoie les résultats à l'agent superviseur qui à son tour fait le tri.
- *La coopération par partage des tâches* : dans ce mode, le problème est distribué entre les différents agents. Les agents travaillent indépendamment les uns des autres. Chaque agent dispose de ressources et de compétences nécessaires pour accomplir la tâche qui lui a été assignée.
- *La coopération par partage des résultats* : dans ce mode les agents ne peuvent accomplir leurs tâches de manière indépendante, ils sont appelés à se transmettre des résultats partiels.

1.4.4.3. Coordination

Les études sur la communication et la coordination ont généralement été effectuées de façon isolée et ont donc produit des résultats et des outils indépendants les uns des autres. Pourtant la communication et la coordination sont complémentaires et les mécanismes les supportant doivent permettre leur intégration au sein d'un même environnement et au service des mêmes procédés [54]. La coordination désigne l'organisation du travail des agents afin d'éviter les interactions négatives et favoriser les interactions utiles. Certaines situations de coopération exigent que les agents se coordonnent pour réaliser leur but global commun. Le problème est de permettre à ces entités autonomes d'articuler leurs actions de telle sorte à rendre cohérent le comportement de l'ensemble sans l'intervention d'un utilisateur externe du SMA

1.4.4.4. Négociation

Les activités des agents dans un système distribué sont souvent interdépendantes et entraînent des conflits. Pour les résoudre, il faut considérer les points de vue des agents, les négocier et utiliser des mécanismes de décision concernant les buts ciblés. Le processus de négociation ne consiste pas forcément à trouver un compromis mais peut s'étendre à la modification des croyances d'autres agents pour faire prévaloir un point de vue [44]. Pour mener à bien ce type de processus, il est nécessaire de suivre un protocole qui facilite la convergence vers une solution.

1.5. Quelques architectures de SIC basées agent

La technologie à base d'agent est devenue un paradigme de conception prometteur et a un appel en croissance à une variété d'applications de systèmes distribués dans plusieurs domaines. Dans la littérature il y a plusieurs travaux qui ont proposé un système multi-agents basé sur une architecture hétérogène et distribuée afin d'assurer la coopération entre plusieurs composants ou systèmes autonomes. Dans ce qui suit nous présentons des systèmes multi-agents qui abordent le problème de la coopération des SIs.

1.5.1. Le système DAARACHE

Dans le système D'AArCHE [54] les SIC sont considérés comme un ensemble d'entités distribuées possédant des spécificités indispensables liées au caractère potentiellement hétérogène des sources d'information. Le système DAARACHE est développé pour la définition, l'administration et la mise en œuvre des SIC. Ce système repose sur une architecture générique dans le sens où elle est indépendante du domaine d'application. Le système DAARACHE est un système multi-agents implanté avec la combinaison de XML et de la technologie CORBA. Cette combinaison garantit l'interopérabilité des SIs composants indépendamment de leurs modèles de données et/ou leurs plates-formes. En plus, les contrôles et les connaissances qui sont distribués entre les agents assurent la flexibilité et supportent la réutilisation.

1.5.2. Le système InfoSleuth

InfoSleuth [38] est un système multi-agents pour la recherche coopérative d'informations dans des bases de données distribuées. L'architecture d'InfoSleuth se compose de trois types d'agents principaux: agents d'utilisateur, agents ressources et agents de noyau. Les agents utilisateurs assistent les utilisateurs pour formuler leurs demandes et transforment aussi les résultats en forme accessible à l'utilisateur. Les agents ressources facilitent l'accès aux bases de données et aux dépôts d'informations. Enfin, les agents de noyau recueillent l'information pour répondre aux demandes des utilisateurs.

1.5.3. Le système NetSA

NetSA [15] fait aussi partie des SIC il s'agit d'un système multi-agents pour la recherche d'informations dans des sources de données hétérogènes et distribuées (Intranet, Extranet ou Internet). Le but principal visé par NetSA est d'avoir un système multi-agents flexible et générique. Le système multi-agents NetSA comporte différents types d'agents dans des

concentrations variables. Ces agents appartiennent tous à une seule unité. De ce fait, chaque agent est responsable uniquement d'une partie de la tâche globale. C'est la coopération entre ces agents qui amène à la réussite de la tâche finale. L'architecture du Netsa est composée de trois couches : la couche de communication avec l'utilisateur, la couche de traitement de l'information et la couche d'interrogation et d'extraction d'informations.

1.6.Conclusion

Dans ce chapitre, nous avons essayé de faire un survol sur notre contexte de travail. Nous avons introduit les concepts qui nous sont nécessaires pour construire notre architecture. Dans un premier lieu, nous avons présenté les SIC en précisant leurs caractéristiques et leur apport sur le développement des entreprises. Puis, une classification de ces derniers est présentée. Enfin, pour montrer la faisabilité de ces systèmes, nous avons introduit le principal support de leur modélisation celui des SMA.

Le chapitre suivant est consacré à la présentation de la notion de sémantique et de l'interopérabilité sémantique qui joue un rôle importante dans la coopération des SIs.

Notion de sémantique et d'interopérabilité sémantique dans les systèmes d'information coopératifs

2.1. Introduction

La croissance exponentielle des informations et des ressources échangées entre les différents composants d'un SIC augmente le taux d'hétérogénéité des informations et rend leur compréhension et leur analyse très difficiles. Un problème crucial découlant de cette hétérogénéité concerne la préservation du sens de l'information échangée. La prise en compte de cette sémantique permet aux composants des systèmes coopératifs de combiner les informations reçues avec des informations locales et de traiter l'ensemble de manière cohérente. C'est ce que l'on appelle l'*interopérabilité sémantique*. Pour assurer cette interopérabilité, l'information échangée entre systèmes doit d'abord être décrite dans une structure formelle permettant de préserver sa sémantique. Ce grand défi est omniprésent dans le domaine de l'ingénierie des connaissances, où des méthodologies et des techniques sont proposées pour percevoir, identifier, analyser, organiser et partager des connaissances entre différentes organisations. Parmi ces techniques, les *ontologies* connaissent un essor très important depuis une dizaine d'années et apparaissent comme le moyen le plus efficace pour la gestion de la sémantique [34].

Dans ce qui suit, nous présenterons la notion de sémantique et d'ontologie. Ensuite, nous exposerons de façon succincte des techniques pour assurer l'interopérabilité sémantique basées sur les ontologies. Puis, nous présenterons quelques architectures qui proposent des solutions aux hétérogénéités sémantiques dans le cadre de la coopération de sources d'information hétérogènes.

2.2. Notion de sémantique

Dans cette section nous abordons la notion de la sémantique, sa typologie, sa classification et les outils utilisés pour sa représentation.

2.2.1. Définition

D'après le dictionnaire(LAROUSSE), La sémantique est l'étude du sens des mots elle constitue une branche de la linguistique. Dans le domaine des systèmes d'information, la sémantique se réfère plus précisément au sens des différents éléments d'un système d'information qui peuvent être des données, des fonctions, voire des processus [28]. La sémantique peut être différée d'un système à un autre, en fonction du contexte d'utilisation. Ces hétérogénéités de type sémantique sont à l'origine des conflits entre systèmes ce qui contraint leur interopération.

2.2.2. Typologie des hétérogénéités sémantiques

Kavouras [30] propose une typologie d'hétérogénéités sémantiques qui prend en considération les termes et leurs définitions associés à des entités de la réalité :

- *Equivalence* : deux entités sont équivalentes si elles ont le même terme et la même définition.
- *Disjonction* : deux entités sont disjointes si elles ont des termes différents et des définitions différentes.
- *Synonymie* : cas où deux entités sont représentées par deux termes différents et une même définition.
- *Homonymie* : cas où deux entités sont représentées par le même terme et deux définitions différentes.
- *Complétude* : cas où les deux entités sont représentées par le même terme, et la définition de l'une est incluse dans l'autre.
- *Spécialisation* : cas où les deux entités sont représentées par deux termes différents, et la définition de l'une est incluse dans l'autre.
- *Chevauchement* : Cas où les définitions des deux entités se recouvrent.

2.2.3. Classification de la sémantique

Uschold et Gruninger [48] proposent une classification de la sémantique qui permet de distinguer les différents types de sémantique qui peuvent exister dans le contexte du Web comme suit :

Chapitre2

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

2.2.4. Représentation de la sémantique

Pour modéliser la sémantique nous pouvons utiliser l'une des deux approches suivante [28] [46]:

- L'approche procédurale qui utilise des procédures ou des règles pour représenter la sémantique.
- L'approche déclarative qui se base sur la modélisation des faits. Cette dernière est meilleure que la première du fait qu'elle présente des avantages tels que standardisation, réutilisation, inférence et flexibilité.

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

2.3. Notion d'ontologie

Dans cette section nous présentons une description détaillée des différents concepts liés aux ontologies.

2.3.1. Définitions

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

- « une ontologie fournit les moyens de décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances », Gomez-Perez [25].

Chapitre2

- « Une ontologie est un ensemble de termes structurés de façon hiérarchique, conçu afin de décrire un domaine et qui peut servir de charpente à une base de connaissances », Swartout [47].
- « Une ontologie est un modèle (au sens large, composé d'objets et de relations) réutilisable et partageable d'un domaine particulier spécifié pour créer un langage commun (éventuellement basé sur des standards basés sur XML) afin de faciliter l'échange d'informations et le partage de fonctionnalités entre les personnes et les systèmes d'information », Izza [28].

2.3.2. Constituants d'une ontologie

Une ontologie est structurée de plusieurs composants qui sont [34] [26] :

- *Les concepts* (termes ou classes) de l'ontologie, constituent les objets de base manipulés par les ontologies. Ils correspondent aux abstractions pertinentes du domaine du problème, retenues en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie.
- *Les relations* traduisent les interactions existant entre les concepts. Ces relations sont formellement définies comme tout sous-ensemble d'un produit cartésien de n ensembles, c'est-à-dire $R : C_1 \times C_2 \times \dots \times C_n$ et incluent 1) la relation de spécialisation (subsumption), 2) la relation de composition (méronymie), 3) la relation d'instanciation, etc. Ces relations nous permettent de capturer, la structuration ainsi que l'interaction entre les concepts, ce qui permet de représenter une grande partie de la sémantique de l'ontologie.
- *Les fonctions* sont des cas particuliers de relations dans les quelles le n ième élément (extrant) de la relation est défini de manière unique à partir des $n-1$ éléments précédents (intrants). Formellement, les fonctions sont définies ainsi : $F : C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$. Comme exemple de fonctions binaires, nous pouvons citer la fonction *mère de*.
- *Les axiomes* permettent de modéliser des assertions toujours vraies, à propos des abstractions du domaine traduites par l'ontologie. Ils permettent de combiner des concepts, des relations et des fonctions pour définir des règles d'inférences et qui peuvent intervenir, par exemple, dans la déduction, la définition des concepts et des relations, ou alors pour restreindre les valeurs des propriétés ou les arguments d'une relation.
- *Les instances ou individus* constituent la définition extensionnelle de l'ontologie. Ils représentent des éléments singuliers véhiculant les connaissances (statiques, factuelles) à propos du problème.

2.3.3. Rôle des ontologies

Une ontologie permet la mise en œuvre de différentes techniques liées aux traitements que l'on peut appliquer à l'information dans des systèmes informatiques. Nous énumérons par la suite, les divers rôles que peuvent jouer les ontologies selon [55] [49].

2.3.3.1. La communication

Les ontologies peuvent intervenir dans la communication entre humains. Dans ce cas, elles servent par exemple, à créer au sein d'un groupe ou d'une entreprise un *vocabulaire* standardisé. Pour de tels besoins, on est plutôt dans le cadre d'une ontologie informelle. Dans le cas de la communication entre êtres humains et ordinateurs, l'ontologie est formelle et sert en général une à tâche précise dans le SBC ou le système d'information.

2.3.3.2. L'aide à la spécification de systèmes

L'ontologie peut servir divers aspects du développement des systèmes d'informations. Premièrement, elle peut assister le processus de construction de spécification de système. L'usage d'ontologie rend les documents du processus plus compréhensibles, évite l'ambiguïté dans la spécification. En outre, une représentation formelle d'ontologie permet un traitement automatique du développement. Elle soutient également l'automatisation du processus de vérification de fiabilité de systèmes.

2.3.3.3. L'interopérabilité

Les ontologies peuvent être utilisées Pour résoudre le problème de compréhension des informations échangées entre des systèmes coopérant à la réalisation d'une tâche globale.

2.3.3.4. L'indexation et la recherche d'information

Plus récemment, les travaux autour du Web sémantique ont réactivé la problématique et l'utilisation des ontologies pour construire les index conceptuels décrivant les ressources sur le Web.

L'indexation de ressources à partir d'une ontologie présente les avantages suivants :

- Aider l'utilisateur à formuler sa requête. En présentant l'ontologie à l'utilisateur. Il est possible de le guider dans le choix des termes de sa requête.
- Faciliter la recherche d'information au sein de collections hétérogènes en indexant tous types de ressource à partir des mêmes concepts.

2.3.4. Dimensions de classification d'ontologies

Les ontologies peuvent être classifiées selon plusieurs critères, afin de déterminer une typologie d'ontologies. Parmi celles-ci, nous en examinerons deux [49] [43] : 1) *Objet de conceptualisation* ; 2) *Niveau de formalisme de représentation*.

2.3.4.1. Classification selon l'objet de conceptualisation

Les ontologies classifiées selon leur objet de conceptualisation (*le but de leur utilisation*) sont les suivantes [36] :

- *Ontologie de représentation des connaissances* : Ce type d'ontologies regroupe les concepts (primitives de représentation) impliqués dans la formalisation des connaissances selon des paradigmes de représentation des connaissances. Un exemple de ce type d'ontologie est la *Frame-ontology* qui intègre les primitives de représentation des langages à base de *frames* : classes, instances, facettes, propriétés/*slots*, relations, restrictions, valeurs permises, etc.
- *Ontologie supérieure ou de Haut niveau* : Cette ontologie est une ontologie générale. Son sujet est l'étude des catégories des choses qui existent dans le monde, soit les concepts de haute abstraction tels que: les événements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés.
- *Ontologie de Tâches* : Ce type d'ontologies est utilisé pour conceptualiser des tâches spécifiques dans les systèmes, telles que les tâches de diagnostic, de planification, de conception, de configuration, soit tout ce qui concerne la résolution de problèmes. Elle régit un ensemble de vocabulaires et de concepts qui décrit une structure de résolution des problèmes inhérente aux tâches et indépendante du domaine.
- *Ontologie du Domain* : Cette ontologie régit un ensemble de vocabulaires et de concepts qui décrit un domaine d'application ou monde cible. Elle permet de créer des modèles d'objets du monde cible. L'ontologie du domaine est une méta-description d'une représentation des connaissances, c'est-à-dire une sorte de méta-modèle de connaissance dont les concepts et propriétés sont de type déclaratif [50]. La plupart des ontologies existantes sont des ontologies du domaine.
- Les ontologies d'application : ce sont les ontologies les plus spécifiques. Elles permettent de décrire des concepts dépendants à la fois d'un domaine et d'une tâche. Ces ontologies

Chapitre2

Notion de sémantique et d'interopérabilité sémantique dans les systèmes d'information coopératifs

décrivent la sémantique des informations et des services manipulés par une ou un ensemble d'applications sur un même domaine.

2.3.4.2. Classification selon le niveau du formalisme de représentation

Selon le niveau du formalisme de représentation du langage utilisé pour décrire l'ontologie, [49] proposent une classification comprenant quatre catégories.

- *Ontologies informelles* : Qui sont des ontologies opérationnalisées dans un langage naturel, ce qui correspond à ce qui est communément appelé la sémantique ouverte.
- *Ontologies semi-informelles* : Ce sont des ontologies qui sont décrites à l'aide d'un langage naturel structuré et limité.
- *Ontologie semi-formelles* : Ce sont des ontologies qui utilisent un langage artificiel défini formellement.
- *Ontologies formelles* : Ce sont des ontologies qui se basent sur l'utilisation d'un langage artificiel contenant une sémantique formelle telles que les logiques de description.

2.3.5. Quelques formalismes de représentation des connaissances

Depuis la naissance de l'intelligence artificielle, plusieurs formalismes de représentation de connaissances ont été développés. Parmi ces formalismes développés au niveau conceptuel pour la modélisation d'ontologie, trois grands modèles sont distingués [19] : les langages de frames, les graphes conceptuels et les logiques de descriptions.

- Le langage des frames, introduits par Minsky en 1975, ont pour but de représenter les connaissances dans un schéma incluant des frames, des slots permettant de décrire ces frames, et des relations d'héritage permettant d'hierarchiser ces frames.
- Les graphes conceptuels (GC), introduits par Sowa en 1984, permettent précisément de gérer la connaissance avec des graphes bipartites finis connexes décrivant comment les connaissances d'un domaine sont organisées. Les deux composants de ces graphes en sont les concepts et les relations. Chaque nœud relation doit relier des concepts par des arcs. Chaque concept peut désigner des entités, des propriétés, des états, voire des événements.
- Les logiques de description (également appelées logiques terminologiques, ou encore langages basés sur les termes), introduit au début des années 90, constituent l'aboutissement des longs travaux de recherche. Ces logiques manipulent trois types d'entités qui sont les individus (entités concrètes de l'univers), les concepts (entités génériques) et les rôles (relations binaires entre individus), et offrent deux types de

Chapitre2

langages pour pouvoir représenter les connaissances: les langages assertionnels et les langages terminologiques.

2.3.6. Langages de représentation des ontologies

Avec l'explosion de l'Internet, plusieurs langages basés Web ou langages d'annotation d'ontologies ont vu le jour pour gérer et exploiter toute la sémantique du Web. La figure suivante (figure 2.1.) permet de résumer les principaux langages de cette catégorie sous forme d'une pile de langages construite sur la base du langage XML (voire HTML) et qui permet de procurer une syntaxe aux documents structurés à travers le Web. Le langage OWL [57] renforce le vocabulaire de description des propriétés et des classes : entre autres, les relations entre les classes (par exemple, la disjonction), la cardinalité (par exemple, un seul exactement), l'égalité, le typage enrichi des propriétés, les caractéristiques des propriétés (par exemple, la symétrie) et les classes énumérées.

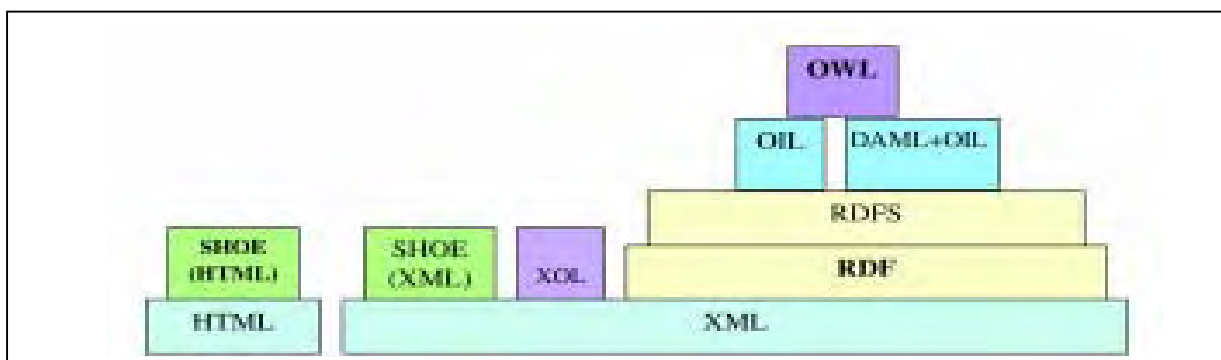


Figure 2.1. Pile des langages d'annotation d'ontologies [14]

SHOE : Le langage SHOE (Simple HTML Ontology Extensions) a été créé en 1996 comme une extension de HTML à l'université de Maryland. Il permet d'insérer des ontologies dans des documents HTML grâce à de nouvelles balises. Il permet de représenter des concepts, des taxonomies, des relations et aussi des règles qui, permettent d'inférer de nouvelles connaissances. En 1998, et avec l'avènement du XML adopté comme un standard pour les échanges d'informations sur le Web par le W3C, SHOE a été modifié pour supporter des documents décrits en XML [28].

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

Chapitre2

Notion de sémantique et d'interopérabilité sémantique dans les systèmes d'information coopératifs

RDF et RDFS : RDF (Resource Description Framework) a été développé par W3C comme un langage basé sur les réseaux sémantiques pour décrire des ressources ou des relations entre ressources, en leur affectant des métadonnées. Le modèle RDF définit trois types d'objets :

- des ressources : les ressources sont tous les objets décrits par RDF ;
- des propriétés : une propriété est un attribut, un aspect, une caractéristique qui s'applique à une ressource ;
- des valeurs : les valeurs en question sont celles particulières que prennent les propriétés. Ces trois types d'objets peuvent être mis en relation par des assertions, c'est à dire des triplets (ressource, propriété, valeur), ou encore (sujet, prédicat, objet). Une description RDF est une suite d'assertions. RDFS (RDF Schema) offre les moyens de définir un modèle (ou bien encore un schéma) de méta données qui permet de :
 - donner du sens aux propriétés associées à une ressource ;
 - formuler des contraintes sur les valeurs associées à une propriété afin de lui assurer une signification.

DAML+OIL : Créer dans le cadre du projet DARPA le langage DAML-ONT (DARPA Agent Markup Language) intervient pour permettre aux agents de partager de la sémantique. DAML est associé à OIL (Ontology Inference Layer), qui est un autre langage de description d'ontologies. Le couple DAML+OIL repose sur RDF. OIL est un langage de description et d'inférence sur les ontologies, basé sur RDF. Il prend appui sur les logiques de description. [61].

OWL : OWL (Ontology Web Language) est un vocabulaire XML basé sur RDF. Il constitue une recommandation du W3C. Il permet de spécifier ce qui peut être compris. Dans le cadre du Web, une ontologie est considérée comme une représentation des termes (information) d'un vocabulaire et ses relations aux autres termes, OWL est utile pour aider les machines à gérer ces informations plutôt que pour les rendre lisibles à l'œil humain. Un document OWL est composé de triplets RDF qui peuvent être écrits dans plusieurs syntaxes RDF [57].

2.3.7. Approches d'utilisation des ontologies

Il existe principalement trois approches d'utilisation des ontologies [53] qui sont : L'approche mono-ontologie, l'approche multi-ontologie et l'approche hybride.

Chapitre2

Notion de sémantique et d'interopérabilité sémantique dans les systèmes d'information coopératifs

2.3.7.1. Approche mono-ontologie

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

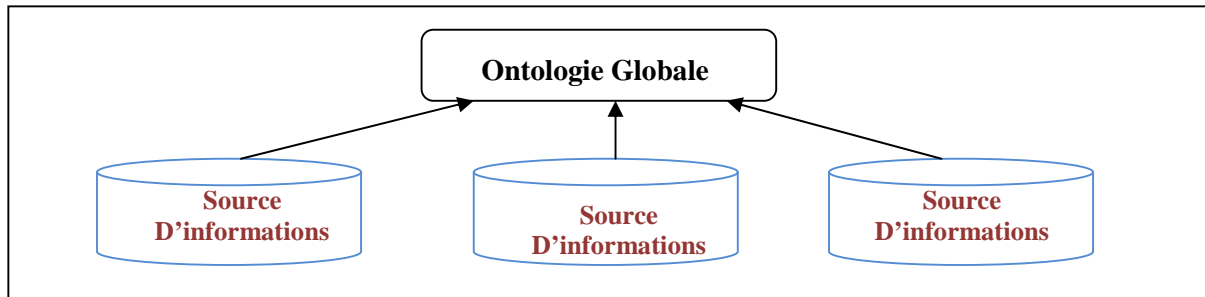


Figure 2.2. Approche mono-ontologie [53]

2.3.7.2. Approche multi-ontologies :

L'approche multi-ontologies (multi-ontology approach) permet d'associer une ontologie locale à chaque Système d'Information. Des mappings inter-ontologiques sont alors nécessaires afin d'établir une interprétation commune des données. Comme exemple d'implémentation, le projet OBSERVER [35] s'inscrit dans cette approche. Il associe à chaque source de données une ontologie locale. L'intérêt de cette approche est le fait que les sources peuvent être développées de façon indépendante, mais l'absence d'un vocabulaire commun conduit généralement à une difficulté extrême pour comparer différentes ontologies sources. Pour pallier cette difficulté, un formalisme de représentation supplémentaire définissant le mapping inter-ontologie est utilisé. Ce dernier identifie sémantiquement les termes correspondants des différentes ontologies source

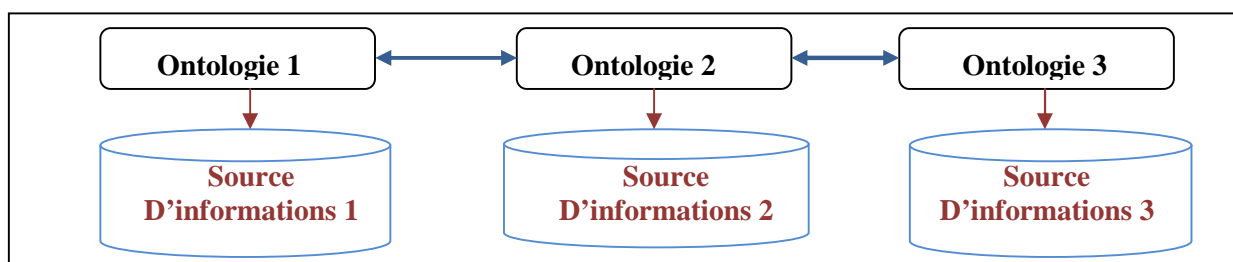


Figure 2.3. Approche multi-ontologies [53]

2.3.7.3. Approche hybride :

Les approches hybrides combinent les deux approches précédentes en utilisant à la fois des ontologies locales et une ontologie partagée (parfois un vocabulaire commun) afin de simplifier les mappings ontologiques. L'ontologie partagée fournit un vocabulaire commun et global ce qui rend les ontologies locales comparables. Cette ontologie partagée comprend les termes de base ou primitives d'un domaine. Afin de construire des termes complexes, ces termes de base sont combinés par des opérateurs, et les termes peuvent être comparés plus facilement que dans une approche multi-ontologies.

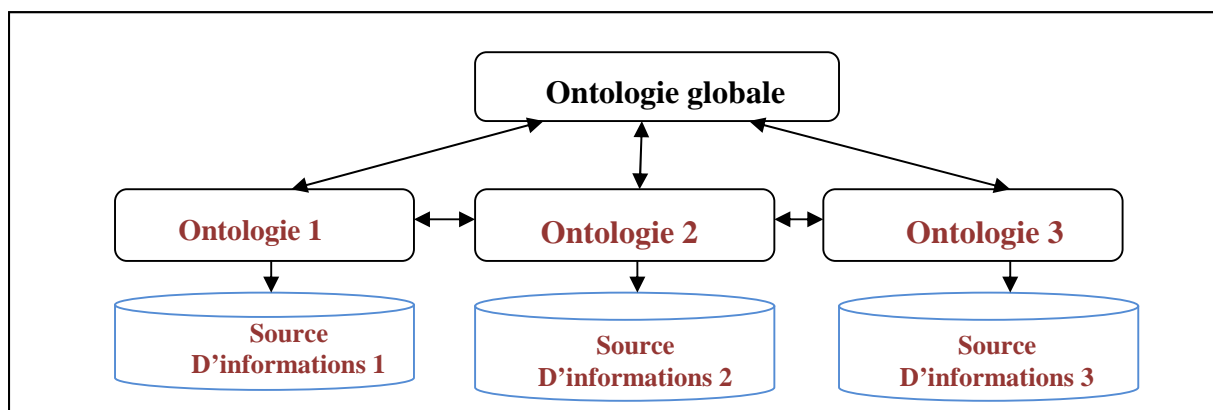


Figure 2.4. Approche hybride [53]

D'après la description des trois approches, Il est clair que l'approche hybride est plus avantageuse du fait que l'ajout de nouvelles sources est simple, et ne nécessite pas de modifications. Aussi, elle supporte l'acquisition et l'évolution des ontologies, et l'utilisation d'un vocabulaire partagé rend les ontologies sources comparables.

2.4. Notion d'interopérabilité

Dans cette section nous abordons la notion de l'interopérabilité et nous présentons particulièrement les outils et les méthodes utilisés pour assurer l'interopérabilité au niveau sémantique qui est au cœur de notre problématique.

2.4.1. Définitions

Le développement rapide des systèmes informatiques a provoqué une grande disparité de types de matériels et de logiciels. Le dialogue entre les systèmes est devenu difficile à cause de cette disparité entre les technologies. Cet aspect rend nécessaire la prise en compte d'un concept appelé interopérabilité. On dit que plusieurs systèmes (identiques ou différents) interopèrent s'ils peuvent communiquer sans aucune ambiguïté. La notion d'interopérabilité

Chapitre2

Notion de sémantique et d'interopérabilité sémantique dans les systèmes d'information coopératifs

est également présente dans les bases de données. Elle consiste à pouvoir acquérir et intégrer des informations, des données et des services issus de bases de données hétérogènes [34]. L'interopérabilité dans le cadre des SIC est indispensable pour masquer les hétérogénéités qui peuvent exister entre les systèmes coopérants. L'EIF [56] (European Interoperability Framework) a proposé trois niveaux d'interopérabilité.

- **Interopérabilité technique**, relative à la mise en oeuvre des technologies de l'information et de la communication concernant les normes pour présenter, stocker, échanger, traiter et communiquer les données au moyen de matériel informatique.
- **Interopérabilité sémantique**, de niveau plus conceptuel, qui doit assurer que les informations échangées sont compréhensibles du point de vue de leur signification et de leur interprétation par les applications qui les utilisent mais ayant été développées pour des objectifs différents. L'interopérabilité sémantique permet à des systèmes de combiner l'information reçue de la part d'autres sources d'information et de la traiter tout en conservant le sens.
- **Interopérabilité organisationnelle**, définissant les responsabilités, les autorisations, les confiances, les aspects légaux, les propriétés intellectuelles et les structures organisationnelles nécessaires à l'acceptation des échanges d'information entre applications par les différents acteurs. Ce niveau d'interopérabilité est particulièrement mis en avant dans le cadre de l'interopérabilité de l'administration électronique et des gouvernements [32].

Nous nous intéressons ici à l'interopérabilité sémantique. Elle représente actuellement un défi dans plusieurs domaines de recherche, en particulier en intelligence artificielle à travers la notion d'ontologie. F.Vernadat explique l'interopérabilité sémantique en une phrase : "*To exchange services and data among systems that make sense (common "meaning")*" [51]. Par conséquent, l'interopérabilité sémantique consiste à donner un sens aux informations échangées et à garantir que ce sens est distribué dans tous les systèmes entre lesquels des échanges doivent être mis en oeuvre [34]. La prise en compte de cette sémantique permet à ces systèmes de combiner les informations reçues avec des informations locales et de les traiter d'une manière appropriée.

2.4.2. Techniques pour l'interopérabilité sémantique basées sur les ontologies

La notion d'interopérabilité sémantique pose un problème de compréhension des informations échangées entre des systèmes coopérant à la réalisation d'une tâche globale. Pour remédier à ce problème, les chercheurs se sont orientés vers deux solutions complémentaires [34] :

- Les informations échangées doivent être structurées de manière à faciliter leur compréhension. Cette structuration amène à utiliser des ontologies qui décrivent, dans un cadre formel, les connaissances d'un domaine.
- Lorsqu'un système reçoit des informations, la compréhension de celles-ci nécessite la mise en correspondances avec d'autres informations ou concepts connus au sein du système, afin de les exploiter. Cette mise en correspondance a donné lieu à la proposition et au développement de nouvelles techniques.

Pour réaliser l'interopérabilité sémantique plusieurs techniques ont été proposées dans la littérature. Elles sont souvent utilisées pour permettre le partage des données entre des bases de connaissance hétérogènes et pour la réutilisation des informations de ces bases. Dans l'ouvrage "Semantic Web Technologies" [7], l'auteur distingue trois catégories principales qui sont : *L'alignement d'ontologies*, *La fusion d'ontologies*, *Le mapping d'ontologies*. Ces trois techniques seront présentées dans les sections suivantes.

2.4.2.1. Alignement d'ontologies

Cette technique est considérée comme le processus qui permet d'amener deux ou plusieurs ontologies hétérogènes à un "accord mutuel" en les rendant ainsi consistantes et mutuellement cohérentes [28]. L'alignement d'ontologies nécessite la transformation des ontologies impliquées en procédant à l'élimination des entités non pertinentes et au rajout des entités manquantes.

2.4.2.2. Fusion d'ontologies

Cette technique est dédiée à la construction d'une nouvelle ontologie à partir de deux ou plusieurs ontologies existantes. La nouvelle ontologie est obtenue via la combinaison des ontologies initiales et quelques concepts supplémentaires nécessaires pour cette combinaison [31]. La fusion d'ontologie est utilisée généralement dans le cadre d'intégration de données, pour obtenir une ontologie globale qui sert d'interface pour un certain nombre d'ontologies locales [10]. La principale limite de cette architecture tient au fait que l'on a besoin de

Chapitre2

développer, et encore plus important, on a besoin de s'entendre sur une ontologie globale, chose qui n'est toujours pas si évidente dans les environnements fortement évolutifs.

2.4.2.3. Mapping d'ontologies

Le mapping d'ontologies repose sur la définition de relations de correspondance entre les entités de deux ontologies qui présentent une similitude [28] [39]. Contrairement à l'alignement d'ontologie le mapping ne modifie pas les ontologies impliquées mais il produit en sortie un ensemble de correspondances. La majorité des architectures dédiées à la coopération des sources d'information sont basées sur la technique de mapping entre ontologies pour assurer l'interopérabilité sémantique. Pour cela nous présentons dans ce qui suit une description du processus et des outils utilisés pour réaliser les mappings entre les ontologies.

2.5. Description générale du processus et des méthodes de Mapping

Dans cette section nous présentons les étapes du processus de mapping et les différentes méthodes qui facilitent la découverte des mappings.

2.5.1. Processus de Mapping

Le processus de mapping comprend généralement les étapes suivantes [60] :

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

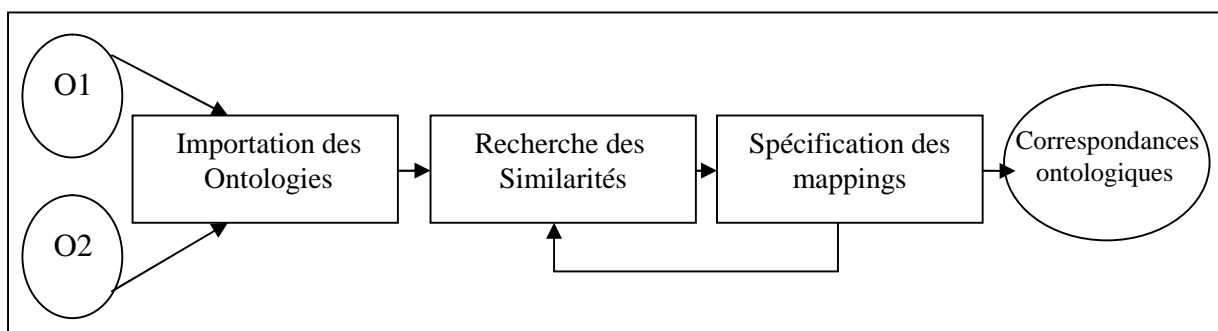


Figure 2.5. Processus de mappings d'ontologies [8]

2.5.2. Méthodes de découverte des mappings

Il existe dans la littérature une multitude de méthodes de découverte de mappings. [63] propose une classification des principales méthodes. Ces méthodes sont classifiées selon deux perspectives : le type de la technique de mapping et le type des objets manipulés. Nous nous intéressons par la seconde classification qui permet de décomposer les méthodes de mappings en : méthodes structurelles, terminologiques, sémantiques, extensionnelles [28].

- *Les méthodes structurelles* sont basées sur la comparaison des structures des concepts à mapper. En fonction de la nature interne ou externe des structures à comparer, on peut alors distinguer :
 - des méthodes structurelles internes : qui utilise la comparaison de structure interne d'un concept (par exemple la comparaison des attributs, des noms, et des types des attributs);
 - des méthodes structurelles externes : qui utilise la comparaison externe en mettant par exemple en jeu la position des concepts dans leur hiérarchie, le voisinage des concepts.
- *Les méthodes terminologiques* reposent principalement sur des comparaisons de texte (chaînes de caractères). Elles peuvent s'appliquer aux noms, aux commentaires, aux propriétés des concepts afin de découvrir les concepts similaires. Parmi cette catégorie, on distingue des méthodes basées sur des textes et des méthodes basées sur des langages.
- *Les méthodes sémantiques* sont des méthodes qui se basent sur des modèles théoriques et se focalisent sur des techniques déductives exploitant très souvent la logique de description (test de subsumption), SAT SATisfiabilité propositionnelle (calcul propositionnel) ou encore SAT modale (calcul des prédicats).
- *Les méthodes extensionnelles* se focalisent sur la comparaison des instances des concepts des ontologies à mapper. En fonction de l'intersection des extensions de deux concepts, on peut distinguer :
 - des méthodes de comparaison d'extensions communes : qui considèrent que deux concepts sont similaires si leur intersection se réduit à l'un des concepts.
 - des méthodes de comparaison d'extensions basée sur la similarité : qui reposent sur des techniques de calcul de similarité entre les instances des deux concepts à mapper;
 - des méthodes de comparaison basée sur l'appariement : qui considèrent que les éléments à comparer sont ceux qui présentent des similarités.

2.6. Quelques travaux portant sur la gestion de la sémantique dans les SIC

L'objectif des premières propositions liées à la coopération de sources d'information a été de traiter les différences techniques, structurelles et syntaxiques entre ces sources. La plupart de ces propositions ne parviennent pas à résoudre les conflits sémantiques. Les systèmes les plus récents sont des systèmes plus évolués dont l'objectif s'oriente vers la résolution des conflits sémantiques grâce à une combinaison de technologies : médiation de l'information, technologie agent et une représentation de la sémantique à l'aide des ontologies. Dans cette section nous exposons quelques travaux qui touchent de près notre recherche ou plus précisément nous présentons des architectures qui proposent des solutions pour le problème d'hétérogénéité sémantique dans le cadre de coopération des sources d'information distribuées et hétérogènes.

2.6.1. L'architecture OWSCIS

Dans cette section nous présentons l'architecture OWSCIS proposée par R.Ghawi [20].

2.6.1.1. Présentation de l'architecture

OWSCIS (Ontology and Web Service based Cooperation of Information Sources) est une architecture qui a pour objectif la mise en œuvre d'un moyen de communication entre des systèmes sources d'information et des utilisateurs sous la forme d'une coopération sémantique basée sur des ontologies. OWSCIS adopte une approche ontologique hybride, dans laquelle la sémantique des sources d'information est décrite explicitement par des ontologies locales. Une ontologie globale est utilisée comme modèle global pour les ontologies locales. L'utilisateur pose ses requêtes en utilisant les termes de cette ontologie globale. Les sources d'informations dans OWSCIS sont organisées dans un ensemble de sites. Chaque site regroupe plusieurs sources. Par conséquent, une seule ontologie locale est utilisée pour décrire la sémantique des sources d'un site. Le fonctionnement global de l'architecture OWSCIS est assuré par un ensemble de modules et de services web. Chaque composant (module ou service web) est destiné à la réalisation d'une tâche bien définie :

- **Base de connaissances** : ce module encapsule l'ontologie globale avec une boîte à outils et un répertoire de mappings. Cette base est utilisée par le service web de requêtes pour localiser les sources pertinentes au traitement de requêtes.
- **Service web de mapping** : ce service est utilisé pour la découverte des correspondances entre l'ontologie globale et les ontologies locales.

Chapitre 2

- **Service web de requêtes** : ce service est utilisé pour traiter les requêtes des utilisateurs, i.e. analyse des requêtes, décomposition des requêtes en sous-requêtes pour les fournisseurs de données adéquates et la recombinaison des résultats des traitements.
- **Service web de visualisation** : ce service est utilisé pour visualiser l'ontologie globale et les réponses finales aux utilisateurs.
- **Fournisseurs de données** : un fournisseur de données est considéré comme un wrapper qui associe les sources d'information d'un site avec leur ontologie locale. Il assure le traitement des sous-requêtes envoyées par le service web de requêtes. Un fournisseur de données contient les composants suivants :
 - Une ontologie locale qui représente un modèle global des sources d'information d'un site.
 - Les mappings entre l'ontologie locale du site et l'ontologie globale.
 - Les mappings entre l'ontologie locale et schémas des sources d'information.
 - Un ensemble d'outils de mappings utilisé pour découvrir les mappings entre l'ontologie locale du site et les schémas des sources d'information.

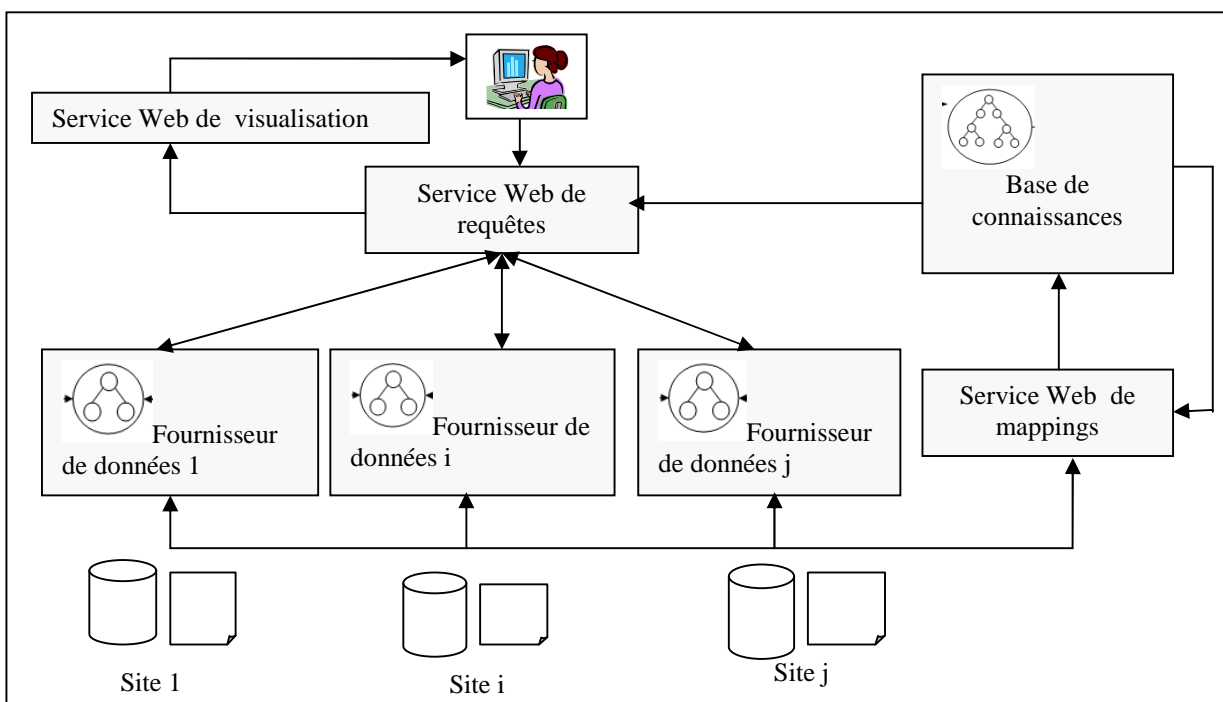


Figure 2.6. Composants de l'architecture OWSCIS [20]

2.6.1.2. Synthèse

OWSCIS propose des solutions aux problèmes liés à la coopération de sources d'information distribuées et hétérogènes. Elle présente des outils efficaces pour résoudre les conflits au niveau sémantique via une approche ontologique hybride. L'ajout d'une nouvelle source d'information à la coopération peut être réalisé facilement via la découverte des mappings entre l'ontologie globale et l'ontologie locale de la nouvelle source. OWSCIS regroupe les sources d'information dans des sites. La sémantique des sources d'information dans chaque site est présentée par une seule ontologie locale contrairement à la majorité des architectures qui proposent l'utilisation d'une ontologie locale pour chaque source d'information, ce qui augmente le nombre de mappings à établir entre les ontologies locales et l'ontologie globale. Le point faible de cette architecture réside dans le fait d'utiliser des services web spécialisés chacun dans la réalisation d'une tâche, ce qui pose les problèmes liés aux approches centralisées.

2.6.2. L'architecture ACSIS

Dans cette section nous présentons l'architecture ACSIS proposée par M.Séguaran [45].

2.6.2.1. Présentation de l'architecture

AC SIS (Agent pour la Coopération de Système d'Information Sécurisée) est une architecture conceptuelle multi-niveaux. Elle permet de résoudre chaque type de conflits : technique, syntaxique et sémantique. Le premier niveau résout les problèmes d'hétérogénéité technique par l'utilisation d'objets spécifiques et du bus Corba. Le deuxième niveau composé d'Objets Descriptifs des Données assure la résolution des conflits syntaxiques. La partie haute de l'architecture est responsable de la résolution des conflits structurels et sémantiques via l'utilisation des agents qui permettent la construction d'une architecture ouverte et évolutive. La gestion de la sémantique dans cette architecture est basée sur l'utilisation des ontologies de domaine. Chaque ontologie est constituée d'un ensemble de liens sémantiques, la résolution des conflits sémantiques est basée sur l'exploitation de l'ensemble de ces liens. L'architecture ACSIS est composée de plusieurs types d'agents :

- des Agents d'Accès aux sources locales (AA) assurent la participation des données locales au sein des processus coopératifs. L'AA est un agent mono-domaine dont l'ontologie est constituée de la hiérarchie d'objets descriptifs et des liens intra-base,

Chapitre 2

Notion de sémantique et d'interopérabilité sémantique dans les systèmes d'information coopératifs

- des Agents Informationnels (AI) regroupent les AA proches sémantiquement dans un réseau d'accointances, découpent les requêtes en sous-requêtes, ont accès aux sources locales par le biais des AA pour répondre à des requêtes d'utilisateur ou d'autres AI et récupèrent les résultats pour obtenir une solution globale. Les AI communiquent entre eux lorsqu'ils n'ont pas la connaissance suffisante pour répondre à une requête. L'AI est un agent multi-domaines (regroupant des domaines proches sémantiquement) dont l'ontologie est constituée des liens inter-bases.
- un troisième type d'agent nommé Agent d'Interface sert d'intermédiaire entre les utilisateurs et les autres agents.

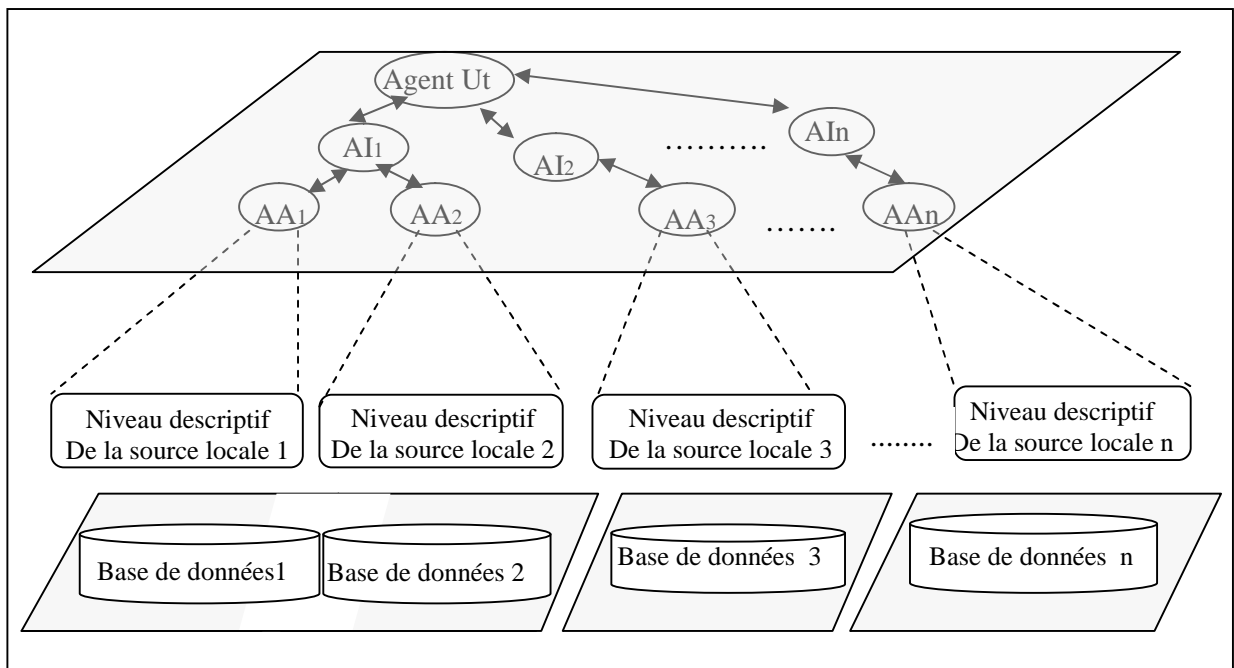


Figure 2.7. Différents niveaux de l'architecture ACSIS [45].

2.6.2.1. Synthèse

ACSIS propose des solutions pour la majorité des conflits sémantiques liés à la coopération des sources d'information contrairement à la majorité des architectures qui traitent seulement les conflits de nommage. ACSIS est basée sur le paradigme multi-agents qui respecte la nature des SIC. Chaque agent de l'architecture encapsule des connaissances sémantiques sous la forme d'une ontologie ce qui fournit une vision distribuée des capacités des différentes sources d'information. Le point faible de cette architecture réside dans le grand nombre de messages échangés entre les différents agents pendant l'ajout d'une nouvelle source

Chapitre2

d'information ou pendant la validation de liens sémantiques au moment du traitement des requêtes.

2.6.3. L'architecture DECA

Dans cette section nous présentons l'architecture DECA proposée par D. Benslimane [2].

2.6.3.1. Présentation de l'architecture

L'architecture DECA élabore une solution aux problèmes d'interopérabilité sémantique des systèmes d'information. Cette architecture adopte une médiation sémantique réalisée autour d'une architecture à base d'agents qui coopèrent «intelligemment» pour la résolution dynamique et décentralisée de requêtes distribuées. Pour homogénéiser l'information échangée et partagée entre les SI hétérogènes la médiation sémantique repose sur le concept d'ontologie qui fournit une description sémantique explicite aux informations échangées. L'interprétation correcte d'une requête est guidée par l'utilisation de règles sémantiques couplées aux concepts afin d'enrichir leur sémantique et permettre la correspondance entre les termes des requêtes et leurs équivalents locaux. DECA est construite à l'aide de quatre types d'agents :

- les agents de coopération (CA) assurent les traductions sémantiques des requêtes et des données. Ils permettent d'apporter une sémantique aux données des sources d'information en exprimant un ensemble de correspondances appelé contexte, entre les objets locaux et les concepts de l'ontologie.
- un agent ontologie (OA) pour la médiation sémantique, l'ontologie est vu comme un vocabulaire commun utilisé par les agents AC pour s'échanger des requêtes sans avoir à travailler sur un schéma global.
- les agents wrapper (WA) jouent le rôle d'une interface entre une source d'information et un agent de coopération. Ils assurent un service d'accès aux données locales d'une base.
- les agents interfaces (UIA) facilitent la formulation de requêtes et la visualisation des résultats.

La stratégie de traitement de requêtes distribuées adoptée dans DECA est basée sur une nouvelle méthode de réduction progressive des requêtes où la décomposition statique est abandonnée au profit d'une décomposition dynamique et décentralisée.

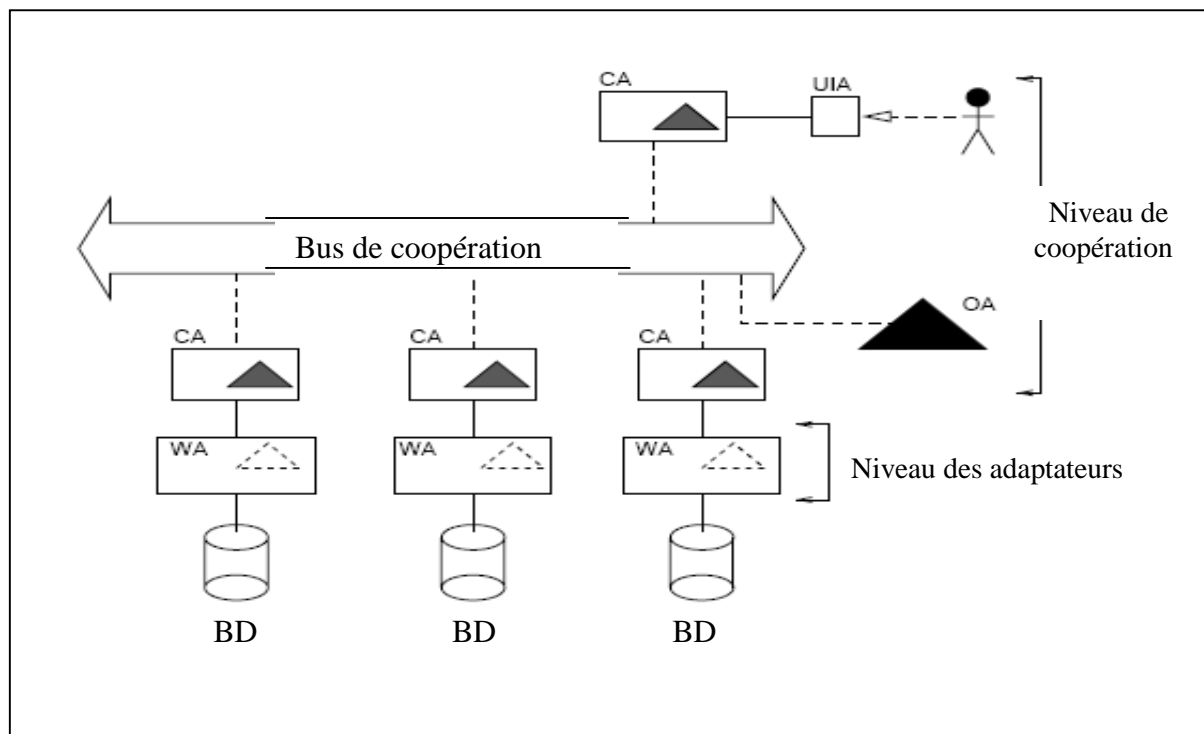


Figure 2.8. Différents composants de l'architecture DECA [2].

2.6.3.1. Synthèse

DECA propose une bonne solution pour la coopération de sources d'information. Elle est construite au tour d'une architecture multi-agents qui respecte la nature distribuée des sources d'information. La résolution des conflits au niveau sémantique est réalisée via la définition des règles sémantiques couplées aux concepts, afin d'enrichir leur sémantique, ce qui facilite la traduction sémantique de requêtes par les agents de coopération. Un des points forts de cette architecture réside dans l'utilisation d'une méthode de décomposition dynamique et décentralisée des requêtes.

2.6.4. Une architecture multi-couches pour le traitement distribué de requêtes

Dans cette section nous présentons une architecture multi-couches proposée par S.M. Benslimane [3] dont le but est d'interroger des sources d'information distribuées et hétérogènes.

2.6.4.1. Présentation de l'architecture

Cette architecture propose une solution au problème d'interopérabilité sémantique entre des sources d'information hétérogènes. Elle est basée sur la découverte de mappings entre les ontologies. Trois couches sont définies pour assurer un traitement sémantique des requêtes :

- La couche des sources d'information (1) inclut un ensemble de sources d'information. Chaque source est liée à une ontologie locale par des relations sémantiques.

Chapitre 2

Notion de sémantique et d'interopérabilité sémantique dans les systèmes d'information coopératifs

- La couche de mappings (2) assure la découverte de mappings entre les différentes ontologies locales. Cette découverte est réalisée par quatre modules : module d'exécution des méthodes de similarité, module de combinaison des méthodes de similarité, module de découvert des mappings, module de présentation des règles de mappings. Les mappings découverts dans ce niveau sont stockés dans le registre de mappings de la couche supérieure.
- La couche de requêtes (3) joue le rôle d'interface entre l'utilisateur et les couches inférieures. Le composant principal de cette couche est le moteur de requêtes qui assure le traitement de requêtes des utilisateurs. Il exploite les mappings stockés dans le registre de mappings afin de décomposer la requête globale en sous-requêtes destinées aux différentes sources d'information.

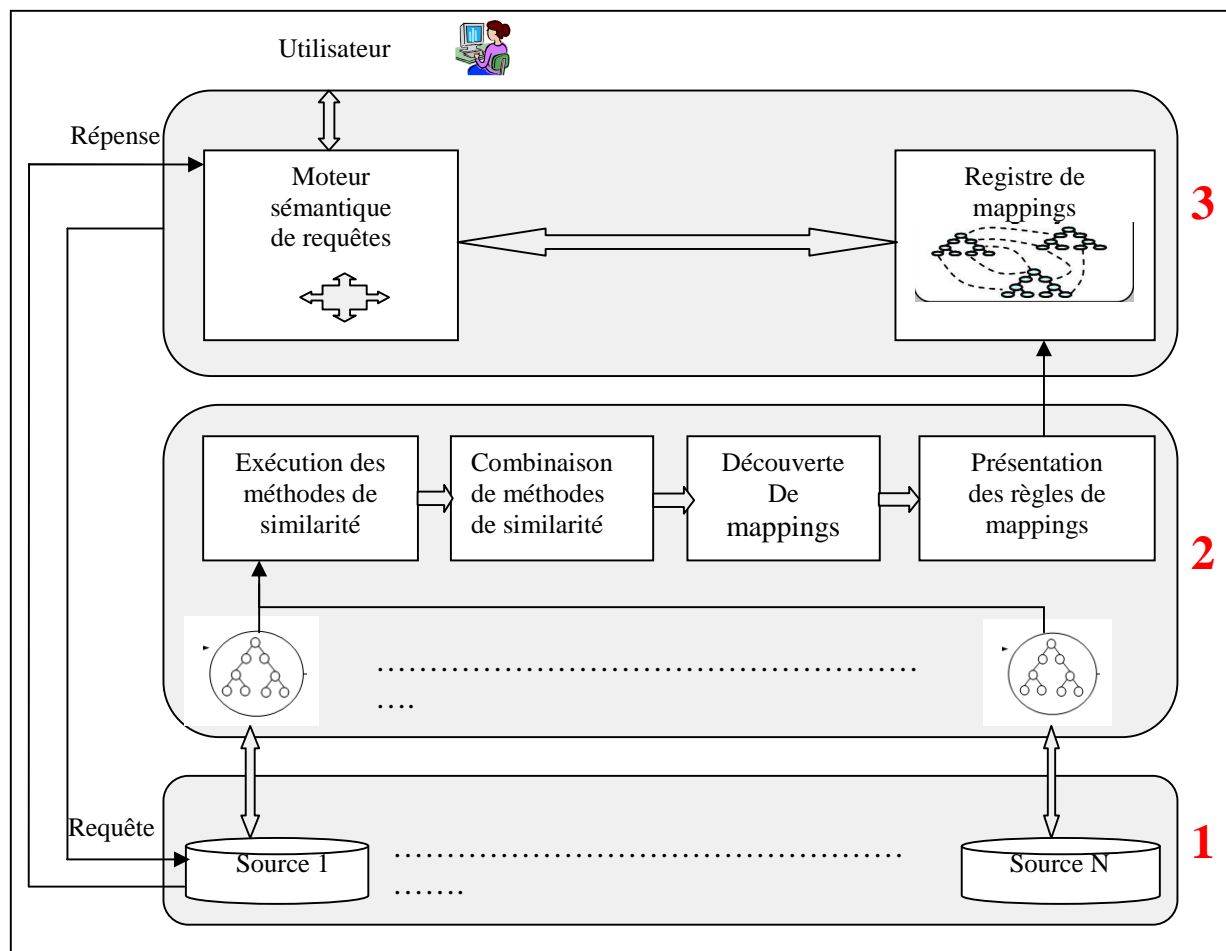


Figure 2.9. Architecture multi-couches pour le traitement distribué de requêtes [3].

Chapitre2

2.6.4.1. Synthèse

L'architecture présentée dans la section précédente propose une bonne solution pour l'interrogation simultanée des sources d'information. La définition d'une couche sémantique entre l'utilisateur et les sources d'information facilite la résolution dynamique des conflits via un ensemble de modules qui assurent des fonctionnalités complémentaires. Le point faible de cette architecture réside dans la centralisation du traitement de requête dans un seul moteur sémantique de requêtes.

2.6.5. Une architecture multi-agents pour le traitement distribué de requêtes

Dans cette section nous présentons une autre architecture multi-agents proposée par N.Alghamdi [1] dont l'objectif d'interroger des sources d'information de type XML et RDF.

2.6.5.1. Présentation de l'architecture

Cette architecture propose une solution multi-agents basée sur une approche ontologique hybride pour faciliter le partage d'informations entre des sources distribuées et hétérogènes. Deux types de sources d'information sont considérés (XML et RDF). Le fonctionnement global de l'architecture est assuré via un ensemble d'interactions entre plusieurs types d'agents : Agent de traitement, Agent de mappings, Agent de transfert, Agent d'importation des ontologies et Agent de schéma. Afin d'expliquer le fonctionnement et le rôle des différents agents nous proposons dans ce qui suit un scénario d'exécution d'une requête posée par un utilisateur.

- L'utilisateur choisi premièrement les termes de sa requête et l'agent de traitement s'occupe de la construction de cette dernière en utilisant un langage classique de requêtes.
- L'agent de traitement passe la requête formulé à l'agent de mappings. Ce dernier s'occupe de la traduction de la requête vers l'ontologie du domaine choisi. si cette ontologie n'est pas présente dans la base des ontologies du domaine, l'agent de mappings invoque l'agent d'importation des ontologies afin de récupérer cette ontologie à partir du répertoire des ontologies.
- L'agent de mappings passe la requête et toutes les informations nécessaires pour le traitement de cette dernière à un agent mobile appelé agent de transfert. Ce dernier s'occupe de la transmission de la requête aux sources d'information adéquates.

Chapitre 2

Notion de sémantique et d'interopérabilité sémantique dans les systèmes d'information coopératifs

- Dans le cas d'une source d'information RDF, l'agent de transfert passe la requête et le module d'ontologie globale à un agent de mappings. Cet agent s'occupe de la découverte de mappings entre l'ontologie globale et locale afin de réaliser les traductions nécessaires sur les termes de la requête. La requête est passée par la suite à un agent de traitement qui transforme cette dernière vers le langage adéquat de la source RDF. Après l'exécution de la requête l'agent de traitement passe les résultats à l'agent de transfert qui va transférer ces dernières vers le site d'utilisateur.
- Dans le cas d'une source XML le schéma de la source est transformé initialement vers le modèle RDF. Cette transformation est réalisée par l'agent de schéma. L'objectif de cette transformation est de construire une ontologie temporaire locale pour cette source XML. Après la construction de cette ontologie temporaire, le traitement de la requête suit les mêmes étapes utilisées dans le cas d'une source RDF.

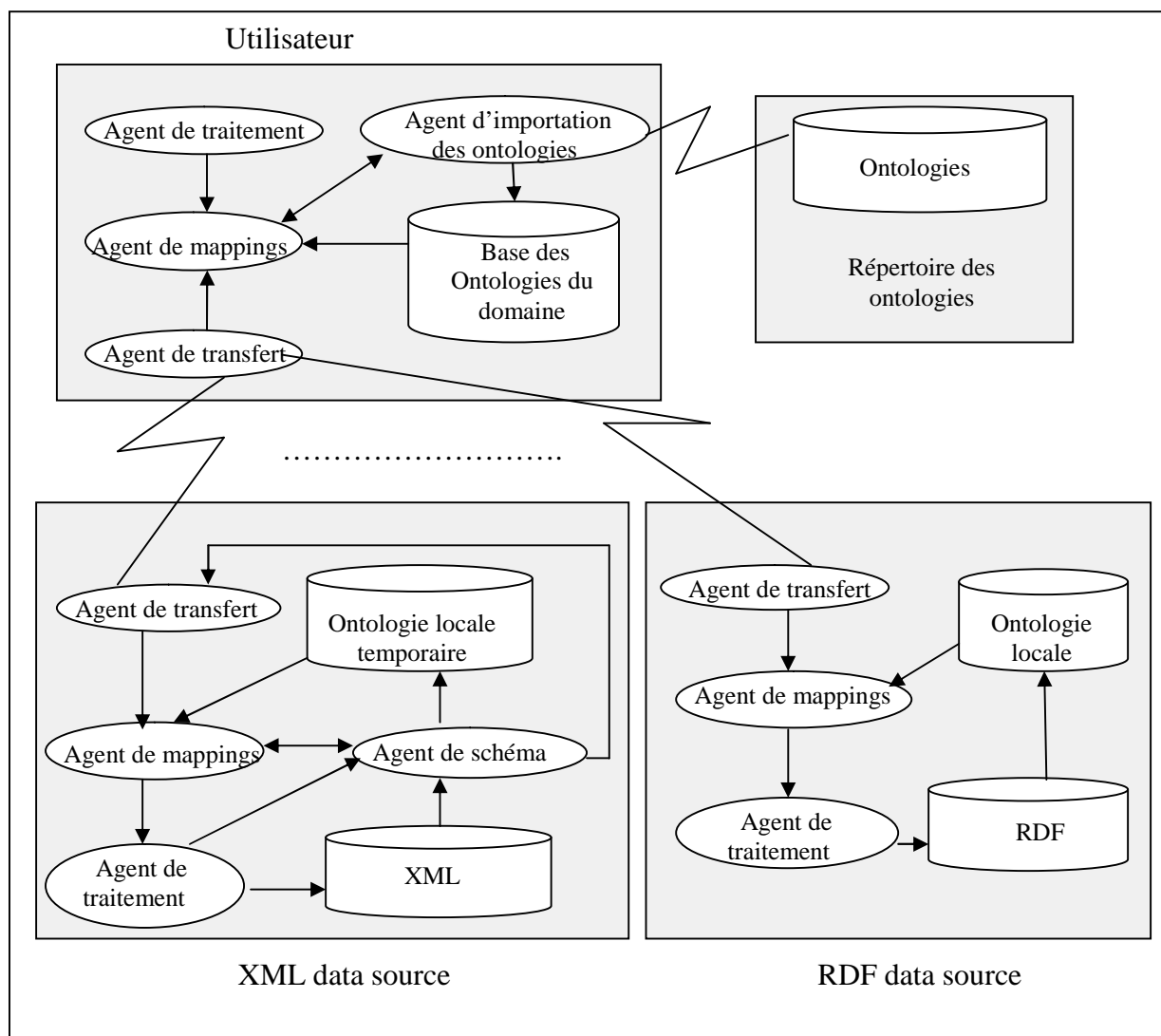


Figure 2.10. Architecture multi-agents pour le traitement distribué de requêtes [1]

2.6.5.2. Synthèse

L'architecture présentée dans la section précédente propose une solution pour l'interrogation sémantique des sources d'information de type RDF et XML. Elle exploite la technologie des agents mobiles ce qui facilite le transfert des requêtes et des résultats entre les différents nœuds de l'architecture. La résolution des conflits sémantiques est réalisée d'une manière dynamique par des agents spécialisés (agent de mappings). Le point faible de cette architecture réside dans la grande quantité d'information transférée entre les différents agents pendant le traitement de requêtes surtout les informations liées à l'aspect sémantique. Aussi, ce système est destiné à l'interrogation de deux types de sources d'information seulement (XML et RDF) ce qui limite la capacité d'interroger les autres types de sources surtout dans le cadre du web caractérisé par une grande masse d'informations hétérogènes.

2.6.6. Discussion

L'étude présentée dans les sections précédentes nous permet de construire une vision concernant la mise en œuvre d'un bon système dédié à la coopération de sources d'information distribuées et hétérogènes. Cette vision est construite à partir des points forts et des points faibles constatés dans chaque architecture. Elle concerne beaucoup plus les aspects liés à la gestion de la sémantique.

Par conséquent, notre vision est caractérisée par les éléments suivants :

- La combinaison d'une approche ontologique hybride pour la résolution des conflits sémantiques avec l'architecture multi-agents semble être un bon candidat pour construire un tel système.
- Le regroupement des sources d'information dans des sites et l'utilisation d'une approche mono-ontologique dans chaque site est avantageux surtout dans le cas des environnements qui regroupent des centaines de sources d'information.
- L'utilisation des composants supplémentaires pour résoudre les conflits sémantiques non traités par les ontologies (ex : conflits d'échelle.) est indispensable pour assurer une transparence totale aux utilisateurs.
- L'indépendance entre le fonctionnement du système et les types des sources d'information interrogées est importante. Il faut que le système soit générique pour faciliter son utilisation dans l'interrogation des différents types de sources (XML, relationnel, RDF...etc.).

2.7. Conclusion

Dans ce chapitre, nous avons exposé la notion de sémantique et sa représentation par les ontologies qui constituent le meilleur outil conceptuel de capture et de description des connaissances. Le chapitre présente aussi une étude de quelques techniques et architectures qui proposent des solutions pour le problème d'hétérogénéité sémantique dans le cadre de coopération des sources d'information. Cette étude est conclue par une discussion qui résume les points forts des architectures présentées. Ces points sont les caractéristiques que nous tentons de vérifier dans l'architecture proposée dans le chapitre suivant.

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

3.1. Introduction

Dans les chapitres précédents nous avons présenté entre autres les SIC et leurs caractéristiques, les SMA et le rôle de ces derniers dans la modélisation des SIC. Nous avons évoqué aussi le problème de sémantique lors de la coopération des SIs. Une étude de quelques systèmes qui propose des solutions à ce problème est présentée. A partir de cette étude une vision globale est construite. Cette vision résume les caractéristiques d'un bon système de coopération de sources d'information. Dans ce chapitre nous essayons de construire notre propre système en se basant sur ces caractéristiques.

Le présent chapitre vise deux objectifs. Le premier consiste à construire une architecture décentralisée pour supporter la coopération des sources d'information distribuées et hétérogènes. Le deuxième permet de résoudre les conflits sémantiques qui gênent la coopération et l'interopérabilité de ces sources. L'architecture proposée est basée sur le paradigme multi-agents qui constitue le point clé dans la modélisation des SIC. Pour résoudre le problème d'hétérogénéité sémantique nous proposons l'utilisation des ontologies.

Ce chapitre est structuré en plusieurs parties. En premier lieu, nous décrivons les principes de notre approche globale. Dans la deuxième partie nous présentons une architecture multi-agent pour l'approche proposée et les architectures internes des différents agents. Ensuite, nous présentons le protocole de coopération utilisé par les agents pendant le traitement des requêtes globales.

3.2. Vue globale de l'approche proposée

Notre contribution porte sur la coopération de systèmes d'information et plus précisément sur la coopération de sources d'information distribuées et hétérogènes. La distribution et l'hétérogénéité au niveau sémantique sont au cœur de notre problématique. Cela interpelle la mise en place d'une approche en se basant sur des outils qui respectent ces deux propriétés (distribution et hétérogénéité). Notre idée générale consiste à modéliser les SIC selon une approche décentralisée où le contrôle et les connaissances sont répartis sur un ensemble d'entités. Ces entités sont distribuées et possèdent des spécificités intrinsèques liées au caractère potentiellement hétérogène des sources d'information. La communication entre ces entités permet aux utilisateurs de manipuler de façon transparente les données issues d'un ensemble de sources d'information.

Les sociétés d'agents semblent être les candidats les plus prometteurs dans notre contexte. Les agents ont la capacité de coopérer en mettant en œuvre des protocoles de communication pour l'échange et le partage des informations avec les autres agents afin de traiter les requêtes globales posées par l'utilisateur.

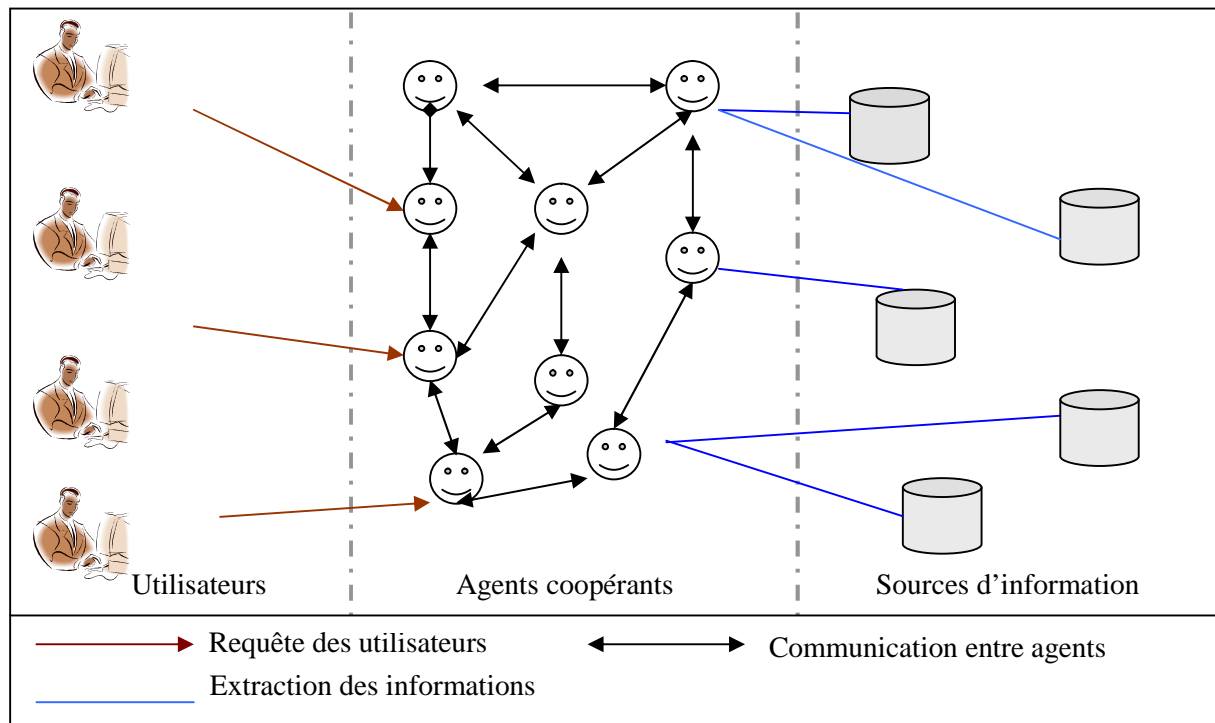


Figure 3.1. Interaction des agents pour le traitement des requêtes.

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

Pour la résolution du problème d'hétérogénéité sémantique nous proposons l'utilisation d'une approche ontologique hybride. Nous supposons l'utilisation de plusieurs ontologies locales. Chaque ontologie décrit la sémantique des données stockées dans plusieurs sources d'information. Pour fournir une vue globale du domaine de coopération nous proposons la construction d'une ontologie globale à partir des ontologies locales. Cette ontologie est utilisée comme un vocabulaire pour formuler les requêtes des utilisateurs.

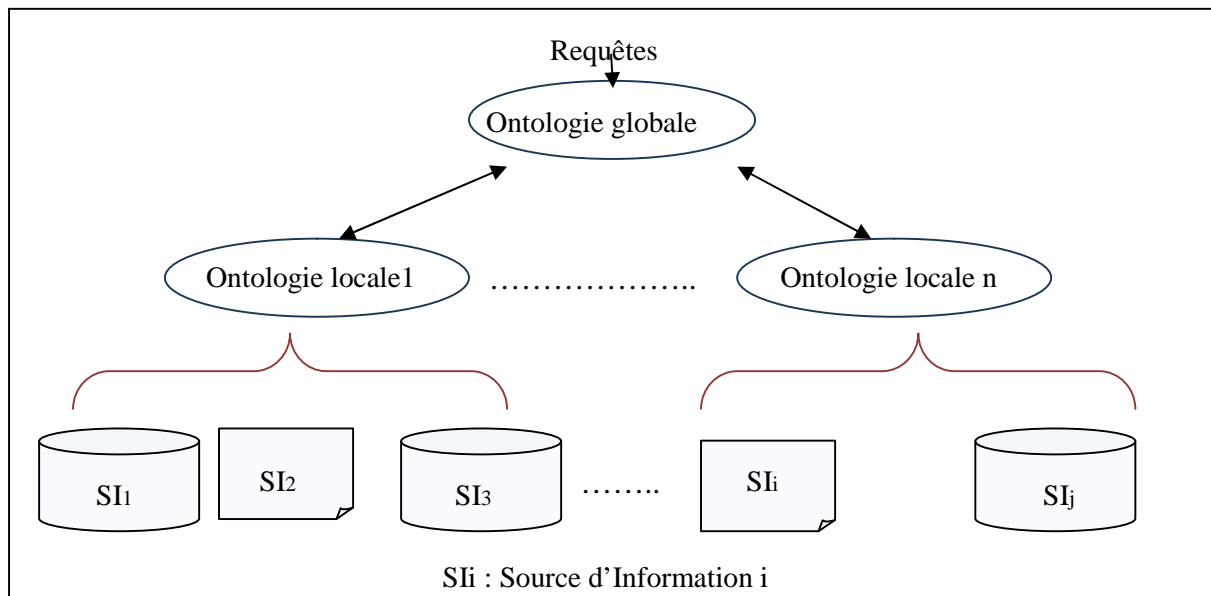


Figure 3.2. Approche ontologique hybride pour l'interopérabilité sémantique.

Par conséquent, le traitement des requêtes est réalisé via les interactions entre les différents agents et via l'exploitation des connaissances sémantiques présentées dans les différentes ontologies (locales et globale). Dans ce qui suit, nous présentons en détails comment le paradigme multi-agents est combiné avec l'approche ontologique hybride pour construire une architecture dédiée à la coopération des sources d'information.

3.3. Une architecture générique pour l'approche proposée

Cette section est dédiée à la description d'une architecture générique dans laquelle plusieurs sources d'information sont prêtes à coopérer. La figure 3.3., donne une vue générale de cette architecture. Elle est structurée en trois niveaux. Le premier niveau regroupe l'ensemble des sources d'information. Le deuxième niveau assure l'homogénéisation des modèles de données locaux des sources d'information. Il est composé d'un ensemble d'agents adaptateur (AA). Chaque agent adaptateur fournit un service d'accès aux données locales d'une source d'information. Notre proposition s'intègre dans le troisième niveau (niveau de coopération).

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

Ce niveau matérialise l'interopérabilité sémantique via l'approche ontologique hybride combinée avec un ensemble d'agents d'information (AI) qui communiquent pour réaliser l'objectif principal de la coopération qui est le traitement de requêtes des utilisateurs d'une manière transparente. Un troisième type d'agent est utilisé, c'est l'agent annuaire (AN) qui assure les fonctionnalités d'un portail pour notre architecture. Dans ce qui suit, une description détaillée de ces trois types d'agents (AA, AI, AN) est présentée.

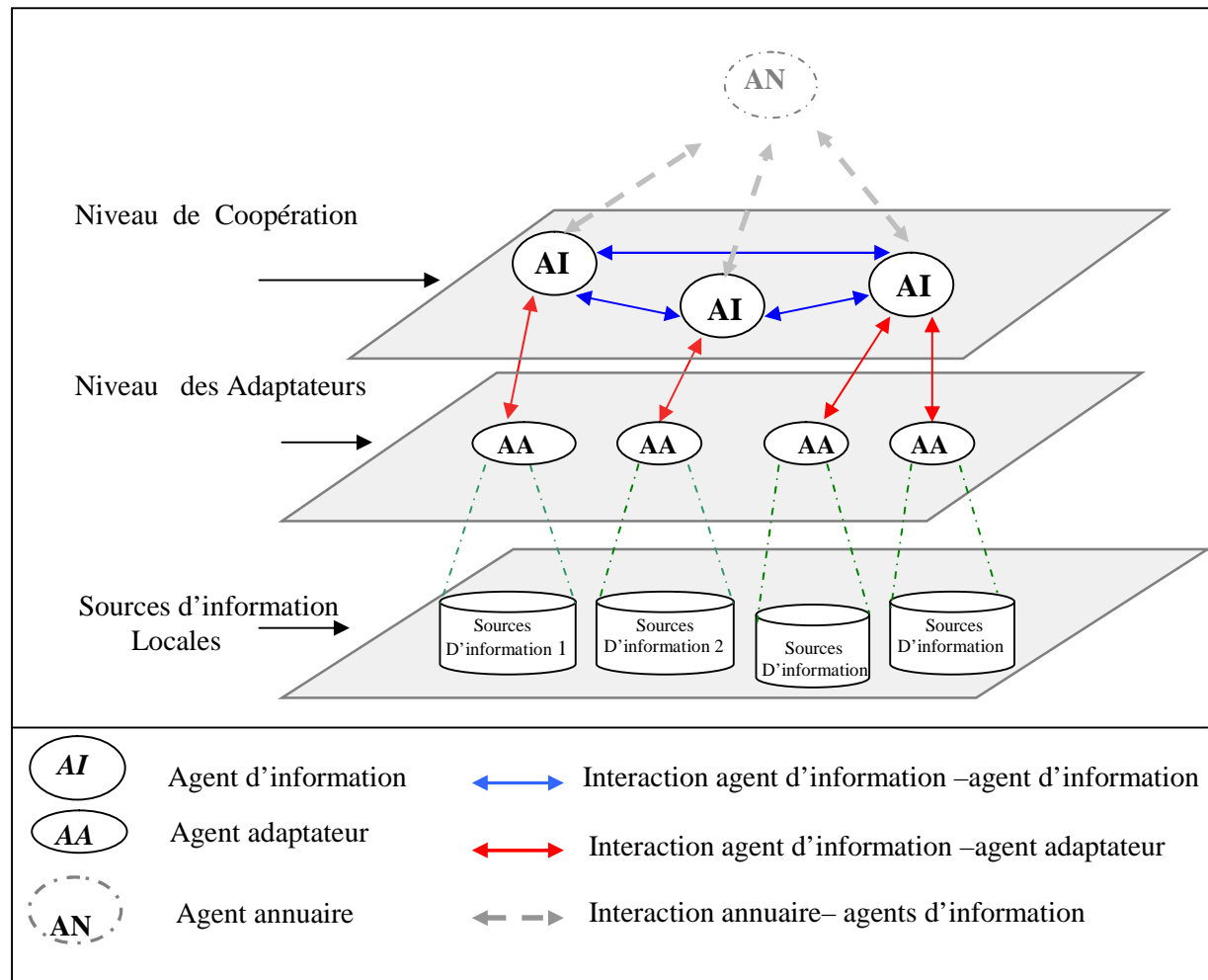


Figure 3.3. Une architecture générique pour les SIC.

3.3.1. Agents d'information

Les agents d'information sont définis selon plusieurs points de vue d'un auteur à l'autre. Dans notre contexte nous considérons ces derniers comme une entité logicielle capable d'accéder à un ensemble de sources d'information hétérogènes et distribuées mais proches sémantiquement. Une des fonctions principales assignées à nos agents informationnels est la résolution des conflits sémantiques pendant le traitement des requêtes. Par conséquent, la structure interne des agents doit permettre une bonne gestion de ces conflits résultant des

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

hétérogénéités sémantiques entre l'ensemble des sources d'information coopérantes. Pour assurer cette fonctionnalité les agents d'information doivent encapsuler des bases de connaissances sémantiques pour gérer l'ensemble de ces conflits.

A partir de cette description générale de nos agents informationnels nous pouvons énumérer les caractéristiques clés de ces derniers comme suit :

- D'après la définition d'un agent fournie par Jacques Ferber [18], nos agents sont caractérisés par les propriétés suivantes : autonomie, action, communication.
- **Autonomie** : les agents peuvent opérer sans l'intervention directe des autres agents ou humains.
- **Action** : L'agent exécute une sous tâche spécifique relative à la solution du problème considéré (sous requêtes négociées).
- **Communication** : les agents peuvent échanger des messages et des données afin d'accomplir les tâches globales.
- Du fait que nos agents informationnels adoptent la responsabilité de résoudre les différents conflits sémantiques via l'ensemble des connaissances encapsulées dans leurs Bases de connaissances internes, nous pouvons classer ces derniers dans la catégorie des **Agents cognitifs**.
- Aussi, dans le cadre d'une dynamique collective, l'agent doit disposer d'un certain nombre de compétences nécessaire pour la coopération. Il doit pouvoir :
 - **Intégrer des informations** venant d'autres agents pour fournir des résultats aux requêtes globales posées par l'utilisateur ;
 - **Déléguer une tâche** qu'il ne sait pas résoudre à un autre agent dont il connaît les compétences.

Toutes ces caractéristiques forment les qualités essentielles de notre agent d'information.

Dans notre architecture, chaque agent d'information contient les composants suivants : *Module de communication, cache sémantique, Base de Connaissances, Base de description des schémas locaux, Bibliothèques de fonctions, Moteur de requêtes globales et un Planificateur local*. Ces composants travaillent ensemble afin de traiter les requêtes des utilisateurs d'une manière transparente.

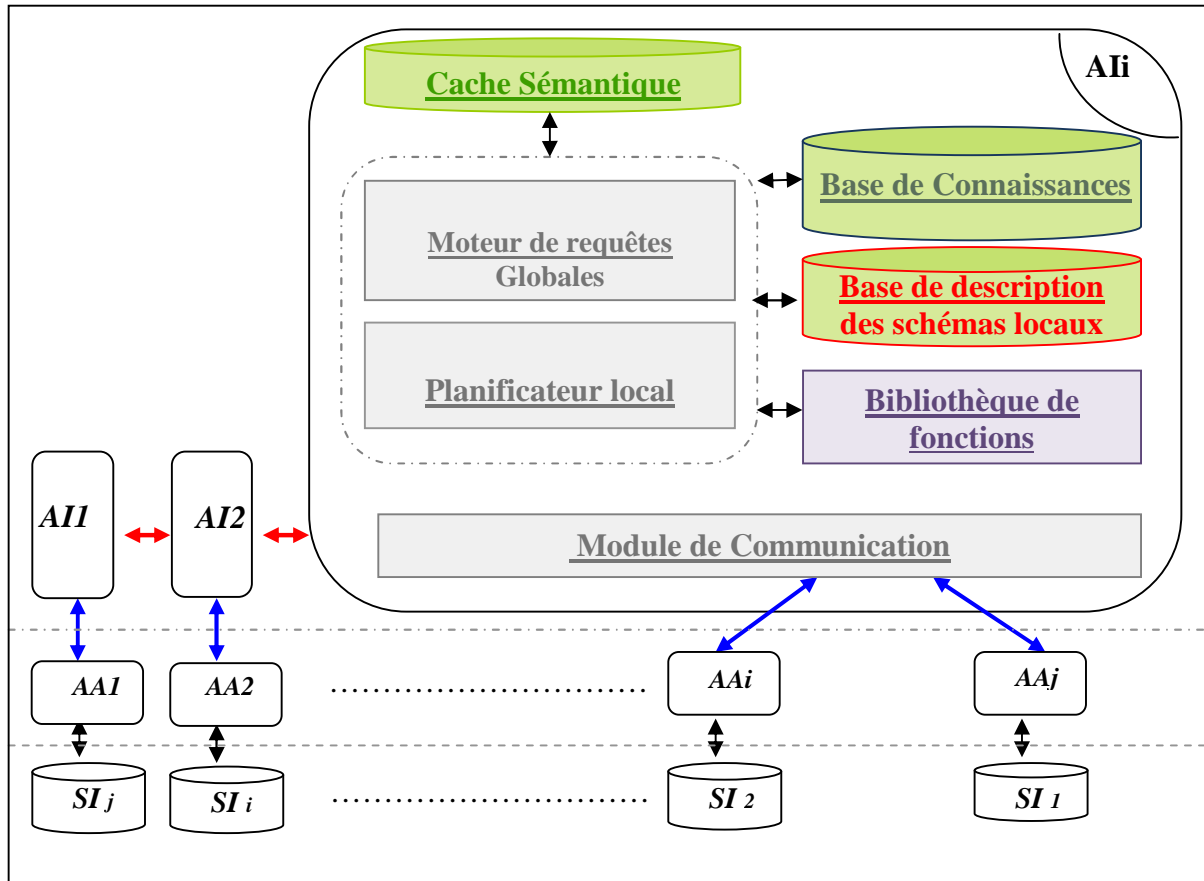


Figure 3.4. Architecture interne d'un agent d'information.

3.3.1.1. Module de communication

Ce module gère les interactions entre un agent d'information et le monde extérieur tel que les autres agents d'information, les agents adaptateurs et l'agent annuaire. Il contient tous les processus de prise en charge des messages.

3.3.1.2. Cache sémantique

Les sources de données accessibles par les agents d'information peuvent être dispersées sur un large réseau dont le temps de réponse par rapport à une base de données locale peut être important. Il est donc avantageux d'utiliser un cache local permettant de stocker temporairement les données envoyées par les sources en réponse à des requêtes, de sorte à limiter le trafic si d'autres requêtes futures accèdent à ces mêmes données.

3.3.1.3. Base de connaissances

Afin de répondre au besoin d'interopérabilité sémantique entre les sources d'informations coopérantes, l'architecture des agents d'information est consolidée par une base de connaissances qui a pour objectif la mise en œuvre d'un moyen de communication entre les

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC agents sous la forme d'une coopération sémantique basée sur les ontologies et sur la mise en place de relations (de mappings) entre ontologies. Pour cela nous avons adopté une approche hybride dans laquelle les connaissances sémantiques de chaque agent d'information sont présentées sous forme d'une ontologie locale et pour donner une vue générale du domaine globale de la coopération nous définissons une ontologie globale construite à partir des ontologies locales. Les requêtes des utilisateurs sont exprimées en termes des éléments de cette ontologie globale. Pour faciliter la traduction des requêtes vers les termes des ontologies locales, chaque agent d'information sauvegarde des correspondances entre son ontologie locale et l'ontologie globale dans un répertoire de mappings. Dans ce qui suit, nous présentons en détail les ontologies locales, les étapes de construction de l'ontologie globale et le formalisme utilisé pour la représentation des mappings.

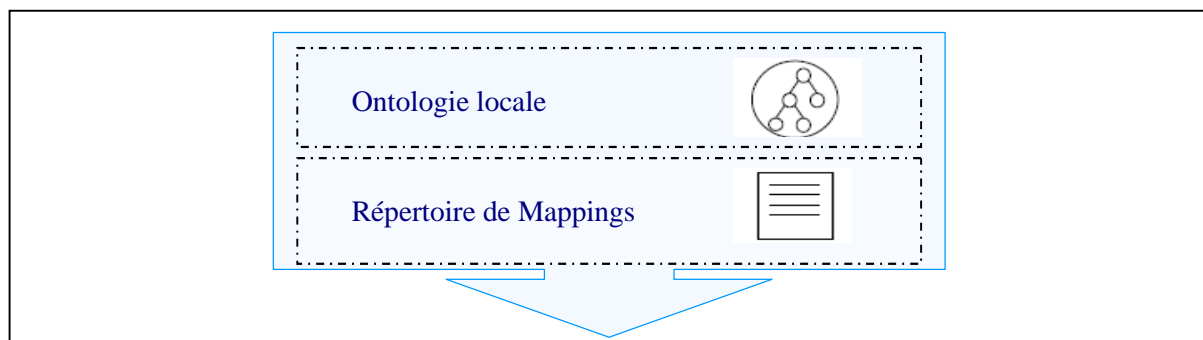


Figure 3.5. Composants de la base de connaissances.

➤ Ontologies locales

Chaque ontologie locale est utilisée pour la description des schémas d'un ensemble de sources d'information. Cette description est présentée sous forme de liens sémantiques stockés dans la base de description des schémas de l'agent concerné par l'extraction des informations à partir de ces sources. Nous adoptons pour la conception des ontologies locales la méthode qui consiste à transformer les modèles de données en une ontologie. Par conséquent, nos ontologies sont classiques, construites à partir du modèle conceptuel des sources d'information. Elles sont toutes décrites à l'aide du langage OWL qui offre d'importantes possibilités de description comme les relations d'équivalence, d'identité et de différence entre classes, propriétés et relations.

➤ Ontologie globale

Dans notre approche, l'ontologie globale est considérée comme une interface qui donne une vue unifiée sur l'ensemble des domaines couverts par les ontologies des différents AI coopérants. En réalité, cette ontologie n'est rien que la fusion des ontologies locales via un

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

ensemble de liens sémantiques. Les étapes de construction de cette ontologie globale sont les suivantes :

- **Etape1 : Analyse des ontologies locales** : cette phase implique l'analyse des ontologies locales existantes. L'objectif de cette analyse est la détection des conflits sémantiques qui peuvent exister entre les différentes ontologies. Les conflits sémantiques traités dans notre approche sont les conflits de synonymie et d'homonymie. Concernant les conflits d'échelle nous proposons l'utilisation d'une bibliothèque de fonction pour faciliter les conversions entre les unités de mesure.

Cette étape peut être réalisée en utilisant les méthodes de découverte des mappings qui facilitent le calcul des similarités et des dissimilarités entre les concepts des différentes ontologies.

- **Etape 2 : Résolution des conflits sémantiques** : Cette phase est dédiée à la résolution des conflits sémantiques détectés dans la phase précédente selon les règles suivantes :

Règle 1 : dans le cas des homonymes, les concepts identiques par leurs nom et sémantiquement différents sont renommés au niveau de l'ontologie globale.

Règle 2 : dans le cas des synonymes, les concepts qui ont des noms différents mais sémantiquement identiques sont exprimés par un seul concept dans l'ontologie globale.

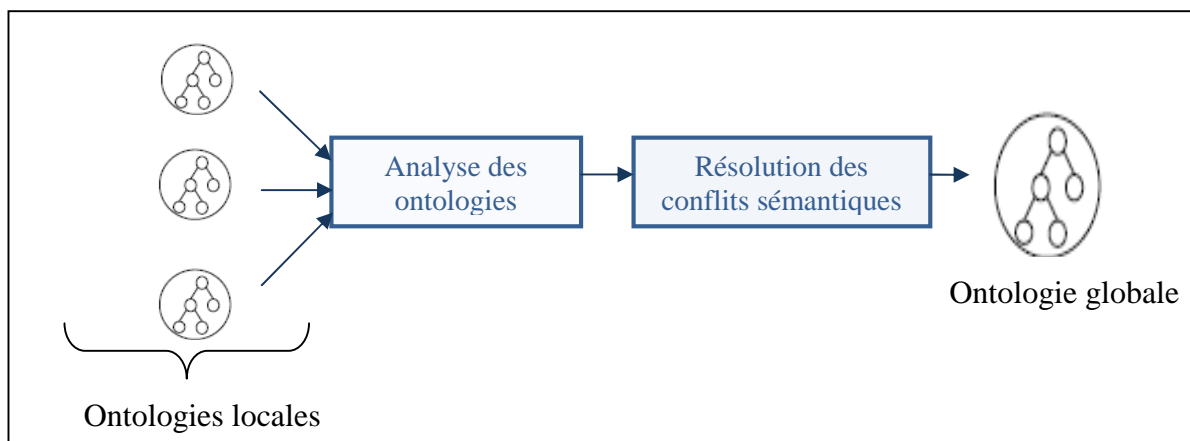


Figure 3.6. Etapes de construction de l'ontologie globale

➤ Répertoire de mappings

En parallèle avec la création de l'ontologie globale, les répertoires des Mappings sont construits. Nos mappings sont exprimés en utilisant le langage OWL car ce dernier fournit des performatifs pour exprimer des relations sémantiques entre les différents termes des ontologies.

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

Notre approche ne traite que les conflits de synonymie et d'homonymie. Alors, les mappings sont définis en utilisant seulement les performatifs (**owl:equivalentClass** et **owl:equivalentproperty**) qui expriment les équivalences entre les éléments (relations, concepts, propriétés) des ontologies.

Exemples:

équivalence entre concepts	<pre><owl: Class rdf: about="&ontologie-globale;Concept1"> <owl: <i>equivalentClass</i> rdf: resource="&ontologies-locale;Concept2"/> </owl: Class></pre>
Équivalence entre propriétés	<pre><owl: DatatypeProperty rdf: about="&ontologie-globale; Propriete1"> <owl: <i>equivalentProperty</i> rdf: resource="&ontologies-locale; Propriete2"/> </owl: DatatypeProperty></pre>

3.3.1.4. Base de description des schémas locaux

Pour faciliter l'interrogation des sources d'information, nous proposons l'utilisation d'une base de description des schémas pour stocker les mappings entre l'ontologie locale de l'agent d'information et les schémas des sources d'information. Par exemple, si nous avons une base de données relationnelle comme sources d'information, les mappings entre le schéma de cette dernière et l'ontologie locale d'un agent d'information peuvent être définis comme suit :

- Chaque table du schéma peut être associée à une classe de l'ontologie.
- Chaque attribut de schémas peut être associé à une propriété de l'ontologie.
- Chaque clé étrangère peut être associée à une relation entre deux concepts de l'ontologie.

3.3.1.5. Moteur de requêtes Globales

Le moteur de requêtes globales s'occupe de toutes les fonctions liées au traitement des requêtes des utilisateurs. Parmi ces fonctions nous citons les suivantes :

- L'exploitation des mappings stockés dans la base de connaissance pour cerner les parties des requêtes globales qui peuvent être traitées localement par l'agent d'information.
- L'exploitation du contenu du cache sémantique pendant le traitement des requêtes.
- la fusion des sous-résultats envoyés par les autres agents d'information pour former un résultat global.

3.3.1.6. Planificateur local

Le planificateur local assure la bonne exécution des parties de requêtes globales cernées par le moteur de requête globale. Il exploite la base de description des schémas pour générer des

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

Plans d'exécution selon la distribution des données dans les sources d'informations.

3.3.1.7. Bibliothèque de fonctions

Cette bibliothèque de fonctions est utilisée dans notre architecture pour résoudre les conflits sémantiques d'échelle. Elle contient les propriétés de l'ontologie locale qui possèdent une unité de mesure et un ensemble de fonction de conversion entre les unités.

3.3.2. Agents Adaptateurs

L'interaction entre les agents d'information et les différentes sources de données est effectuée par un agent nommé *adaptateur*. L'adaptateur traite de *l'hétérogénéité* des sources. Il traduit le langage commun utilisé par les agents d'information en langage spécifique à la source à laquelle il est connecté. De la même façon, les résultats rendus par la source sont traduits dans un modèle de données commun utilisé par les agents d'information. Outre le passage des requêtes et des résultats, l'interaction entre les agents d'information et les adaptateurs peut aussi servir à communiquer des informations relatives aux sources de données. Ces informations peuvent être des *descriptions des métadonnées* (schémas gérés par la source) et des *descriptions de capacités des sources* (types de requête que la source est capable de traiter). L'architecture interne de nos agents adaptateurs est composée des deux modules suivants.

3.3.2.1. Module de communication

Ce module est utilisé pour communiquer avec les agents d'information de la couche supérieure et aussi pour communiquer avec les sources d'information de la couche inférieure.

3.3.2.2. Traducteur syntaxique

Ce module est le cœur de l'adaptateur, responsable de la traduction des sous-requêtes provenant des Agents d'Information (1) en requêtes exprimées dans le langage propre de manipulation de données de la source locale (3). Cette requêtes est envoyée (4) par la suite à la source correspondante en utilisant le protocole adéquat pour l'exécuter sur ses données locales. Les résultats (5) renvoyés par la source sont exprimés en termes des concepts de son modèle de données locale, avant de les envoyer (8) à l'agent d'information (source de la requête) une traduction de ces derniers vers le modèle de données de cet agent est indispensable (7).

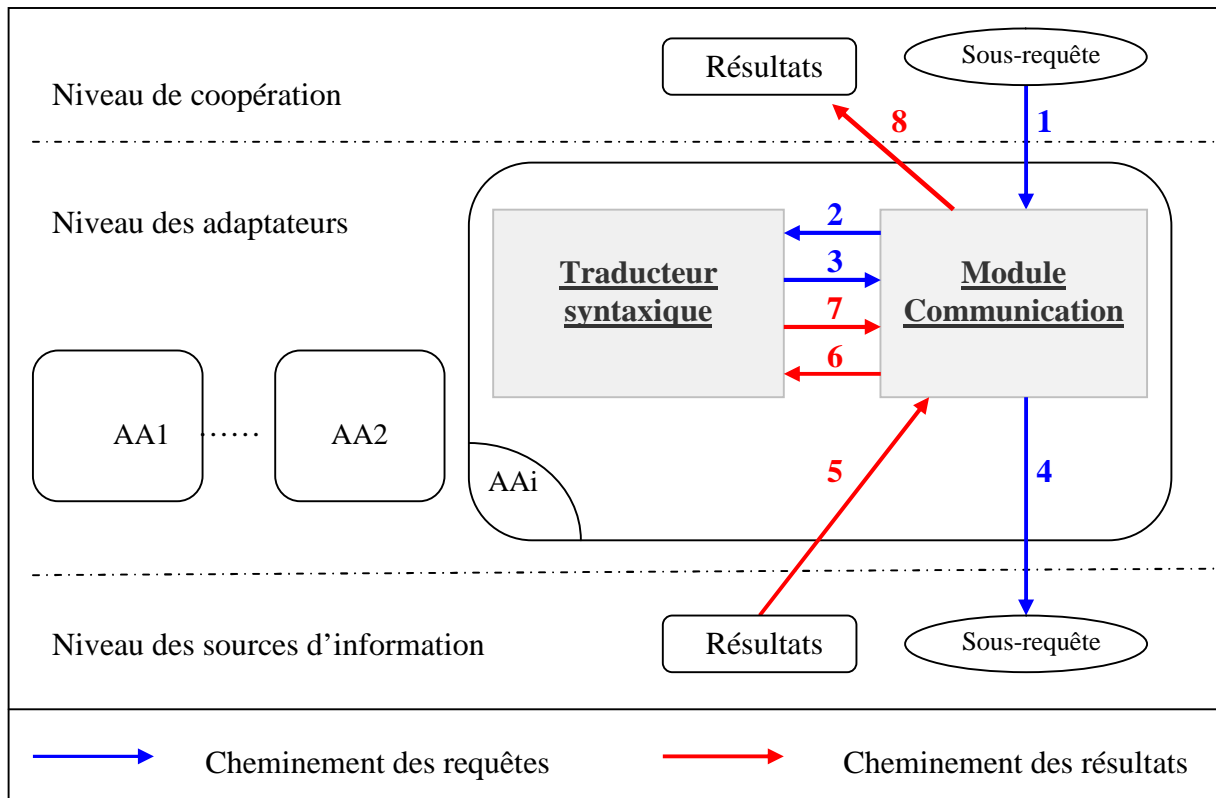


Figure 3.7. Architecture interne d'un adaptateur

3.3.3. L'agent Annuaire

Dans notre architecture, l'agent Annuaire joue le rôle d'un portail pour le SIC. Il centralise une partie des informations de ce dernier afin de minimiser les flux de communication. Il stocke des informations concernant les capacités de chaque agent d'information. En effet, il permet d'éviter un recours systématique à la diffusion lors de prospection d'informations portant sur les compétences des agents ou simplement la recherche de la localisation (l'adresse) d'un agent. Notre agent annuaire est constitué des trois composants suivants : Registre d'expertise, Module de gestion et un Module de communication.

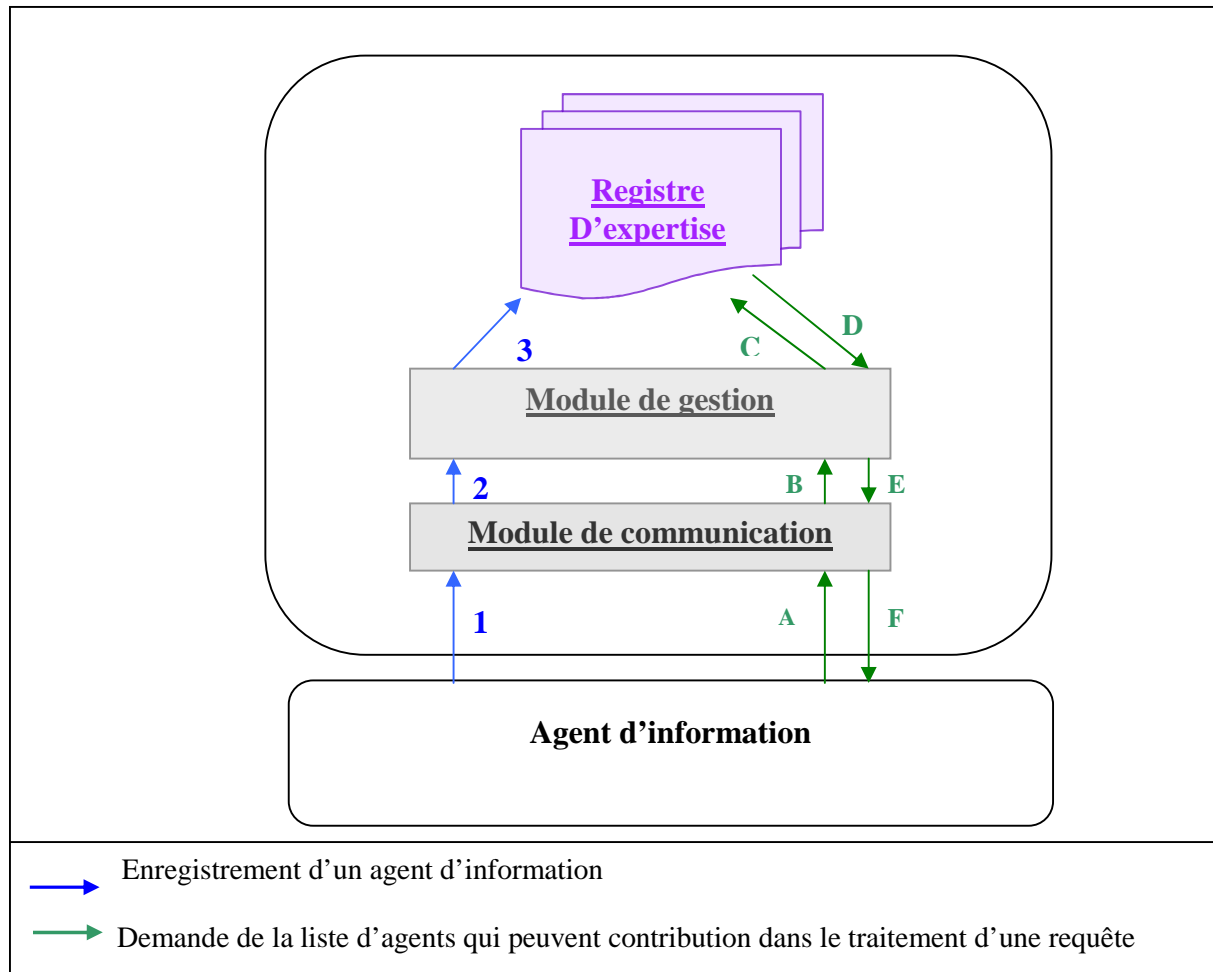


Figure 3.8. Architecture interne de l'agent annuaire

3.3.3.1. Registre d'expertise

Ce registre contient un ensemble d'enregistrements chacun concerne un agent d'information.

Les champs principaux d'un enregistrement sont les suivants :

- *Localisation de l'agent* : ce champ contient des informations générales sur l'adresse de l'agent dans le réseau.
- *Expertise de l'agent* : la notion d'expertise d'un agent est inspirée de celle utilisée dans [11] dans le cadre des systèmes pair à pair de gestion de données, dans ce travail, l'expertise est considérée comme un résumé du *schéma de donnée* du pair en termes de couples de nœuds supportés par ce schéma. Dans notre cadre nous considérons l'expertise d'un agent d'information comme la liste des concepts (Classe) de l'ontologie globale qui ont un correspondant avec les concepts de l'ontologie locale de l'agent d'information.

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

Exemple1 :

Soit AI_i un agent d'information, $Expertise(AI_i) = \{C_1, C_2, C_6, C_8\}$ tel que C_1, C_2, C_6 et C_8 sont des concepts de l'ontologie globale qui ont des correspondants dans l'ontologie locale de l'agent AI_i .

Quand un agent d'information reçoit une requête globale qui dépasse ses capacités, il communique l'annuaire pour avoir des informations sur les agents qui peuvent contribuer dans le traitement de cette dernière (pour éviter une diffusion à tous les agents d'information). L'agent annuaire compare les concepts de cette requête avec les éléments (concepts) qui décrivent l'expertise de chaque agent en consultant les différents enregistrements du registre d'expertise et il ne fournit que la liste d'agents qui peuvent contribuer dans le traitement de la requête posée.

3.3.3.2. Module de gestion

Ce module gère toutes les opérations concernant l'enregistrement des expertises des agents d'information. Il répond aussi aux requêtes des agents qui veulent récupérer la liste des membres qui peuvent contribuer dans le traitement d'une requête.

3.3.3.3. Module de Communication

Cette interface contient tous les processus de prise en charge des messages provenant des agents d'information.

3.3.4. Communication entre les agents de l'architecture

La réussite du fonctionnement de l'architecture proposée passe par une communication efficace entre les différents agents. Le mode de communication que nous avons adopté est celui par envoi de messages. Ce mode s'adapte bien avec notre architecture caractérisée par un comportement globale distribué sur les différents agents qui y sont impliqués, contrairement au mode de partage qui impose une centralisation des données.

3.4. Formulation des requêtes

Pour expliquer le fonctionnement global de notre architecture, nous proposons dans ce qui suit ; une grammaire simple pour la formulation des requêtes. Dans notre approche, nous supposons que l'utilisateur dispose d'une interface de construction de requêtes. Cette interface s'occupe aussi de la transmission de la requête formulée à un des agents d'information.

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

3.4.1. Grammaire utilisée

Après la description de la sémantique des différentes sources d'information par des ontologies locales et la construction de l'ontologie globale, l'utilisateur peut poser ses requêtes en se référant aux éléments de l'ontologie globale. Nos requêtes comprennent deux parties : une partie propriétés recherchées, définie par la clause `<propriete_concept>`, et une partie condition, définie par la clause `<condition_composee>`. Elle est de la forme :

$\text{Requete} ::= \text{'SELECT'} \langle \text{propriete_concept} \rangle \text{'WHERE'} \langle \text{condition_composee} \rangle$
<ul style="list-style-type: none"> • $\langle \text{propriete_concept} \rangle ::= \langle \text{propriete} \rangle (', ' \langle \text{propriete_concept} \rangle)^*$ • $\langle \text{condition_composee} \rangle ::= \text{relation}(\text{'AND'} \langle \text{condition_composee} \rangle)^* / \text{condition_simple}(\text{'AND'} \langle \text{condition_composee} \rangle)^*$ • $\langle \text{condition_simple} \rangle ::= \langle \text{propriete} \rangle \langle \text{opérateur} \rangle \langle \text{valeur} \rangle (\langle \text{echelle} \rangle)^*$

Chacun des composants de la requête est décrit dans le tableau suivant :

Composant de la requête	Description
Propriete	Exprime une relation entre une classe et une propriété. Soit une classe de l'ontologie O, une relation entre une propriété P _i et la classe C _j est dénotée par C _j .P _i .
Relation	Exprime une relation entre deux classes. Soient C _i , C _j deux classes de l'ontologie, une relation entre ces deux classes est dénotée par R (C _i , C _j).
Operateur	Un opérateur logique pouvant prendre les valeurs suivantes : '<', '>', '≤', '≥', '≠', '='.
Valeur	Représente une valeur atomique
Echelle	Représente l'échelle utilisée pour mesurer la valeur d'une propriété.
/	Le symbole «/» définit une alternative.
*	Le symbole «*» signifie que l'élément est optionnel ou répétitif.

Tableau 3.1. Description des composants de la requête.

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

3.4.2. Sujet d'une requête

Pour faciliter le traitement des requêtes globales nous adoptons la notion du sujet de requête [11]. Nous définissons le sujet d'une requête comme la liste des concepts utilisés pour exprimer cette dernière. Ce sujet est envoyé à l'agent annuaire au lieu d'envoyer la requête complète. L'agent annuaire fait les comparaisons entre les expertises des agents d'information et le sujet de requête pour identifier les agents concernés par le traitement de cette dernière.

Exemple 2:

soit C_1, C_2, C_3 trois concepts de l'ontologie globale, $R_1 (C_1, C_2), R_2 (C_2, C_3)$ deux relations entre ces concepts, $C_1.P_j, C_2.P_i$ et $C_3.P_k$ trois propriétés. Supposant que l'utilisateur pose la requête suivante :

$Q = \text{SELECT } C_2.P_i \text{ WHERE } R_1 (C_1, C_2) \text{ AND } R_2 (C_2, C_3) \text{ AND } C_1.P_j < 20 \text{ g AND } C_3.P_k = 3 \text{ Kg}$
--

Le sujet de la requête Q est le suivant: $\text{Sujet}(Q) = \{C_1, C_2, C_3\}$.

soit AI_1, AI_2, AI_3 trois agents d'information qui coopèrent pour le traitement des requêtes globales. L'expertise de ces agents est défini comme suit :

- $\text{Expertise}(AI_1) = \{C_1, C_2, C_8\}$.
- $\text{Expertise}(AI_2) = \{C_4, C_5\}$.
- $\text{Expertise}(AI_3) = \{C_2, C_3, C_5, C_6\}$.

En comparant le sujet de la requête Q et les expertises des trois agents nous constatons que seulement les agents d'information AI_1, AI_3 peuvent contribuer dans le traitement de cette dernière. Car chacun des deux agents fournit au moins des données sur un des concepts du sujet de la requête contrairement à l'agent AI_2 .

Après cette description des éléments nécessaires pour la formulation d'une requête, nous présentons dans ce qui suit ; le protocole adopté pour le traitement des requêtes globales.

3.5. Spécification du protocole coopératif de traitement des requêtes globales

Le traitement des requêtes dans notre architecture est réalisé selon un processus coopératif basé sur les interactions entre les différents agents de l'architecture. L'utilisateur pose ses

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

requêtes en utilisant les termes de l'ontologie globale. L'agent d'information récepteur de la requête traite cette dernière selon deux chemins.

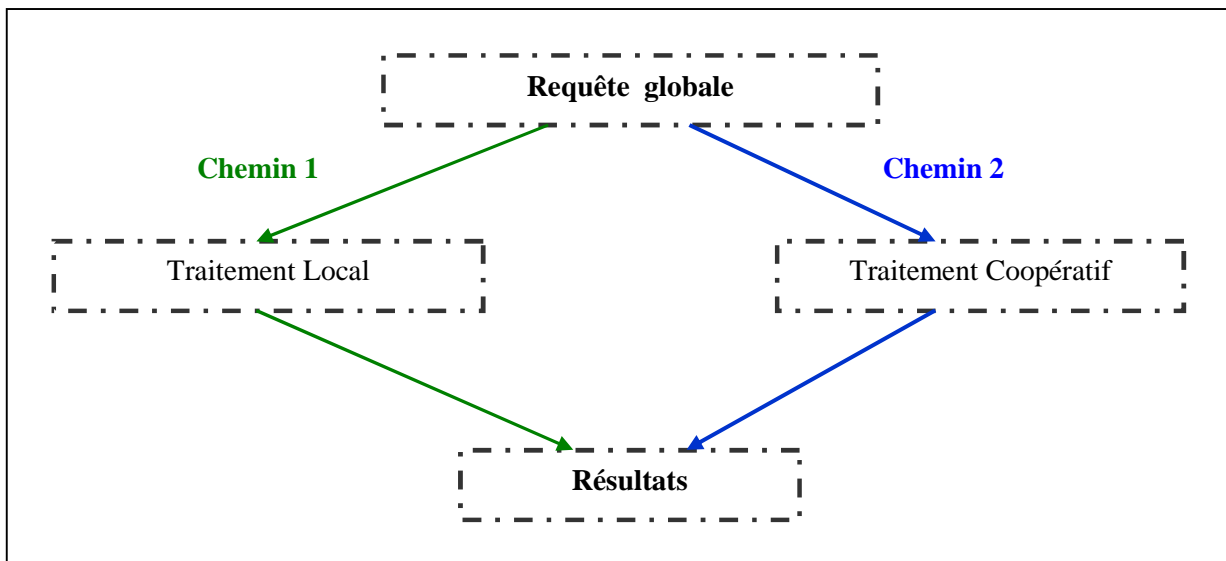


Figure 3.9. Traitement des requêtes globales.

3.5.1. Traitement locale de la requête

L'agent d'information récepteur de la requête étudie initialement la possibilité de traiter localement cette dernière. Pour cela il exploite premièrement le cache sémantique pour vérifier si cette requête est déjà traitée. Dans le cas contraire il procède à une exploitation de sa base de connaissance et il vérifie si tous les termes de la requête sont des éléments de son ontologie locale. Dans ce cas, l'agent décide de traiter la requête localement. Cette requête peut subir une autre décomposition (au niveau du planificateur local) selon la distribution des données dans les sources d'information car chaque agent d'information est responsable de l'extraction des informations à partir d'un ensemble de sources d'information via les agents adaptateurs de chaque source. Cette décomposition est réalisée en utilisant les mappings définis dans la base de description des schémas locaux. La décomposition dans ce niveau ne fait pas l'objet de notre travail. Nous supposons alors que l'AI envoie directement la requête aux AA concernés.

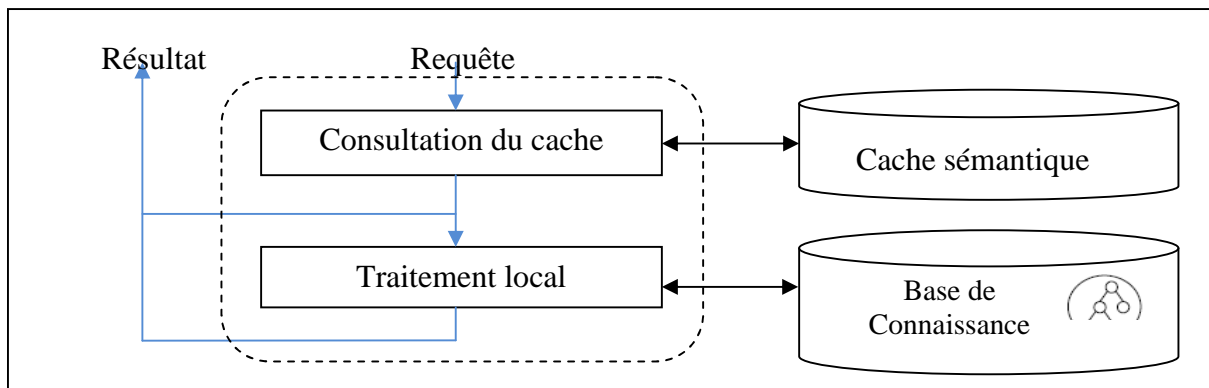


Figure 3.10. Traitement local.

3.5.2. Traitement coopératif de la requête

Si l'agent d'information récepteur de la requête détecte quelques éléments de la requête (concepts, relations ou propriétés) qui ne figurent pas dans son ontologie locale, il déclenche un processus coopératif pour impliquer les autres agents d'information dans le traitement. Comme le montre la figure 3.11 notre protocole inclut trois phases : *la phase d'initialisation, la phase des traitements locaux de la requête, la phase de recomposition des résultats.*

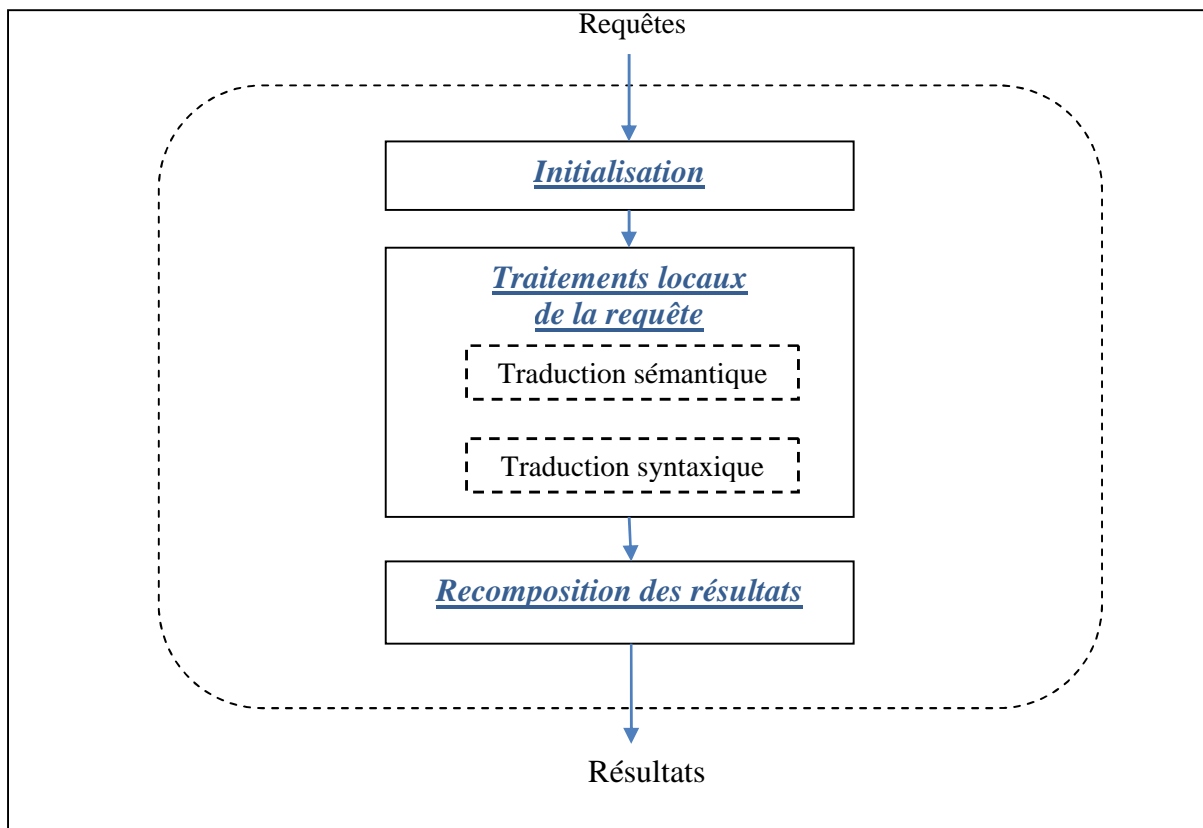


Figure 3.11. Etape du protocole de coopération.

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

3.5.2.1. La phase d'initialisation

L'AI récepteur de la requête extrait le sujet de cette dernière et envoie ce sujet à l'agent annuaire pour récupérer la liste des AI qui peuvent contribuer dans le traitement. L'agent annuaire consulte son registre d'expertise et compare le sujet de la requête avec les expertises des agents et ne sélectionne que les agents pertinents.

3.5.2.2. La phase des traitements locaux de la requête

L'agent récepteur de la requête diffuse cette dernière aux agents d'information sélectionnés par l'agent annuaire dans l'étape précédente. Chaque AI cerne et traite la sous-requête qui le concerne selon deux étapes :

➤ *Traduction sémantique:*

Cette traduction est réalisée au niveau du moteur de requêtes globales. Elle consiste à faire des substitutions entre les noms d'éléments de l'ontologie globale qui composent la requête et leurs équivalents dans l'ontologie locale de l'AI. La traduction est basée sur l'exploitation des Mappings définis dans la base de connaissance de chaque AI. Les éléments de la requête qui sont traduits sont les termes qui composent la sous-requête qui sera traitée localement par l'AI. Par conséquent, chaque AI ne conserve que les éléments de la requête sur lesquels il dispose des connaissances. Cette sous-requête cernée est complétée par les propriétés clés des concepts pour faciliter la fusion des résultats dans la phase suivante. Ces propriétés clés sont extraites de l'ontologie locale de chaque agent d'information. Comme c'est mentionné dans la section précédente la décomposition des requêtes au niveau du planificateur local de chaque agent ne fait pas l'objet de notre travail. Dans ce cas nous supposons que chaque agent d'information envoie directement la sous-requête cernée dans cette étape de traduction sémantique aux agents adaptateurs concernés.

Exemple 4:

Nous considérons la requête et les agents d'information présentés dans l'exemple 2.

```
Q = SELECT C2.Pi WHERE R1 (C1,C2) AND R2 (C2,C3) AND C1.Pj < 20 g AND C3.Pk = 3 Kg
```

D'après l'agent annuaire les agents d'information qui peuvent contribuer dans le traitement de cette requête sont AI₁ et AI₃. Les répertoires de mappings de ces deux agents sont les suivants :

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

Répertoire de mappings de AI1
<pre><owl: Class rdf: about="&ontologie-globale;C1"> <owl: <i>equivalentClass</i> rdf: resource="&ontologies-locale1;C5"/> </owl: Class></pre>
<pre><owl: Class rdf: about="&ontologie-globale;C2"> <owl: <i>equivalentClass</i> rdf: resource="&ontologies-locale1;C8"/> </owl: Class></pre>
<pre><owl: DatatypeProperty rdf: about="&ontologie-globale; C1.Pj"> <owl: <i>equivalentProperty</i> rdf: resource="&ontologies-locale1; C5.Pj "/> </owl: DatatypeProperty></pre>
<pre><owl: DatatypeProperty rdf: about="&ontologie-globale; C2.Pi"> <owl: <i>equivalentProperty</i> rdf: resource="&ontologies-locale1; C8.Pi "/> </owl: DatatypeProperty></pre>
<pre><owl: ObjectProperty rdf: about="&ontologie-globale; R1"> <owl: <i>equivalentProperty</i> rdf: resource="&ontologies-locale1; R9 "/> </owl: DatatypeProperty></pre>

Tableau3.2. table de mappings de l'agent AI1

Répertoire de mappings de AI3
<pre><owl: Class rdf: about="&ontologie-globale;C2"> <owl: <i>equivalentClass</i> rdf: resource="&ontologies-locale2;C7"/> </owl: Class></pre>
<pre><owl: Class rdf: about="&ontologie-globale;C3"> <owl: <i>equivalentClass</i> rdf: resource="&ontologies-locale1;C9"/> </owl: Class></pre>
<pre><owl: DatatypeProperty rdf: about="&ontologie-globale; C2.Pi"> <owl: <i>equivalentProperty</i> rdf: resource="&ontologies-locale1; C7.Pi "/> </owl: DatatypeProperty></pre>
<pre><owl: DatatypeProperty rdf: about="&ontologie-globale; C3.Pk"> <owl: <i>equivalentProperty</i> rdf: resource="&ontologies-locale1; C9.Pk "/> </owl: DatatypeProperty></pre>
<pre><owl: ObjectProperty rdf: about="&ontologie-globale; R2"> <owl: <i>equivalentProperty</i> rdf: resource="&ontologies-locale1; R5 "/> </owl: DatatypeProperty></pre>

Tableau 3.3. Table de mappings de l'agent AI3

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

L'agent d'information AI₁ consulte son répertoire de mappings et il génère la sous-requête suivante :

Sous-requête cernée
Q1 = SELECT C2.Pi WHERE R1 (C1,C2) AND C1.Pj<20g
Sous-requête après traduction et ajout des propriétés clés
Q1 = SELECT C8.Pi, C5.cle,C8.cle WHERE R9 (C5,C8) AND C5.Pj<20g

L'agent d'information AI₃ consulte son répertoire de mappings et il génère la sous-requête suivante :

Sous-requête cernée
Q2= SELECT C2.Pi WHERE R2 (C2,C3) AND C3.Pk=3Kg
Sous-requête après traduction et ajout des propriétés clés
Q2= SELECT C7.Pi,C7.cle,C9.cle WHERE R5 (C7,C9) AND C9.Pk=3Kg

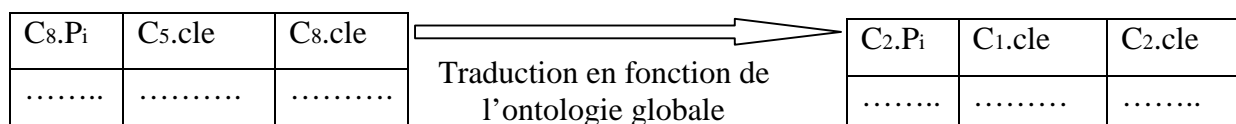
➤ *Traduction syntaxique :*

L'hétérogénéité syntaxique est résolue au niveau des adaptateurs. En effet, ces derniers adaptent les sous-requêtes de l'étape précédente au langage de requête des sources. Les résultats renvoyés par la source sont alors acheminés vers l'agent d'information qui va construire une réponse pour sa sous-requête.

A l'issue de cette étape chaque AI envoie les résultats du traitement de sa sous requête à l'AI récepteur de la requête globale. Ces résultats sont exprimés en fonction de l'ontologie locale de chaque agent. Alors, avant de les envoyer à l'agent récepteur de la requête globale chaque agent doit traduire ses résultats en fonction de l'ontologie globale pour faciliter la construction d'une réponse homogène. Cette traduction est réalisée en utilisant les tables de mappings.

Exemple 4 :

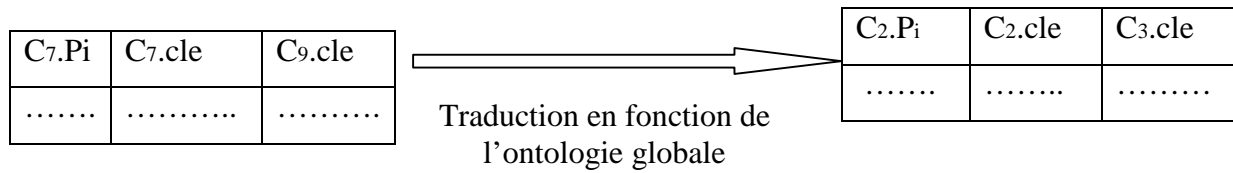
Nous supposons que les résultats de l'agent AI₁ sont présentés dans la table suivante :



Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

Les résultats de l'agent AI3 sont présentés dans la table suivante :



3.5.2.3. La phase de recomposition des résultats

Concernant la construction d'une réponse globale nous considérons que cette réponse est obtenue par des jointures naturelles et des unions entre les résultats générés par les différents agents d'information intégrés dans le traitement de la requête globale. (En réalité la construction de la réponse globale est plus compliquée qu'une simple jointure entre les différentes tables. Ce point fera l'objet de travaux futurs)

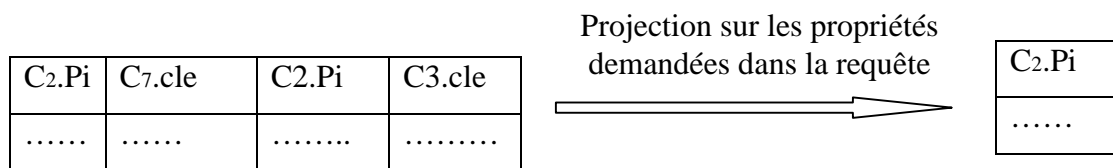
Par exemple, la jointure entre les tables de résultats de l'agent AI1 et l'agent AI2 donne la table suivante :

C2.Pi	C1.cle	C2.cle	C3.cle
....

Le résultat final est obtenu via une projection sur les propriétés demandées dans la requête.

Dans la requête de notre exemple une seule propriété C2.Pi est demandée:

Q = **SELECT** C2.Pi **WHERE** R1 (C1,C2) **AND** R2 (C2,C3) **AND** C1.Pj < 20 g **AND** C3.Pk = 3 Kg



3.6. Conclusion

Nous avons présenté dans ce chapitre une architecture pour la coopération des sources d'informations distribuées et hétérogènes qui présente les spécificités suivantes :

- Elle repose sur le paradigme multi-agents et une approche ontologique hybride pour la résolution des conflits au niveau sémantique.
- Elle est générique est indépendante des types de sources d'information à interroger.

Chapitre 3

Une architecture multi-agents pour la mise en œuvre de l'interopérabilité sémantique dans les SIC

- Elle repose sur une décomposition dynamique des requêtes selon un protocole coopératif entre les différents agents.

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

Chapitre 4

Etude de Cas et perspectives d'implémentation :

4.1. Introduction

Dans le chapitre précédent nous avons présenté une architecture multi-agents dédiée à la coopération des sources d'information distribuées et hétérogènes. Trois types d'agents sont utilisés pour la construction de cette dernière. Ces agents jouent des rôles complémentaires afin de traiter les requêtes des utilisateurs d'une manière transparente. Le problème d'interopérabilité au niveau sémantique est résolu via l'utilisation d'une approche ontologique hybride.

Afin de comprendre plus aisément les principes de l'architecture proposée, nous présentons une étude de cas et une simulation du fonctionnement globale de l'architecture qui feront l'objet du présent chapitre. Notre étude de cas est inspirée des exemples présentés par D. Benslimane [4] dans le cadre de la proposition d'une architecture multi-agents pour l'interopérabilité des systèmes d'information géographique. Nous avons opté pour la coopération de trois sources d'information issues du domaine touristique et du domaine de la location des maisons. Ces deux domaines sont en pleine évolution dans nos jours. Cette évolution est reflétée par le nombre important des agences immobilières ou bien touristiques qui proposent des services différents. Actuellement, et avec l'évolution des technologies de communication et de l'internet des centaines de sources d'informations (sites, base de données...Etc.) sont consultables facilement à travers le Net. Chaque source propose des services variés dont l'objectif est de faciliter la recherche aux clients. Faire coopérer ces sources d'information fournit un bon support pour répondre aux besoins complexes des utilisateurs.

4.2. Etude de cas : Coopération des sources d'information du domaine touristique et du domaine de la location des maisons.

Avec l'évolution des conditions de vie durant ces dernières années, les besoins des clients dans le domaine de la location des maisons sont aussi en pleine évolution. Ces clients n'ont plus besoin de trouver une maison avec un petit loyer mais ils cherchent des maisons qui respectent leurs besoins selon un ensemble de critères. Généralement, les critères exigés par les clients ne concernent pas seulement les caractéristiques d'une maison (loyer, surface, nombre de pièces... etc.) mais aussi celles des auxiliaires (piscine, jardin...etc.) ou bien celles des communes où ces maisons se trouvent (par exemple la commune où il ya un bon hôpital, une petite population, les beaux paysages, ...etc.). Dans ce cadre, nous constatons que les requêtes dans ce domaine sont plus complexes qu'une simple recherche traditionnelle dans la page des petites annonces d'un journal. Par contre les besoins des clients s'orientent vers des critères liés au domaine touristique qui ne sont pas gérés par les sources d'information du domaine de la location de maison. Pour cela nous proposons la coopération de trois sources d'information issues du domaine touristique et du domaine de la location de maison afin de satisfaire les besoins des clients. Dans ce qui suit nous présentons les parties des schémas des sources d'information qui nous intéressent. Ces schémas sont inspirés de quelques sites internet [65] [66] [67] [68] qui fournissent ces types de services pour les clients.

Pour faciliter la construction des ontologies nos sources d'information sont présentées sous forme de trois bases de données relationnelles simples. Chaque base est décrite par un ensemble de tables. Chaque table est identifiée par une clé primaire et éventuellement par des clés étrangères. Pour raison de simplification, nos clés primaires sont de type entier (i.e. les instances des tables sont identifiées par des numéros 1, 2,3...etc.).

➤ **Une base de données des petites annonces d'un journal :** Cette base contient les petites annonces envoyées par des personnes qui veulent louer leurs maisons. Dans notre contexte nous présentons la partie du schéma de la base de données qui nous intéresse :

- | |
|--|
| <ul style="list-style-type: none">- Petite-annonce (<u>Num-annonce</u>, loyer, <u>Num-maison</u>, date-libre).- Maison (<u>Num-maison</u>, Surface, nbr-pieces, <u>Num-commune</u>, <u>Num-prop</u>).- Propriétaire (<u>Num-prop</u>, Nom-prop, Prenom-prop, Num-telephone).- Commune (<u>Num-commune</u>, Nom-commune, Surface-commune, Population). |
|--|

- **Une base de données d'un site immobilier :** C'est une base de données relationnelle. Elle contient une description détaillée des maisons et surtout une description des auxiliaires associées à chaque maison, tel que les piscines et les jardins, elle fournit même des photos qui facilitent la recherche aux clients. La partie du schéma qui nous intéresse et la suivante :

- Annonce-de-location (Num-location, loyer, Num-maison, date_libre).
- Maison (Num-maison, Surface, nbr-pieces, Num-prop, Num-commune).
- Propriétaire (Num-prop, Nom-prop, Prenom-prop, Num-telephone).
- Commune (Num-commune, Nom-commune, Surface-commune, Population).
- Auxiliaire (Num-aux, type-aux, Surface-aux).

- **Une base de données d'un site touristique :** Cette base est utilisée dans le cadre d'un site touristique, elle fournit des informations sur les services disponibles dans les différentes villes. L'objectif principal de ce type de site est d'attirer les touristes vers les différentes villes d'un pays via la description des beaux monuments et des services généraux qui facilitent le séjour des touristes dans chaque ville. Ce type de base de données touristique semble être très intéressant dans le cadre de la location des maisons car il facilite le choix d'une maison selon les critères de la ville où elle se trouve. Une partie du schéma de cette table est la suivante :

- Ville (Num-ville, Nom-ville, Surface-ville, Population).
- Restaurant (Nom-restaurant, Type, Num-propr, Num-ville).
- Propriétaire (Num-propr, Nom, Prenom, Age, Nationalité).

4.3. Outils d'implémentation utilisés

Cette section est dédiée à la présentation des outils utilisés pour simuler le fonctionnement global de notre architecture selon l'étude de cas proposée.

4.3.1. La plate forme JADE

Afin de simuler les idées et les concepts de notre architecture, il est pratique d'utiliser une plateforme Multi-agents. Nous avons choisi la plateforme JADE. Cette plate forme (Java Agent Development framework) [52] est un environnement de développement d'agents implanté totalement dans le langage java. Elle facilite la mise en place d'un SMA tout en répondant aux spécifications de FIPA à travers un ensemble d'outils.

Il est important de mentionner que JADE comprend tous les composants qui permettent la gestion de la plateforme selon FIPA. Ses composants se résument en :

- ACC « Agent Communication Channel » : c'est l'agent qui fournit la route pour les interactions de base entre les agents dans et hors de la plateforme.
- AMS « Agent Management system » : c'est l'agent qui exerce le contrôle de supervision sur l'accès et l'usage de la plateforme.
- DF « Directory Facilitator » : c'est l'agent qui fournit un service de pages jaunes à la plateforme. Cet agent dispose d'un tableau dans lequel il inscrit les agents qui peuvent fournir des services à d'autres agents. Ainsi, un agent qui cherche la réalisation d'une tâche va seulement contacter le DF pour lui remettre la liste des noms des agents qui peuvent satisfaire ses besoins.

Pour traiter les requêtes globales, les agents doivent s'échanger des messages. L'utilisation de FIPA-ACL [64] comme langage de communication inter-agents dans JADE nous a permis d'implémenter le protocole de coopération que nous suggérons. Un message ACL comprend plusieurs éléments. Nous présentons dans le tableau 4.1. Les champs qui nous intéressent.

Elément	Signification
Performatif	type de l'acte communicatif.
Sender	l'émetteur du message.
Receiver	le destinataire du message.
Content	le contenu du message (l'information transportée par la performative).
Ontology	le nom de l'ontologie utilisé pour donner un sens aux termes utilisés dans le <i>content</i> .

Tableau 4.1. Structure d'un message FIPA-ACL. [61]

4.3.2. Protégé

La construction des ontologies dans notre étude de cas est réalisée avec l'éditeur d'ontologies protégé. Protégé est un outil d'édition, visualisation et contrôle d'ontologies. Nos bases de données sont relationnelles. Alors, chaque table (relation) est représentée par une classe **owl:Class**. Les attributs des classes sont représentés par **owl:DatatypeProperty**. Si l'attribut est contraint d'être clé primaire alors la caractéristique fonctionnelle sera ajoutée. Nous utilisons aussi **owl:ObjectProperty** pour représenter les attributs qui sont sous une contrainte de clé étrangère.

4.4. Simulation des agents avec JADE

Dans cette section nous présentons les différents agents de notre architecture selon l'étude de cas proposée. La figure 4.1., présente l'implantation de l'architecture en utilisant JADE, où chaque agent est lancé dans un hôte séparé et dans un conteneur différent (i.e. un environnement distribué). Cinq agents sont utilisés. La distribution de ces agents selon les niveaux de l'architecture est la suivante :

4.4.1. Niveau de coopération

D'après la description des différentes sources d'information nous avons décidé d'utiliser deux agents d'information. Chaque agent est responsable de l'extraction des informations à partir des sources d'information de un des deux domaines.

- **AI1** pour le domaine de la location des maisons ;
- **AI2** pour le domaine touristique.

4.4.2. Niveau des adaptateurs

Trois agents adaptateurs sont utilisés dans ce niveau. Chacun facilite l'accès à une des trois sources d'information :

- **AA1** pour la base de données des petites annonces ;
- **AA2** pour la base de données du site immobilier ;
- **AA3** pour la base de données du site touristique.

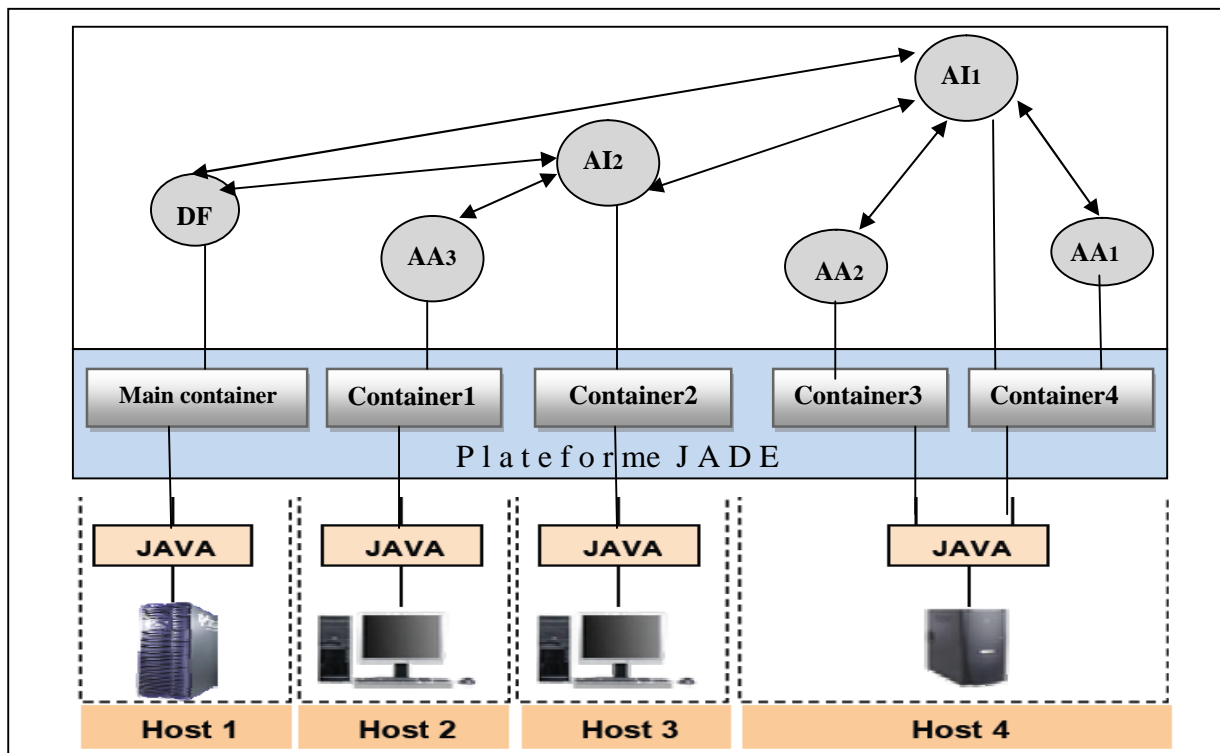


Figure 4.1 lancement des agents dans des containers différents.

4.4.3. L'agent annuaire

L'utilisation de JADE nous a permis de bénéficier de beaucoup d'outils qui nous ont facilité la simulation des services fournis par notre architecture. Par exemple le DF (Directory Facilitator) modélise quelques aspects de l'agent annuaire. Pour cela nous supposons l'utilisation de l'agent DF comme agent annuaire de notre architecture.

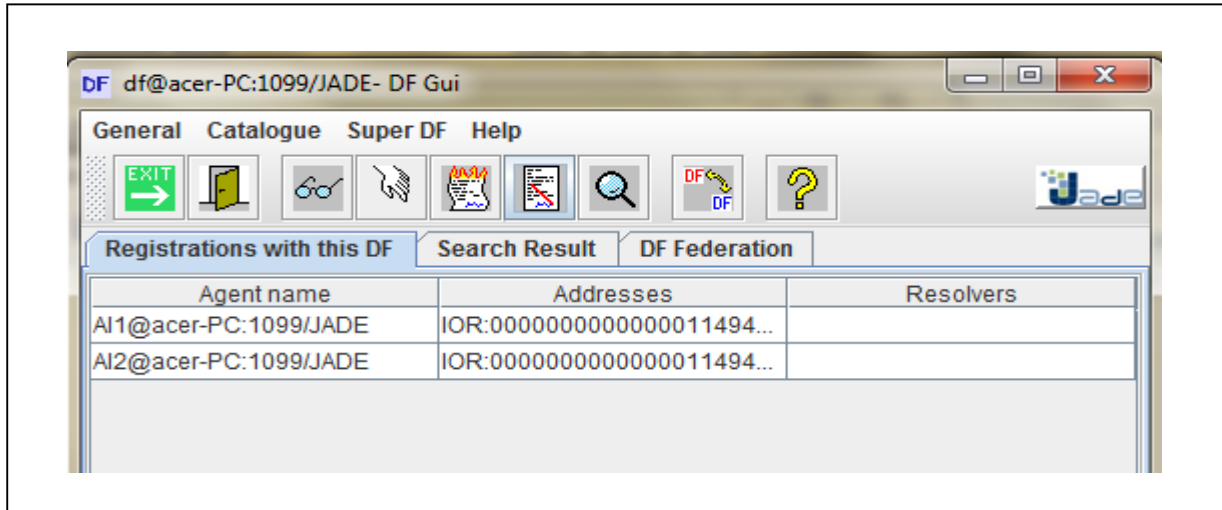


Figure 4.2. Inscription des agents d'information AI1 et AI2 au près du DF.

La Figure 4.3., montre les agents AI1, AI2, AA1, AA2 et AA3 via GUI (Globally Unique Identifier) de la plateforme JADE.

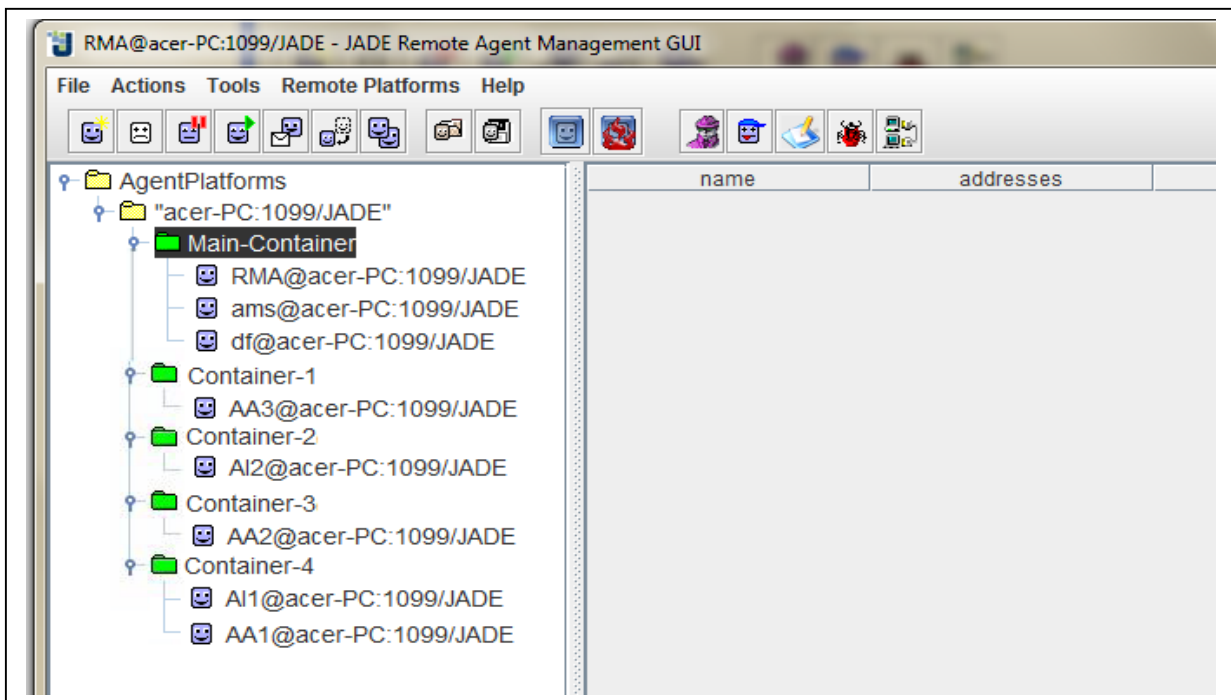


Figure 4.3. Simulation des agents avec GUI de JADE.

4.5. Construction des ontologies

Avant de simuler les interactions des agents pendant le traitement des requêtes posées par les utilisateurs nous présentons dans cette section les ontologies utilisées pour décrire la sémantique des trois sources d'information et les tables de mappings des deux agents d'information AI₁ et AI₂. (Pour des raisons de simplification nous présentons seulement les classes et les relations qui constituent les ontologies).

4.5.1. Ontologie locale 1

Cette ontologie décrit la sémantique des sources d'information du domaine de la location de maisons. Les éléments (classes, relations et propriétés) de cette ontologie sont définis à partir des schémas relationnels des deux sources. Les classes et les relations qui composent cette ontologie sont présentées dans la figure 4.4.

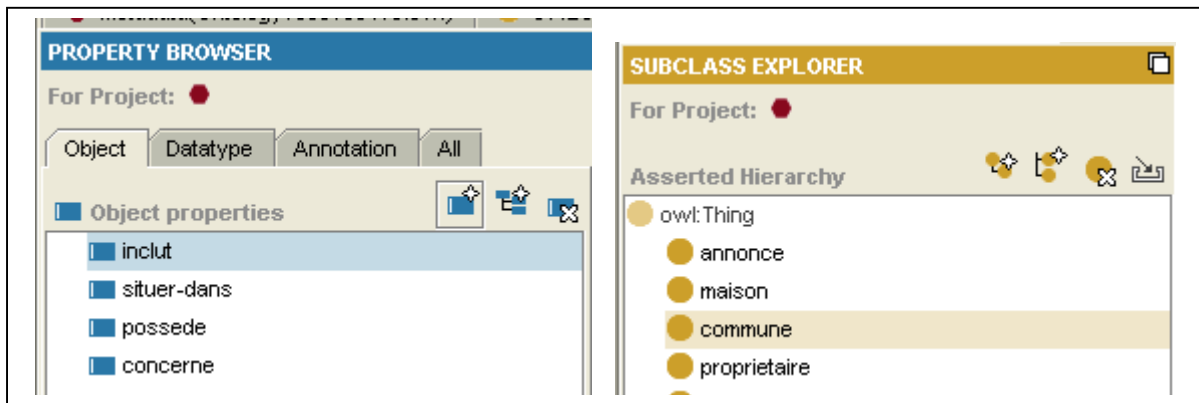


Figure4.4. Classes et relations de l'ontologie1.

4.5.2. Ontologie locale 2

Cette ontologie décrit la sémantique de la source d'information du domaine touristique. Les éléments (classes, relations et propriétés) de cette ontologie sont définis à partir du schéma relationnel de la source. Les classes et les relations sont présentées dans la figure 4.5.

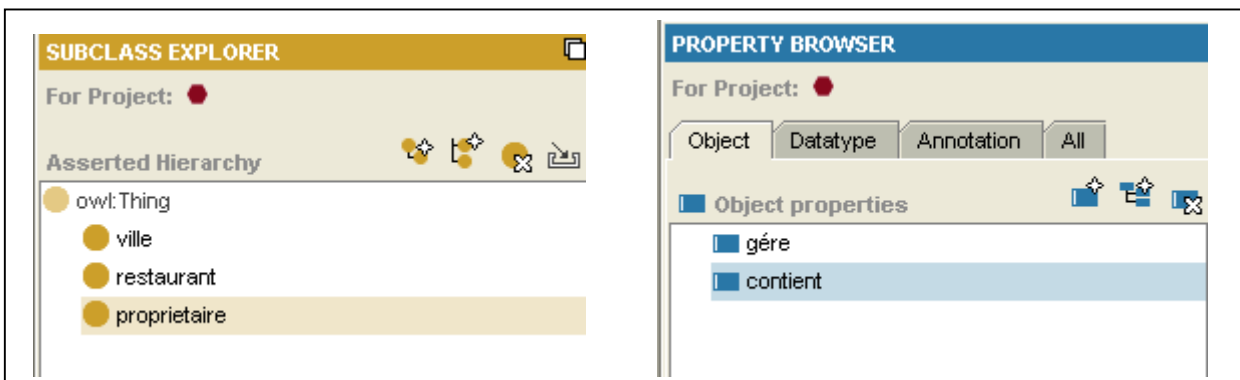


Figure4.5. Classes et relations de l'ontologie2.

4.5.3. Ontologie globale

L'ontologie globale est construite selon les étapes du processus présenté dans le chapitre 3.

Etape1 : Analyse des ontologies locales

D'après les deux ontologies locales de l'étape précédente, nous constatons une relation de synonymie entre le concept « commune » de l'ontologie1 et le concept « ville » de l'ontologie 2. Les propriétés de ces deux concepts sont aussi des synonymes ; par exemple : la propriété « Nom-commune » du concept « commune » et synonyme de la propriété « Nom-ville » du concept ville. Aussi une relation d'homonymie est détectée entre le concept « propriétaire » d'une maison dans l'ontologie1 qui est différent du concept « propriétaire » d'un restaurant dans l'ontologie 2.

Etape2 : Résolution des conflits sémantiques

Dans cette étape l'ontologie globale est construite via la fusion des deux ontologies locales avec la prise en compte des conflits détectés dans l'étape précédent. Par exemple, les deux concepts synonyme « commune, ville» sont présentés par un seul concept « cité » dans l'ontologie globale. Les deux concepts « propriétaire » homonyme dans les deux ontologies sont renommés dans l'ontologie globale par deux concepts différents. « Proprietaire-maison » pour la classe propriétaire de l'ontologie1 et « propriétaire-restaurant » pour la classe propriétaire de l'ontologie2.

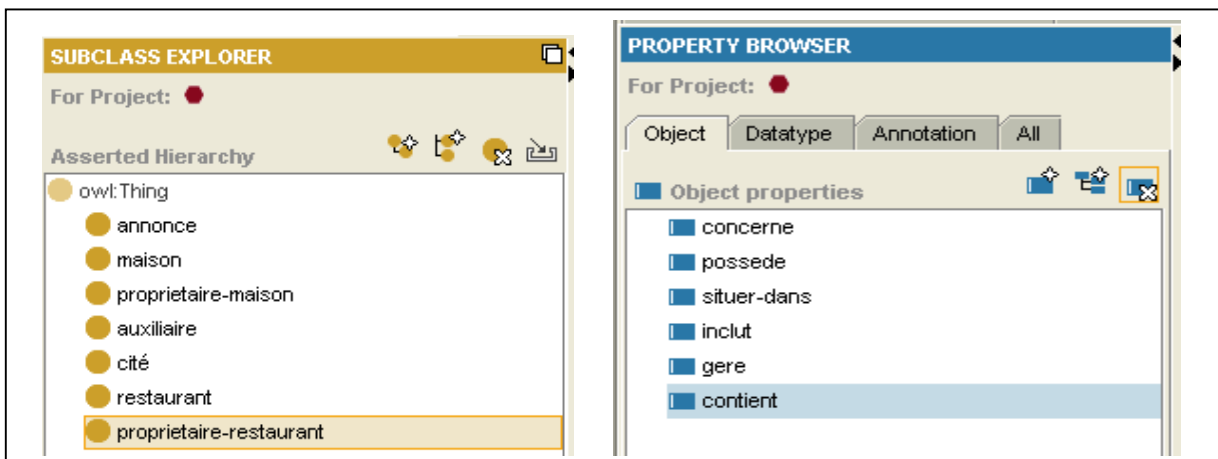


Figure4.6. Classes et relations de l'ontologie globale.

4.6. Construction des tables de mappings

D'après la description de l'architecture du système deux agents d'information sont utilisés dans le niveau de coopération. Alors, la base de connaissances de chaque agent doit encapsuler en

plus des ontologies locales une table de mappings qui contient les correspondances avec l'ontologie globale.

Un extrait du contenu des tables de mappings des deux agents AI₁ et AI₂ est résumé dans le tableau 4.2.

Table de mappings de AI ₁
<pre><owl: Class rdf: about="&ontologie-globale;cité"> <owl: <i>equivalentClass</i> rdf: resource="&ontologies-locale1;commune"/> </owl: Class></pre>
<pre><owl: Class rdf: about="&ontologie-globale; propriétaire-maison "> <owl: <i>equivalentClass</i> rdf: resource="&ontologies-locale1; propriétaire "/> </owl: Class></pre>
<pre><owl: DatatypeProperty rdf: about="&ontologie-globale; Num-cité"> <owl: <i>equivalentProperty</i> rdf: resource="&ontologies-locale1; Num-commune"/> </owl: DatatypeProperty></pre>
Table de mappings de AI ₂
<pre><owl: Class rdf: about="&ontologie-globale;cité"> <owl: <i>equivalentClass</i> rdf: resource="&ontologies-locale2;ville"/> </owl: Class></pre>
<pre><owl: Class rdf: about="&ontologie-globale; propriétaire-restaurant "> <owl: <i>equivalentClass</i> rdf: resource="&ontologies-locale2;propriétaire"/> </owl: Class></pre>
<pre><owl: DatatypeProperty rdf: about="&ontologie-globale; Num-cité"> <owl: <i>equivalentProperty</i> rdf: resource="&ontologies-locale2; Num-Ville "/> </owl: DatatypeProperty></pre>

Tableau 4.2. Extrait des tables de mappings des deux agents AI₁ et AI₂

4.7. Exemples de traitement de requêtes

Après la construction des différentes ontologies (locales et globale) nous présentons maintenant quelques exemples de traitement de requête en utilisant La grammaire présentée dans le chapitre 3. Le premier exemple concerne le traitement d'une requête par un seul agent d'information (traitement locale). Le deuxième concerne une requête qui dépasse les capacités de traitement de chaque agente (traitement coopératif). La communication entre les différents agents est réalisée via l'envoi des messages FIPA-ACL.

Exemple1: Nous supposons que l'agent AI₁ reçoit la requête Q₁ suivante :

Donner la liste des annonces qui concernent des maisons composée de quatre pièces. Cette requête peut être exprimée en utilisant les termes de l'ontologie globale comme suit :

```
SELECT annonce. Num-annonce WHERE concerne (annonce,maison)
AND maison.nbr-piece=4.
```

A la réception de cette requête l'agent AI₁ consulte premièrement le cache sémantique (nous supposons que AI₁ n'a pas traité précédemment cette dernière). Puis, il consulte sa base de connaissance où il constate que tous les termes (concepts, propriétés et relations) de la requête sont des éléments de son ontologie locale. AI₁ décide de la traiter localement et envoie cette dernière aux agents adaptateurs AA₁ et AA₂ responsables de l'extraction des informations à partir de la base de données des petites annonces et la base de données du site immobilier respectivement. La figure 4.7. Illustre un exemple de requête envoyée par l'agent AI₁ à l'agent AA₁. Cette requête est encapsulée dans un message ACL en utilisant la performatif *Query-ref*.

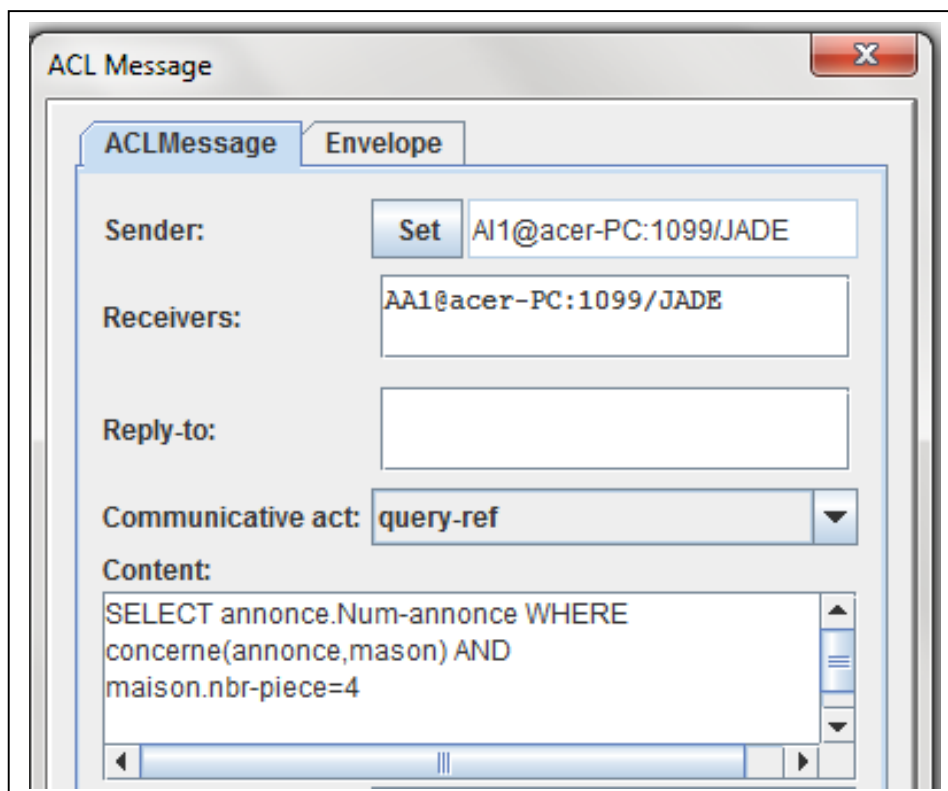


Figure 4.7. Requête envoyé par AI₁ à AA₁.

Les sources d'informations liées aux agents adaptateurs AA₁ et AA₂ sont relationnelles. Alors, AA₁ et AA₁ expriment la requête en utilisant par exemple SQL et envoient cette

dernière aux sources correspondantes pour l'extraction des informations. Après la récupération des résultats, AA1 et AA2 envoient ces résultats à AI2 dans un message FIPA-ACL en utilisant la performatif *Inform*. l'agent AA1 fusionne les résultats envoyés par les deux agents adaptateur pour construire une réponse globale.

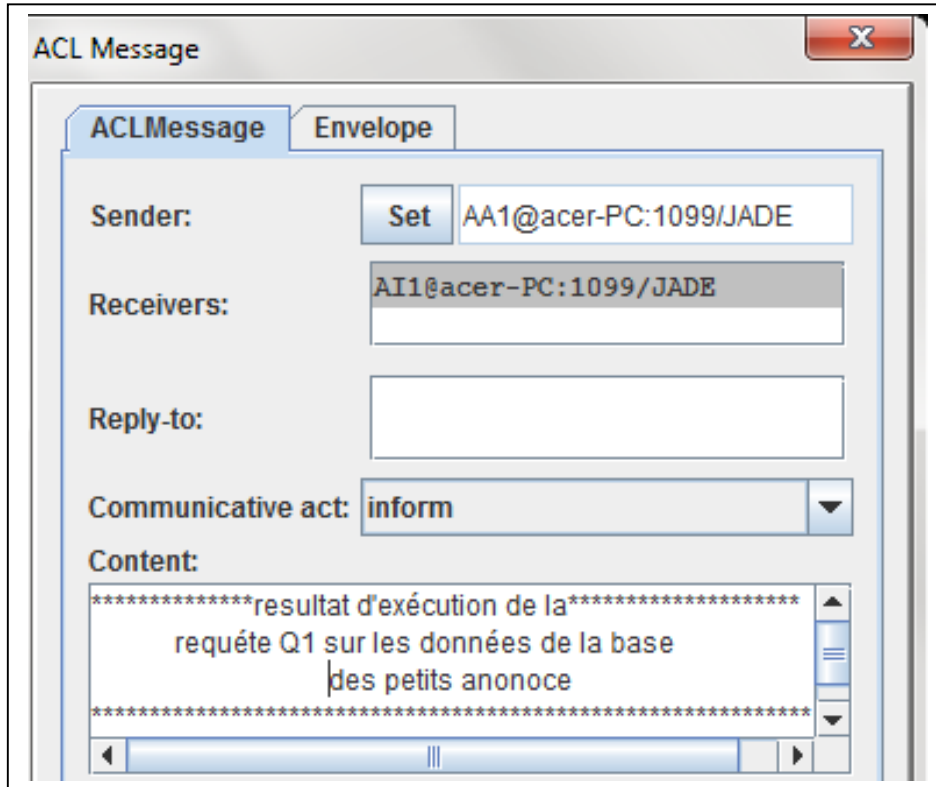


Figure 4.8. Résultats envoyés par AA1 à AI1.

Exemple2 : Nous supposons que l'agent AI₁ reçoit la requête Q₂ suivante :

Donner la liste des maisons composées de 4 pièces et qui se trouvent dans une commune qui contient des restaurants Italiens. Cette requête peut être exprimée en utilisant les termes de l'ontologie globale comme suit :

```
SELECT maison.Num-maison WHERE maison.nbr-piece= 4 AND
situer-dans (maison,cité) AND contient (cité, restaurant) AND restaurant.type= "italien".
```

Premièrement, l'agent AI₁ consulte le cache sémantique puis sa base de connaissance. Il constate que la requête dépasse ses capacités de traitement car il ya des termes (restaurant) qui ne possèdent pas des correspondants dans sa base de connaissances. Dans ce cas l'implication des autres agents d'informations dans le traitement est nécessaire. Dans ce qui suit, nous présentons les étapes de traitement de la requête Q₂ selon le protocole de coopération proposé dans le chapitre précédant.

Phase d'initialisation

AI₁ extrait le sujet de la requête et envoie ce dernier au DF pour récupérer la liste d'agents qui peuvent contribuer dans le traitement de cette requête.

Sujet (Q1)= { maison, cité, restaurant }.

Dans notre architecture nous avons deux agents d'information seulement. Alors, AI₂ est le seul agent qui peut contribuer dans le traitement de la requête. AI₁ envoie la requête à AI₂ pour que ce dernier cerne et traite la partie qui le concerne. La figure 4.9. Illustre la requête envoyée par AI₁ encapsulée dans un message ACL en utilisant la performatif *Query-ref*.

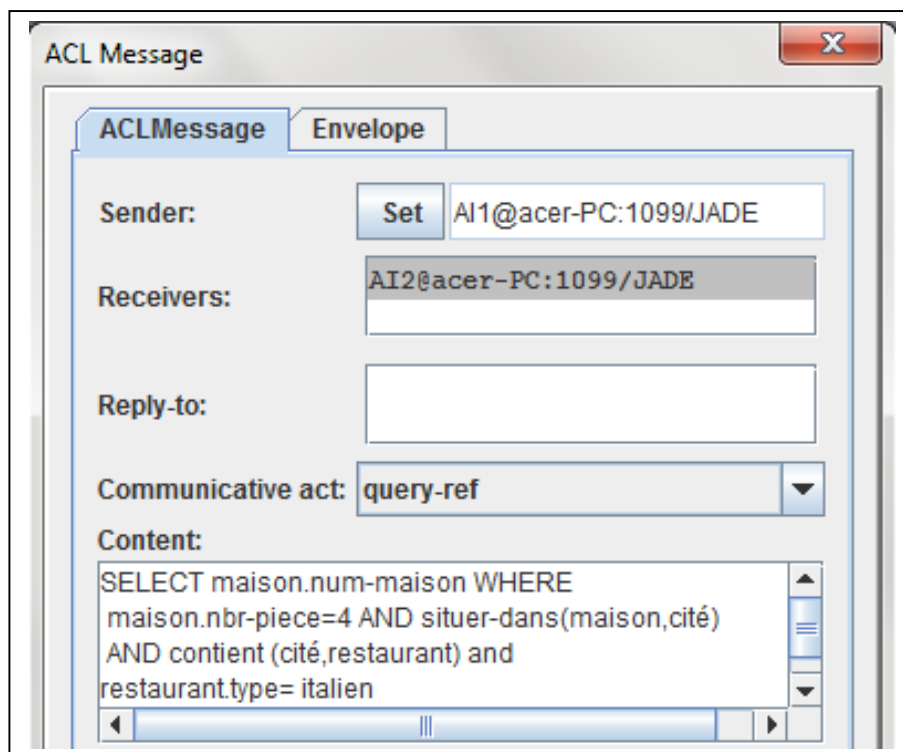


Figure4.9. Requête envoyée à l'agent AI₂.

Phase des traitements locaux de la requête

Chacun des agents AI₁ et AI₂ Cerne la partie de la requête qui le concerne en utilisant sa base de connaissances (ontologies locale et répertoire de mappings). Les parties cernées sont traduites en utilisant les termes des ontologies locales de chaque agent puis elles sont complétées par les clés des concepts pour faciliter la fusion des résultats dans l'étape suivante. Les sous-requêtes cernées par les deux agents AI₁ et AI₂ sont les suivantes :

- L'agent AI₁ fournit la liste des maisons composées de 4 pièces et pour chaque maison il donne aussi la ville où elle se trouve.

La partie de la requête cernée par AI1
SELECT maison.num-maison WHERE maison.nbr-piece= 4 AND situer-dans (maison,cité)
La sous requête après traduction et ajout des clés des concepts
SELECT maison.num-maison,commune.num-commune WHERE maison.nbr-piece= 4 AND situer-dans (maison, commune)

- L'agent AI2 fournit la liste des restaurants italiens et pour chaque restaurant il donne aussi la ville où il se trouve.

La partie de la requête cernée par AI2
SELECT * WHERE contient (cité, restaurant) AND restaurant.type= "italien".
La sous requête après traduction et ajout des clés des concepts
SELECT ville.num-ville, restaurant.nom-restaurant WHERE contient (ville, restaurant) AND restaurant.type= "italien".

Ces deux sous-requêtes générées dans cette phase sont envoyées aux adaptateurs adéquats. Par exemple, l'agent AI2 envoie sa sous-requête à l'agent adaptateur AA3 et l'agent AI1 envoie sa sous-requête aux agents adaptateurs AA2 et AA1. Elles sont encapsulées d'abord dans des messages FIPA-ACL en utilisant le performatif *Query-ref* comme dans l'exemple 1.

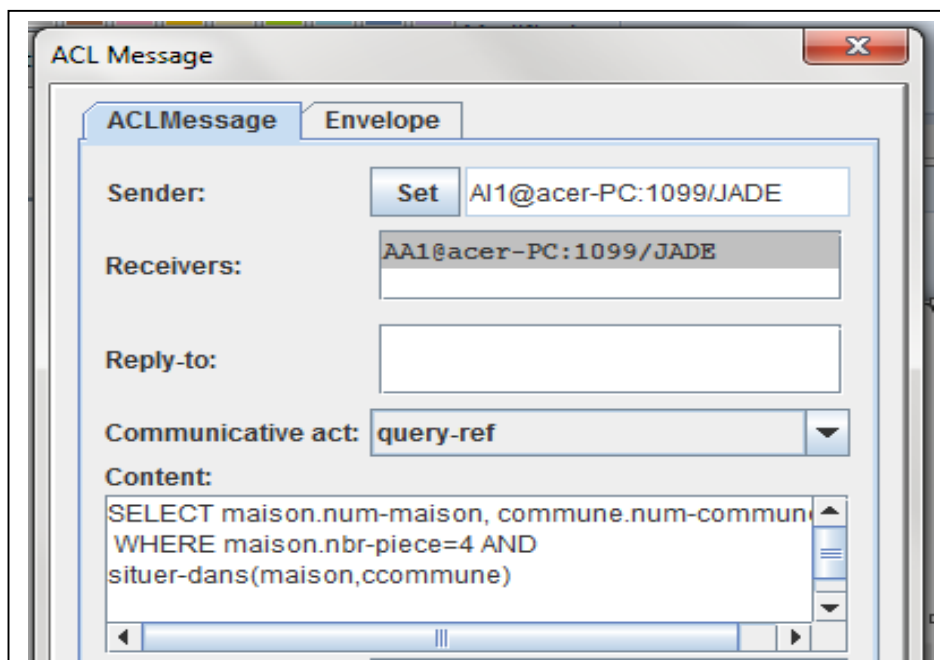


Figure 4.10. Requête envoyée par AI1 à AA.

Après l'exécution des sous-requêtes au niveau des sources d'information chaque agent adaptateur (AA1, AA2 et AA3) envoie son résultat à l'agent d'informations concerné (AA1 et

AA2 envoie leurs résultats à AI1 et AA3 envoie ses résultats à AI2). Par exemple l'agent AA1 envoie le résultat de traitement à AI1 dans un message ACL en utilisant la performatif *Inform*.

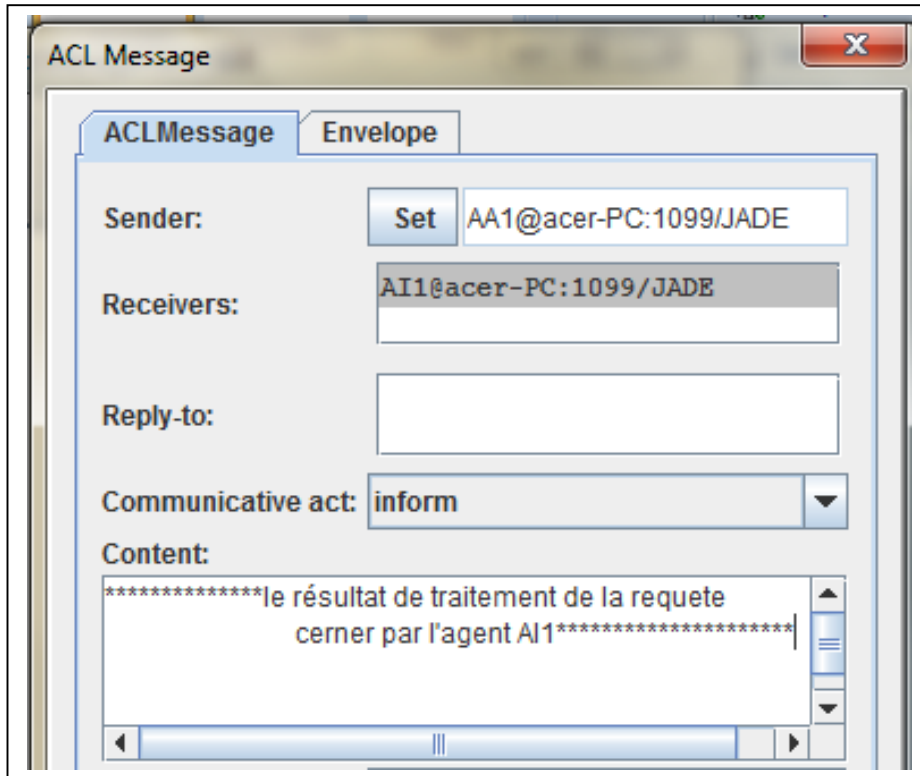


Figure4.11. Résultats envoyés par l'agent AA₁ à AI₁.

Après la réception des résultats envoyés par les différents agents adaptateurs, chaque agent d'information (AI₁ et AI₂) formule un résultat final pour sa sous-requête et traduit ce résultat en utilisant les termes de l'ontologie globale.

Phase de recomposition des résultats :

Dans notre exemple l'agent AI₁ est le récepteur de la requête donc il est responsable de la fusion de ses résultats avec les résultats envoyés par AI₂ pour formuler un résultat final. D'après la sous-requête traitée par chaque agent, nous constatons que le résultat final peut être obtenu par une jointure simple. En réalité, l'obtention d'un résultat nécessite la mise en œuvre de mécanismes plus évolués qu'une simple jointure surtout dans le cas des requêtes complexes qui nécessitent une bonne coordination et des plans d'exécution optimisés. Ce point fait l'objet de nos futurs travaux.

4.7. Conclusion

Nous avons présenté dans ce chapitre une étude de cas qui a illustré les différents concepts de notre approche. Nous avons en particulier, présenté la plate- forme JADE utilisée pour la simulation et les performatifs principaux utilisés pendant le protocole de coopération. Les différentes ontologies sont construites en utilisant le logiciel protégé. Le traitement coopératif de quelques requêtes est simulé en utilisant les différents composants de la plate forme jade qui facilitent la formulation des messages et la visualisation des interactions entre les différents agents.

Conclusion Générale

Nous rappelons que notre travail de recherche s'inscrit dans le cadre des systèmes d'information coopératifs. L'objectif principal est la proposition d'une approche Multi-Agents pour la mise en œuvre des mécanismes d'interopérabilité sémantique entre plusieurs sources d'information distribuées et hétérogènes. Notre ambition à travers ce travail est de proposer une architecture pour la modélisation des SIC et en même temps de consolider cette architecture avec les composants nécessaires pour répondre à la problématique d'interopérabilité sémantique pendant le traitement des requêtes globales.

Pour cela nous avons établi une étude du paradigme agent qui nous a permis de constater l'apport des SMA dans la modélisation des systèmes d'information coopératifs. Concernant les technologies permettant l'interopérabilité sémantique, nous avons constaté aussi que ces derniers ont pris une énorme importance de part leur création, normalisation et standardisation par de grands constructeurs et organisations. Dans cette étude, nous avons focalisé sur la notion d'ontologie qui est jugée comme l'un des moyens les plus efficaces pour la représentation formelle de la sémantique. Trois techniques d'interopérabilité sémantique basées ontologies sont présentées : l'alignement d'ontologie, la fusion d'ontologie et le mapping d'ontologies. Dans notre architecture, nous avons opté pour le mappings d'ontologies vue le nombre des architectures basées sur cette technique surtout dans le cadre de la coopération de sources d'information hétérogènes.

L'approche que nous avons proposée a conduit à une architecture générique dédiée à la coopération des sources d'information. Nos principales contributions sont résumées comme suit :

- La proposition d'une architecture composée d'un ensemble d'agents organisés dans des niveaux différents et selon des fonctionnalités complémentaires. Les agents d'information sont la base de cette architecture car ils sont responsables de la gestion du processus global de coopération selon une approche sémantique.
- L'adoption d'une approche ontologique hybride pour la résolution des conflits au niveau sémantique.

- La spécification d'un protocole de coopération pour traiter les requêtes globales posées par les utilisateurs.

Dans le but d'une continuité de nos travaux de recherche, nous envisageons de compléter la résolution de tous les conflits sémantiques pour assurer une transparence complète à l'utilisateur. Vu que l'évolutivité et la scalabilité sont essentielles dans le cadre des SIC, nous envisageons l'amélioration de l'architecture pour qu'elle soit capable de traiter ces deux propriétés surtout dans un contexte multi-domaines. Nos perspectives concernent aussi l'étendue du rôle des AI à la négociation des résultats provenant des différentes sources d'information et l'amélioration du protocole de coopération par des mécanismes de coordination. Enfin, nous envisageons l'implémentation complète et réelle de tous les composants de notre architecture et surtout ceux liées à l'aspect sémantique qui est le cœur de notre problématique car les simulations présentées dans ce chapitre ne couvrent pas le raisonnement via les connaissances sémantiques.

Références Bibliographiques

- [1] Al-ghamidi N., Saleh M., Eassa F., “Ontology-Based Query in Heterogeneous & Distributed Data Sources”, *International Journal of Electrical & Computer Sciences IJECS-IJENS* Vol.10, No.06, 2010, pp. 94-115.
- [2] Benslimane D., Yétongnon K., Chraïbi S., Leclercq E., Abdelwaheb E., “Les systèmes d’information coopératifs : le projet DECA”, *Technique et science informatiques*. Vol.19, No.07, 2000, pp. 1009-1043,
- [3] Benslimane S.M., MERAZI A., MALKI M., BENSABER D.A., “Ontology mapping for querying heterogeneous information sources”. *Evolutionary Engineering and Distributed Information Systems Laboratory, University of Sidi Bel Abbes*, 2007.
- [4] Benslimane D., Leclercq E., Yetongnon K., “Une architecture multi-agents pour l’interopérabilité des SIG”. *LE2I - Equipe Ingénierie Informatique, Université de Bourgogne*, 1999.
- [5] Boulanger D., Dubois G., “Objets et coopération de systèmes d’information, Ingénierie Objet : Concepts et techniques”, *InterEditions*, Mai 1997.
- [6] Brodie M.L., Seri C. “On intelligent and cooperative information Systems”, *A workshop summer, International Journal of cooperative information systems*, Juin 1992.
- [7] Bruijn J., Ehrig M., Feier C., Martin-Recuerda F., SCHARF F., “Semantic Web Technologies, trends and research in ontology-based systems”, chapter *Ontology Mediation, Merging, and Aligning*, pp 95–113, *WILEY*, 2006.
- [8] Bruijn J., Lausen H., Krummenacher R., Polleres A., Kifer M., Fensel D., “The Web Service Modeling Language WSML”, *Deliverable D16.1v0.2, Final Draft, WSMO project*, 2005.
- [9] Busse S., Kutsche R., Leser U., “Weber H. federated information systems: Concepts, terminologies and architectures”, *Technical Report Nr.99-9, Berlin University*, 1999.
- [10] Calvanese D., De Giacomo G., Lenzerini M., “A framework for ontology integration”, *Proceedings of the 1st Internationally Semantic Web Working Symposium (SWWS)* pp 303-317, 2001.
- [11] Celestin D. “Médiation de données sémantique dans SenPeer, un système pair-à-pair de gestion de données”, *Thèse de Doctorat, Université de Nantes*, 2007.

- [12] Chaib-Draa B., Jarras I., Moulin B. “Systèmes multi-agents : principes généraux et applications. Principes et architecture des systèmes multi-agents”, Chapitre 1, J. P. Briot, Y. Demazeau (dir), Paris, Hermès Science Publications, 2001.
- [13] [Chen D., Panetto H., “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, 2005.](#)
- [14] Corcho O., Fernandez-Lopez M., Gomez-Pérez A., “Methodologies, tools and languages for building ontologies: Where is their meeting point ?,” *Data & Knowledge Engineering* Vol.46, No.1 pp.,2003, 41-64, Elsevier,
- [15] Coté M., Chaib-draa B., Troudi N., “ NetSA: Une architecture multiagent réutilisable pour les environnement riche en information. *Information-Interaction-Intelligence* ”, Vol.1, No2, Cépadués-Edition, 2001.
- [16] [Crubézy M., Musen M. A., “Ontologies in Support of Problem Solving”, In Staab S. and Studer R., editors, *Handbook on ontologies*, pp. 321-342 Springer, 2003.](#)
- [17] [Dou D., McDermott D., and Qi P., “Ontology translation on the semantic web”. In *International Conference on Ontologies, Databases and Applications of Semantics \(ODBASE'03\)*, Catania, Italy, 2003.](#)
- [18] Ferber J., “Les systèmes multi-agents : vers une intelligence collective”, InterEditions, 1995.
- [19] Gaelle L., “État de l’art ontologies et intégration/fusion d’ontologies, 2002.
- [20] Ghawi R., “Ontology-based Cooperation of Information Systems”, Thèse de Doctorat, Université de Bourgogne, 2010.
- [21] Ghomari A.R. “Approche Méthodologique d’Acquisition de Connaissances Agrégées à base d’Agents cognitifs coopérants pour les systèmes d’aide à la décision stratégiques”, Thèse de Doctorat, INI, 2006.
- [22] Gleizes M.P., “Vers la résolution de Problèmes par émergence”, Habilitation à Diriger des Recherches, Université Paul Sabatier, Toulouse, 2004.
- [23] [Goh C.H. “Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources”, Phd Thesis, MIT, 1997.](#)
- [24] Gomez-Perez A., “Ontology Engineering”, Springer Verlag, 2003.
- [25] Gomez-Perez A., “ Evaluation of Ontologies. *International Journal of Intelligent Systems*”, Vol.16, No.3, 2001.
- [26] Gomez-Perez A., “ Développements récents en matière de conception, de maintenance et d'utilisation d'ontologies”, 3èmes rencontres Terminologie et Intelligence Artificielle TIA, Vol 19, pp. 9-20, 2000.

- [27] Hérin D., Espinasse B., Andonoff E., Hannachi C., “Des systèmes d’information coopératifs aux agents informationnels”, chapitre 8 du livre ingénierie des systèmes d’information, 2001.
- [28] Izza S., “Intégration des systèmes d’information industriels, une approche flexible basée sur les services sémantiques”, Thèse de Doctorat, Ecole Nationale Supérieure des Mines de Saint –Etienne, France, 2006.
- [29] Jennings N.R., Wooldridge M., Sycara. K., “A roadmap of agent research and development”, In *International journal of Autonomous Agents and Multi- Agents Systems*, Vol.1, No1, pp. 7-38, 1998.
- [30] Kavouras M., “A unified ontological framework for semantic integration. International Workshop on Next Generation Geospatial Information”, Cambridge, UK, 2003.
- [31] Klein M., “Combining and relating ontologies: an analysis of problems and solutions”, International Joint Conferences on Artificial Intelligence (IJCAI), Workshop on Ontologies and Information Sharing, Seattle, USA, 2003.
- [32] Klischewski R., “Information integration or process integration: How to achieve interoperability in administration”, EGOV04 at DEXA, 30 August - 3 September, Zaragoza, Spain, 2004.
- [33] Maedche A., Motik B., Silva N., Volz R., “MAFRA MAPPING FRAMework for Distributed Ontologies”, In *Proc. of the 13th European Conf. on Knowledge Engineering and Knowledge Management EKAW-02*, 2002.
- [34] Mellal N., “Réalisation de l’interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d’information”, Thèse de Doctorat, Université de Savoie, 2007.
- [35] Mena E., Illarramendi A., Kashyap V., Sheth A.P., “Observer: An approach for query processing in global information systems based on interoperation across pre-existing ontologies”, *Distributed and Parallel Databases*, Vol.8, No.2, pp. 223-271, 2000.
- [36] Mizoguchi R., Bourdeau J., “Using ontological engineering to overcome common ai-ed problems”, IJAIED, 2000.
- [37] Müller J.P., “The Design of Intelligent Agents, A layered Approach”, *Lecture Notes in artificial Intelligence*, 1996.
- [38] Nodin, M., Unruh A., “Facilitating open communication in agent systems: The InfoSleuth infrastructure”, *Lecture Notes in Computer Science*. 1365, p. 281-290, 1998.
- [39] Noy N.F., “Semantic integration: a survey of ontology-based approaches”, *SIGMOD Record*, Special Issue on Semantic Integration, Vol. 33, No.4, pp. 65- 70, 2004.
- [40] Noy N.F., Musen M., "The Prompt suite: Interactive tools for ontology merging and

mapping', *International Journal of Human Computer Studies*, Vol.59, No.6, pp. 983-1024, 2003.

- [41] Ozsu M.T., Valduriez P., "Principles of Distributed Database Systems", Prentice Hall, 1999.
- [42] Palme J., "Personal notes from the 3 International Conferences on Cooperative Information Systems", NEW York, August 1998.
- [43] Psyche V., Mendes O., Bourdeau J., "Apport de l'ingénierie ontologique aux environnements de formation à distance", *Revue STICEF*, 10, 2003.
- [44] Saidane S., "Formalisation de Familles d'Architecture Logicielles Coopérative : démarches, Modèles et Outils", Thèse de Doctorat, Université Joseph Fourier-Grenoble I, 2005.
- [45] Séguran M., "Résolution des conflits sémantiques dans les systèmes d'information Coopératifs", Thèse de doctorat, Université Jean Moulin, 2003.
- [46] Singh M., Huhns M., "Service-Oriented Computing, Semantics, processes, agents", Wiley, 2005.
- [47] Swartout B., Patil K., Knight K., Russ T., "Use of Large-Scale Ontologies". *Spring Symposium Series on Ontological Engineering*. Stanford University, CA, pp. 138-148, 1997.
- [48] Uschold M., Gruninger M., "Creating Semantically Integrated Communities on the World Wide Web", In *Semantic Web Workshop*, Honolulu, Hawaii-Invited Talk, 2002.
- [49] Uschold M., Gruninger M., "Ontologies: principles, methods, and applications". *Knowledge Engineering Review*, Vol.11,No2, pp.93-155, 1996.
- [50] Van Heijst G., Schreiber A., Wielinga B.J. "Using explicit ontologies in KBS development", *International Journal of Human-Computer Studies*, Vol.46,No.2, pp.183-292, 1997
- [51] Vernadat F., "Interoperable enterprise systems: Architectures, methods and metrics", *Rapport technique*, LGIPM, Université de Metz, France, 2007.
- [52] Vokrinek J., JADE Tutorial. <http://agents.felk.cvut.cz/teching/ui2/JADEtutorial.doc>.
- [53] Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., "Ontology-based Integration of Information - A Survey of Existing Approaches", In: *Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing*, Seattle, WA, Vol.2, pp. 108-117.2001.
- [54] Zarour N., "Contribution à la modélisation de la coopération des systèmes d'information distribués et hétérogènes", Thèse de Doctorat, Université Mentouri de Constantine, 2004.

- [55] Zouggar N., Vallespir B., Chen D., “ Enrichissement de la MODélisation d’Entreprise par les Ontologies”, LAPS/GRAI Université Bordeaux1, ENSEIRB, UMR CNRS 5131. 6^e Conférence Francophone de MODélisation et SIMulation - MOSIM’06 ,2006.
- [56] [On line] WWW.Comptia.org/
European Interoperability Framework, - ICT Industry Recommendation, Brussels,18 February, 2004.
- [57] [On Line] <http://www.w3.org/TR/2004/REC-owl-guide/>
OWL Coalition, "OWL - Web Ontology Language - (W3C Recommendation 10 February 2004)". W3C, 2004.
- [58] [On Line] www.interop-noe.org/
INTEROP, "Deliverables of INTEROP Project", 2004.
- [59] [On Line] <http://turing.cs.pub.ro/auf2/home.html>
Agents Intelligents - Cours Web Interactif, Politechnica University of Bucharest – 2002.
- [60] [On Line] <http://www.sektproject.com>
Sekt, “Deliverables of SEKT Project”, 2004.
- [61] The Workflow Management Coalition, “*Workflow Management Coalition Terminology and Glossary*”, Technical Report, WFMC-TC-1011, 1999.
- [62] Wikipédia: <http://fr.wikipedia.org/wiki/Groupware>.2011.
- [63] [On Line] <http://knowledgeweb.semanticweb.org/>
KnowledgeWeb, "Deliverables of KWEB Project". EU-IST- 2004-507482, 2004
- [64] [On Line] www.FIPA.org/specifications/index.html
FIPA -ACL Interaction Protocol Specification.
- [65] http://www.acheter-louer.fr/location_maison.html
- [66] <http://www.seloger.com/immobilier/locations/ville/bien-maison/>
- [67] <http://www.tourisme.fr/>
- [68] <http://www.quebecweb.com/tourisme/>

Résumé

La coopération des systèmes d'information est devenue désormais un besoin crucial surtout dans le cadre du développement rapide des technologies de communication et la croissance du besoin d'échange d'information. Cette coopération peut être considérée dans plusieurs niveaux. Dans le cadre de ce travail, nous nous intéressons à la coopération des sources d'information distribuées et hétérogènes.

Notre objectif principal est de répondre au besoin d'accès uniforme et coopératif à plusieurs sources d'information distribuées et de gérer l'hétérogénéité de ces sources au niveau sémantique. Pour répondre à cet objectif, nous proposons une architecture basée sur le paradigme multi-agents dans laquelle trois types d'agents sont utilisés (agent d'information, agent adaptateur et agent annuaire) afin de traiter les requêtes complexes posées par les utilisateurs. Ce traitement est assuré par des interactions entre l'ensemble de ces agents selon un protocole de coopération. Concernant l'hétérogénéité sémantique une approche ontologique hybride est adoptée dont le but est de faciliter la découverte des correspondances (mappings) sémantiques entre les différentes sources d'information.

Mots clés : Systèmes d'information coopératifs, Agent, Sémantique, Interopérabilité, Ontologie, Mapping.

Abstract

The cooperation of information systems became henceforth a crucial need especially with the fast development of the technologies of communication and the growth of the need of information exchange. This cooperation can be considered in several levels. In this work, we are interested in the cooperation of the distributed and heterogeneous information sources.

Our main objective is to answer the need of uniform and cooperative access to several distributed information sources and to manage the heterogeneity of these sources at the semantic level. To answer this objective, we propose an architecture based on multi-agents paradigm in which three types of agents are used (information agent, wrapper agent and directory agent) to treat the complex requests of users. This treatment is insured by interactions between agents according to a protocol of cooperation. Concerning the semantic heterogeneity an ontological hybrid approach is adopted to facilitate the discovery of semantic correspondences (mappings) between the various information sources.

Keywords: Cooperative information systems, Agent, Semantic, Interoperability, Ontology, Mapping.