

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Doctorale de l'Est en Informatique
Pole Annaba
Centre Universitaire de Souk Ahras
Institut des sciences exactes

N° d'ordre :

Série :

Mémoire

Présenté en vue de l'obtention du diplôme de

Magister en informatique

Option : intelligence artificielle

Une Approche Multi-Ontologie Pour Des Systèmes Interopérables Basés Agent

Présenté par : M^F Lakhdar Goudjil

Dirigé par : Dr Okba Kazar

Soutenu-le : / / 2009 devant le jury composé de:

Président :

Pr Mahmoud BOUFAÏDA Professeur, Université Mentouri de Constantine

Rapporteur :

Dr Okba KAZAR Maitre de conférences, université mohamed khider Biskra

Examineurs :

Dr Nadir FARAH Maitre de conférences, Université Badji Mokhtar Annaba

Dr Nacer ZAROOUR Maitre de conférences, Université Mentouri de Constantine

Dr Abdelkrim AMIRAT Maitre de conférences B, Centre Universitaire de Souk Ahras

Résumé

Les utilisateurs du web sont devenus très exigeant en qualité d'information notamment après les échecs des approches basées sur la syntaxe. Certaines requêtes nécessitent des réponses contenues sur plusieurs sites distincts. Cela exige de recomposer les réponses élémentaires pour former un résultat final à l'utilisateur. Nous proposons un modèle d'interopérabilité entre systèmes avec des ontologies locales.

Dans notre modèle on utilise avec une approche multi-ontologies des mappings point-à-point bidirectionnels qui permettent de créer des mappings entre chaque paire d'ontologies. Cette approche permet d'apporter plus de flexibilité dans les environnements aussi complexes. Une difficulté extrême pour comparer différentes ontologies sources.

Pour pallier à cette difficulté nous avons intégré des agents ; dans notre cas les agents sont cognitifs ; ils manipulent des connaissances ; ces connaissances sont représentées sous forme d'ontologies. Une ontologie est propre à l'agent ontologique utilisé pour fonctionner le mapping entre les différentes ontologies en collaboration avec un agent interface.

Pour créer des agents aux ontologies différentes de façon systématique, automatique et intégrée, nous utilisons une nouvelle méthodologie que nous avons intégrée dans la plateforme JADE et l'éditeur d'ontologies Protégé. Nous avons proposé pour gérer le mapping point à point bidirectionnel de notre approche, un nouveau protocole respectant un formalisme basé sur le langage de communication FIPA-ACL, et qui permet également de spécifier l'ontologie utilisée pour exprimer le contenu du message.

MOTS CLES : Web Sémantique, Multi ontologie, Mapping d'ontologie, Interopérabilité Sémantique, Systèmes Multi-Agent.

Abstract

Web users have become very demanding as information especially after the failure of approaches based on the syntax. Some requests require responses contained several separate sites. This requires answers to recompose the basic training for a final result to the user. We propose a model for interoperability between systems with local ontologies.

In our model uses a multi-ontology mappings point-to-point bidirectional that create mappings between each pair of ontologies. This approach will provide more flexibility in such complex environments. Extremely difficult to compare different ontologies sources. To overcome this difficulty by integrating the agents.

In our case the agents are cognitive handling of knowledge, which knowledge is represented in the form of ontologies. Each ontology their own ontology agent used to operate the mapping between different ontologies and, in collaboration with an agent interface. To create different ontologies agents by systematic and automated way, we use a new methodology; integrated into the platform JADE editor Protégé. We have proposed to manage the mapping bidirectional point to point of our approach, a new protocol within a formalism based on the communication language FIPA-ACL, and also allows you to specify the ontology used to express the message content.

KEYWORDS: Semantic Web, Multi ontology, Ontology Mapping, Semantic Interoperability, Multi-Agent Systems.

ملخص

إن مستخدمي شبكة الإنترنت قد أصبحت غايتهم الدقة في المعلومات وخاصة بعد فشل النهج المعتمد على التركيب. بالإضافة إلى ذلك فإن بعض الطلبات تحتاج إلى العديد من الردود الواردة من مواقع مختلفة لتحصيل الأجوبة الأساسية لتكوين النتيجة النهائية للمستخدم ، من أجل ذلك نحن نقترح نموذجاً للتبادلية بين النظم باستخدام خرائط مفاهيم محلية .

في نموذجنا نستخدم متعدد خرائط المفاهيم ببناء ربط بينهم من نوع نقطة إلى نقطة ثنائي الاتجاه التي تجعل هناك ربط بين كل زوج من خرائط المفاهيم ، هذا النهج سيوفر قدراً أكبر من المرونة في مثل هذه البيئات المعقدة. من الصعب للغاية مقارنة خرائط المفاهيم لمصادر مختلفة للتغلب على هذه الصعوبة نقوم بدمج وكلاء؛ في حالتنا فإن عملاء (الوكلاء) قادرين على المعرفة و الإدراك و المستخدمين من أجل التعامل مع المعارف، والتي تتمثل في شكل خرائط المفاهيم.

لكل خريطة مفاهيم وكيل وجودي خاص بها ويستخدم لتشغيل وربط، بين مختلف خرائط المفاهيم ، وذلك بالتعاون مع وكيل واجهة لإنشاء وكلاء لخرائط مفاهيم مختلفة بطريقة منتظمة ومتكاملة ، إستخدمنا منهجية جديدة ، حيث قمنا بإضافة منصة JADE إلى محرر خرائط المفاهيم Protégé. من أجل التواصل بين مختلف خرائط المفاهيم قمنا باقتراح بروتوكول جديد يستخدم وكلائهم بتبادل رسائل مبنية على أساس لغة الاتصال FIPA ACL ، حيث يسمح ذلك بتحديد خريطة المفاهيم المستخدمة للتعبير عن مضمون الرسالة.

الكلمات الرئيسية : الواب الاستدلالي ، متعدد خرائط المفاهيم ، علم الربط بين خرائط المفاهيم ، الإسدال البيئي ، الأنظمة متعدد الوكلاء .

Dédicaces

Je dédie ce mémoire à:

À ma mère et mon père
À mes frères et mes sœurs
À ma future femme Sana
À tous ceux qui me sont cher

Remerciements

En tout premier lieu, je remercie mon dieu, tout puissant, qui m'a éclairé le bon chemin et qui m'a aidé à réaliser ce travail dans les meilleures conditions.

Il me serait impossible de citer nommément toutes les personnes qui m'ont aidé, encouragé et soutenu afin que ce travail puisse voir le jour. Que toutes ces personnes trouvent ici l'expression de ma sincère reconnaissance.

Je tiens tout d'abord à remercier mon encadreur Monsieur Okba KAZAR, Maitre de conférences, université Mohamed khider de Biskra pour avoir accepté de diriger mes travaux de recherche. Son aide, sa disponibilité, ses conseils, sa gentillesse, et son soutien durant toute cette période, m'ont toujours redonné confiance et volonté, ainsi que de ses précieuses remarques qui ont grandement contribué à améliorer la qualité de ce mémoire. Qu'il trouve ici l'expression de ma sincère reconnaissance et de ma profonde gratitude.

Je tiens à remercier vivement Monsieur Mahmoud BOUFAÏDA, professeur à l'université Mentouri de Constantine, pour m'avoir fait l'honneur d'accepter de présider le jury de ma soutenance.

Je tiens aussi à remercier monsieur FARAH Nadir, Maitre de conférences à l'université Badji Mokhtar d'Annaba, et monsieur ZAROUR Nacereddine, Maitre de conférences à l'université Mentouri de Constantine et monsieur Abdelkrim AMIRAT, Maitre de conférences B à Centre Universitaire de Souk Ahras pour m'avoir fait l'honneur d'accepter d'examiner mon mémoire de magister.

Mes remerciements vont également à mes beaux parents, mes beaux frères, et mes belles sœurs.

Enfin je n'omettrais pas de remercier tous mes amis de l'école doctorale de l'est en informatique avec eux j'ai partagé de très beaux moments de bonheur, je cite entre autres: Kamel TEBASSI, Rachid MEHMOUDI et aussi tous mes amis : Kamel BENIDIR, Amar ELAMRAOUI, Brahim SAHRAOUI, Abderrahim BELKHIR, Youcef ABOUD, Hichem OUAR, Adel BOUZIDI, Ali MANSAR, Abdelghani DOUKHI, Abdelkader HAMEL.

SOMMAIRE

INTRODUCTION GÉNÉRALE	01
CHAPITRE 01 : INTEROPERABILITE ET MAPPING D'ONTOLOGIE	04
I. INTRODUCTION	04
II. INTEROPERABILITE	04
II.1 Définition	04
II.2 Dimensions d'Interopérabilité	05
II.3 Types d'interopérabilité	05
III. POUR QUOI L'APPROCHE ONTOLOGIQUE ?	06
IV. ARCHITECTURE DES ONTOLOGIES POUR L'INTEGRATION	06
IV.1 L'approche mono-ontologie	07
IV.2 L'approche multi-ontologies ?	07
IV.3 L'approche hybride	08
IV.4 Comparaison entre les différentes approches	09
V. INTEGRATION DES ONTOLOGIES	09
V.1 Hétérogénéité des ontologies	10
V.2 Niveau pragmatique	12
IV. PROCESSUS D'INTEGRATION DES ONTOLOGIES	12
VI. TECHNIQUES POUR L'INTEROPERABILITE SEMANTIQUE	13
VI.1 La fusion d'ontologies	13
VI.2 Transformation d'ontologies	14
VI.3 Mapping d'ontologies	14
VI.3.1 Définition de mapping d'ontologies	14
VI.3.2 Classification des mappings d'ontologies	15

V.3.3 Formalisation de la notion de mapping	16
V.3.4 Processus de mapping d'ontologies	17
VII.MESURES DE SIMILARITE ET DE DISTANCES	18
VIII.DECOUCVERTE DE MAPPINGS	19
VIII.1 Découverte basée sur des heuristiques ou l'apprentissage	19
VIII.2 Méthodes de découverte de mappings	20
VIII.UTILS DE MAPPING	22
IX. CONCLUSION	23
CHAPITRE 02 : SYSTEMES MULTI-AGENTS POUR MULTI ONTOLOGIE 24	24
I. INTRODUCTION	24
II. SYSTEMES MULTI-AGENTS ET ONTOLOGIES	24
II.1 Ontologies pour les systèmes multi-agents	24
II.2 Ontologies pour les agents	25
II.3 Systèmes multi-agents pour les ontologies	25
III. ETAT DE L'ART	26
IV. NOTRE APPROCHE	30
V. ANALYSE COMPARATIVE	31
VI. PARTAGE DU SENS ENTRE AGENTS	35
VI.1 L'approche "NASA / Knowledge Evolution, Inc" : ONP.....	35
VI.2 L'approche "MERIT / Infonomics".....	35
VI.3 L'approche "Alignement réciproque d'ontologies".....	36
VI.4 Analyse comparative (synthèse)	36
VII. CONCLUSIONS	37

CHAPITRE 03 : MODELISATION D'UNE APPROCHE MULTI ONTOLOGIES BASE AGENT	38
I. INTRODUCTION	38
II. LA PLANIFICATION ET LE WEB SEMANTIQUE	38
III. L'INGENIERIE DES BESOINS	39
IV. CONSTRUCTION DES ONTOLOGIES	39
IV.1 Cycle de développement ontologique	40
IV.2 Principes pour la construction d'ontologies	41
V. AGENTS LOGICIELS, CONNAISSANCES ET COMMUNICATION	41
V.1 L'agent d'utilisateur	42
V.2.1 Définition et description	42
V.1.2 Architecture interne	42
V.2 L'agent d'ontologie	43
V.2.1 Définition et description	43
V.1.2 L'architecture d'agent d'ontologie	43
V.3 Diagramme de classes	47
VI. MAPPINGS POINT À POINT BIDIRECTIONNELS	48
VI.1 Description des services de base	48
VI.1.1 Accès à une ontologie	49
VI.1.2 Manipulation d'ontologies	49
VI.1.3 Interopérabilité d'ontologie	49
VI.2 Mapping d'ontologies	49
VI.2.1 Approches de découverte mappings	50
VI.2.2 Mapping deux ontologies	50
VI.2.3 Le multi-mapping d'ontologies	52
VII. PROTOCOLE DE MAPPING D'ONTOLOGIES	53
VII.1 Architecture et représentation	53

VII.2	Les règles	53
VII.3	Contrôle de la conversation	53
VII.4	Nomenclatures	54
VII.5	Services fournis par Le module de services mapping	55
VII.5.1	Trouver une ontologie	55
VII.5.2	Mapping d'ontologies	55
VII.5.3	Traduction d'un message	56
VII.5.4	Programme à partir d'un mapping	56
VII.5.6	Ajouter un mapping à la bibliothèque	56
VII.5.7	Obtenir un mapping	57
VIII.	SCENARIO D'UTILISATION	57
IX.	CONCLUSION	58
	CHAPITRE 04 : ETUDE DE CAS	60
I.	INTRODUCTION	60
II.	LA PLANIFICATION DU DOMAINE	60
III.	L'INGENIERIE DES BESOINS	60
IV.	LA CONSTRUCTION DES ONTOLOGIES LOCALES	61
IV.1.	Construction d'un Glossaire de termes	62
IV.2	construction des hiérarchies de concepts	63
IV.3	Construction du diagramme des relations binaires	64
IV.4	Construction du dictionnaire des concepts	66
IV.5	La table de relations binaires	68
IV.6	La table des attributs	69
IV.7	La table d'axiomes logiques	70
IV.8.	La table des instances	73
IV.9.	La table des assertions	78
a)	<i>représentation de la partie terminologique T-Box</i>	78

a).1	Définition de concepts	79
a).2	Définition de rôles	81
b)	<i>Représentation de la partie assertionnelle A-BOX</i>	81
V.	CONCLUSION	84
 CHAPITRE 05: IMPLEMENTATION		 85
I.	INTRODUCTION	85
II.	L'IMPLEMENTATION DES ONTOLOGIES	85
II.1.	Définir les classes et la hiérarchie des classes	86
II.2.	Définir les propriétés des classes – attributs	88
II.3.	Création des rôles entre les classes	89
II.4.	descriptions des classes	91
II.5.	Créer les instances	93
II.6.	Code OWL généré par Protégé 3.3.1	94
III.	L'IMPLEMENTATION DES AGENTS	94
III.1	Les agents d'ontologies	94
III.2	Description et Architecture logiciel de la plate-forme JADE	95
IV.	SERVICES DE MAPPINGS D'ONTOLOGIES	96
V.	SCENARIO D'UTILISATION	96
VI.1	Les acteurs du système	96
V.2	Déroulement du scénario	96
VI.	CONCLUSION	99
CONCLUSION ET PERSPECTIVES		100
REFERENCES		103

LISTES DES FIGURES

Figure 1.1 : Les aspects de l'interopérabilité	06
Figure 1.2 : approche mono-ontologie	07
Figure 1.3: approche multi-ontologies	07
Figure 1.4 : Approche hybride	08
Figure 1.5 : Trois dimensions d'hétérogénéités conceptuelles	11
Figure 1.6 : Méthodologie d'intégration d'ontologies	12
Figure 1.7 : Le principe de la fusion d'ontologies	13
Figure 1.8 : Le principe de la Transformation d'ontologies	14
Figure 1.9 : Le mapping d'ontologies	15
Figure 1.10 : Processus générique de mapping	16
Figure 1.11 : Processus de mapping d'ontologies	17
Figure 3.1 : la planification	38
Figure 3.2 : la construction d'ontologie	39
Figure 3.3. Cycle de vie d'ontologie	40
Figure 3.4. Cycle de vie fusionné	40
Figure 3.5 : architecture d'agent utilisateur	42
Figure 3.6 : architecture d'agent d'ontologie	44
Figure 3.7 : Le niveau conceptuel de diagramme de classe pour notre approche.....	47
Figure 3.8 : Le niveau implémentation de diagramme de classe pour les différents composants de notre approche	48
Figure 3.9 : Architecture générale de notre approche Mappings Point A Point Bidirectionnels	50
Figure 3.10 : Processus générique de mapping de deux ontologies	51
Figure 3.11 : Principe d'identification des interlocuteurs pour le suivi de la cohérence des échanges dans un langage de communication entre agents	54
Figure 3.12 : Protocole de mapping d'ontologie proposé pour notre approche	59
Figure 4.1 : la planification du notre domaine	60
Figure 4.2 : La hiérarchie des concepts pour le système enseignant	64
Figure 4.3 : La hiérarchie des concepts pour le système étudiant	64
Figure 4.4: diagramme des relations binaires pour le système enseignant	65
Figure 4.5 : diagramme des relations binaires pour le système étudiant	65
Figure 5.1: L'interface du protégé 3.3.1	86
Figure 5.2 : La hiérarchie des classes d'ontologie Encadreur	87
Figure 5.3 : La hiérarchie des classes d'ontologie Etudiant	87
Figure 5.4 : les propriétés des classes pour l'ontologie Encadreur	88
Figure 5.5 : les propriétés des classes pour l'ontologie Etudiant	89
Figure 5.6 : des rôles entre les classes d'ontologie Encadreur	89
Figure 5.7 : la création du rôle Travail_avec dans d'ontologie Encadreur	90
Figure 5.8 : des rôles entre les classes d'ontologie Etudiant	90
Figure 5.9 : la création du rôle Etudier_avec dans d'ontologie Etudiant	91
Figure 5.10 : la description de la classe Encadreur	92
Figure 5.11 : la description de la classe Etudiant	92
Figure 5.12 Création des instances pour l'ontologie Encadreur	93
Figure 5.13 : Création des instances pour l'ontologie Etudiant	93
Figure 5.14 : Architecture logiciel de La plate-forme JADE	95

LISTES DES TABLES

Table 1.1 : Évaluation des approches ontologiques	09
Table 2.1 : Comparaison entre les différentes approches ontologique, Langage d'otologie et Editeur d'Ontologie	
Table 2.2 : Panorama des principaux outils de mapping	32
Table 2.2 : Comparaison des mécanismes de création d'ontologies, mapping Inter-ontologie et l'application de domaine	33
Table 2.3 : Comparaison vue la classification et les méthodes de Mapping d'ontologie	34
Table 3.1 : Performatifs FIPA-ACL utilisés dans notre protocole	54
Table 3.2 : Pseudo-performatifs PMAO, spécifiques à notre protocole	54
Table 4.1 : Glossaire de termes	63
Table 4.2 : Dictionnaire des concepts pour le système enseignant	66
Table 4.3 : Dictionnaire des concepts pour le système enseignant	67
Table 4.4 : Table de relations binaires pour le système enseignant	68
Table 4.5 : Table de relations binaires pour le système étudiant	69
Table 4.6 : Table des attributs pour le système enseignant	69
Table 4.7 : Table des attributs pour le système étudiant	70
Table 4.8 : Table d'axiomes logiques pour le système enseignant	71
Table 4.9 : Table d'axiomes logiques pour le système étudiant	72
Table 4.10 : Table des instances pour le système Enseignant	75
Table 4.11 : Table des instances pour le système Etudiant	77
Table 4.12 : T-Box pour le système Enseignant	78
Table 4.13 : T-Box pour le système Etudiant	79
Table 4.14 : Définition de concepts pour le système Enseignant	80
Table 4.15 : Définition de concepts pour le système Etudiant	80
Table 4.16 : Définition de rôles pour le système Enseignant	81
Table 4.15 : Définition de rôles pour le système Etudiant	81
Table 4.16 : Table A-BOX pour le système Enseignant	83
Table 4.16 : Table A-BOX pour le système Etudiant	84

INTRODUCTION GÉNÉRALE

Le web actuel montre déjà ses limites, que ce soit par rapport à la recherche sur le texte, l'automatisation des recherches, et la hiérarchisation des données. Il faut donc trouver un moyen d'améliorer la recherche d'information sur internet, un outil permettra de faire des requêtes toujours plus complexes tout en obtenant des réponses toujours plus adaptées à nos besoins ,un outil qui permettra de faire évoluer le Web, pour l'amener à prendre possession de ses pleines capacités. Cet outil a déjà imaginé, il se somme le web sémantique.

Le principe de base du web sémantique consiste à décrire des relations existant entre des différents concepts dans un domaine de connaissance données. C'est ce qu'on appelle un "ontologie". Dans le cadre d'applications distribuées, l'évolution des récents développements des systèmes à base de connaissances fait apparaître la nécessité de spécifier une interprétation commune des informations échangées. Cela peut être effectué en utilisant des standards ou en adaptions ces systèmes pour qu'ils interprètent les informations sans ambiguïté, on parle alors d'interopérabilité.

Plus précisément, la croissance exponentielle des informations et des ressources échangées entre les différents systèmes, qu'ils soient publics ou privés (Internet, bases de données, etc), 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. C'est ce que l'on appelle l'interopérabilité sémantique. 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 œuvre. L'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, à travers la notion d'ontologie.

Dans le cas où les sources concernent des systèmes interopérables, L'approche multi-ontologies permet d'associer une ontologie locale à chaque Système à base des connaissances. L'avantage est que chaque ontologie source peut être définie sans prendre en considération les autres sources ou les autres ontologies. Mais le manque d'un vocabulaire commun conduit à une difficulté extrême pour comparer différentes ontologies sources. Pour surmonter ce problème, un formalisme de représentation additionnelle définissant le mapping inter-ontologie est fourni. On peut avoir dans ce cadre des architectures, des mappings point-à-point qui permettent de créer des mappings entre chaque paire d'ontologies. Bien qu'elle soit relativement lourde, l'approche multi-ontologies permet de préserver la nature décentralisée des systèmes d'information à intégrer. Mais,

le mapping point-à-point est difficile à définir à cause de divers problèmes d'hétérogénéité sémantique qui peuvent se produire.

Face aux problèmes précédemment évoqués, la problématique abordée ici pour pallier cette difficulté par l'intégration des agents ; ils sont des programmes autonomes et communicants, conçus pour interagir avec d'autres agents. Comme nous venons de le voir, les systèmes multi-agents gagnent à s'appuyer sur des ontologies pour fonctionner et le mapping entre ontologies est une des conditions de leur bonne communication. Dans notre cas les agents sont cognitifs ; ils manipulent des connaissances ; ces connaissances sont représentées sous forme d'ontologies.

Comme de coutume, ce mémoire commence par un premier chapitre dédié à un problème crucial concerne l'interopérabilité (*syntaxique* et *sémantique*) ces la capacité de deux applications (ou plus) à coopérer et à échanger de l'information afin d'atteindre un but global. Nous présentons ensuite, recense trois approches possibles pour l'intégration des données qui sont : *l'approche mono-ontologie*, *l'approche multi-ontologies* et *l'approche hybride*. Nous entendons par les processus d'intégration des ontologies tel que Le mapping d'ontologies, qui interroger des bases de connaissances hétérogènes en transformant des données entre différentes représentations des ontologies. Ainsi Le processus mapping nécessite généralement de mettre en œuvre le calcul d'une distance de similarité qui est calculée plusieurs façons.

Le chapitre 2 est structuré en deux grandes parties. La première partie porte l'état de l'art sur les différents travaux présente les relations entre les systèmes multi agents et les ontologies dans les systèmes interopérables, et plus particulièrement, l'utilisation des systèmes multi-agents pour la mapping entre les ontologies qui est le but recherché dans nos travaux. La deuxième partie est consacrée à quelques propositions pour le partage du sens entre agents. Et dans chaque partie, nous intéressons à comparer notre approche avec les autre approches précédentes.

Le chapitre 3 comprend notre approche méthodologique, où nous essayerons de détailler la démarche suivie, et la description du chaque étape à réalisé pour géré le mapping dans l'architecture multi ontologies. Notre objectif est de modéliser les connaissances traduisant les différents points de vue de ces systèmes au sein des ontologies locales afin de pouvoir comparer et rendre interopérables les conclusions fournies par différents systèmes. Pour le mapping d'ontologie nous avons proposé un nouveau protocole pour gérer la communication entre les agents, eux s'échangent des messages en respectant un formalisme basé sur le langage de communication FIPA-ACL.

Le chapitre 4, Afin de plus aisément comprendre le mapping point à point dans l'approche multi ontologie basé agent, et à travers le processus proposé dans le chapitre précédent, Comme il nécessitait choisir un domaine de connaissance, nous avons choisi « l'encadrement universitaire » comme domaine applicatif.

Le chapitre 5, sur l'implémentation, le premier objectif est de générer un code OWL pour créer les ontologies, nous utiliserons pour ce la protégé 3.3.1. Dans le deuxième objectif, Nous étudions dans les différents scénarios de communication entre les agents d'ontologies de notre cas d'utilisations, au cours de laquelle notre protocole proposé sera amené à être utilisé pour accélérer le mapping entre les ontologies locales.

Enfin la conclusion résume les apports, et nos perspectives sur le travail, et une prévision sur les travaux futurs. Nous discutons également des perspectives de ce travail.

CHAPITRE 01 :

INTEROPERABILITE ET MAPPING D'ONTOLOGIE

I. INTRODUCTION

Les systèmes ont généralement besoin de communiquer, d'échanger de l'information et des ressources sous forme de programmes, de données ou de services. Ces systèmes sont appelés systèmes distribués et l'exemple le plus connu n'est autre que le réseau Internet. Le concept de communication est à l'origine des transferts d'information dans les systèmes distribués. En termes simples, celui-ci est vu comme un ensemble de sous-systèmes logiciels et matériels, physiquement reliés par des canaux de communication. Sur cet ensemble de ressources réparties ou distribuées, les applications mises en oeuvre sont appelées applications distribuées. De nombreuses applications distribuées connaissent un véritable succès, telles que le World Wide Web(WWW) sur lequel un nombre quelconque de navigateurs ou de clients peuvent se connecter.

Parmi les applications distribuées, un nombre non négligeable s'oriente vers la réalisation d'une tâche commune. Cette tâche est généralement associée à un objectif que l'on qualifie de but global du système distribué. Cet aspect est au coeur de notre problématique, puisque nous allons nous intéresser aux systèmes distribués dans lesquels chaque sous-système est dédié à la réalisation d'un fragment d'une tâche globale et possède ses propres objectifs. Dans ce domaine, un problème crucial concerne la capacité de deux applications (ou plus) à coopérer et à échanger de l'information afin d'atteindre un but global. On parle alors d'interopérabilité entre les systèmes pour la réalisation du but.

II. INTEROPERABILITE

II.1 Définitions

On dit que plusieurs systèmes (identiques ou différents) *interopèrent* s'ils peuvent communiquer sans aucune ambiguïté. Cette *interopérabilité* des systèmes informatiques est limitée aux aspects d'hétérogénéité et de normalisation. Elle est assurée au niveau des couches basses des systèmes informatiques par la normalisation des interfaces physiques s'appuyant sur des standards concernant les couches hautes. A titre d'exemple, nous pouvons citer les protocoles du réseau Internet (TCP/IP, HTTP, etc) qui permettent à des ordinateurs utilisant des technologies et des systèmes d'exploitation différents d'échanger sans ambiguïté de l'information.

En ce qui concerne l'intelligence artificielle, et plus particulièrement la représentation des connaissances et le raisonnement, l'interopérabilité apparaît comme une étape cruciale vers une unification de la sémantique des connaissances distribuées. Les ontologies sont précisément un des moyens contribuant à faciliter la compréhension des informations échangées entre les systèmes interopérables en essayant de standardiser la représentation des concepts et de leurs relations.

II.2 Dimensions d'Interopérabilité

L'interopérabilité peut être analysée selon trois principales dimensions, qui sont la *distribution*, *l'autonomie* et *l'hétérogénéité*.

a) **La distribution** identifie les interactions entre différentes composantes; d'une manière plus fine, ces interactions peuvent se décomposer selon plusieurs axes (qui, pourquoi, quoi, quand, comment, où).

b) **L'autonomie** peut se présenter sous plusieurs formes. Au niveau de la conception, elle désigne le choix du domaine de gestion et la conceptualisation du contexte. En relation avec d'autres systèmes, elle représente la capacité d'un système ou d'un composant de choisir "ses partenaires" pour l'échange de ressources, ainsi que les modes de réaliser ces échanges. La communication représente la capacité, pour un noeud du réseau (système ou composant) de communiquer avec d'autres noeuds, tandis que l'exécution concerne la capacité pour un noeud d'exécuter des opérations locales indépendamment des composantes externes.

c) **L'hétérogénéité** se manifeste à plusieurs niveaux et peut concerner les systèmes (plateformes techniques, systèmes logiciels), comme l'information elle-même, pour des problématiques relevant de la syntaxe, de la structure et de la sémantique.

II. 3 Types d'interopérabilité

L'interopérabilité amène deux problèmes majeurs, les conflits syntaxiques et les conflits sémantiques ; Ces deux types de conflits, nécessitent de caractériser *l'interopérabilité syntaxique* et *l'interopérabilité sémantique*.

a) **L'interopérabilité syntaxique** : Les conflits syntaxiques résultent de l'utilisation de modèles de données distincts entre systèmes. Par exemple, des modèles de représentation différents sont utilisés pour structurer un même concept. Ce conflit, nécessite de caractériser *l'interopérabilité syntaxique* ; dans ce type la syntaxe des informations échangées est définie et doit être respectée par les différents systèmes qui interopèrent.

b) **L'interopérabilité sémantique** : Les conflits sémantiques sont issus des différences de compréhension et d'interprétation entre les informations provenant de divers domaines d'application.

Ce conflit, nécessitent de caractériser *l'interopérabilité sémantique* ; Dans ce type l'interopérabilité vise à explorer les problèmes de cohérence dans le sens où les informations échangées doivent avoir la même signification au sein de ces systèmes. L'interopérabilité sémantique doit également préserver la sémantique des informations échangées (voir la figure 1.1).

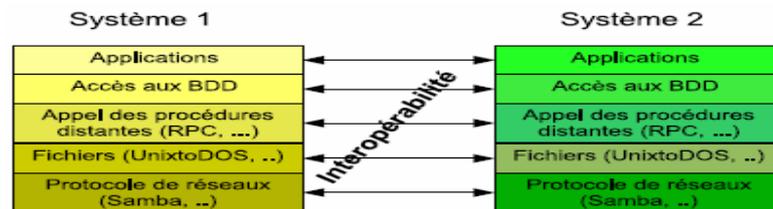


Figure 1.1 : Les aspects de l'interopérabilité [MELLAL 2007]

Nous nous intéressons ici à *l'interopérabilité sémantique* ; **F.Vernadat** explique l'interopérabilité sémantique en une phrase : " *To exchange services and data among systems that make sense (common "meaning")*" [F. VERNADAT 2007] .

III. POUR QUOI L'APPROCHE ONTOLOGIQUE ?

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 œuvre. 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 dans le respect de cette sémantique, qui représente actuellement un défi dans plusieurs domaines de recherche, en particulier en intelligence artificielle à travers la notion *d'ontologie*.

Les ontologies sont étudiées par les chercheurs travaillant en intelligence artificielle, sur la représentation de la connaissance, et maintenant, sur le Web sémantique. Une définition des ontologies consensuelle, précise et complète dans le contexte du Web sémantique n'existe pas encore. Cependant, la définition la plus citée dans notre domaine de recherche est celle de [Gruber, 1993]: « *Une ontologie est une spécification explicite et formelle d'une conceptualisation d'un domaine de connaissance* ».

IV. ARCHITECTURE DES ONTOLOGIES POUR L'INTEGRATION

Il existe plusieurs approches d'utilisation des ontologies pour l'explicitation de la sémantique. Une première approche intuitive est celle qui consiste à utiliser une ontologie unique qui permet de capturer toute la sémantique de la réalité considérée. D'autres approches consistent à utiliser, non pas une seule, mais plusieurs ontologies. Ces dernières approches sont notamment motivées par le fait qu'il est souvent impossible voire irréaliste de s'entendre sur une seule ontologie, ou à la limite sur un nombre assez réduit d'ontologies. Dans le domaine de l'intégration des données, [Wache et al. 2001] recense

trois approches possibles qui sont : *l'approche mono-ontologie*, *l'approche multi-ontologies* et *l'approche hybride*[Saïd IZZA 2006].

IV.1 L'approche mono-ontologie

L'approche mono-ontologie (single-ontology approach) permet de mettre en œuvre une ontologie unique et partagée entre les différents systèmes à bases des connaissances, fournissant ainsi un vocabulaire partagé pour la spécification de la sémantique. Toutes les bases des connaissances sont reliées à une ontologie globale (figure 1.2).

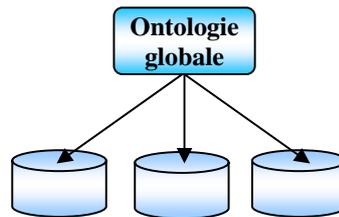


Figure 1.2 : **approche mono-ontologie** [Wache et al. 2001]

Les approches mono-ontologie sont les approches les plus simples à mettre en œuvre lorsque les bases des connaissances se réfèrent à des domaines similaires.

IV.2 L'approche multi-ontologies ?

a) Pour quoi l'Approche multi-ontologies ?

Dans le cas où les sources concernent des systèmes interopérables, il devient difficile voire impossible à concrétiser un engagement ontologique. Une autre limite importante de cette approche tient du fait que la modification des sources peut affecter l'ontologie globale. et la solution d'utiliser *l'Approche multi-ontologies*

b) L'approche multi-ontologies

L'approche multi-ontologies (multi-ontology approach) permet d'associer une ontologie locale à chaque Système à base des connaissances (figure : 1.3). Cette approche convient dans le cas où il devient difficile de trouver une ontologie commune résultant des grandes différences sémantiques existant entre les systèmes. Des mappings inter-ontologiques sont alors nécessaires afin d'établir une interprétation commune des données. Comme exemple d'implémentation, le projet OBSERVER [Mena et al. 2000] s'inscrit dans cette approche. Il associe à chaque source de données une ontologie locale.

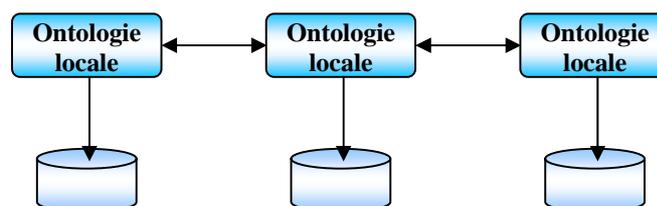


Figure 1.3: **approche multi-ontologies** [Wache et al. 2001]

L'intérêt de cette approche est le fait que les sources peuvent être développées de façon indépendante, mais l'absence d'un vocabulaire commun conduit généralement à une difficulté extrême pour comparer différentes ontologies sources. Pour pallier cette difficulté, un formalisme de représentation additionnelle définissant le mapping inter-ontologie est utilisé. Ce dernier identifie sémantiquement les termes correspondants de différentes ontologies sources [Saïd IZZA 2006].

c) Le projet OBSERVER

Dans le cadre du projet Observer [Mena et al, 2000]: Observer (ontology based system enhanced with relationships for vocabulary heterogeneity resolution) une ontologie est une description des concepts et des relations qui peuvent exister pour un agent ou une communauté d'agents. Les termes d'un vocabulaire, liés entre eux par des relations sémantiques (synonymies, polysémie, etc.), constituent l'ontologie. Observer utilise des ontologies existantes pour l'accès à des sources de données autonomes et hétérogènes. Ces ontologies servent à décrire le contenu des sources de données qui donnent une vue conceptuelle de leurs contenus ; Ces ontologies sont décrites en utilisant un langage de la logique de descriptions.

IV.3 L'approche hybride

L'approche hybride combine les deux approches précédentes en utilisant à la fois des ontologies locales et une ontologie partagée (parfois un vocabulaire commun) afin de simplifier les mappings ontologiques (figure :1.4). 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[Saïd IZZA 2006].

Comme exemples de projets associés à cette approche, nous pouvons citer le projet COIN [Goh 1997], le projet MECOTA [Wache et al. 1999], et le projet BUSTER [Visser et stuckenschmidt 2002] et qui utilisent respectivement la notion de contexte local, des mécanismes d'annotation, des raffinements d'une ontologie générale pour décrire localement les sources de données.

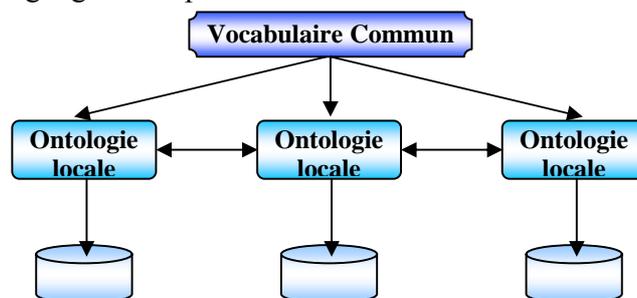


Figure 1.4 : Approche hybride (adapté de [Wache et al. 2001])

IV.4 Comparaison entre les différentes approches

Nous avons brièvement reproduit dans le tableau suivant (tableau :1.1) les principales caractéristiques de chacune de ces approches.

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

Tableau : 1.1 Évaluation des approches ontologiques [Wache et al. 2001]

Dans cette thèse nous avons affaire à l'approche de multi ontologie où chaque du fait qu'elles comportent plus d'une ontologie, les approches à ontologies multiples peuvent être confrontées au problème d'hétérogénéité ontologique [Klein 2001]. Dans ce cas, il devient nécessaire d'intégrer et de faire interopérer des ontologies. Cet aspect est traité dans la section suivante.

V. INTEGRATION DES ONTOLOGIES

Le mot intégration est utilisé avec plusieurs significations dans le domaine d'ontologie. [A. GÓMEZ-PÉREZ,al 1995]définie trois différentes significations :

a) Intégration : Construction d'une nouvelle ontologie réutilisant (en assemblant, étendant, spécialisant, ou adaptant) d'autres ontologies disponibles. Ces différentes ontologies font partie de la nouvelle ontologie.

b) Fusion : Construction d'une ontologie par la fusion de différentes ontologies dans une seule ontologie qui les unifie.

c) Utilisation : Construction d'une application utilisant une ou plusieurs ontologies pour spécifier ou implémenter un système à base de connaissance.

Plusieurs travaux existent qui étudient le problème d'intégration des ontologies ; et qui proposent des solutions pour résoudre les différentes hétérogénéités qui peuvent exister au niveau des ontologies. Dans cette section nous allons étudier les hétérogénéités d'ontologies et nous allons ensuite présenter les solutions typiques utilisées pour leur résolution.

V.1 Hétérogénéité des ontologies

Il existe dans la littérature plusieurs classifications d'hétérogénéités ; on a retenu dans le cadre de nos travaux la classification proposée dans le cadre du projet Knowledge Web [KnowledgeWeb 2004] qui distingue fondamentalement quatre niveaux d'hétérogénéités qui sont

- les hétérogénéités de niveau syntaxique (ou niveau de langage);
- les hétérogénéités de niveau terminologique
- les hétérogénéités de niveau conceptuel
- et des hétérogénéités de niveau pragmatique.

a) Niveau syntaxique

Le premier type d'hétérogénéité est lié aux caractéristiques des langages utilisés pour représenter les ontologies. Les langages peuvent différer dans leur syntaxe, mais plus important encore dans les concepts et aussi dans les contraintes utilisés dans les deux langages : certains concepts et/ou contraintes d'un langage ne sont pas disponibles dans un autre langage. Ce type d'hétérogénéité soulève le problème de translation ou de traduction d'ontologies [Corcho et al. 2003-b] qui constitue un cas particulier du problème d'intégration d'ontologies. Ce problème est généralement résolu de façon relativement simple par la mise en oeuvre d'un processus de translation ou de traduction qui se base souvent sur des techniques de normalisation qui permettent ainsi de résoudre les différences syntaxiques pouvant exister entre deux ontologies données. Dans le cadre des travaux que nous menons, cet aspect n'est pas considéré dans la mesure où l'on impose *l'utilisation d'un langage unique*.

b) Niveau terminologique

Le second type d'hétérogénéité concerne toutes les différences liées au processus de nomination des entités (classes, propriétés, etc.). A ce titre, il est possible d'établir les conflits typiques suivants :

- *Synonymie* : où plusieurs mots qui désignent la même entité;
- *Polysémie* : où un même mot est utilisé pour dénommer différentes entités;
- *Multilinguisme* : où des mots de différents langages sont utilisés pour nommer des entités;
- *Et des variations syntaxiques d'un même mot* (abréviations, etc.)

Comme le précise [KnowledgeWeb 2004], il est important de retenir que du point de vue complexité, les hétérogénéités terminologiques ne sont pas aussi complexes que les hétérogénéités conceptuelles. Cependant, la plupart des hétérogénéités qui occurrent dans la réalité se situent à ce niveau. Ce qui rend ce dernier assez critique.

c) Niveau conceptuel

Les hétérogénéités conceptuelles concernent les différences de couverture de l'univers du discours, le degré de granularité et le point de vue de l'ontologie (figure : 1.5) :

- *Couverture* : qui désigne le fait que les ontologies couvrent différentes portions de l'univers du discours. Par exemple, on peut définir une ontologie sur la fabrication de composants électroniques qui inclut la description des opérateurs de maintenance, alors qu'on peut également avoir une autre ontologie qui ne décrit pas les opérateurs.

- *Granularité* : qui signifie que les entités des ontologies peuvent décrire les objets de la réalité à des degrés de détails différents. Par exemple : on peut avoir une ontologie d'équipement qui ne s'intéresse qu'aux différents type de machines dont on dispose, alors que l'on peut définir une autre ontologie qui inclut tous les équipements individuels avec leurs caractéristiques spécifiques;

- *Perspective* : qui signifie que les ontologies couvrent des points de vue différents.

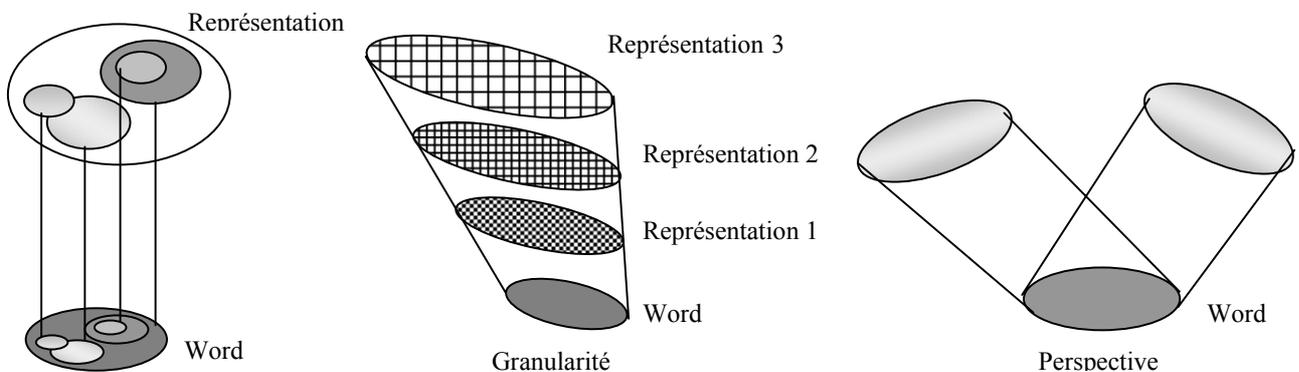


Figure 1.5 : Trois dimensions d'hétérogénéités conceptuelles [KnowledgeWeb 2004]

IV.1.4. Niveau pragmatique

Le niveau pragmatique est le niveau le plus complexe. Il concerne les hétérogénéités d'interprétation d'une ontologie qui peuvent survenir lorsque des individus ou des communautés différentes interprètent différemment l'ontologie selon différents contextes. Dans le cadre de nos travaux, on ne s'intéresse pas à cet aspect des choses, qui relève plutôt de l'intégration au niveau des connaissances.

V. PROCESSUS D'INTEGRATION DES ONTOLOGIES

La médiation d'ontologies qui est le processus qui permet de réconcilier différentes ontologies, se base généralement sur le principe de la combinaison d'ontologies. Ce processus de combinaison est très souvent appelé intégration d'ontologies qui signifie à la fois le fait que les ontologies peuvent être fusionnées en une ontologie unique nouvellement créée, et aussi le fait que les ontologies peuvent être mises en correspondance sans création de nouvelle ontologie [Klein 2001]. Dans tous les cas, les ontologies à intégrer doivent être réconciliées dans le sens où elles doivent être ramenées à un engagement ontologique mutuel.

Il est généralement admis que la démarche d'intégration d'ontologies est basée sur trois sous-processus (figure :1.6) [Kavouras 2003] :

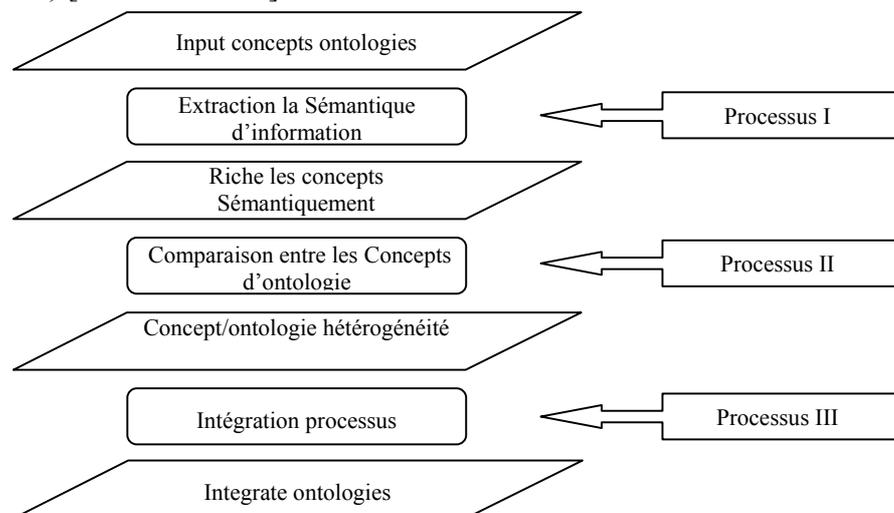


Figure 1.6 : Méthodologie d'intégration d'ontologies (adapté de [Kavouras 2003])

- *Le processus d'extraction* correspond à la modélisation de la sémantique effectuée sur la base d'ontologies.
- *Le processus de comparaison* d'ontologies permet de quantifier le degré de similarité ou d'hétérogénéité des concepts des ontologies à intégrer, et ce en se basant très souvent sur des calculs de distances [Euzenat et Valtchev 2004][Maedche et Staab 2002][Sheth et Kashyap 1993].
- Et enfin, le dernier processus permet d'intégrer les ontologies concernées.

VI. TECHNIQUES POUR L'INTEROPERABILITE SEMANTIQUE

Un certain nombre de techniques ont été proposées dans la littérature pour réaliser l'interopérabilité. 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 [Saïd IZZA 2006], l'auteur distingue **trois catégories principales** qui sont :

1. **La fusion d'ontologies**, qui permet de créer une nouvelle ontologie, appelée *l'ontologie fusionnée* capturant les connaissances des ontologies d'origine. Le défi est alors d'assurer que toutes les correspondances et les différences entre les ontologies soient correctement prises en compte dans l'ontologie résultante.

2. **Transformation d'ontologies** qui permet de changer la structure des ontologies en conservant au maximum sa sémantique.

3. **Le mapping d'ontologies**, qui a comme objectif la représentation des correspondances entre les ontologies. Ceci permet, par exemple, d'interroger des bases de connaissances hétérogènes en utilisant une interface commune ou en transformant des données entre différentes représentations.

Ces trois techniques seront détaillées dans les sections suivantes.

VI.1 La fusion d'ontologies

[NAMYOUUN et YEOL et HYOIL 2006] donne la définition de la fusion d'ontologie suivante : « *Ontology merging is the process of generating a single, coherent ontology from two or more existing and different ontologies related to the same subject* ». La fusion d'ontologies représente la création d'une nouvelle ontologie à partir de deux ontologies ou plus. L'ontologie résultante unifie et remplace les ontologies d'origine (Voir la figure 1.7). Cette définition ne précise pas comment l'ontologie résultante est reliée aux ontologies originales pour laisser ouvert le problème du choix de la méthode de fusion.

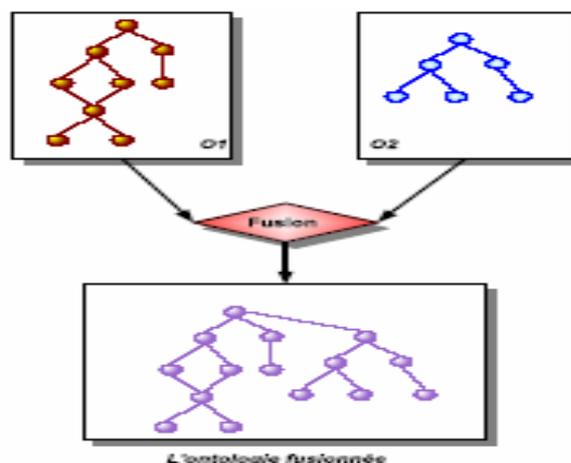


Figure 1.7: le principe de la fusion d'ontologies [MELLAL 2007]

Les approches les plus courantes utilisent l'union ou l'intersection. Dans l'approche par union, l'ontologie résultante contient l'union des entités provenant des ontologies originales et suppose résolues les différences de représentation d'un même concept. Dans l'approche de type intersection, l'ontologie résultante ne contient que les parties communes des ontologies originelles.

Plusieurs approches mettant en oeuvre la fusion d'ontologies ont été proposées telles que PROMPT [NOY et MUSEN 2000] ,CHIMAERA [MCGUINNESS], FCA-Merge [STUMME et MAEDCHE 2001] et OntoMerge [DOU, MCDERMOTT et Peishen QI 2002].

VI.2 Transformation d'ontologies

La transformation d'ontologies est un processus qui permet de changer la structure des ontologies en conservant au maximum sa sémantique (figure 1.8). Selon que la transformation se fait sans pertes ou avec perte d'informations, on parle de transformation sans pertes (lossless) et de transformation avec pertes (lossy)

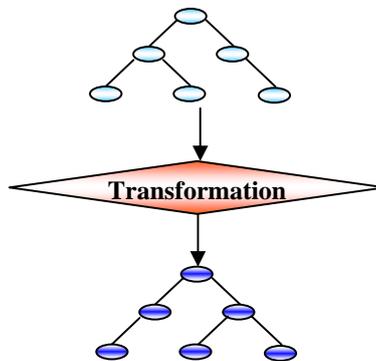


Figure 1.8: le principe de la Transformation d'ontologies

VI.3 Mapping d'ontologies

VI.3.1 Définition de mapping d'ontologies

La définition la plus pertinente est probablement celle de Noy pour qui le mapping d'ontologies est un processus qui spécifie une convergence sémantique entre différentes ontologies afin d'en extraire les correspondances entre certaines entités.

Ces correspondances sont exprimées en introduisant des axiomes formulés dans un langage spécifique.

Trois phases principales peuvent être distinguées dans ce processus (voir la figure 1.9) :

- la découverte du mapping ;
- la représentation du mapping ;
- l'exploitation et l'exécution du mapping.

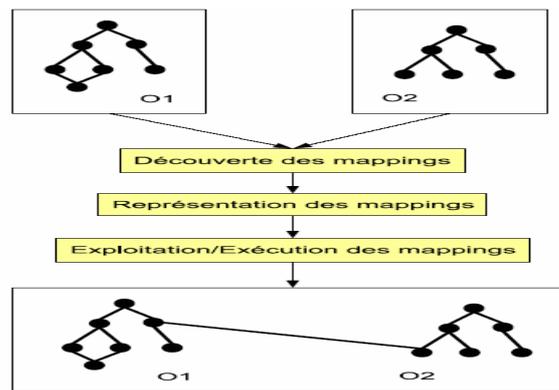


Figure 1.9 : mapping d'ontologies [MELLAL 2007]

Dans la suite de ce document nous allons nous restreindre principalement au mappings des ontologies qui constitue à notre connaissance l'approche la plus adaptée.

Les outils et méthodologies, parmi les plus significatifs dans cette catégorie, sont MAFRA [Alexander M; al 2002], IF-Map [Yannis K ;al 2003],RDFT [B OMELAYENKO;al 2003],C-OWL [P.BOUQUET,al 2003] et OntoMap [J. ANGELE ;al 2005].

VI.3.2 Classification des mappings d'ontologies

Il existe deux types de classification des mappings : La première classification basée sur le mode de liaison des entités des ontologies impliquées dans le mapping. Aussi, on distingue principalement deux types de mappings qui sont [Bruijn et Polleres 2004] :

- Les mappings unidirectionnels : permettent de spécifier des relations entre deux ontologies uniquement dans une seule direction.ces mappings sont relativement simples à mettre en œuvre.
- Les mappings bidirectionnels : permettent de définir des relations entre deux ontologies dans les deux directions.ces mappings, sont généralement plus flexibles, mais ils demandent souvent plus d'effort en matière de spécification que dans le cas des mappings unidirectionnels.

Notons que les mappings bidirectionnels sont à notre sens la catégorie de mappings les plus favorables. Ils permettent de définir l'approche la plus réaliste et également la plus flexible. C'est la raison principale qui nous pousse à nous intéresser dans les prochains chapitres à ce type de mappings.

La deuxième approche des mappings d'ontologies repose sur des architectures d'ontologies mettant en jeu deux ou plusieurs ontologies. Ce sont donc les approches multi-ontologies qui peuvent être concernées par ce type de mappings, on peut avoir dans le cadre des architectures, des mappings point-à-point (un-à un ou one-to-one mapping) qui permettent de créer des mappings entre chaque paire d'ontologies. Par exemple, l'approche utilisée dans le projet OBSERVER [Mena et al. 2000] repose sur

des mappings de type point-à-point. le nombre de mappings créés par l'approche multi-ontologies est de l'ordre de $o(n^2)$ où n est le nombre d'ontologies impliquées. Bien qu'elle soit relativement lourde, l'approche multi-ontologies permet de préserver la nature décentralisée des systèmes d'information à intégrer. Mais, on reproduit ainsi la complexité des architectures point-à-point.

V.3.3 Formalisation de la notion de mapping

Formellement, il est possible de définir la notion de mapping d'ontologies comme un ensemble de correspondances au sens mathématique entre un certain nombre d'ontologies [KnowledgeWeb 2004]. Autrement dit, un mapping entre deux ou plusieurs ontologies est un ensemble de liens (nommées aussi mappings). Un mapping désigne une expression formelle qui établit une relation sémantique entre deux ontologies. Quand la relation inter-ontologique est orientée, on parle très souvent de mapping au sens mathématique, c'est-à-dire de la notion de fonction ou de morphisme mathématiques [Kalfoglou et Schorlemmer 2003].

Du point de vue abstrait, [KnowledgeWeb 2004] définit un mapping de deux ontologies O et O' comme un opérateur $\alpha(O, O')$ qui :

- à partir des deux ontologies O et O' de couverture différente, permet de réconcilier l'utilisation simultanée des deux ontologies pour la description de l'univers du discours;
- à partir des deux ontologies O et O' de granularité différente, permet d'associer aux entités de l'ontologie O les entités homologues dans O' ;
- à partir des deux ontologies O et O' de point de vue différent, permet de déduire le point de vue de O' associé à une entité de l'ontologie O .

Le processus de mapping consiste donc à générer une correspondance M' à partir de deux ontologies O et O' . Aussi, [KnowledgeWeb 2004] considère le processus de mapping comme une fonction f qui à partir d'une paire d'ontologies O et O' à réconcilier, un mapping initial M , un ensemble de paramètres p , un ensemble de ressources r , produit un nouveau mapping M' entre ces deux ontologies :

$$M = f(O, O', M, p, r)$$

Ce processus peut être représenté comme suit (figure 1.10) :

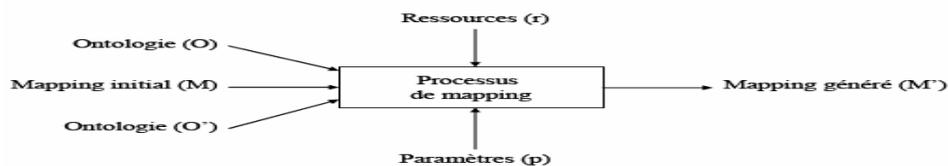


Figure 1.10 : Processus générique de mapping (adapté de [KnowledgeWeb 2004])

Il est bien entendu possible de définir le mapping de plusieurs ontologies, ce dernier est alors appelé le multi-mapping d'ontologies. Et il peut être considéré comme étant une extension de la fonction précédente, en insérant comme arguments les n ontologies concernées par le mapping :

$$M' = f(O1, O2, \dots, On, M, p, r)$$

Le mapping initial M peut être un mapping injecté en entrée en vue de le mettre à jour ou le compléter. Les ressources r peuvent être les ressources matérielles, logicielles et humaines utilisées pour exécuter le processus de mapping. Quant aux paramètres p , ces derniers peuvent être les hypothèses effectuées comme le seuil de vraisemblance à utiliser, etc.

V.3.4 Processus de mapping d'ontologies

De façon générale, le processus de mapping ontologique peut se résumer principalement aux étapes suivantes [McGuinness et al. 2000]:

1. Trouver les zones de recouvrement des ontologies sources (à intégrer)
2. Relier les concepts qui sont sémantiquement proches en utilisant des relations sémantiques (relations d'équivalence, de subsumption, etc.)
3. Vérifier la consistance, la cohérence et non redondance du résultat obtenu.

De façon plus détaillée, le processus de mapping comprend généralement les étapes suivantes (figure 1.11):

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

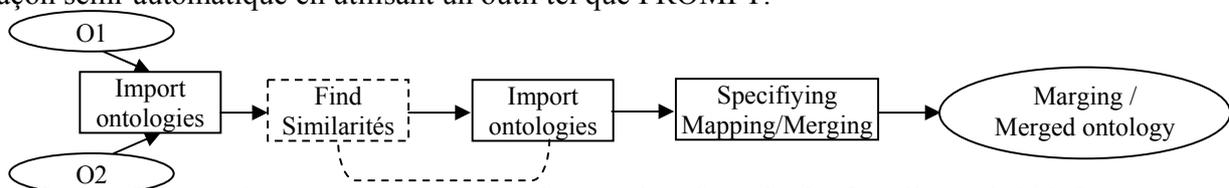


Figure 1.11 : Processus de mapping d'ontologies [Bruijn et al. 2005]

Le processus de mapping est rarement un processus complètement automatique. Comme le précise [Klein 2001], le mapping d'ontologies constitue une tâche complexe du fait qu'il nécessite la compréhension de la sémantique liée aux concepts à lier. Ceci implique que le processus de mapping ne peut être complètement automatisé. Il nécessitera sans doute toujours l'intervention humaine. C'est

d'ailleurs pour cette raison que tous les outils existants ne peuvent que, dans les meilleurs des cas, suggérer à l'utilisateur des mappings potentiels. Nous devons noter que les mappings d'ontologies peuvent conduire parfois à des changements dans au moins une ontologie source. Ces changements peuvent conduire à leur tour à de nouvelles versions d'ontologies. De plus, la translation d'ontologies est parfois opérée dans le cas d'utilisation de langages ontologiques différents. Mais ce dernier point ne nous intéresse pas dans la mesure où l'on s'insère dans le cadre d'un langage de représentation unifié.

VII. MESURES DE SIMILARITE ET DE DISTANCES

Le processus mapping nécessite généralement de mettre en œuvre le calcul d'une distance de similarité qui est calculée plusieurs façons (similarité syntaxique / linguistique, similarité sémantique, etc.). Mais avant de rentrer dans le détail des méthodes utilisées dans le mapping, définissons la similarité et quelques fonctions de distance qui sont souvent utilisées.

Définition 1 (similarité) : une mesure de similarité $\sigma : O \times O \rightarrow \mathbb{R}$ est une fonction qui exprime la similarité entre deux entités tels que :

$$\begin{aligned} \forall x, y \in O, \sigma(x, y) &\geq 0 \text{ (Positivité)} \\ \forall x \in O, \forall y, z \in O, \sigma(x, x) &\geq \sigma(y, z) \text{ (Maximalité)} \\ \forall x, y \in O, \sigma(x, y) &= \sigma(y, x) \text{ (Symétrie)} \end{aligned}$$

De façon duale, il est possible de définir la relation de dissimilarité

Définition 2 (dissimilarité) : une mesure de dissimilarité $\delta : O \times O \rightarrow \mathbb{R}$ est une fonction qui exprime la différence entre deux objets tels que

$$\begin{aligned} \forall x, y \in O, \delta(x, y) &\geq 0 \text{ (Positivité)} \\ \forall x \in O, \delta(x, x) &= 0 \text{ (Minimalité)} \\ \forall x, y \in O, \delta(x, y) &= \delta(y, x) \text{ (Symétrie)} \end{aligned}$$

Définition 3 (distance) : une distance ou métrique $\delta : O \times O \rightarrow \mathbb{R}$ est une fonction de dissimilarité qui vérifie les relations suivantes

$$\begin{aligned} \forall x, y \in O, \delta(x, y) &= 0 \text{ ssi } x = y \\ \forall x, y, z \in O, \delta(x, y) + \delta(y, z) &\geq \delta(x, z) \text{ (Inégalité triangulaire)} \end{aligned}$$

Définition 4 (ultra-métrique) : une ultra-métrique $\delta : O \times O \rightarrow \mathbb{R}$ est une métrique qui vérifie

$$\forall x, y, z \in O, \delta(x, y) \leq \max(\delta(x, z), \delta(y, z)) \text{ (Inégalité ultra-métrique)}$$

De plus, on met généralement en œuvre des fonctions normalisées dans la mesure où on restreint l'ensemble image à $[0, 1]$. Les similarités (dissimilarités) normalisées sont respectivement notées σ (resp. δ). Il est également trivial de constater que $\sigma = 1 - \delta$.

Toutes les distances présentées précédemment sont relativement simples. Or la réalité est plus complexe, les objets ne sont pas aussi simples, ils sont souvent décrits sous forme de vecteurs pour prendre en compte la complexité des paramètres qui les caractérisent. Aussi, les métriques précédentes sont généralement étendues à n dimensions. On peut citer, et à titre d'exemple, les deux distances les plus simples, celle de Minkowski et la somme pondérée.

Définition 5 (distance de Minkowski) : Soit O un ensemble d'objets qui peuvent être analysés selon n dimensions. La distance de Minkowski δ_M entre deux objets x et y est définie en fonction de la distance traditionnelle δ (dissimilarité) comme suit :

$$\forall x, y \in O, \delta_M(x, y) = (1/p) \sum \delta(x_i, y_i)^p$$

où p est un paramètre réel. ($p=2$: distance euclidienne, $p=\infty$: distance de Chebichev)

Définition 6 (somme pondérée) : Soit O un ensemble d'objets qui peuvent être analysés selon n dimensions. La distance somme pondérée δ_S entre deux objets x et y est définie en fonction de la distance traditionnelle δ (dissimilarité) comme suit :

$$\forall x, y \in O, \delta_S(x, y) = \sum \omega_i * \delta(x_i, y_i)$$

VIII. DECOUVERTE DE MAPPINGS

Selon l'avis de plusieurs auteurs, la découverte de mappings constitue l'un des points fondamentaux de l'intégration sémantique [Kalfoglou et Schorlemmer 2003-b], [Noy 2004], [KnowledgeWeb 2004]. La découverte de mappings consiste à chercher des correspondances entre les ontologies. Cette recherche peut se faire manuellement, de façon interactive, de façon semi automatique, ou de façon automatique. On distingue principalement deux approches de découvertes ainsi qu'une panoplie de méthodes pouvant être utilisées pour chacune de ces approches [Noy 2004] : la découverte basée sur une ontologie partagée; et la découverte basée sur des techniques heuristiques ou l'apprentissage.

La première approche repose sur l'utilisation d'une ontologie supérieure et qui est ensuite spécialisée en fonction des spécificités de chaque application. Dans ce cas, la recherche de correspondances entre deux ontologies est alors facilitée par cette ontologie supérieure partagée. La seconde approche comprend des techniques basées sur des heuristiques ou des techniques d'apprentissage qui sont utilisées dans le processus de découverte de mappings entre les ontologies c'est le cas de l'architecture multi ontologie.

VIII.1 Découverte basée sur des heuristiques ou l'apprentissage

Cette approche concerne la découverte de mappings ontologiques sans utilisation de vocabulaire en commun c'est l'approche qui est utilisée dans l'architecture multi ontologie. D'une manière générale, les approches basées sur des techniques heuristiques utilisent des techniques exploitant les composants lexicaux et structurels des définitions des concepts afin de chercher des correspondances entre les concepts. Dans le cadre des approches basées ontologies (on se restreint au domaine ontologique), l'approche basée sur des heuristiques exploite la sémantique véhiculée par les caractéristiques des ontologies à mapper et qui peuvent être [Noy 2004] :

- noms des concepts ou la description des concepts en langue naturelle
- la taxonomie des concepts (relations is-a)
- les autres relations (lien part-of, instance-of, etc.)
- définitions des propriétés (contraintes: domaines, rangs, restrictions)
- les instances des concepts
- la description des classes (outils basés DL).

VIII.2 Méthodes de découverte de mappings

Il existe dans la littérature une multitude de méthodes de découverte de mappings [Sekt 2004][KnowledgeWeb 2004]. [KnowledgeWeb 2004] propose une classification des principales méthodes. La classification selon le type de technique de mapping utilisée permet de distinguer deux grandes catégories de méthodes : les méthodes individuelles et les méthodes combinées.

a- Les méthodes individuelles : reposent sur l'utilisation d'une seule technique, et on distingue les méthodes basées sur des instances, et les méthodes basées sur des schémas.

1- les méthodes basées sur des instances: ces méthodes exploitent les instances des bases de données pour définir des mappings en se basant fondamentalement sur des similarités linguistiques;

2-les méthodes basées sur des schémas : ces méthodes exploitent les schémas de bases de données pour définir des mappings en utilisant principalement des techniques heuristiques ou des techniques formelles pour calculer le rapprochement entre les éléments des schémas.

b- Les méthodes combinées permettent d'associer plusieurs techniques en vue de définir les mappings. On distingue les méthodes hybrides et les méthodes composites.

1- Les méthodes hybrides : sont des méthodes basées sur des techniques qui sont considérées comme une combinaison de deux ou plusieurs approches individuelles;

2- Les *méthodes composites* sont celles qui sont basées sur un assemblage (séquence) de deux ou plusieurs méthodes. Les méthodes hybrides ne sont pas des méthodes composites dans la mesure où elles sont élémentaires non décomposables.

La seconde classification permet de décomposer les méthodes de mappings en : méthodes terminologiques, structurelles, extensionnelles, et sémantiques.

a- 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*, ou alors des *méthodes basées sur des langages*.

b- Les méthodes structurelles se focalisent 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 :

1- 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);

2- 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, etc.

C- 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 :

1- des méthodes de comparaison d'extensions communes : qui considèrent que deux concepts sont similaires si leur intersection se réduit à l'un des concepts (en termes d'instances);

2- 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;

3- des méthodes de comparaison basée sur l'appariement : qui considèrent que les éléments à comparer sont ceux qui présentent des similarités.

D - Les méthodes sémantiques sont très souvent 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).

- En plus de ces méthodes, il y a lieu d'ajouter les méthodes d'apprentissage. Ces méthodes exploitent les techniques développées dans le domaine d'apprentissage comme l'analyse formelle des concepts (Formal Concepts Analysis) [Stumme et Madche 2001], apprentissage Bayésien [Berlin et Motro 2002] ou les réseaux de neurones [Li et Clifton 1994].

VIII. OUTILS DE MAPPING

Les différents types de méthodes précédemment présentés ne sont pas exclusifs dans la mesure où plusieurs d'entre eux peuvent être mis en œuvre simultanément. Le tableau suivant (tableau 1.2) permet de résumer le croisement des principaux outils de mappings avec les principales méthodes présentées.

Les colonnes de ce tableau correspondent aux types de méthodes de mappings précédemment présentés :

- T : Terminologique
- TS : Terminologique basée Texte,
- TL : Terminologique basée Langage (TL));
- I : Structurelle Interne;
- S : Structurelle Externe
- ST : Structurelle Terminologique,
- SC : Structurel Cyclique),
- E: Extensionnelle,
- M: Sémantique (basée Modèle),
- U : Interaction avec l'Utilisateur.

Nous avons choisi seulement de présenter succinctement quelques exemples d'outils qui nous paraissent les plus représentatifs, à savoir : **FCA-Merge**, **IF-Map**.

L'outil **FCA-Merge** (Formal Concept Analysis - Merge) [Stumme et Madche 2001] permet de comparer deux ontologies qui possèdent un ensemble d'instances partagées ou un ensemble de documents annotés. Basé sur des techniques formelles d'analyse des concepts (et aussi des techniques extensionnelles et structurelles), cet outil permet de prendre en charge les équivalences ainsi que les liens de spécialisation. Les résultats de cette méthode peuvent être utilisés par un cogniticien afin d'effectuer la fusion d'ontologies.

IF-Map [Kalfoglou et Schorlemmer 2003] permet d'identifier automatiquement les mappings en se basant sur la théorie des flux d'information en exploitant des techniques structurelles et extensionnelles. IF-Map permet de générer à partir de deux ontologies un isomorphisme logique (mapping ontologique), et qui sont ensuite traduits en mappings en utilisant la théorie des flux.

SYSTEME	T	TS	TL	I	S	ST	SC	E	M	U
Multikat	X		X	X	X	X				
FCA-Merge						X		X		
IF-map						X		X		
APrompt	X			X	X	X				X
Cupid	X	X	X	X	X	X				
QOM	X	X		X	X	X	X	X		
OLA	X	X		X	X	X	X	X		
Rondo		X			X		X			X
T-tree					X			X		
S-match	X	X	X			X			X	
Buster				X		X			X	
Glue								X		
Coma	X	X			X					X

Table 1.2 : Panorama des principaux outils de mapping (adapté de [KnowledgeWeb 2004])

IX. CONCLUSION

Atteindre l'interopérabilité entre systèmes représente un défi majeur. L'interopérabilité comporte deux aspects, un aspect syntaxique et un aspect sémantique. L'écart syntaxique entre concepts de différents systèmes résulte de l'utilisation de modèles de données différents d'un système à l'autre. Quant à la différence sémantique, elle est issue des différences de compréhension et d'interprétation des informations partagées entre différents domaines d'application.

Nous nous intéressons plus particulièrement à l'interopérabilité sémantique qui vise à donner une sémantique aux informations échangées et à s'assurer que cette sémantique soit commune à tous les systèmes entre lesquels des échanges doivent être mis en œuvre. Parmi les différentes techniques pour réaliser l'interopérabilité sémantique, le mappings d'ontologies est le plus adapté à notre problématique, son objectif principal étant la découverte de correspondances entre les ontologies.

Dans le chapitre suivant l'utilisation des systèmes multi-agents pour la mapping entre les ontologies qui est le but recherché dans nos travaux et la conceptualisation de notre approche, nous intéressons ici à comparer avec les approches précédentes.

CHAPITRE 02 :

SYSTEMES MULTI-AGENTS POUR MULTI ONTOLOGIE

I.INTRODUCTION

Les avantages d'un SMA sont multiples ; Ils sont particulièrement bien adaptés pour les environnements distribués. Ils permettent de gérer la distribution des données et des traitements. Tout d'abord, la technologie "*agent*" permet une approche modulaire du problème, De plus, la capacité à communiquer entre eux sans intervention externe. Cela rend possible un échange d'informations. En effet, le processus doit être exécuté dans sa globalité, chaque agent effectuant un ensemble de tâches bien défini et transmettant éventuellement aux autres agents ses résultats. Enfin, le concept de mobilité rend possible le déplacement des agents vers les sources de données.

II.SYSTEMES MULTI-AGENTS ET ONTOLOGIES

Depuis quelques années, les termes « systèmes multi-agents » et « ontologies » sont souvent mentionnés ensemble dans la littérature. Sans y prêter attention, nous pourrions simplement supposer qu'il s'agit là de deux domaines en vue pour expliquer cette présence conjointe. Toutefois, en y regardant de plus près, il s'agit bien d'éléments complémentaires. L'usage veut que les ontologies soient utilisées pour permettre le fonctionnement des systèmes multi-agents. En effet, les systèmes multi-agents, en particulier s'ils sont ouverts, soulèvent de nombreuses questions liées à la communication. Or, les ontologies permettent de s'attaquer à des problèmes communs de langage et de représentation du monde. De plus, comme nous l'avons vu au chapitre précédent, les ontologies ont un coût de construction et de maintenance qu'il convient de réduire au maximum. Des systèmes décentralisés comme les systèmes multi-agents sont alors une possibilité pour atteindre ce but, tout comme l'utilisation de techniques d'apprentissage.

Nous allons donc examiner les différentes relations présentes entre les systèmes multi agents et les ontologies, et plus particulièrement, l'utilisation des systèmes multi-agents pour la communication entre les ontologies qui est le but recherché dans nos travaux.

II.1 Ontologies pour les systèmes multi-agents

Le lien le plus répandu entre systèmes multi-agents et ontologies est l'utilisation d'ontologies pour faire fonctionner les systèmes multi-agents [Gandon, 2002], en particulier pour ce qui concerne les agents dans le Web sémantique et les Web services [Greenwood et al., 2007]. Dans ce

cas, les agents se réfèrent à une ontologie connue pour mettre en oeuvre un langage commun et être capable d'interagir. Les ontologies sont alors un moyen de médiation entre agents logiciels, ce qui est une des raisons de leur définition.

Il existe également d'autres types d'usage des ontologies à l'échelle du système. Par exemple, un ensemble fixe d'ontologies donné initialement au système sera utilisé par tous les agents pour effectuer un traitement sur un ensemble de données [Elmore et al., 2003]. Dans ce cas le langage des agents est commun, et les ontologies leur permettent uniquement d'interpréter un flux de données en provenance de l'extérieur du système.

Un autre système utilisant des ontologies est COMMA[Bergenti et al., 2002]. Dans ce dernier, chaque ontologie est encapsulée dans un agent et, en collaboration avec un agent interface, aide l'utilisateur à ajouter des documents dans une base de connaissance, ou à retrouver des documents. Pour cela, l'agent ontologie mis en oeuvre est une aide pour construire correctement les méta-données du nouveau document, ou la requête de recherche.

II.2 Ontologies pour les agents

Parfois, chaque agent est porteur de sa propre ontologie et est capable de la modifier en fonction des interactions avec d'autres agents [Siebes et van Harmelen, 2002]. Dans ce cas, nous sommes plus proches d'ontologies utilisées comme représentation du monde pour chacun des agents d'un système ouvert. Les formalisations employées sont généralement basées sur XML, comme OWL, aujourd'hui la plus répandue. L'inconvénient principal de leur utilisation provient de l'impossibilité d'avoir un point de vue unique sur un domaine.

Ainsi, pour un système ouvert, il est tout à fait possible d'avoir plusieurs ontologies d'un même domaine donné. On retrouve alors des problèmes de communication entre ces agents, ce qui justifie le besoin de système d'alignements d'ontologies. Aujourd'hui, la majorité de ces systèmes est centralisée, mais des systèmes décentralisés commencent à apparaître pour effectuer cette tâche [Laera et al., 2007]. L'intérêt pour les agents de s'appuyer sur des ontologies explicites et au même format réside justement dans cette possibilité de comparer les représentations du monde auxquelles ils font référence.

II.3 Systèmes multi-agents pour les ontologies

Comme nous venons de le voir, les systèmes multi-agents gagnent à s'appuyer sur des ontologies pour fonctionner et l'alignement entre ontologies est une des conditions de leur bonne communication. On trouve donc des systèmes multi-agents prenant en charge des calculs sur des ontologies différentes afin de les aligner [Lister et al., 2006]. Ces systèmes jouent alors un rôle de médiateur entre des ontologies en identifiant leurs points communs, en utilisant par exemple des métriques appropriées [Hernandez, 2005] sans qu'eux-mêmes ne disposent de connaissance sémantique.

En revanche, il est un peu plus rare de trouver, dans la littérature, des exemples où les systèmes multi-agents sont une aide pour la construction de ressources ontologiques. Dans ce cas, ils sont le plus souvent utilisés comme support pour des ateliers de construction d'ontologies collaboratifs et n'exploitent pas de résultat en provenance de textes [Bao et Honavar, 2004]. Ils sont essentiellement là pour assurer la cohérence du produit modélisé en suivant les modifications concurrentes des utilisateurs.

III. ETAT DE L'ART

La recherche concernant mapping d'ontologie entre les différentes relations sémantiques en utilisant des ontologies, taxonomies lexicales, sémantiques similitudes, les similitudes linguistiques des termes, les relations taxonomiques et des textes d'information [Jiang et Conrath, 1997], [Klein et al., 2002], [Rodriguez et Egenhofer, 2003], [Maedche et Staab, 2001], [Mitra et Wiederhold, 2002], [Guarino et Welty, 2001] a été menée par plusieurs groupes. Les objectifs de recherche ainsi que les domaines d'application sont très diverses.

Dans le cas du SMA ouvert hétérogène, le FIPA a identifiée et analysé les Différents types de problèmes d'interopérabilité qui se posent et a, par conséquent, proposé la création d'un agent d'ontologie (AO) pour aider la communauté des agents dans le mapping des ontologies. Dans cette section, nous fournissons une brève synthèse des principales contributions dans ce domaine, qui sont:

✚ [Steels, 1998] propose un système adaptatif complexe approche d'une ontologie partagée et un lexique dans une population d'agents distribués seulement aux interactions locales et aucun organisme central de contrôle. Les ontologies sont adaptatives. Les agents ont une connaissance limitée, ils ne peuvent pas inspecter les états internes des autres agents. De plus, les agents locaux se limiter à des interactions avec d'autres agents et sont autonomes, ils acquièrent leur propre savoir et de décider par eux-mêmes ce qu'il faut faire et comment communiquer. L'interaction entre les agents est modélisée comme un jeu, de sorte que la langue est un jeu de langage. Pour effectuer un acte de langage, l'orateur doit conceptualiser les objets de manière à trouver une description du sujet qui le distingue d'autres objets dans le contexte. Cela exige une ontologie, un ensemble de traits distinctifs Fonctionnalités. L'intervenant doit trouver les mots pour coder les caractères distinctifs ainsi constaté et de transmettre ces paroles à l'auditeur. L'auditeur reçoit le message transmis, il décode en une ou plusieurs interprétations possibles et vérifie si les interprétations sont conciliables avec la situation actuelle.

Chaque ensemble de mots et d'un lexique qui est initialement vide a son propre agent. Un agent peut associer un seul mot à plusieurs sens et un sens de plusieurs mots. Les mots sont jumelés en utilisant les mesures de distance. Lors de l'exécution d'expériences avec tous les agents SMA utiliser les mêmes mécanismes de création d'ontologies.

✚ [Bailin et Truszkowski, 2002] décrivent une approche de l'ontologie de négociation qui permet d'information sur Internet pour résoudre les inadéquations des agents en temps réel, sans intervention humaine. Le système utilise la base de données lexicale WordNet comme source de données pour étendre le concept d'ontologie dans chaque répertoire. Toutefois, l'essentiel repose sur le mécanisme d'échanges d'information entre les agents depuis WordNet par lui-même ne permet pas d'interpréter les agents des uns et des autres concepts. Ces échanges de données sont structurées par les règles Ontologie du processus de négociation (ONP) et de permettre à chaque agent de demander des éclaircissements sur les messages précédents et de la confirmation ou de correction de la tentative d'interprétation. Le projet consiste ONP des tâches d'interprétation, de clarification, d'évaluation et de la pertinence évolution d'ontologie.

Bien que l'interprétation et la clarification des tâches peut prendre la forme d'une simple substitution par des synonymes, le protocole prévoit des formes plus complexes telles que la logique formelle des définitions, des descriptions et des évaluations opérationnelles à une notion de sens. Dans cet ouvrage, les auteurs explorent la possibilité d'une ontologie de l'évolution en cours sous le contrôle d'un agent autonome dédié. En outre, ils définissent un outil pour fournir des API Fonctionnés pour soutenir l'évolution d'ontologie. Les termes échangés entre les agents standards sont soit des questions à poser ou des réponses à des questions. Tant les contenus peuvent être considérés comme des mots clés qui décrivent le document qui est soient désirés (requête) ou trouvé (réponse). Le processus de négociation se termine par un ou deux agents de modifier leur ontologie d'introduire un nouveau concept, une nouvelle caractéristique ou tout simplement un nouveau terme pour un concept existant.

✚ [Van Eijk et al., 2001] mis au point un mécanisme de communication dans lequel les traducteurs entre les vocabulaires d'agents sont générés. Ces traducteurs ne sont pas définis par le programmeur, mais au lieu de cela, ils sont construits dynamiquement pendant l'exécution du système et sont fondées sur les renseignements aux agents de change et sur leurs sous-tendent les ontologies. La dynamique de construction de la traduction a lieu pendant les étapes successives de la communication, dans laquelle une information est échangée entre les agents. Cela signifie que les informations fournies par un agent révélateur sont traduit dans le vocabulaire de la demande agent. Au cours de chaque étape de communication, les traductions de la précédente communication de tours affinée afin d'inclure les présenter de l'information échangée.

Les auteurs supposent que les ontologies sont représentées par une théorie du premier ordre et, dans la situation typique dans laquelle cette théorie est finie, simplement par une formule du premier ordre. Dans cette approche, il n'ya pas de partage mondial d'ontologie mais chaque ontologie a sa propre agent. Initialement, avant de lancer le système, pas de connexions entre les différentes ontologies, des agents sont pris en charge. En revanche, les connexions sont construites dynamiquement pendant l'exécution. De plus, ces connexions sont établies sur une base axée sur la

demande, seules les connexions qui sont nécessaires au cours de certaines étapes de communication sont établies entre les ontologies des deux agents.

✚ [Tzitzikas et Meghini, 2003] envisager de peer-to-peer dans les systèmes de pairs qui emploient des taxonomies pour décrire le contenu de leurs objets et de la formulation sémantique fondée sur les requêtes d'autres pairs du système. Chacun utilise ses propres pairs en matière de taxonomie et est équipé d'inter-mappings de la taxonomie afin de mener à bien les tâches de traduction. Comme ces systèmes sont ad-hoc, les pairs devraient être en mesure de créer ou de réviser des mappings à la demande et au moment de l'exécution. Puisqu'il n'y a pas de serveur central ou d'un médiateur, chaque participant doit avoir la source (ou être en mesure de créer) des mappings entre ses articulations ou modèle conceptuel et les modèles conceptuels de ses voisins afin d'être en mesure de traduire les requêtes reçues à des questions qui peuvent être Compris (et donc une réponse) par le destinataire sources. Ce travail présente un *data-driven* méthode automatique de taxonomie pour l'articulation, que les auteurs appellent une méthode illustrative parce que la signification de chaque terme est expliquée par ostension.

Le mapping est découvert, en examinant comment les termes sont utilisés à l'indexation des objets. Cette méthode ne fait aucune hypothèse sur la façon dont les participantes taxonomies sont construits ou comment elles sont utilisées, mais il exige la présence de deux bases de données qui contiennent plusieurs objets communs.

✚ [Burnstein et al., 2003] Ont esquissé une approche de la dérivation automatique de glue code " pour la traduction des programmes de production d'une source d'entrée à l'agent de la représentation d'un agent cible. Les auteurs décrivent comment les agents peuvent tirer autonome transformation fonctions, Pour traduire les calculs entre ontologies hétérogènes. Dans cette approche on suppose qu'il existe une théorie commune sous-jacente à laquelle les privés ontologiques des agents sont liés. Toutefois, l'approche ne fournit pas de méthode permettant d'établir une mapping.

La théorie commune de la demande de domaine fournit des symboles des concepts, des opérations et des relations dans le domaine. Ses axiomes limiter la signification du vocabulaire.

✚ [Doherty et al., 2004] combinent la logique basée sur des techniques de raisonnement approximatif. Cette proposition prévoit des logiciels ou des agents robotisés avec la possibilité de poser mutuellement des questions concernant approximative termes inconnus ou peu clairs et des actions. Agents locaux ont une ontologie composée de concepts / relations, qui est un sous-ensemble d'une ontologie globale. Bien que les agents ont des concepts communs, il existe quelques concepts disjoints / relations. Chaque agent peut communiquer dans la langue des autres agents (à l'aide d'un premier ordre ou fixpoint formule) et a une fonction de médiation.

Les formules générées peuvent être comprises par l'agent parce qu'elles sont formulées en utilisant ses propres concepts / relations. Par conséquent, ces formules peuvent être utilisées pour interroger la base de données relationnelle de l'agent. Les données de l'intimée ou de bases de connaissances associées à ces agents sont formalisées et estimé que les bases de données et les questions qui peuvent être posées sont représentés comme premier ordre ou fixpoint requêtes. Les plus faibles et les plus fortes conditions suffisantes nécessaires sont utilisés pour modéliser des requêtes approximatives avec des sous-langues communes à chaque paire d'agents impliqués dans un énoncé.

✚ [Williams et al., 2003] proposé une méthodologie pour les agents de développement local consensuel les ontologies comme moyen de favoriser la communication au sein d'un système multi-agents des agents de B2B. Les agents doivent être en mesure de trouver des services connexes (ontologie de concepts) entre une organisation d'ontologies inter-et intra-organisation Agents ontologiques peuvent avoir diverses ontologies, mais, avant avoir une vaste traduction, elles doivent être en mesure de lier les services à un niveau légèrement plus élevé. L'approche permet de trouver des similitudes sémantiques et syntaxiques, en comparant, pour chaque paire d'agents, les deux ontologies sans l'utilisation d'une ontologie globale et de la fusion de ces ontologies dans une locale consensuelle ontologie. Ils fournissent un outil simple pour créer des ontologies qui sont stockées en XML et représentés. Les auteurs montrent comment cette approche fonctionne par fusion de deux services de messagerie Web de diverses ontologies.

✚ [Wiesman et Roos, 2004] proposent un domaine indépendant de la méthodologie de traitement des problèmes d'interopérabilité en apprenant des correspondances entre ontologies. La méthode d'apprentissage est basée sur l'échange d'exemples de concepts définis dans les ontologies. Ils se concentrent sur l'établissement d'une correspondance entre deux concepts, un de chaque ontologie. Aucune restriction n'est placée sur la structure d'un concept. Un concept peut être défini comme une agrégation d'attributs et de sous-concepts. Cette agrégation peut même, directement ou indirectement, de contenir le concept qui est en cours de définition.

La manière dont les agents établir une mapping est un jeu inspiré de la langue appelé l'attention conjointe, où un agent essaie d'interpréter les déclarations d'un autre agent de réalisation et d'évaluation des associations entre les déclarations reçues et catégorisations des entités observées. Après avoir établi les concepts correspondants, l'un des agents propose des associations entre les étiquettes de deux déclarations sur la base de la proportion correspondante de mots que les deux marques ont en commun.

✚ [Van Diggelen et al., 2005] résoudre le problème de l'établissement d'une communication appropriées dans un vocabulaire formel et abstrait. Chaque agent a une ontologie qui est incompréhensible pour les autres agents. Elles supposent qu'il existe un terrain commun dans lequel

l'ontologie privé (ontologie de chaque agent) est enracinée. Ce terrain permet de découvrir les relations entre les concepts étrangers et leurs ontologies privé. Tous les termes utilisés dans une ontologie peuvent être exprimée comme une combinaison complexe de termes fondamentaux définis dans le sol ontologie. Les ontologies sont formalisées en utilisant L, qui est égale à la description logique de la langue - Attributs Concept Langue Compléments, mais avec sans rôles. Le vocabulaire de la communication est également formalisé comme une ontologie. Les agents s'adapter à un vocabulaire de la communication, ils doivent être en mesure de découvrir les relations entre les concepts de communication dans le vocabulaire et les concepts dans leur propre ontologie.

Le privé concepts de chaque agent peut éventuellement être défini par le biais de l'ontologie partagée terrain puisque chaque concept primitif est présent sur ce terrain ontologie. Pour préserver la solidité, les agents traduire (l'adoption d'une fonction de distribution) des privés concepts équivalents ou plus généralement partagée des concepts. Comme une communication optimale, le vocabulaire devrait contenir des concepts bien précisés, les concepts partagés ne devraient pas être trop générale. Un agent d'envoi devrait être autorisé la traduction d'une affirmation privé une concept plus général dans le vocabulaire de la communication aussi longtemps que le concept reste traduit équivalent à la conception originelle relative à l'agent de l'ontologie.

✚ [Malucelli et Oliveira, 2006] est centré sur la résolution de conflits de la négociation dans un domaine B2B. ils définissent un ensemble de services pour faire une solution pour les problèmes de l'interopérabilité qui se posent au cours de la communication inter-agent. Le plus important est le service de résolution de conflit ontologique.une méthodologie permettant d'évaluer la similitude entre les concepts représentés dans les différentes ontologies sans avoir besoin de construire a priori une ontologie. Les similitudes identifiés sont considérés comme des ponts entre les éléments impliqués les ontologies et peuvent être utilisés pour soutenir le processus de négociation inter-agent. cette approche utilise un médiateur appelé agent OSAg.

OSAg est un agent responsable de la résolution de tous les conflits de négociations qui ont lieu au sein du SMA. Le privé des ontologies reste inchangé pendant toute la durée du processus de négociation. Les concepts sont appariés par le OSAg mémorisé et conservé pour les futurs cycles de négociation. Le mapping entre les ontologies est établi en comparant, pour chaque paire de concepts, les attributs (regroupés par type de données), et la relation a-part et aussi la description des concepts. Le rapport comprend à la fois des dimensions syntaxique et sémantique. Les auteurs développés des stratégies de dialogue dans lequel les agents peuvent demander d'établir un vocabulaire de communication optimale commun.

IV. NOTRE APPROCHE

Certaines requêtes nécessitent des réponses contenues sur plusieurs sites distincts ; cela exige un traitement au préalable de la requête et un autre traitement post recherche qui consiste à

recomposer les réponses élémentaires pour former un résultat final à l'utilisateur. Cela consiste à développer un modèle de recherche d'information qui intègre des systèmes à base des connaissances ou chacun a sa propre ontologie. Notre objectif est de modéliser les connaissances traduisant les différents points de vue de ces systèmes au sein des ontologies locales afin de pouvoir comparer et rendre interopérables les conclusions fournies par différents systèmes. Nous focalisons dans le cadre de nos travaux sur l'approche d'intégration des ontologies basée sur des mappings. Cette approche permet d'apporter plus de flexibilité dans les environnements aussi complexes.

Les mappings bidirectionnels sont à notre sens la catégorie de mappings les plus favorables. Ils permettent de définir l'approche la plus réaliste et également la plus flexible. Les approches multi-ontologies sont les architectures d'ontologies qui peuvent être concernées, L'intérêt de cette approche est le fait que les sources peuvent être développées de façon indépendante, on peut avoir dans le cadre des architectures, des mappings point-à-point (un-à un ou one-to-one mapping) qui permettent de créer des mappings entre chaque paire d'ontologies. Bien qu'elle soit relativement lourde, l'approche multi-ontologies permet de préserver la nature décentralisée des systèmes d'information à intégrer. Mais, on reproduit ainsi la complexité des architectures point-à-point. Une difficulté extrême pour comparer différentes ontologies sources.

Nous pallier cette difficulté par l'intégration des agents ; ils sont des programmes autonomes et communicants, conçus pour interagir avec d'autres agents. Dans notre cas les agents sont cognitifs ; ils manipulent des connaissances ; ces connaissances sont représentées sous forme d'ontologies. Chaque ontologie leur propre agent ontologique (AO) les systèmes multi-agents gagnent à s'appuyer sur des ontologies pour fonctionner le mapping entre les différents ontologies et, en collaboration avec un agent utilisateur, et utilise pour communiquera avec l'utilisateur. Pour communiquer entre eux les agents s'échangent des messages en respectant un formalisme basé sur la théorie des actes de langage. Le langage de communication FIPA-ACL (le plus en vogue actuellement), nous proposons un protocole de mapping qui permet également de spécifier l'ontologie utilisée pour exprimer le contenu du message.

V. ANALYSE COMPARATIVE

La section précédente résume les travaux les plus pertinents qui abordent la communication entre les agents et les ontologies. Dans particulier, nous nous intéressons à comparer ces déférentes approches certaines questions importantes se posent:

1. Est-ce que les ontologies ont leurs propres agent Privé (AP) ou / et Instances partagées (SI) et/ ou Modalités partagées Ground (SGT)?
2. Est-ce que les auteurs utilisent les outils de développement de l'ontologie (Ontology Editor (OE)) pour créer les ontologies? Si oui, quels sont les éditeurs ontologie utilisé?

3. Quel Langage d' Ontologie (AL) est utilisé pour représenter les ontologies?
4. Y est-il des mécanismes de création d'ontologies Obligatoire (MOCM), un rédacteur et un langage spécifique, impliqués pour l'ensemble des agents dans le SMA?
5. Chaque agent est Individuel (IndA) responsables de la construction de la mapping existante entre les différentes ontologies ou y est-il une ontologie globale Mediator (MA), et un agent chargé de fournir ce service?
5. S'agit-il d'une application B2B (B2B-AD)?
7. Comment sont créées les ontologies classées (Light weight Ontologies (LwO) ou Heavy weight Ontologies (HwO))?
6. Quelles sont les méthodes de mapping des ontologies (OMM) utilisé?

Nous avons créé trois tables où ces principales caractéristiques sont comparées. Certaines des approches analysées mettre certains détails au sujet de certains éléments comparatifs.

Approches	AP	SI	SGT	OL	OE
Steels, 1998	X		X	Language game	
Bailin and Truskowski, 2002	X			API	
van Eijk et al., 2001	X			First-order theory	
Tzitzikas and Meghini, 2003	X	X	X		
Burnstein et al., 2003	X		X	Data structure	
Doherty et al., 2005	X		X		
Williams et al., 2003	X	X	X	XML	
Wiesman and Roos, 2004	X	X			
van Diggelen et al., 2005	X		X	L	
Malucelli and Oliveira, 2006	X			OWL/Free	Protégé/Free
Notre approche	X			OWL	Protégé

Table 2.1 : comparaison entre les différentes approches ontologique, Langage et Editeur d'ontologie

Approches	MOCM	IndA	MA	B2B-AD	Other AD
Steels, 1998	X	X	X		X
Bailin and Truszkowski, 2002	X	X			X
van Eijk et al., 2001	X	X			
Tzitzikas and Meghini, 2003	X	X	X		X
Burnstein et al., 2003	X		X		X
Doherty et al., 2005	X	X	X		X
Williams et al., 2003	X	X	X	X	
Wiesman and Roos, 2004	X	X		X	X
van Diggelen et al., 2005	X	X		X	X
Malucelli and Oliveira, 2006			X	X	
Notre approche	X	X			

Table 2.2 : comparaison des mécanismes de création d'ontologies, mapping Inter-ontologie et l'application de domaine

Ces tableaux récapitulent les principales caractéristiques de l'état de l'art dans notre domaine de recherche. Nous avons conclu que: la majorité des approches n'utilise pas les éditeurs à construire ontologies; les langues d'ontologie diffèrent, et il n'existe pas de norme de l'ontologie, chacun de ces systèmes utilise un langage d'ontologie spécifique pour faciliter l'interopérabilité. En conséquence, elles nécessitent l'utilisation de la même tous les mécanismes visant à créer des ontologies impliqués. Le mapping ontologique est présente uniquement dans certaines démarches et, si elle est disponible, le mappage est effectué par les différents agents. En conséquence, tout agent qui souhaite participer à ces SMA doit être équipé d'une méthodologie de mapping d'ontologie.

Éviter les approches existantes, soit la résolution de l'hétérogénéité des problèmes ou élaborer une ontologie de domaine à être utilisé par tous les agents de la SMA. Un exemple de la première affaire est l'oeuvre de [Dignum, 2001], qui souligne la nécessité de disposer d'une ontologie commune pour toutes les parties au sein de l'institution, décrivant la fois généraux et concepts tributaires de domaine. Ce dernier cas est représenté par [Vázquez-Salceda et Dignum, 2003]. Ces auteurs utilisent la spécification des normes électroniques au sein des organisations, qui sont formés au sein d'une IE. Ils commencent par un résumé du niveau institutionnel et de l'adresse de la traduction en actes concrets les normes abstraites, qui sont décrits en termes d'une e-organisation de

l'ontologie. Un domaine-ontologie définit le vocabulaire à utiliser par tous les agents dans un e-organisation.

ApprAOches	LwO	HwO	OMM
Steels, 1998	X		Communication linguistique, la distance entre les objets et d'association-set.
Bailin and Truszkowski, 2002	X		Requêtes sur la base de données lexicale WordNet, Trouver des synonymes dans la source agent de l'ontologie, et de fournir des exemples de généralisation à partir des sources de l'ontologie.
van Eijk et al., 2001	X		Le mapping consiste en un ensemble de la traduction des formules qui expriment chacune une équivalence entre les expressions.
Tzitzikas and Meghini, 2003	X		Illustrative articulation de la taxonomie des sources.
Burnstein et al., 2003		X	Pas de correspondance entre les ontologies est fourni.
Doherty et al., 2005	X		Pas de mapping entre les ontologies .
Williams et al., 2003	X		Syntaxique et sémantique similitude et relation sémantique concerne la découverte.
Wiesman and Roos, 2004	X		Méthode d'apprentissage basée sur l'échange d'exemples de concepts défini dans les ontologies.
van Diggelen et al., 2005	X		Fonction de distribution (communication vocabulaire).
Malucelli and Oliveira, 2006	X		Syntaxique et sémantique de similarité entre les attributs, les relations et les descriptions.
Notre approche	X		les systèmes multi-agent à utiliser pour fonctionner et le mapping entre les différentes ontologies.

Table 2.3: Comparaison vue la classification des méthodes de Mapping.

VI. PARTAGE DU SENS ENTRE AGENTS

Nous présentons dans cette partie différentes approches qui abordent le même problème que celui que nous traitons : la compréhension partagée des connaissances au sein d'un système multi-agents (SMA), et dans lequel les acteurs manipulent des ontologies pour représenter leurs connaissances.

VI.1 L'approche "NASA / Knowledge Evolution, Inc" : ONP

L'objectif de cette approche est bien la compréhension de messages. ONP suit une démarche de négociation "terme à terme" qui est dépendante des ressources utilisées pour l'interprétation : Il est en effet difficile de conserver la sémantique d'un terme en lui choisissant un synonyme au hasard, même si ce synonyme est ensuite confronté à l'ontologie de laquelle est issu le terme initial. Signalons que seul le concepteur d'une ontologie connaît suffisamment le sens d'un terme de son ontologie et est donc à priori le seul capable de trouver un terme, manuellement, qui soit équivalent au terme original avec une précision complètement fiable.

Le protocole présenté dans ce projet s'attaque à un problème très similaire à celui que nous traitons, à savoir permettre l'exploitation de bases de connaissances distribuées par des agents logiciels. Il traite également de mise en correspondance d'ontologies, mais utilise d'autres moyens que ceux que nous envisageons dans notre approche. De plus cette approche ne tire pas profit de façon optimale de l'interaction entre deux agents: même si les ontologies sont mises à jour, seuls des nouveaux éléments sont assimilés, sans aucun lien avec l'agent ayant permis cet apprentissage.

VI.2 L'approche "MERIT / Infonomics"

Cette approche, tout comme les autres, s'attaque au problème d'hétérogénéité dans la représentation des connaissances. Le postulat de départ est ici que les alternatives existantes nécessitent des connaissances à priori trop contraignantes, d'où l'hypothèse de l'apprentissage automatique de mappings : cela permet de s'affranchir de la nécessité d'avoir des concepts partagés, des ontologies dérivées les unes des autres, et de l'intervention humaine pour spécifier manuellement les relations entre différentes ontologies. La méthode se base sur la mise en correspondance d'instances de concepts afin de pouvoir communiquer à propos d'un concept subsumant. Elle se propose ainsi de résoudre une partie du problème de l'hétérogénéité des sources de connaissances: les conflits de noms, structurels et représentationnels.

Cette approche est une solution générique au problème d'hétérogénéité des ontologies, mais elle souffre toutefois de quelques limitations. Tout d'abord les mappings obtenus ne sont pas bidirectionnels. De plus ils nécessitent un temps de calcul élevé à cause du parcours des concepts, des envois successifs d'ensembles { concepts + instances }, et du traitement des chaînes

de caractères. La nécessité que les ontologies aient des instances de concepts en commun pour guider les associations limite également son utilisation dans le cas général.

VI.3 L'approche "Alignement réciproque d'ontologies"

Cette approche présente une méthode d'alignement également basée sur l'échange d'instances d'ontologies, mais elle utilise, contrairement aux autres, une représentation ensembliste et définit des catégorisations du domaine qui servent à raisonner sur l'alignement à un niveau logique. Lors d'un "jeu d'alignement", des concepts sont échangés entre deux agents. Selon l'appartenance ou non d'un concept à une catégorie les liens entre les concepts sont mis à jour. Cette méthode repose donc sur l'échange d'objets représentationnels pour mener une action collective coordonnée au sein d'un système multi agents. L'objectif est la convergence des agents vers une ontologie partagée, à partir des catégorisations.

Le processus décrit dans cette approche, comme dans l'approche MERIT, est basé sur la manipulation d'instances ; il faut que des objets soient connus de part et d'autre. Il s'agit donc d'une approche fondée sur une vision extensionnelle. Il n'est pas non plus possible d'automatiser la communication entre deux agents quelconques, mais le processus permet toutefois l'adaptation d'ontologies et la convergence vers une ontologie commune à un groupe d'agents ; la convergence absolue n'a cependant pas été prouvée de manière formelle.

VI.4 Analyse comparative (synthèse)

Nous remarquons cependant que ces approches regroupent un certain nombre de limitations et qu'aucune d'entre elles ne se propose de résoudre le problème dans le cas général. Résumons ici les principales remarques d'ordre général que l'on peut faire en se basant sur ces approches, et afin de concevoir un système qui puisse traiter le problème d'interopérabilité entre agents d'une manière générale. Il est souhaitable d'éviter que :

- * Tous les agents du système aient besoin d'implémenter un mécanisme d'apprentissage ou de raisonnement spécifique (du type jeu de langage entre deux agents), ce qui est une hypothèse trop lourde. Cela nuit également à leur indépendance.
- * Les sources de connaissances soient manipulées au niveau global du système.
- * Des objets représentationnels, des instances de concepts, soient nécessairement connus de différents agents (exprimées en fonction de l'ontologie de chacun).
- * des recherches de correspondances entre concepts ne se basent que sur des données terminales, ou ne prennent même simplement pas en compte les relations existantes entre les concepts.
- * trop de connaissances a priori soient nécessaires.

- * pas d'interaction optimale entre les différents agents.
- * les mappings obtenus ne sont pas bidirectionnels.

C'est sur cette analyse que nous nous basons pour présenter notre approche.

VII. CONCLUSIONS

Nous avons donné une brève synthèse des principales contributions dans le domaine d'utilisation des SMA pour les ontologies, Les différentes approches attaquent au problème de l'hétérogénéité des connaissances et au d'interopérabilité regroupent un certain nombre de limitations. Les systèmes multi-agents à conditions de leur bonne communication. Et sur cette analyse que nous nous basons pour présenter notre approche pour gérer le mapping d'ontologies.

Dans le chapitre suivant comprend notre approche méthodologique, où nous présentons les différentes étapes de notre approche, nous essayerons de donner l'allure suivie, pour le mapping dans l'architecture multi ontologies.

CHAPITRE 03 : *MODELISATION*

D'UNE APPROCHE MULT ONTOLOGIES BASE AGENT

I. INTRODUCTION

Les agents sont des programmes autonomes et communicants, conçus pour interagir avec d'autres agents. Ils manipulent des connaissances ; ces connaissances sont représentées sous forme d'ontologies. Pour communiquer entre eux les agents s'échangent des messages en respectant un formalisme basé sur la théorie des actes de langage. Le langage de communication FIPA-ACL (le plus en vogue actuellement), permet également de spécifier l'ontologie utilisée pour exprimer le contenu du message.

Les différentes approches qui s'attaquent au problème de l'hétérogénéité des connaissances et au d'interopérabilité regroupent un certain nombre de limitations et sur cette analyse que nous nous basons pour présenter notre approche. Nous présentons dans ce chapitre les différentes étapes de notre approche, pour gérer le mapping dans l'architecture multi ontologies.

II. LA PLANIFICATION ET LE WEB SEMANTIQUE

La planification s'intéresse particulièrement à l'utilisation de techniques et de méthodologies pour déterminer les différents systèmes dans un domaine donné (figure 3.1), en précisant quel est le comportement attendu de ces systèmes, ces systèmes échangent, en général, des services afin d'accomplir un objectif global. La distribution identifie les interactions entre différentes composantes; d'une manière plus fine, ces interactions peuvent se décomposer selon plusieurs axes (qui, pourquoi, quoi, quand, comment, où). L'autonomie peut se présenter sous plusieurs formes. Au niveau de la conception, elle désigne le choix du domaine de gestion et la conceptualisation du contexte. En relation avec d'autres systèmes, elle représente la capacité d'un système ou d'un composant de choisir "ses partenaires" pour l'échange de ressources, ainsi que les modes de réaliser ces échanges.

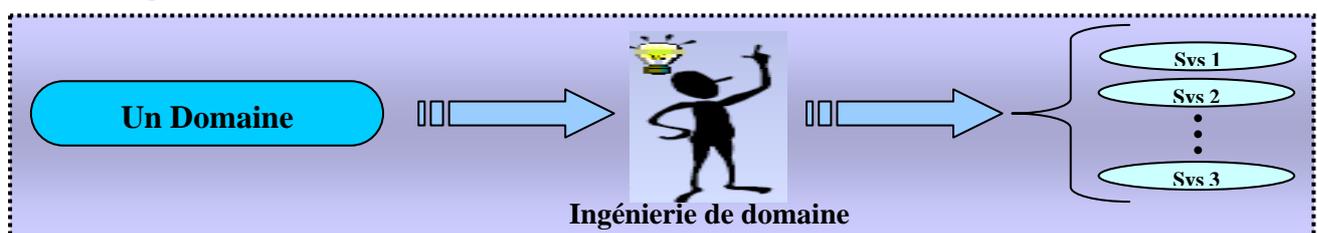


Figure 3.1 : La planification

III. L'INGENIERIE DES BESOINS

L'ingénierie des besoins, autre discipline de l'intelligence artificielle, un système à base de connaissances est bien sûr limité à un domaine bien circonscrit de la connaissance humaine. Les connaissances à modéliser ne sont jamais considérées dans l'absolu mais dans un but opérationnel précis. L'ingénierie des besoins tente de surpasser la vue fonctionnelle de la modélisation conceptuelle. Nous mettons en avant deux dimensions autour desquelles cette tentative est faite :

- L'ingénierie des besoins ne se limite plus à prendre en compte ce que doit faire le système mais cherche à comprendre le "pourquoi" du système. On répond à la question du "pourquoi" dans des termes d'objectifs organisationnels et de l'impact de ces objectifs sur le système de l'organisation.

- L'ingénierie des besoins ne s'intéresse pas directement aux fonctionnalités d'un système, ni à ce qui doit être réalisé par le système tel que cela a été conceptualisé par un analyste en partant d'un ensemble de besoins donnés. Il faut effectuer une exploration détaillée des différentes manières dont le système sera utilisé ainsi que les activités qu'il doit supporter.

Afin de répondre à tous ces objectifs pour chaque système hétérogène de notre domaine notre objectif est de modéliser les connaissances traduisant les différents points de vue de ces systèmes au sein des ontologies locales afin de pouvoir comparer et rendre interopérables les conclusions fournies par différents systèmes.

IV. CONSTRUCTION DES ONTOLOGIES

La construction d'ontologies faite par l'Ingénierie des connaissances peut se révéler complexe, impliquant plusieurs acteurs, diverses connaissances et formalismes. Afin de maîtriser cette complexité, la mise en place d'une méthodologie peut se révéler nécessaire. A l'heure actuelle on peut recenser dans la littérature une multitude de méthodologies, il n'existe pas encore de consensus en matière de normes de construction, et par conséquent il n'existe pas encore de méthode universellement reconnue pour la construction d'ontologies. Ceci relève beaucoup plus du savoir-faire que de l'ingénierie (figure 3.2).

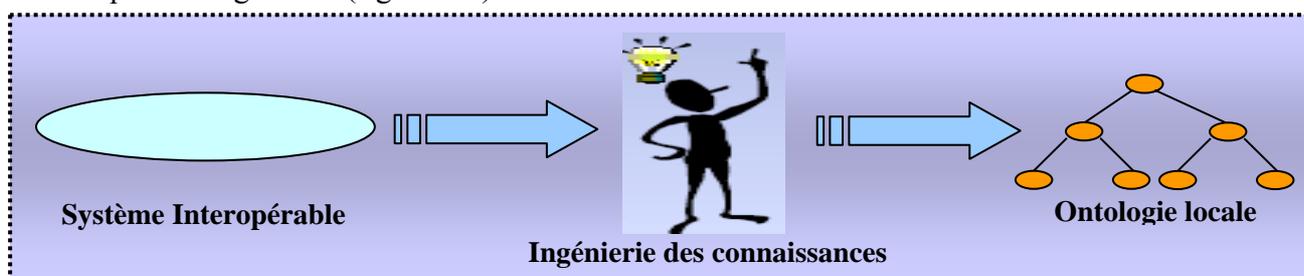


Figure 3.2 : La construction d'ontologie (vue globale)

IV.1 Cycle de développement ontologique

Du fait que les ontologies peuvent s'apparenter à des composants logiciels, Nous devons retenir le fait que le processus de développement n'est fondamentalement pas linéaire et que de nombreux allers-retours sont a priori nécessaires pour bâtir une ontologie locale opérationnelle adaptée aux besoins du chaque système interopérable du domaine donnée. Il ne s'agit donc pas nécessairement d'un cycle en cascade, mais plutôt d'un *cycle en spirale* qui est donc itératif et incrémental. Remarquons également que le modèle de construction d'ontologie tel que présenté part des connaissances à représenter, pour aboutir à une représentation formelle.

Donc le cycle de vie qui nous paraît intéressant, est celui proposé par l'équipe de Fernandez [Fernandez et al. 1997]. Il est fortement inspiré du *cycle en spirale* utilisé en génie logiciel, et permet de développer des ontologies d'une manière prototypique (figure 3.3).

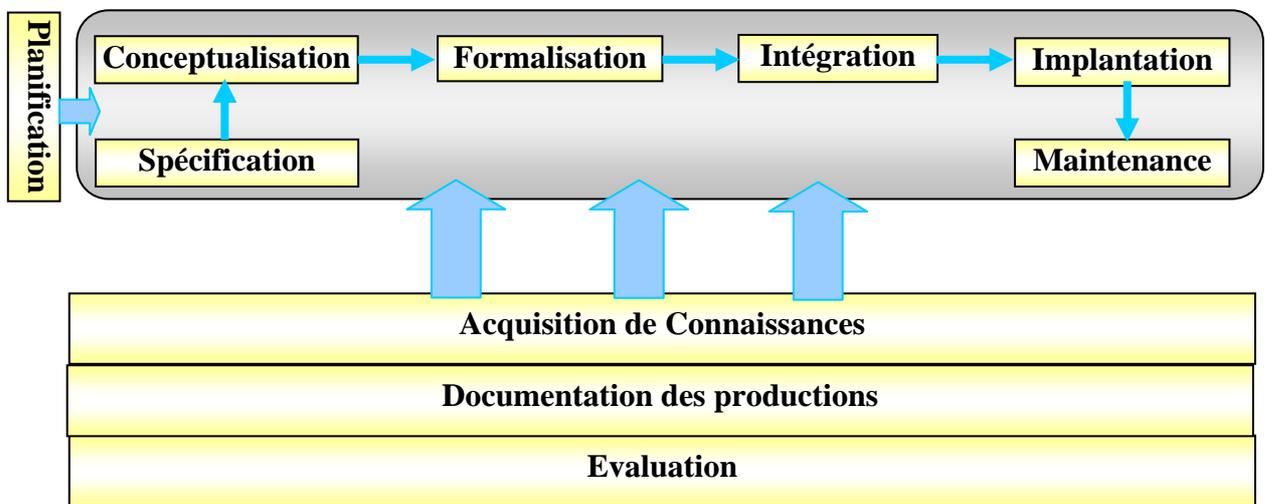


Figure 3.3 : Cycle de vie d'ontologie [Fernandez et al. 1997]

Une fusion de ce dernier cycle de vie a été proposée dans [Gandon 2002] dans le cadre du projet COMMA mené par l'équipe ACACIA de l'INRIA. Ce cycle de vie propose, comme le montre la figure 3.4, une meilleure articulation entre les différentes activités liées à la construction de chaque ontologie locale.

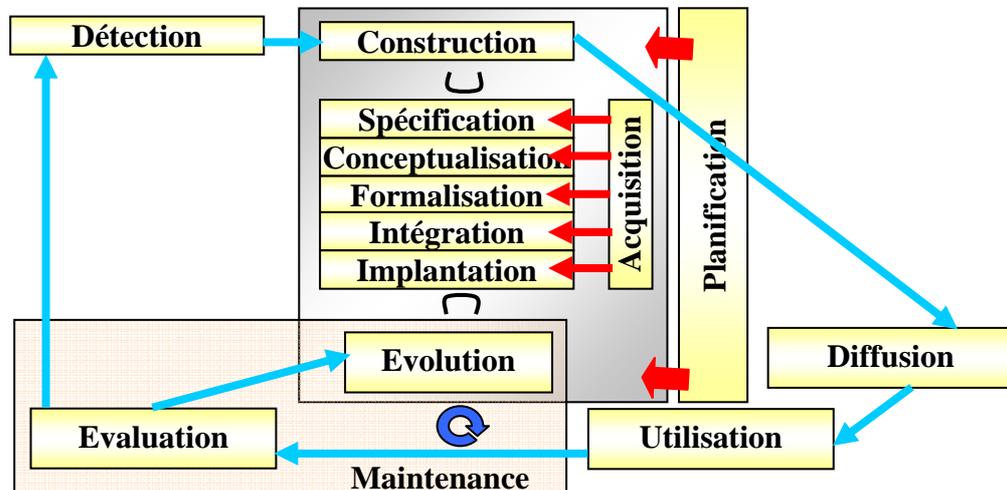


Figure 3.4 : Cycle de vie fusionné [Gandon 2002]

IV.2 Principes pour la construction d'ontologies

Pour mieux guider la construction de chaque ontologie locale, plusieurs auteurs ont proposé quelques principes qui peuvent s'avérer efficaces durant la conception d'ontologies (ontologisation). Nous ne citerons que ceux qui sont à notre sens assez représentatifs, et il s'agit essentiellement des principes proposés par [Gruber 1993] et [Bachimont 2001]. Gruber propose cinq critères génériques permettant de guider le processus d'ontologisation. Il s'agit de [Gruber 1993]:

- *Clarté et objectivité* : l'ontologie devrait fournir des définitions claires et objectives pour les termes, indépendantes de tout choix d'implémentation.

- *Cohérence* : consistance des axiomes afin de pouvoir formuler par la suite des inférences cohérentes.

- *Extensibilité*: c'est à dire la possibilité d'étendre l'ontologie locale sans modification.

- *Minimalité des postulats d'encodage*: ce qui assure une bonne portabilité.

- *Minimalité de l'engagement ontologique*: c'est-à-dire l'expressivité maximum de chaque terme qui implique la minimalité et la consistance du vocabulaire utilisé.

Pour l'identification des concepts, les auteurs proposent trois stratégies: *une approche descendante, une approche ascendante, et une approche mixte*. Dans notre approche on utilise *l'approche mixte* dans laquelle les concepts les plus importants sont identifiés et auxquels on applique les techniques de spécialisation et/ou de généralisation pour déterminer les autres concepts de chaque ontologie locale.

V. AGENTS LOGICIELS, CONNAISSANCES ET COMMUNICATION

Les agents de cette proposition possèdent plusieurs propriétés qui sont conformes aux propriétés générales des agents. Ces propriétés sont pour l'essentiel :

1. *Portabilité* : pour assurer sa portabilité, l'agent est programmé avec le langage Java;
2. *Autonomie* : l'agent capable de prendre certaines décisions basées sur leur état interne. Dans une certaine mesure, ils peuvent prendre des décisions sans l'intervention directe d'un humain.
3. *Stabilité* : une bonne gestion des exceptions permet à l'agent de demeurer dans un état stable;
4. *Persistance* : l'agent reprend le cours normal de ses opérations après une panne du système informatique;
5. *Flexibilité* : les agents de notre approche doivent être en mesure de pouvoir être reconfigurés afin qu'ils puissent être facilement utilisés pour rechercher des informations dans la plupart des applications.
6. *Sociabilité* : l'agent peut communiquer avec les autres, et possiblement avec des humains, grâce à un langage de communication pour réaliser les tâches pour lesquelles ils ont été conçus.

V.1 L'agent d'utilisateur

V.2.1 Définition et description

L'agent est une entité informatique ou électronique ; cela ne signifie pas que l'être humain soit exclu du monde des agents mais, pour intégrer l'utilisateur dans la modélisation, on peut lui associer un agent qui lui offre une interface plus ou moins évoluée. On peut alors voir l'utilisateur comme une ressource plutôt "capricieuse" connectée au système par l'intermédiaire de l'agent qui le représente.

L'agent utilisateur, de notre approche est responsable du dialogue avec l'utilisateur : il reçoit les requêtes initiales formulées par les utilisateurs, présente les ; Il est responsable de la gestion du modèle utilisateur : il enregistre dans sa base de connaissances les données initialement fournies par l'utilisateur sur lui-même par l'intermédiaire d'un formulaire de saisie et mémorise les interactions avec l'utilisateur (requêtes). Les agents " utilisateur " sont des agents réactifs qui possèdent très peu de connaissances mais peuvent faire appel à certains agents d'ontologies que nous décrivons dans le paragraphe suivant.

V.1.2 Architecture interne

L'agent utilisateur est la porte d'entrée des requêtes externes au système. Il fournit à l'utilisateur la bonne interface lui permettant de faire une requête. L'agent utilisateur s'occupe de transmettre à l'utilisateur les réponses requises pour la requête posée. L'architecture interne de l'agent utilisateur, quant à elle, est composée de trois modules principaux et d'un registre de sauvegarde, comme l'indique la **Figure 3.5**.

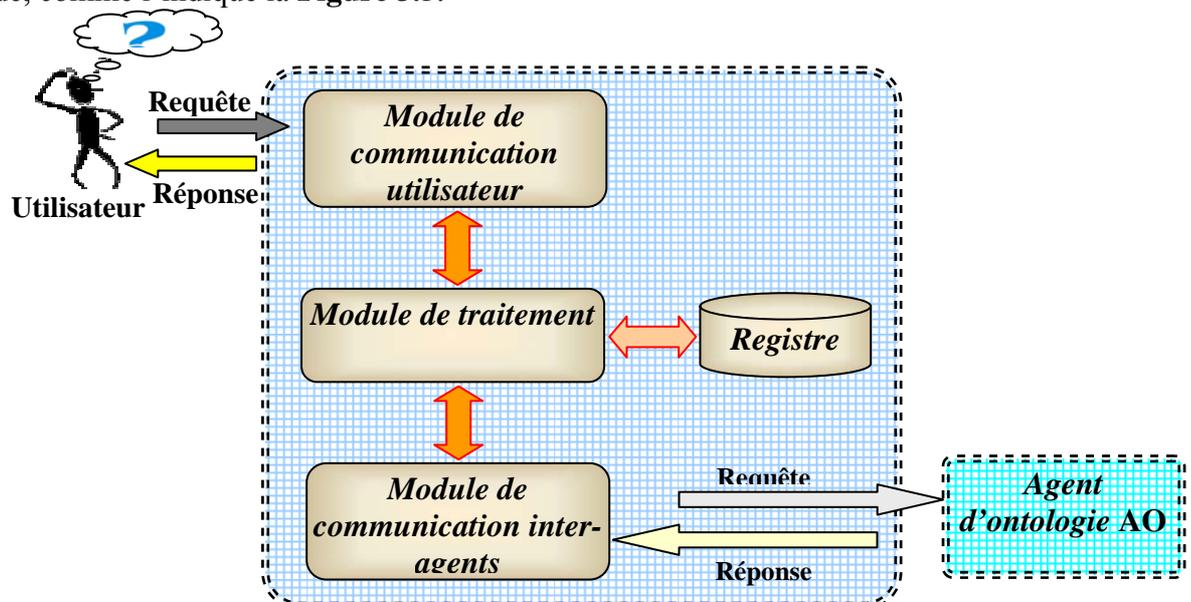


Figure 3.5 : Architecture d'agent utilisateur

Les trois modules donnent à l'agent une plus grande extensibilité étant donné leur découpage selon les tâches. Les trois modules sont comme suit :

1) **Le module de communication utilisateur** : Ce module reçoit les requêtes de l'utilisateur et les transfère au module de traitement. L'opération inverse est également disponible, c'est-à-dire

que le module peut recevoir de l'information du module de traitement pour la montrer à l'utilisateur.

2) *Le module de traitement* : Il reçoit des données du module de communication utilisateur et les sauve dans le registre de l'agent. Ce registre contient toutes les informations recueillies par l'agent sur l'utilisateur du système multi agent. Il détermine ensuite si toutes les informations nécessaires à la réponse de la requête sont disponibles.

3) *Le module de communication inter-agents* : Il reçoit du module de traitement, des demandes de transmission de messages vers les différents agents d'ontologies .Il transfère également les informations reçues des agents du système multi-agent au module de traitement.

4) *Le registre de sauvegarde* : Il a pour rôle de sauvegarder les données que l'utilisateur a fournies, de manière à regrouper les réponses de différents agents d'ontologies.

V.2 L'agent d'ontologie

V.2.1 Définition et description

L'agent d'ontologie est un agent responsable de la reformulation des requêtes. Il utilise le modèle utilisateur, le modèle de domaine et le thésaurus pour proposer une requête étendue. Nous définissons les agents d'ontologies de notre système comme des agents cognitifs (au sens de l'intelligence artificielle), le fait que les agents soient cognitifs implique l'utilisation d'un langage élaboré qui puisse supporter la puissance de raisonnement mise en jeu. Ils sont aussi des agents qui interagissent entre eux par envois de messages .Leur nature communicative doit aussi être supportée par un modèle de communication efficace. Ce sont ces aspects que nous allons aborder dans les paragraphes suivants.

V.1.2 L'architecture d'agent d'ontologie

Notre modèle d'agent propose une architecture générique (figure 3.6). Les agents sont homogènes dans leur structure et se différencient par leurs connaissances (ontologie locale) de domaine. Un agent d'ontologie est composé de *quatre* modules: *le Module de planification, le module de service mapping, le Module de manipulation, le module de communication inter -agents*. Seules les *connaissances du domaine (les ontologies locales)* différencient les agents d'ontologies. Les agents d'ontologies du système interagissent entre eux et avec l'agent utilisateur. Ces interactions sont gérées par des mécanismes de *contrôle inter-agents*. A l'intérieur d'un même agent d'ontologie AO, les interactions entre les quatre modules sont gérées par le *contrôle intra-agent*.

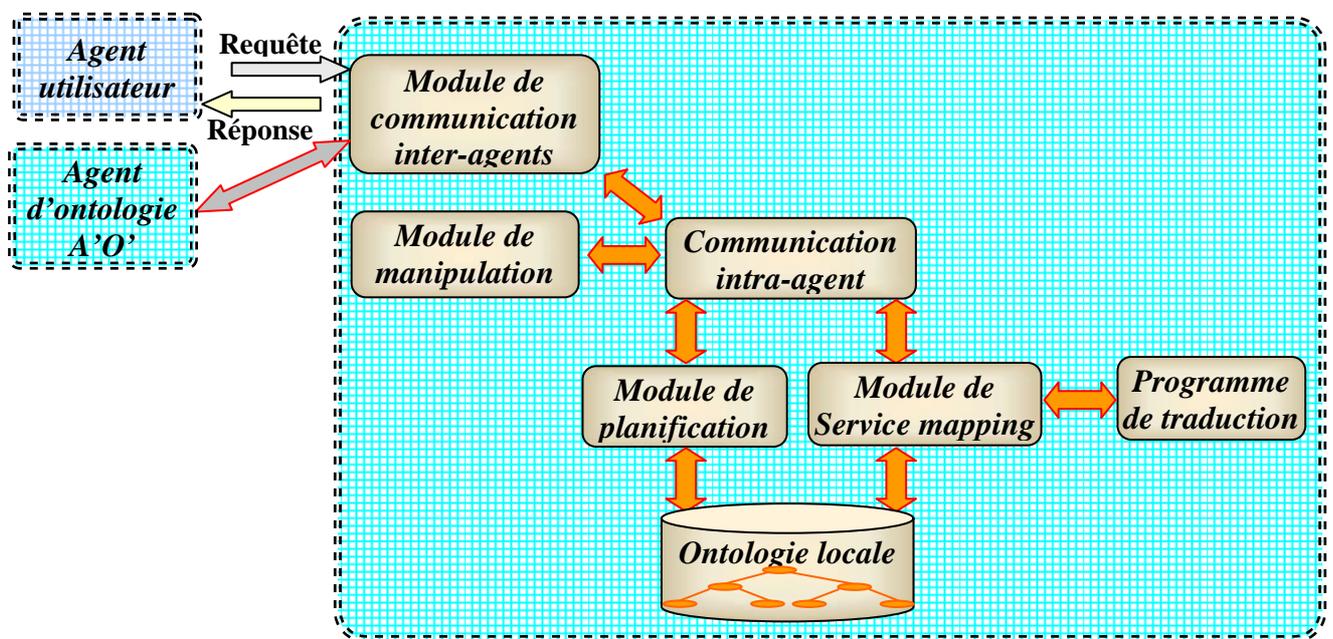


Figure 3.6 : Architecture d'agent d'ontologie

Les quatre modules partagent une structure de données commune (*tableau noir* ou *blackboard*) contenant le(s) plan(s) en cours d'exécution. Les modules échangent des messages de synchronisation (*notifications*) grâce à des *primitives de communication intra-agent*.

1) *Le module de planification*

Le module de planification définit la stratégie d'exécution de plans de l'agent. Le choix d'un plan se fait indépendamment de la manière dont seront traitées ultérieurement les éventuelles actions abstraites qu'il contient. Le module de planification peut prendre deux types de décisions: lancer des plans, ou arrêter des plans. Le lancement de nouveaux plans est motivé soit : par l'arrivée de nouveaux problèmes, soit pour raffiner des sous plans abstraits. Et l'arrêt de plans peut être décidé soit : pour des plans devenus inutiles, soit pour permettre de lancer des plans plus prioritaires. Il est à noter que plusieurs plans peuvent être exécutés en parallèle par un même agent d'ontologie.

Pour choisir entre plusieurs plans, le module de planification doit d'abord les comparer. Une étape d'évaluation de plans est donc nécessaire. Elle aboutit à l'attribution d'un score à chaque plan candidat. Ce score tient compte des caractéristiques propres aux plans mais aussi de l'état actuel de l'agent d'ontologie.

2) *Le module de Service mapping*

La planification s'intéresse à la description de l'ensemble des buts des agents dans un système donné, en précisant quel est le comportement attendu de ces agents. Chaque agent d'ontologie a un but final à atteindre (mappings), puis de le décomposer en une séquence d'actions ; ces actions sont organisées sous formes des plans de mapping syntaxiques et des plans de mapping sémantiques pour effectuer des mappings bidirectionnels entre les différents agents d'ontologies et qui

permettant donc de réaliser le mapping entre les différentes ontologies locales. Pour cela, un but est décrit par une séquence d'actions ordonnées :

But = < Achieved (A1), Achieved (A2), ..., Achieved (An) >, où les A_i représentent les actions, chacune d'entre elles étant associée à un sous-but.

Le module de Service mapping ces le corps cognitif de l'agent d'ontologie ; une méthodologie qui permet le mappings ontologiques repose sur l'utilisation d'un *programme de traduction* et propose alors au module de planification des plans d'action. Ces plans sont accompagnés d'une fiche d'identité contenant les critères extra structurels. Ce corps cognitif gère également les objets du domaine utilisés lors de l'instanciation (exécution) des plans. Nous considérons que chaque événement arrivant à l'agent déclenche un plan.

3) *Le programme de traduction*

L'agent d'ontologie à la possibilité de pouvoir traduire un message à n'importe quel moment, programme de traduction doit permettre à un agent d'ontologie AO de pouvoir transformer localement un message exprimé en fonction d'une ontologie O en un nouveau message exprimé selon une ontologie O' qui utilise par la communication avec l'agent d'ontologie A'O'. Le programme lui-même doit être écrit dans le langage java , mais l'agent qui impose un programme de traduction devra bien sûr s'assurer qu'il connaît le langage et pourra donc exécuter, ou interpréter, le programme.

Fonction de requête d'un programme de traduction sous la forme suivante :

program (A'O', AO, Ms') \rightarrow Ms

où

- O et O' sont deux ontologies,
- Ms' est le message de l'agent d'ontologie A'O'
- Ms est le message résultant.

program (M, Ms') \rightarrow Ms

où

- M est un Mapping,
- Ms' est le message de l'agent d'ontologie A'O'
- Ms est le message résultant.

Signalons que (M, Ms') \rightarrow Ms est équivalent à program(Mapping(AO,AO'), lg).

4) *Le module de manipulation*

Le module de manipulation gère les plans au niveau de leur structure. Le module de manipulation lance l'exécution du plan ou d'une partie d'un plan lorsque les préconditions de déclenchement sont satisfaites, il assure le suivi de l'exécution. En cas de contradictions structurelles entre plans, c'est le module de manipulation qui se chargera de les éliminer.

5) *Le module de communication*

Deux types de communication coexistent dans le système: communication inter-agents et communication intra-agent. La communication inter-agents est à la charge du module de communication et est effectuée par échange messages. Ce module ne traite donc que les échanges entre l'agent et les autres agents. Il utilise pour cela un module d'accointances structuré en une liste des agents connus avec leurs compétences (l'ensemble des transitions qu'ils sont capables de raffiner).

La communication inter-agents peut être de différents modes: POINT TO POINT, MULTICAST, et mode BROADCAST. Dans notre approches pour gérer les mappings bidirectionnels nous utilisons le mode POINT TO POINT; l'agent d'ontologie AO1 confie où délègue une tâche à un agent d'ontologie particulier AO2 pour faire des mappings entre les deux ontologies locales O et O' ou pour donner un résultat à l'agent utilisateur (communication sélective).

Nous définissons le type de message échangés entre agents d'ontologies REQUEST TO INFORM: requête d'un agent à un autre agent. L'agent à l'initiative de la communication a besoin d'une information ou de déléguer une partie de l'exécution d'un plan à un autre agent, puis intégrer la réponse à son propre travail. Le module de communication garde une trace des messages qu'il envoie. Ceci permet à l'agent de gérer les messages en attente de réponse et au module de communication de surveiller les temps de réponse à ses requêtes. La trace d'un message contient l'identifiant unique généré au moment de l'émission du message, la date de l'envoi, le module d'origine et un script d'arrivée.

Le module de communication échange uniquement des plans. Toute information est encapsulée dans un plan pour pouvoir être transmise. Le module ne gère pas la communication intra-agent, elle est à la charge de chaque module. Dans notre vue présentée, chaque agent exécute un plan mapping. Ce plan général est raffiné à l'arrivée d'une requête sur l'ontologie locale.

Pour assurer la réalisation d'échange des messages et les négociations entre les différents agents, en utilisant le « Agent Communication Language ». Ce langage d'agent que nous avons présenté ACL-FIPA, il a été développé de façon à permettre l'interaction entre agents artificiels. Les langages d'agents, et plus particulièrement l'ACL-FIPA, ont été développés de façon à servir les différentes fonctions des agents et ont été adaptés aux besoins des agents artificiels. Ils ne cherchent

pas à suivre une théorie de l'interaction humaine (malgré les emprunts à la théorie des actes de langage), et donc, il est normal qu'ils n'y soient que peu adaptés.

6) *La communication intra-agent*

Les quatre modules de l'agent sont indépendants, fonctionnent en parallèle et communiquent de façon asynchrone. On peut lancer plusieurs exécutions parallèles de sous-plans indépendants. C'est le module de planification qui détermine les sous-plans vérifiant ces contraintes et c'en tenant compte des critères structurels. Le suivi de l'exécution se fait à l'aide d'une structure de données partagée (tableau noir) contenant le plan en cours d'exécution. Lorsque les unités d'exécution se terminent, le tableau noir contient le plan avec le résultat de l'exécution. Les données écrites sur le tableau noir sont lues par les modules de manipulation et de planification et module de service mapping. Le premier s'en sert pour enchaîner l'instanciation d'unités d'exécution du même plan. Le second les utilise pour décider des plans ultérieurs à exécuter.

V.3 Diagramme de classes

UML et par extension Agent UML permettent de représenter plusieurs niveaux d'abstraction lors de la conception des diagrammes de classes. Nous nous intéressons aux deux niveaux suivants : le niveau conceptuel et le niveau implémentation. Le niveau conceptuel est une vue assez haute du système multi-agents éliminant toutes les informations qui sont superficielles pour comprendre la structure du système : les attributs, les opérations (**Figure 3.7**).

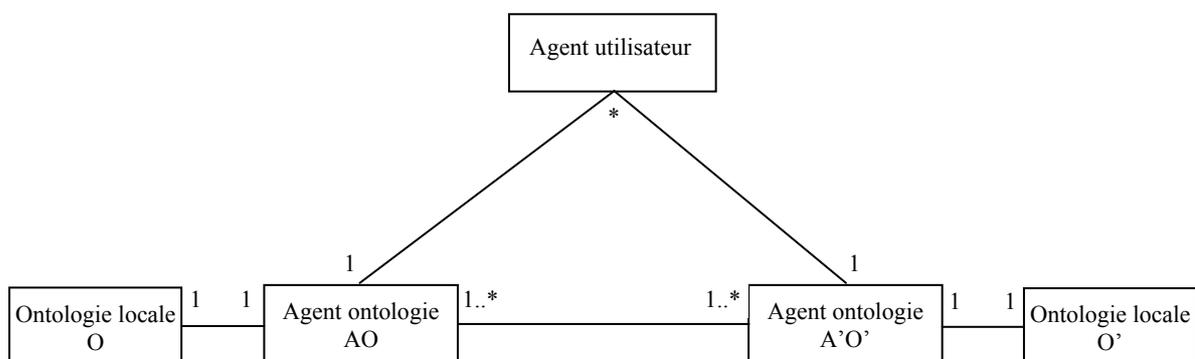


Figure 3.7. Le niveau conceptuel de diagramme de classe pour notre approche

Le second niveau que nous considérons est le niveau implémentation qui donne une vue détaillée des agents de notre approche, des classes et des différentes relations (**Figure 3.8**).

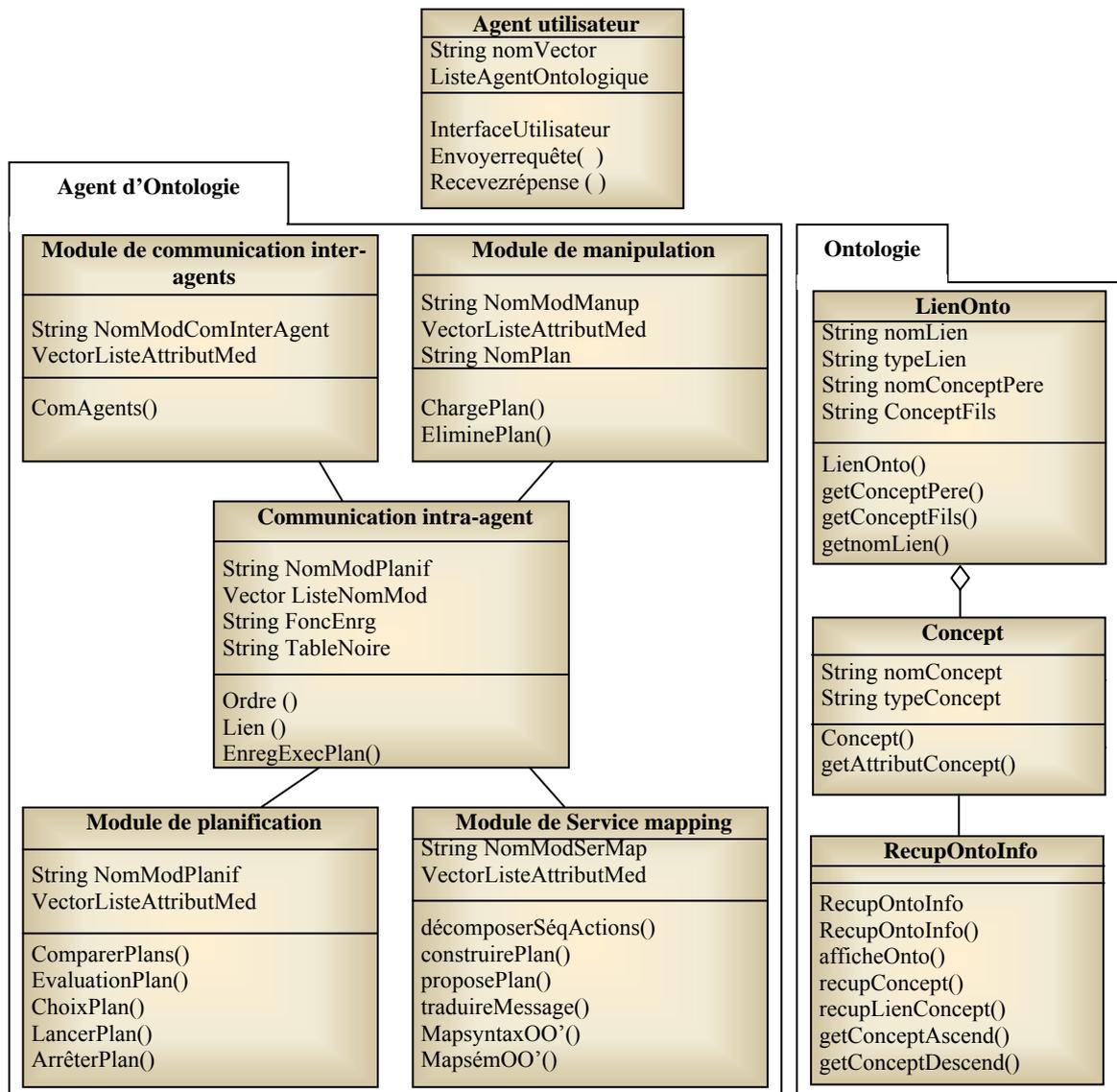


Figure 3.8. Le niveau implémentation de diagramme de classe pour les différents composants de notre approche

VI. MAPPINGS POINT À POINT BIDIRECTIONNELS

VI.1 Description des services de base

Nous avons dit dans le chapitre 1 il y a trois catégories principales proposées pour réaliser l'interopérabilité et qui sont : *Le mapping d'ontologies*, *La fusion d'ontologies*, et *la Transformation d'ontologies*. Comme il a été évoqué précédemment, nous nous focalisons dans le cadre de nos travaux sur le mapping d'ontologies par l'utilisation des mappings point-à-point (un-à-un ou one-to-one mapping) bidirectionnels qui permettent de créer des mappings entre chaque paire d'ontologies. Cette approche permet d'apporter plus de flexibilité dans les environnements aussi complexes par exemple les secteurs industriels. Nous allons dans ce qui suit, étudier de plus près cette approche. Et pour simplifier le mapping nous proposons dans notre approche chaque ontologie à son propre agent d'ontologie AO (agent cognitif) et, en collaboration avec un agent d'utilisateur (agent réactif), qui utilise pour communiquer avec l'utilisateur.

VI.1.1 Accès à une ontologie

Nous devons prendre en compte l'accès aux ontologies. Nous entendons par accès le fait qu'un agent, un agent d'ontologie OA1 peut éventuellement manipuler une ontologie O1 qu'il ne veut pas mettre en libre accès, pour des raisons de confidentialité par exemple ; un agent peut également se référer à une ontologie locale qu'un autre agent ne pourra pas manipuler. Il est donc nécessaire de tester cette accessibilité, ainsi que de permettre la reformulation d'un message en se référant à une autre ontologie : cette reformulation doit pouvoir se faire soit à partir d'une autre ontologie, qu'il convient donc d'identifier, soit à partir d'un mapping existant entre l'ontologie O et une autre ontologie O'.

L'accès à une ontologie tierce pouvant être initié par l'un ou l'autre des agents dialoguant, les processus de manipulation d'ontologies et de mapping doivent être conçus avec une grande flexibilité au niveau des paramètres de configuration.

VI.1.2 Manipulation d'ontologies

Plusieurs ontologies locales peuvent interopérer uniquement si les correspondances (mapping) entre leurs agents ontologies sont clairement identifiées (analyse des termes employés dans les différentes ontologies locales au moyen de *module de service mapping* dans les agents des ontologies. Cette phase de mise en correspondance est établie par des outils d'analyse de la similarité en fonction d'une métrique. Dans la phase de correspondance, il y a deux techniques possibles pour manipuler les différentes ontologies : *la comparaison d'ontologie* permet la personnalisation de parcours de formation ; et *l'interopérabilité d'ontologies* permet la coopération d'entrepôts, et c'est l'approche qui nous intéressons dans le cadre de nos travaux.

VI.1.3 Interopérabilité d'ontologie

La coopération d'entrepôts est réalisée soit en conservant les ontologies locales, soit en les enrichissant. L'interopérabilité d'ontologies n'est envisageable que lorsque le vocabulaire utilisé pour décrire les concepts est proche et précis. L'interopérabilité n'est pas réaliste en cas d'ontologies radicalement différentes. Il y a deux approches Interopérabilité d'ontologie : Approche enrichissant les ontologies : Elles sont enrichies par le biais de deux techniques : la composition d'ontologies et l'intégration d'ontologies. Approche conservant les ontologies : deux techniques sont possibles: la copie de métadonnées et le Mapping d'ontologies. L'approche qui nous intéressons c'est l'approche conservant les ontologies en utilisant le Mapping d'ontologies.

VI.2 Mapping d'ontologies

Le Mapping de deux ontologies consiste à définir les correspondances logiques (mapping) entre les différentes ontologies locales. Il s'agit donc d'identifier les concepts similaires entre chaque deux ontologie locale. Dans ce cas, les métadonnées ne sont pas dupliquées (pas de copie). Lorsque l'utilisateur recherche une ressource, il exprime sa recherche sous forme

d'une requête en utilisant les concepts définis dans son modèle de domaine .Il faut alors nécessairement disposer d'outils (semi-)automatiques de définition de mapping. Cette automatisation est possible si les concepteurs d'ontologies des différents systèmes se sont mis d'accord sur la définition des termes et concepts utilisés (figure 3.9).

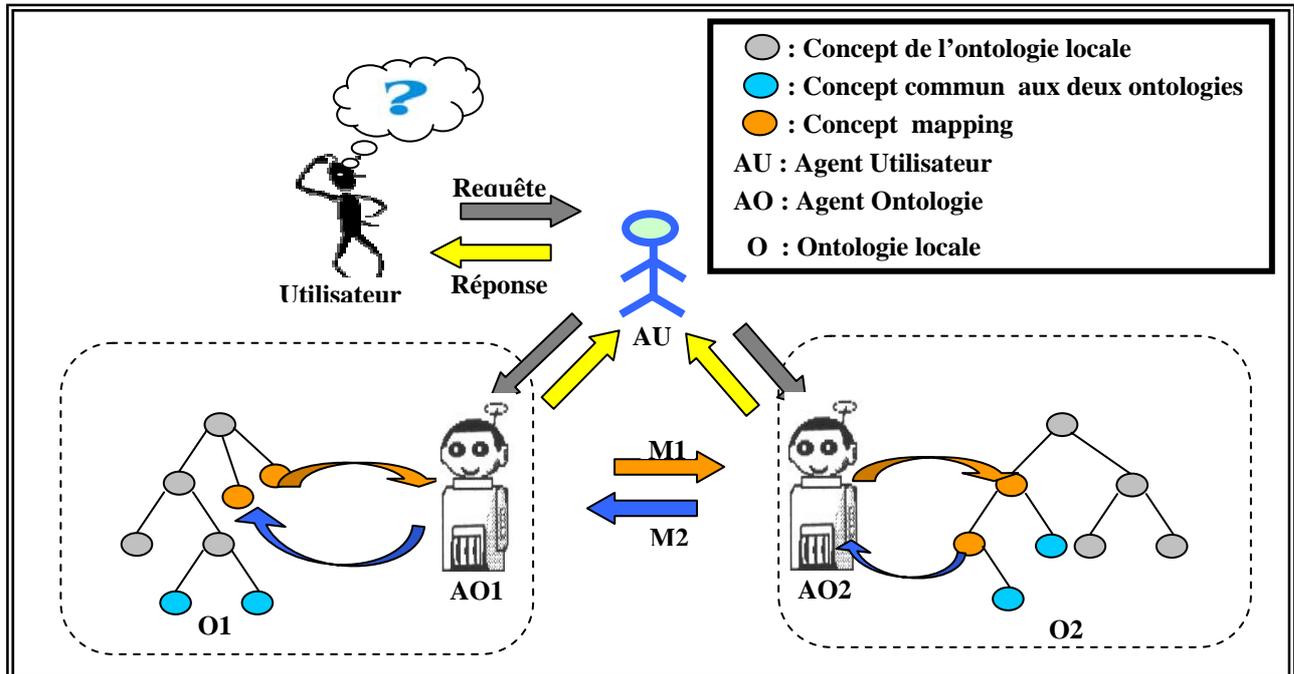


Figure 3.9 : Architecture générale de notre approche Mappings Point A Point Bidirectionnels

Comme on l'a vu jusqu'à présent, les agents sont amenés à manipuler des ontologies et des mappings. Nous décrirons donc, dans les paragraphes suivants, les fonctionnalités qu'ils doivent mettre à la disposition des composants du système.

VI.2.1 Approches de découverte mappings

Dans notre approche qui utilise l'architecture multi ontologie, on utilise la méthode de *découverte basée sur des heuristiques ou l'apprentissage*, cette approche comprend des techniques basées sur des heuristiques ou des techniques d'apprentissage, ces techniques exploitant les composants lexicaux et structurels des définitions des concepts afin de chercher des correspondances entre les concepts. Dans le domaine ontologique, l'approche basée sur des heuristiques exploite la sémantique véhiculée par les caractéristiques des ontologies à mapper et qui peuvent être: noms des concepts en langue naturelle ; la taxonomie des concepts sous forme des relations is-a , lien part-of, instance-of, etc ; les instances des concepts ; les définitions des propriétés et la description des classes.

VI.2.2 Mapping deux ontologies

Le mapping est la fonction centrale qui permet à un agent d'ontologie AO1 de demander à un autre agent d'ontologie AO2 pour faire le mapping de deux ontologies (ontologie O1 et ontologie O2). Formellement, il est possible de définir la notion de mapping d'ontologies comme un ensemble

de correspondances et des liens au sens mathématique et des relations sémantique entre un certain nombre d'agents.

Du point de vue abstrait, en définit un mapping de deux ontologies O et O' avec l'utilisation des agents ontologiques A et A' comme un opérateur α ($AO, A'O'$) qui :

- à partir des deux ontologies O et O' de couverture différente, permet de réconcilier l'utilisation simultanée des deux agents d'ontologies AO et $A'O'$ pour la description de l'univers du discours;
- à partir des deux ontologies O et O' de granularité différente, permet d'associer aux agent d'ontologie AO les entités homologues dans $A'O'$;
- à partir des deux ontologies O et O' de point de vue différent, permet de déduire le point de vue de $A'O'$ associé à l'agent d'ontologie AO .
- à partir des deux ontologies O et O' de couverture différente, permet de utiliser simultanée des mapping bidirectionnels utilisés la communication entre les deux agents d'ontologies AO et $A'O'$;

Le processus de mapping consiste donc à générer une correspondance M' à partir de deux agents d'ontologies AO et $A'O'$.

Aussi, la considération de processus mapping comme une fonction f qui à partir d'une paire d'ontologies O et O' à réconcilier, les agents d'ontologies A et A' , un mapping initial M , un ensemble de paramètres p , un ensemble de ressources r , produit un nouveau mapping M' entre ces deux ontologies :

$$M' = f(O, O', AO, A'O', M, p, r)$$

Ce processus peut être représenté comme suit :

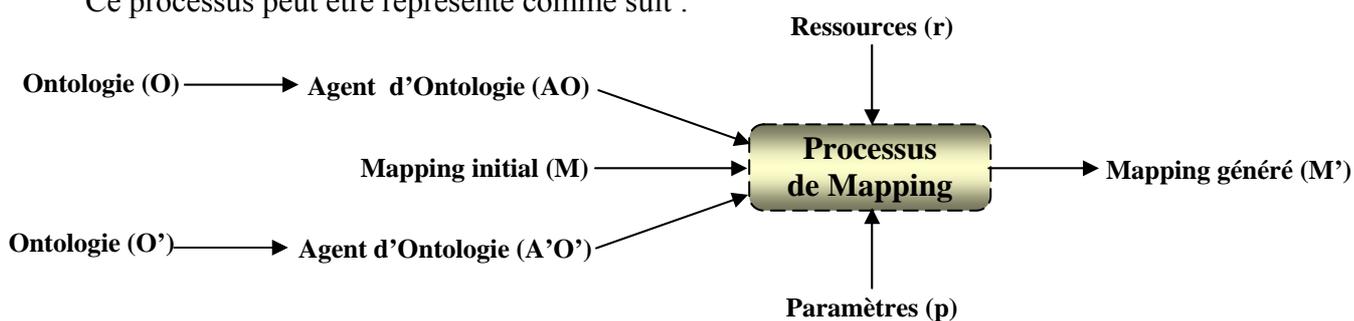


Figure 3.10 : Processus générique de mapping de deux ontologies

a) Construire les mappings syntaxiques

Cette étape est importante, elle permet la construction des mappings syntaxiques (et plus exactement des mappings syntaxe-sémantique) nécessaires pour définir des correspondances entre les concepts des ontologies locales.

On peut distinguer principalement deux types de correspondances entre les ontologies locales :

- correspondance type-concept, permettant d'utiliser les agents d'ontologies pour définir les différentes mappings syntaxiques entre les concepts d'ontologies locales ;
- correspondance attribut-propriété, permettant d'utiliser les agents d'ontologies pour définir les différentes mappings syntaxique plus fins entre les propriétés d'ontologies locales.

b) Construire les mappings sémantiques

Cette étape est aussi importante, elle permet la construction des mappings sémantiques nécessaires pour entre les différentes ontologies locales. On peut distinguer principalement deux types de correspondances sémantiques:

- correspondance pour concept, permettant d'utiliser les agents d'ontologies pour définir les différentes mappings sémantique entre les concepts d'ontologies locales ;
- correspondance pour propriété, permettant d'utiliser les agents d'ontologies pour définir les différentes mappings sémantique plus fins entre les propriétés d'ontologies locales.

VI.2.3 Le multi-mapping d'ontologies

Il est bien entendu possible de définir le multi-mapping d'ontologies M' (mapping de plusieurs ontologies), en insérant comme arguments les n ontologies concernées, les n agents d'ontologies :

$$M' = f(O1, O2, \dots, On, AO1, AO2, \dots, AOn, M, p, r)$$

Noter que :

- ✓ le mapping initial M peut être un mapping injecté en entrée en vue de le mettre à jour ou le compléter.
- ✓ Les agents d'ontologies $AO1, AO2, \dots, AOn$
- ✓ Les ressources r peuvent être les ressources matérielles, logicielles et humaines utilisées pour exécuter le processus de mapping.
- ✓ les hypothèses p qui peuvent être effectuées comme le seuil de vraisemblance à utiliser.

Dans cette thèse nous avons affaire, des mappings point-à-point (architecture multi-ontologies) qui permettent de créer des mappings entre chaque paire d'ontologies. Le nombre de mappings créés par l'approche multi-ontologies est de l'ordre de $O(n^2)$ où n est le nombre d'ontologies impliquées

Aussi, la considération le nombre de mapping comme une fonction g qui à partir d'une paire d'un ensemble ontologies $O1, O2, \dots, On$; les agents d'ontologies $A1, A2, \dots, An$; des mappings initiaux $M1, M2, \dots, Mn$; un ensemble de paramètres p , un ensemble de ressources r , produit un nouveau mapping M entre ces ontologies :

$$M = g(O1, O2, \dots, On, AO1, AO2, \dots, AOn, M1, M2, \dots, Mn, p, r)$$

VII. PROTOCOLE DE MAPPING D'ONTOLOGIES

Dans cette partie nous étudions notre protocole de mapping en détail, c'est-à-dire les échanges de messages possibles entre les différents systèmes, afin de déterminer si les termes utilisés dans un message permettent au récepteur de ce message de l'interpréter : ceci revient à déterminer, comme on l'a évoqué jusqu'à maintenant, si chacun des concepts du message peut être relié à d'autres connaissances déjà assimilées, grâce à une mapping.

Il est important de rappeler que tous les échanges de messages décrits dans cette partie ont lieu en amont de toute action liée à l'exécution / interprétation du contenu du message lui-même.

VII.1 Architecture et représentation

Nous présentons le protocole, dans les paragraphes suivants, sous forme de règles qui définissent les échanges de messages possibles entre les acteurs du système sous certaines conditions. Cette représentation est tirée de [Euzenat, 1997] et semble être la plus adaptée à la description de notre protocole grâce à sa simplicité d'écriture et de lecture, ainsi que pour la rigueur de sa notation qui facilite le codage. Signalons que la notation AUML est une alternative très intéressante à la notre. On la préférera cependant dans une deuxième étape d'écriture, une fois que toutes les interactions sont bien définies, afin de ne pas se perdre dans la notation graphique qui, de prime abord, peut sembler déroutante pour des novices de la modélisation. Mais une fois l'outil maîtrisé, AUML se révèle tout aussi exploitable pour la transmission de protocoles, et peut être même plus efficace pour avoir un aperçu rapide du déroulement d'un protocole. Rappelons que les protocoles d'interaction du modèle FIPA-ACL sont spécifiés en AUML.

VII.2 Les règles

La partie supérieure de la règle définit l'envoi d'un message de X vers Y (X et Y sont l'agent d'ontologie AO et l'agent d'ontologie A'O'), alors que la partie inférieure définit la réponse qui doit être envoyée par Y à Z (Z est éventuellement X) si la condition (C) exprimée à droite de la règle est vérifiée. Avant l'envoi de sa réponse, Y peut effectuer un certain nombre d'actions (a_1, \dots, a_n).

$$\text{Nom de la règle : } \frac{X - \text{Message} \rightarrow Y}{a_1, \dots, a_n, Y - \text{Réponse} \rightarrow Z}$$

VII.3 Contrôle de la conversation

Nous considérons que les échanges de messages décrits dans les paragraphes suivants utilisent et respectent les informations d'identification de messages spécifiées par les performatifs FIPA-ACL de contrôle de la conversation. Nous partons ainsi de l'hypothèse qu'un agent B répond à un agent A en lui précisant l'identifiant de réponse correspondant au message initial, émit par l'agent A. Nous posons cette condition pour éviter tout problème lié à des croisements de messages ; Les communications doivent donc respecter la règle illustrée par l'exemple de la figure 3.11.

<pre>(QUERY-REF :sender A :receiver B :content :reply-with id-123456) Message 1 l'agent A interroge l'agent B</pre>	<pre>(INFORM :sender B :receiver A :content :in-reply-to id-123456) Message 2 . l'agent B répond à l'agent A</pre>
--	---

Figure 3.11. : Principe d'identification des interlocuteurs pour le suivi de la cohérence des échanges dans un langage de communication entre agents.

VII.4 Nomenclatures

Nous donnons dans les tableaux suivant la liste des performatifs et pseudo-performatifs utilisés dans les règles qui seront présentées par la suite. Nous distinguons les performatifs issus du modèle FIPA-ACL de ceux que nous introduisons spécifiquement pour ce projet et que nous identifierons par le label PMAO (pour Protocole de Mapping entre Agents d'Ontologies). Notons que ces derniers sont plus des actions qui sont considérées par FIPA-ACL comme un langage de contenu de type *Speech Acts*.

Performatif	Utilisation dans notre protocole
<i>request</i>	L'émetteur demande au récepteur d'effectuer une action.
<i>inform</i>	Utilisé en réponse à une requête.
<i>refuse</i>	Refuse d'effectuer une action
<i>failure</i>	Echec d'une action tentée par le récepteur suite à la requête de l'émetteur.
<i>query-if</i>	L'émetteur demande au récepteur d'effectuer un test.
<i>Not-understood</i>	Un paramètre est inconnu du récepteur.
<i>confirm</i>	Confirme qu'un tuple est bien présent dans une structure de données.

Table 3.1 : Performatifs FIPA-ACL utilisés dans notre protocole.

Pseudo-performatif	Signification
<i>map</i>	Demande d'effectuer un mapping d'ontologies
<i>inaccessible</i>	La ressource transmise est inaccessible. (problème réseau, droit d'accès...)
<i>inconsistent-set</i>	L'ensemble transmis n'est pas cohérent pour l'exécution de la tâche préalablement requise.
<i>get-axioms</i>	Demande d'un ensemble d'axiomes.
<i>find</i>	Demande de rechercher une ontologie.
<i>tsl</i>	Demande de traduction d'un message
<i>declare</i>	Déclaration d'une donnée à mémoriser
<i>is-Map</i>	Transmet une interrogation sur l'existence d'un mapping.
<i>unknown-pair</i>	Indique qu'un couple de données (ici d'ontologies) est inconnu.

Table 3.2 : Pseudo-performatifs PMAO, spécifiques à notre protocole

VII.5 Services fournis par Le module de services mapping

VII.5.1 Trouver une ontologie

Afin de pouvoir mapping une ontologie O avec une autre O', il faut d'abord la trouver. Nous avons donc permis à un agent de cette ontologie AO de trouver un agent d'ontologie A'O' pour envoyer son message. De plus, le module de service mapping de l'agent émetteur dans sa recherche d'un tel agent ontologie, il est conseillé d'inclure un sous-protocole du type contract-net qui permette l'interrogation d'autres acteurs du système, par l'intermédiaire d'un appel d'offre (CFP "Call For Proposal").

(search-success)	$\frac{X - request (find (O', T)) \rightarrow Y}{Y - inform (match (O', T)) \rightarrow X}$	O' est accessible, $\forall t \in T, t \in O'$
(search-access-denied)	$\frac{X - request (find (O', T)) \rightarrow Y}{Y - refuse (inaccessible (O')) \rightarrow X}$	O' est inaccessible
(search-bad-set)	$\frac{X - request (find (O', T)) \rightarrow Y}{Y - failure (inconsistent - set (T)) \rightarrow X}$	O' est accessible, $\exists t \in T / t \notin O'$

VII.5.2 Mapping d'ontologies

Dans le mapping d'ontologies, les agents d'ontologies déterminent : Les paramètres de mapping de deux ontologies O et O', un mapping initialement vide et un ensemble de paramètres comme le niveau de mapping, une valeur de seuil pour filtrer les paires selon leur similarité, un sous-ensemble de termes à considérer ou un algorithme de mapping. Des valeurs par défaut doivent bien sûr être gérées par les modules de service mapping. Signalons peut-être un mapping partiel existant entre O et O', mais il peut aussi être vide.

(mapping-success)	$\frac{X - request (map (O, O', a, P)) \rightarrow Y}{Y - inform (map (O, O', a, P)) \rightarrow X}$	O et O' sont accessibles, le module de service mapping interprète les paramètres de l'ensemble (P) sans problème et peut donc mapping MOO'.
(map-access-denied)	$\frac{X - request (map (O, O', a, P)) \rightarrow Y}{Y - refuse (inaccessible (R)) \rightarrow X}$	R est la ressource inaccessible (R est soit O, soit O', soit a, soit P)
(map-bad-set)	$\frac{X - request (map (O, O', a, P)) \rightarrow Y}{Y - failure (inconsistent - set (P)) \rightarrow X}$	O et O' sont accessibles, P n'est pas conforme pour MOO'.

VII.5.3 Traduction d'un message

La traduction d'un message M peut se faire : en précisant l'agent d'ontologie (et donc l'ontologie locale) à partir de laquelle le message M a été exprimé.

Dans ce cas de figure, la requête est faite à partir du message M , de l'ontologie O et d'un mapping m . Si tous les concepts du message M apparaissent dans l'ontologie O alors le message peut être traduit. Précisons que la référence à l'ontologie O permet de rechercher les concepts de M dans le mapping (m), et ainsi d'identifier les concepts équivalents dans l'autre ontologie (celle qui

$$\text{(trans-map-success)} \quad \frac{X - request (tsl(M, O, m)) \rightarrow Y}{Y - inf orm (translate (M, O, m)) \rightarrow X} \quad \forall e \in M / e \in m$$

$$\text{(trans-map-bad-set)} \quad \frac{X - request (tsl(M, O, m)) \rightarrow Y}{Y - refuse (inconsiste nt - set(m)) \rightarrow X} \quad \exists e \in M / e \notin m$$

n'est pas O dans le mapping). Si un concept de O présent dans M n'apparaît pas dans le mapping(m), alors le message de réponse fera état de cette incohérence dans les données.

VII.5.4 Programme à partir d'un mapping

La requête d'un programme se fait directement à partir d'un couple d'ontologies ; un cas supplémentaire peut donc se produire : celui d'une ontologie inaccessible. Dans les règles suivantes, lg représente le langage dans lequel doit-être exprimé le programme de traduction de l'agent Y ; $Y(L)$ représente l'ensemble L associé à Y , c'est-à-dire l'ensemble des langages que Y supporte, et l'appel de $program(m, lg)$ retourne le programme de traduction lui-même.

$$\text{(trans-onto-success)} \quad \frac{X - request (tsl(M, O, O')) \rightarrow Y}{Y - inf orm (translate (M, O, O')) \rightarrow X} \quad O \text{ et } O' \text{ sont accessibles}$$

$$\text{(trans-onto-access-denied)} \quad \frac{X - request (tsl(M, O, O')) \rightarrow Y}{Y - refuse (inaccessib le(O*)) \rightarrow X} \quad \begin{array}{l} O^* \text{ est inaccessible} \\ (O^* \text{ est soit } O \text{ soit } O') \end{array}$$

VII.5.6 Ajouter un mapping à la bibliothèque

Rappelons tout d'abord que $Y(M)$ est l'ensemble des Mappings dont MOO' (mapping de O et O') dispose dans sa bibliothèque de Y . Dans ce cas de figure, puisque le mapping m n'appartient pas aux mappings que MOO' considère comme étant associés aux ontologies O et O' , Y l'ajoute à son ensemble M de triples et répond à X que le mapping a bien été mémorisé :

$$\text{(declare-map-success)} \quad \frac{X - request (declare (O, O', m)) \rightarrow Y}{Y(M) := Y(M)U \langle O, O', m \rangle, \quad Y - confirm (O, O', m) \rightarrow X} \quad \begin{array}{l} \langle O, O', m \rangle \notin Y(M), \\ \text{verifSources}(m, O, O') \end{array}$$

Si le triplet est déjà connu de Y , alors Y envoie le même message que dans le cas précédent ; cela revient à confirmer à X que $Y(M)$ contient le triple initialement transmis.

$$\text{(declare-map-existing)} \quad \frac{X - request (declare (O, O', m)) \rightarrow Y}{Y - confirm (O, O', m) \rightarrow X} \quad \langle O, O', m \rangle \in MOO'(M)$$

Si le mapping transmis ne correspond pas aux ontologies également transmises, alors la règle suivante sera déclenchée :

$$\text{(declare-map-bad-set)} \quad \frac{X - \text{request}(\text{declare}(O, O', m)) \rightarrow Y}{Y - \text{refuse} - \text{set}(m) \rightarrow X} \quad \begin{array}{l} \langle O, O', m \rangle \notin Y(M), \\ \text{not}(\text{verifSources}(m, O, O')) \end{array}$$

VII.5.7 Obtenir un mapping

L'obtention directe d'un mapping, survenant préférentiellement juste après un test d'existence, permet de récupérer un mapping existant. Dans les règles suivantes, Y et X sont également des agents d'ontologies. Dans ce premier cas de figure, un mapping existe :

$$\text{(get-map-success)} \quad \frac{X - \text{request}(\text{get} - \text{map}(O, O')) \rightarrow Y}{Y - \text{inform}(\text{getMap}(O, O')) \rightarrow X} \quad \langle O, O', _ \rangle \in \text{MOO}'$$

Dans le deuxième cas de figure, aucun Mapping ne correspond au couple d'ontologies (O, O') ; un message adapté doit donc être transmis en réponse :

$$\text{(get-map-unknown)} \quad \frac{X - \text{request}(\text{get} - \text{map}(O, O')) \rightarrow Y}{Y - \text{refuse}(\text{unkown} - \text{pair}(O, O')) \rightarrow X} \quad \langle O, O', _ \rangle \notin \text{MOO}'$$

VIII. SCENARIO D'UTILISATION

Dans cette partie nous considérons les différents scénarios de notre approche, sans toutefois être implantable directement. Nous partons en effet d'hypothèses relatives à l'environnement technique dans lequel se déroule le scénario. Les différentes entités intervenant au sein du scénario sont : un agent utilisateur AU et des agents d'ontologies et comme un exemple nous utilisons deux agents d'ontologies AO et l'agent d'ontologie A'O'.

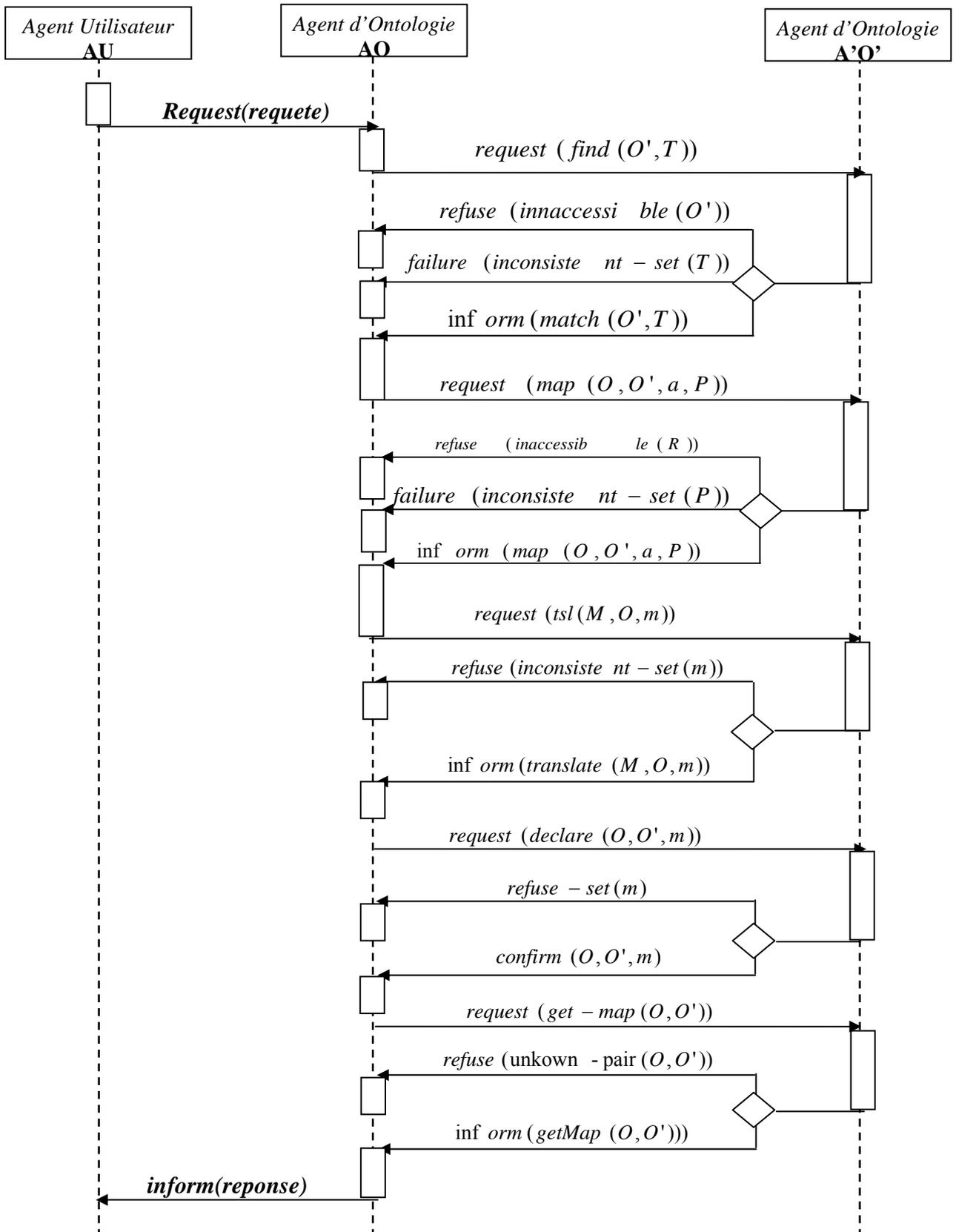


Figure 3.12. Protocole de mapping d'ontologie proposé pour notre approche

IX. CONCLUSION

Dans ce chapitre nous avons détaillé la démarche suivie, et la reproduction du chaque étape à réalisé notre approche. Dans la planification, en déterminant quel est le comportement attendu de chaque sous système, ces systèmes échangent, en général, des services afin d’accomplir un objectif

global. Afin de répondre à tous ces objectifs, une méthode pour l'ingénierie des besoins, comprenant une démarche, un ensemble de modèles et des outils logiciels, doit être développée. .

La construction d'ontologies faite par l'Ingénierie des connaissances. Le cycle de vie qui nous avons utilisé permet de développer des ontologies d'une manière prototypique, c'est celui proposé par l'équipe de Fernandez. Les agents d'ontologies sont des agents cognitifs ayant des vocabulaires particuliers différents, ils sont communiquent entre eux et avec l'agent utilisateur par envois de messages. La construction des mappings sémantiques nécessaires entre les différentes ontologies locales. On peut distinguer principalement la correspondance sémantique pour concept, et la *correspondance sémantique pour propriété*, permettant d'utiliser les agents d'ontologies pour décrire les différentes mappings sémantique entre les ontologies locales. Pour le mapping d'ontologie nous avons proposé un nouveau protocole pour gérer la communication entre les agents.

Dan le chapitre suivant, Afin de plus aisément comprendre le mapping point à point dans l'approche multi ontologie basé agent, et à travers le processus proposé dans ce chapitre, et pour motiver notre travail d'un point de vue pratique, le chapitre suivant décrit l'application du processus proposé avec une étude de cas.

CHAPITRE 04:

ETUDE DE CAS

I. INTRODUCTION

Afin de plus aisément comprendre le mapping point à point dans l'approche multi ontologie basé agent, et à travers le processus proposé dans le chapitre précédent, Comme il nécessitait choisir un domaine de connaissance, nous avons choisi « *l'encadrement universitaire* » comme domaine applicatif. Dans les paragraphes suivants, une étude de cas qui consiste à construire deux ontologies locales de l'encadrement universitaire (ontologie Etudiant et ontologie Enseignant).

II. PLANIFICATION DU DOMAINE

La planification s'intéresse comme nous avons vue pour déterminer les différents systèmes dans un domaine donné, dans notre domaine **l'encadrement universitaire** il y a deux systèmes interopérables principales : le premier système c'est l'enseignant et la deuxième c'est l'étudiant (**figure 1**).



Figure 4.1 : *Planification du notre domaine*

Ces deux systèmes échangent des informations afin d'accomplir un objectif global (*l'encadrement*). Un étudiant est encadré par un enseignant (encadreur) et un encadreur encadre au moins un étudiant.

III. L'INGENIERIE DES BESOINS

L'ingénierie des besoins a pour objectif de décrire ce que le système doit faire, c'est à dire ses fonctionnalités. Parmi les hypothèses (le plus souvent implicites) sur lesquelles est basée la modélisation conceptuelle, il y en a deux qui nous paraissent essentielles : Les besoins relatifs à les deux systèmes interopérables (enseignant et étudiant) sont donnés au départ à travers les différents besoins exprimés par les utilisateurs, le problème se réduit à construire le système qui répond à ces besoins ; et ce sont les ontologies qui sont les plus adaptées pour ce travail.

Dans cette étape l'ingénierie des besoins, qui a pour but d'établir un document de spécification des besoins. Ce dernier permet de décrire chaque ontologie locale à construire à travers les cinq

aspects (i.e. propriétés) suivants : le domaine de connaissance que va couvrir l'ontologie locale, l'objectif de l'ontologie locale, les futurs utilisateurs de l'ontologie locale, les sources de l'information et la portée de l'ontologie locale qui consiste à déterminer à priori la liste des termes (les plus importants) pour le domaine à représenter.

IV. LA CONSTRUCTION DES ONTOLOGIES LOCALES

Dans notre approche en utilisant le cycle de vie qui proposé par l'équipe de Fernandez cette méthodologie propose les étapes suivantes:

1) Identifier le but et la portée de l'ontologie (planification du domaine et L'ingénierie des besoins)

2) la construire, de chaque ontologie locale ; Le processus de construction de l'ontologie comprend plusieurs étapes, La première étape c'est l'étape de conceptualisation qui consiste à identifier et à structurer, à partir des sources d'informations, les connaissances du domaine et à décrire la sémantique dans un langage semi-formel, en indiquant leurs propriétés, leurs instances connues et les liens qu'ils entretiennent entre eux.

On distingue dans cette étape plusieurs activités qui sont:

- 1) Construction d'un *glossaire de termes*.
- 2) Classifier les concepts dans *des hiérarchies de concepts*.
- 3) Représenter les relations qui existent entre les différents concepts par *un diagramme de relations binaires*.
- 4) Identifier les concepts par leurs instances, leurs attributs ainsi que leurs concepts synonymes dans *un dictionnaire de concepts (DC)*.
- 5) Décrire les relations dans *une table de relations binaires*.
- 6) Spécifier des contraintes sur les attributs dans *une table d'attributs*.
- 7) Spécifier des axiomes sur les concepts dans *une table d'axiomes logiques*.
- 8) Décrire les instances des concepts dans *une table d'instances*.
- 9) Décrire les assertions entre instances dans *une table des assertions*.

IV.1. Construction d'un Glossaire de termes

Le glossaire recueille et décrit tous les termes qui sont utiles et potentiellement utilisables dans le domaine que l'on investit.

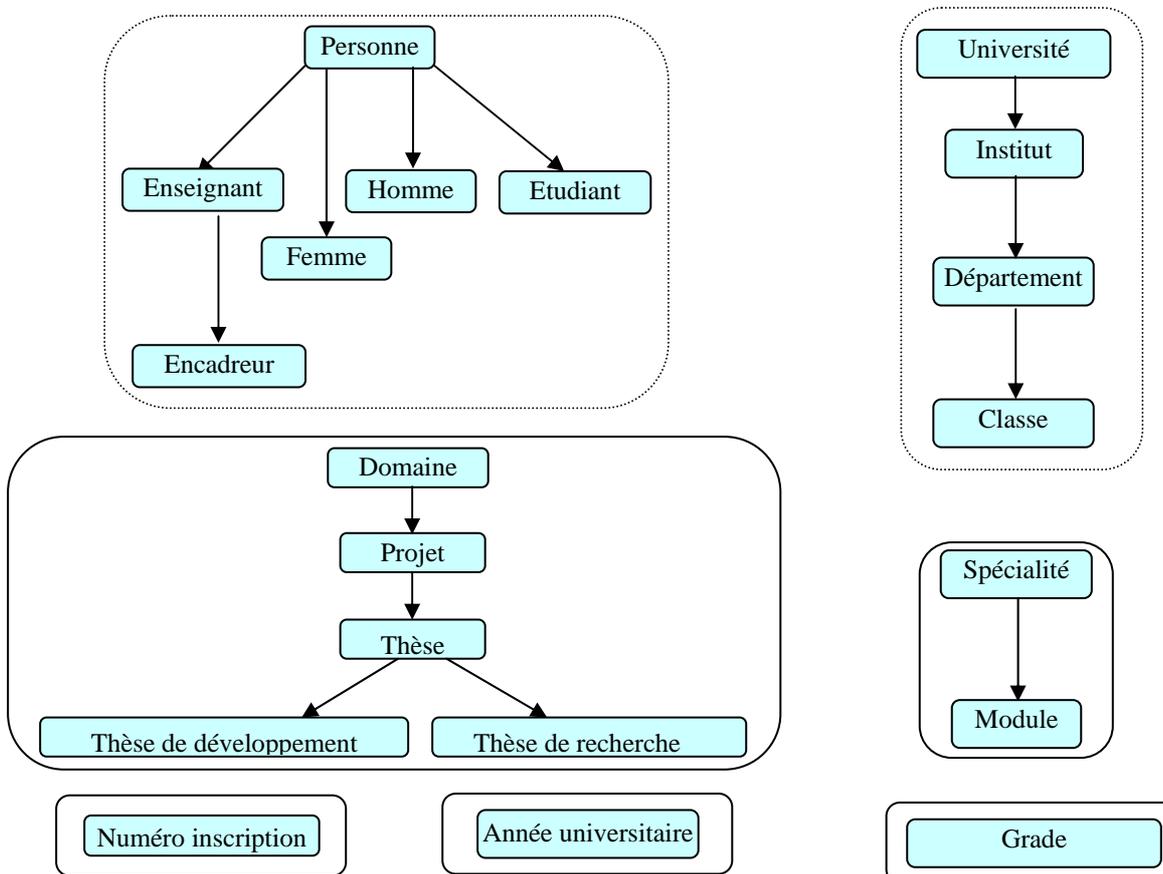
Termes	Description	Référence
Personne	<ul style="list-style-type: none"> ▪ Est un humain qui est caractérisé par un nom, un prénom et un téléphone, une Adresse et un e-mail. 	
Homme	<ul style="list-style-type: none"> ▪ Est une personne de sexe masculin 	
Femme	<ul style="list-style-type: none"> ▪ Est une personne de sexe féminin 	
Enseignant	<ul style="list-style-type: none"> ▪ Est une personne qui travail dans un département avec un ou plusieurs enseignants. ▪ Il doit avoir un grade. ▪ Il doit enseigne au moins une spécialité. ▪ Il donne un cours. ▪ Il doit travail au moins dans une classe ▪ Il collabore dans un institut. ▪ Il dirige un projet. ▪ Il oriente une thèse. 	
Grade	<p>Peut être : magister, Docteur, Maitre de conférences, Professeur</p> <ul style="list-style-type: none"> ▪ Caractérisé par un nom. 	
Classe	<ul style="list-style-type: none"> ▪ Est un ensemble d'étudiants. ▪ Elle reliée à un département. 	
Etudiant	<ul style="list-style-type: none"> ▪ Est une personne qui suit des études supérieures et qui étudier dans une classe d'étudiants. ▪ Il doit avoir au moins une spécialité. ▪ Il doit contient un numéro d'inscription. 	
Spécialité	<ul style="list-style-type: none"> ▪ Est enseignée par des enseignants dans une classe. ▪ Elle est composition des modules. ▪ Elle est ensemble des années universitaires. 	
Département	<ul style="list-style-type: none"> ▪ Est un collectivement des classes. ▪ Il doit être lié à un institut. 	
Institut	<ul style="list-style-type: none"> ▪ Est un ensemble de départements. ▪ Caractérisé par un nom, une adresse, un téléphone et un fax. 	
Encadreur	<ul style="list-style-type: none"> ▪ Est un enseignant qui Oriente une thèse. 	

Domaine	<ul style="list-style-type: none"> Il propose un ensemble des projets. 	
Projet	<ul style="list-style-type: none"> Diriger par au moins un enseignant. Et sur un domaine précis 	
Thèse	<ul style="list-style-type: none"> Est une démonstration d'un projet. Elle est orientée par un Encadreur 	
Thèse de développement	<ul style="list-style-type: none"> Est un raisonnement de développement d'une thèse. 	
Thèse de recherche	<ul style="list-style-type: none"> Est un raisonnement de recherche d'une thèse 	
Université	<ul style="list-style-type: none"> Est une ensemble d'instituts. Caractérisé par un nom, une adresse, un téléphone et un fax. 	

Table 4.1 : Glossaire de termes

IV.2 Construction des hiérarchies de concepts

Une hiérarchie de concepts organise un groupe de concepts entre eux sous forme d'une taxonomie. Les hiérarchies de concepts sont utilisées non seulement pour spécifier les relations taxonomiques entre concepts, mais aussi pour la modularité de la connaissance de domaine.



▪ *La hiérarchie des concepts :*

Nous avons déjà identifié dans la planification du notre domaine « l' **encadrement universitaire** » deux systèmes principales : le premier système c'est l'enseignant et la deuxième c'est l'étudiant ; dans cette étape nous créons la hiérarchie des concepts pour chaque système.

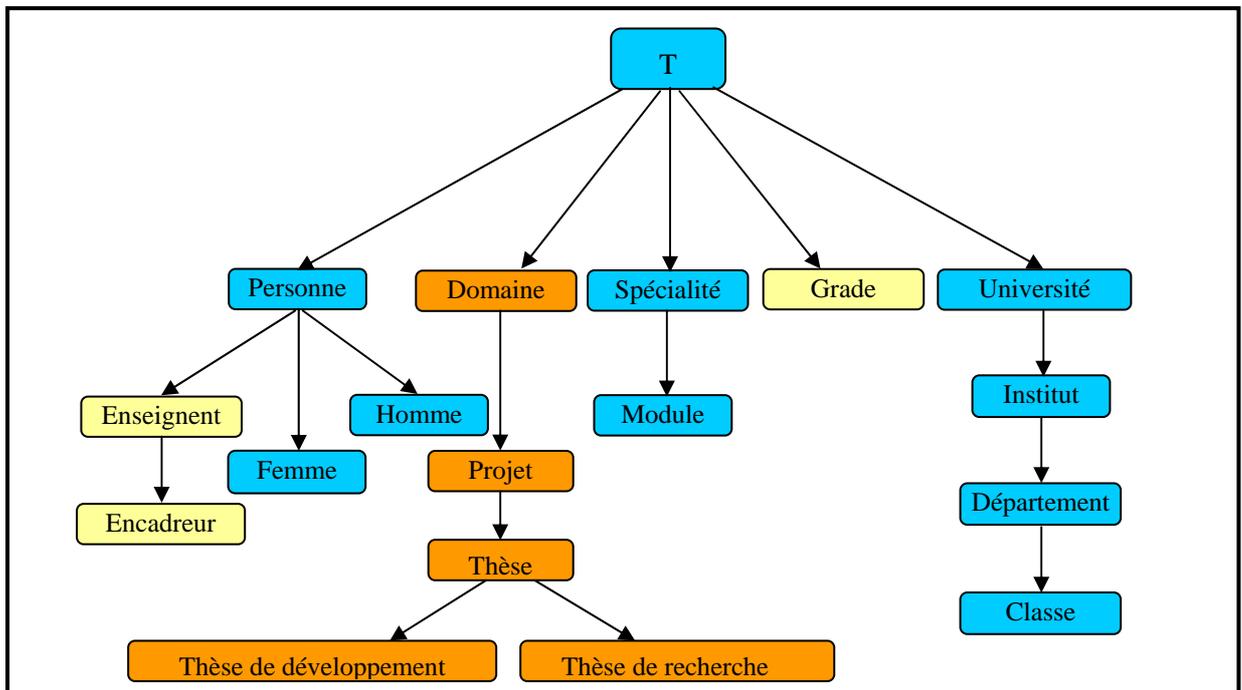


Figure 4.2 : La hiérarchie des concepts pour le système enseignant

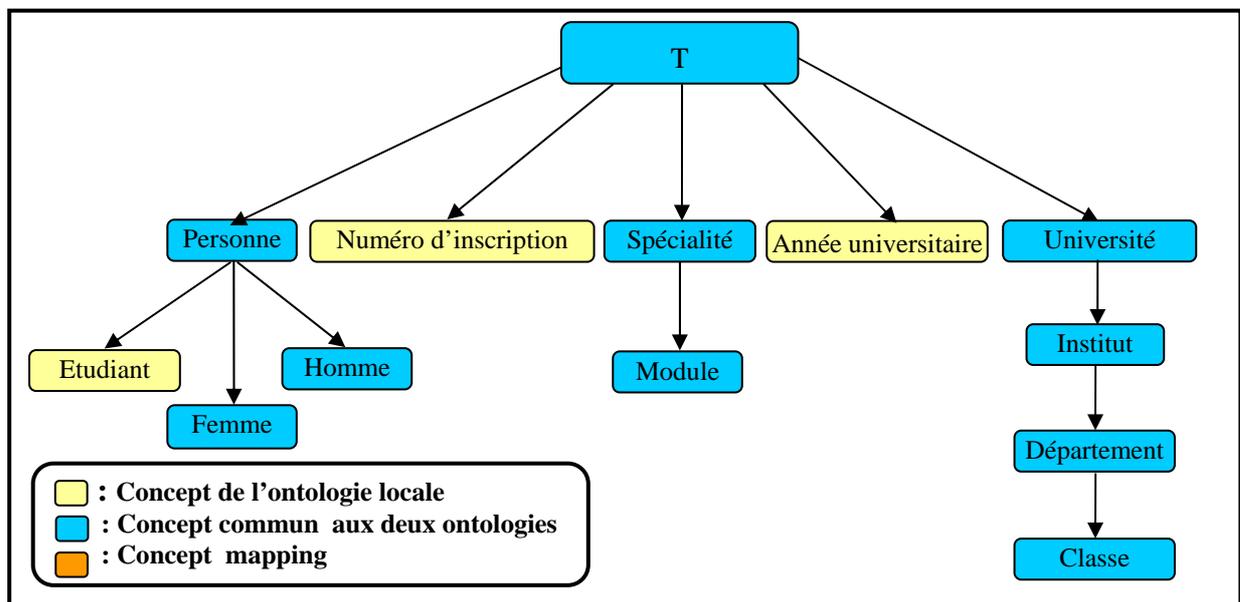


Figure 4.3 : La hiérarchie des concepts pour le système étudiant

IV.3 Construction du diagramme des relations binaires

Cette phase consiste à construire un diagramme de relations binaires. Ce diagramme permet de représenter de manière graphique les diverses relations qui existent entre les divers concepts de même ou de différentes hiérarchies.

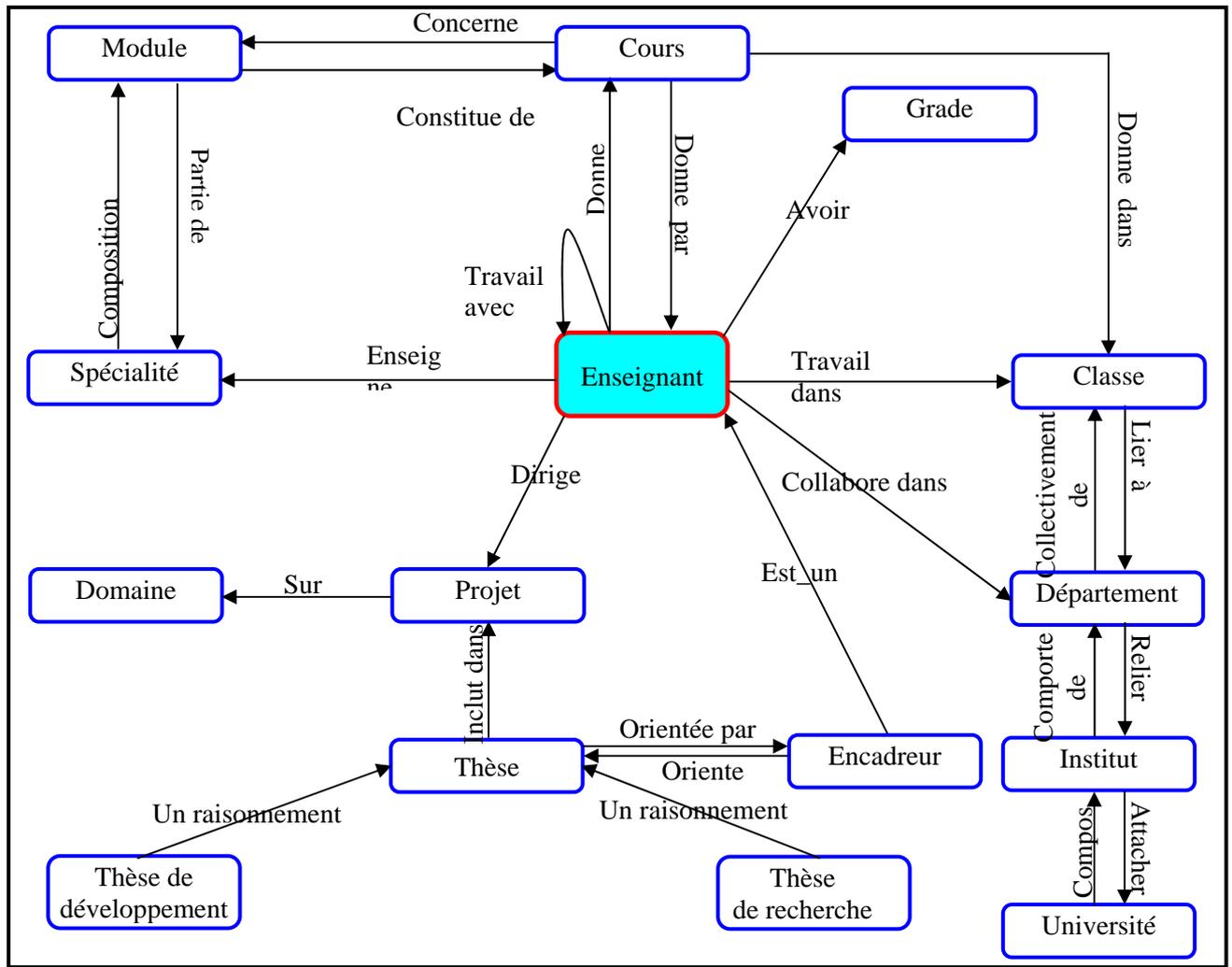


Figure 4.4: diagramme des relations binaires pour le système enseignant

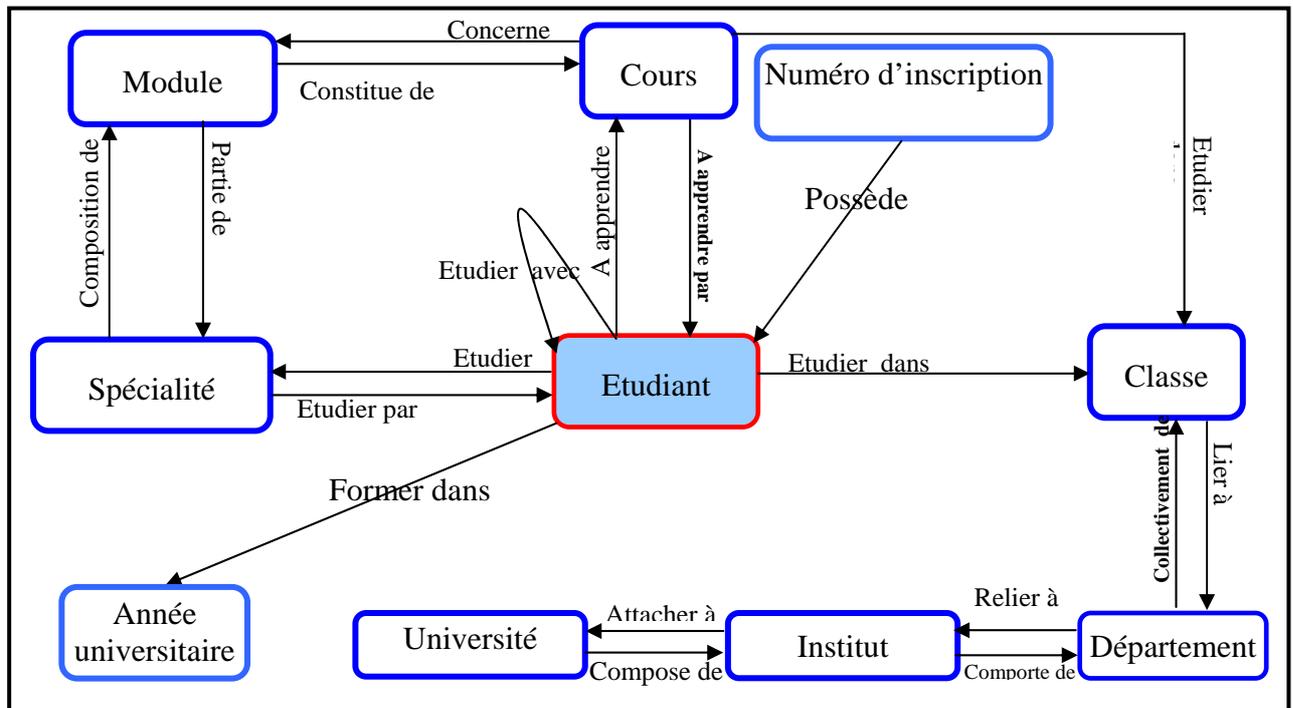


Figure 4.5 : diagramme des relations binaires pour le système étudiant

IV.4 Construction du dictionnaire des concepts

Le dictionnaire de concepts contient tous les concepts du chaque système, et pour chaque concept ses instances connus, ses propriétés (attributs) et ses concepts synonymes.

Termes	Synonyme	Attributs	instances
Personne	Individus Humain	Nom Prénom Adresse Téléphone e-mail	<ul style="list-style-type: none"> • Kazar Okba
Homme		Nom Prénom Adresse Téléphone e-mail	<ul style="list-style-type: none"> • Kazar Okba
Femme		Nom Prénom Adresse Téléphone e-mail	
Enseignant	Professeur Éducateur	Nom Prénom Téléphone Adresse e-mail grade	<ul style="list-style-type: none"> • Kazar Okba • Sahnoun Zaidi • Laskri M^{ed}.T
Encadreur		Nom Prénom Téléphone e-mail grade	<ul style="list-style-type: none"> • Kazar Okba • Sahnoun Zaidi • Laskri M^{ed}.T
Cours		Titre	Type d'agent
Grade	Niveau, Rang, Échelon	Nom	<ul style="list-style-type: none"> • Magister • Docteur d'état • Professeur
Classe	Groupe	Nom	Salle 3
Spécialité	Branche	Nom	Intelligence artificielle
Département		Nom Adresse Téléphone Fax	Math et informatique
Institut	Faculté, Établissement, École	Nom Adresse Téléphone Fax	Séances exactes
Université		Nom Adresse Téléphone Fax E_mail Site	université de biskra

Table 4.2 : dictionnaire des concepts pour le système enseignant

Termes	Synonyme	Attributs	instances
Personne	Individus Humain	Nom Prénom Adresse Téléphone e-mail	
Homme		Nom Prénom Adresse Téléphone e-mail	Goudjil Lakhdar
Femme		Nom Prénom Adresse Téléphone e-mail	
Cours		Titre Duré	Type d'agent
Classe	Groupe	Nom, numéro	Salle 3
Etudiant	Elève	Nom Prénom Adresse Téléphone e-mail	Goudjil lakhdar
Spécialité	Branche	Nom	Intelligence artificielle
Département		Nom Adresse Téléphone Fax	Informatique
Institut	Faculté, Établissement, École	Nom Adresse Téléphone Fax	Séances exactes
Université		Nom Adresse Téléphone Fax Site	Université de biskra C.U de Souk-Ahras
Année universitaire	Niveau	Nom	Deuxième
Numéro d'inscription		Numéro	10/2006

Table 4.3 : dictionnaire des concepts pour le système enseignant

IV.5 La table de relations binaires

Nom relation	Source	Destination	Cardinalité (min/max)	Relation inverse	Cardinalité (min/max)
Ensemble_de	Spécialité	Module	5-N	Partie_de	1-N
collabore_dans	Enseignant	Institut	1-N	-	1-N
Travail-avec	Enseignant	Enseignant	1-N	Travail-avec	1-N
Donne	Enseignant	Cours	1-N	Donne par	1-N
Concerne	Cours	Module	1-1	Constitue de	1-N
Avoir	Enseignant	Grade	1-1	-	1-N
Travail dans	Enseignant	Classe	1-N	-	1-N
Enseigne	Enseignant	Spécialité	1-N	Enseignée par	1-N
Orienté	Encadreur	Thèse	1-N	Orientée par	1-N
Donne dans	Cours	Classe	1-N	-	1-N
Dirige	Enseignant	Projet	1-N	Dirige par	1-N
Est_un	Encadreur	Enseignant	1-1	-	1-1
Sur un	Projet	Domaine	1-N	-	1-N
Inclut dans	Thèse	Projet	1-N	-	1-N
Un raisonnement de	Thèse de développement	Thèse	1-1	-	1-1
Un raisonnement de	Thèse de recherche	Thèse	1-1	-	1-1
Collectivement de	Département	Classe	1-N	Lier à	1-N
Comporte de	Institut	Département	1-N	Relier à	1-N
Compose de	Université	Institut	1-N	Attacher à	1-N

Table 4.4 : table de relations binaires pour le système enseignant

Nom relation	Source	Destination	Cardinalité (min/max)	Relation inverse	Cardinalité (min/max)
Ensemble de	Spécialité	Module	5-N	Partie_de	1-N
Former dans	Etudiant	Année universitaire	1-1	-	-
Etudier dans	Etudiant	Classe	1-N	-	1-N
Etudier avec	Etudiant	Etudiant	1-N	Etudier avec	1-N
A apprendre	Etudiant	Cours	1-N	A apprendre par	1-N
Concerne	Cours	Module	1-1	Constitue de	0-N
Possède	Etudiant	Numéro d'inscription	1-1	-	1-1
Etudier dans	Etudiant	Classe	1-N	-	1-N
Etudier	Etudiant	Spécialité	1-N	Etudier par	1-N
Collectivement de	Département	Classe	1-N	Lier à	1-N
Comporte de	Institut	Département	1-N	Relier à	1-N
Compose de	Université	Institut	1-N	Attacher à	1-N

Table 4.5 : table de relations binaires pour le système étudiant

IV.6 La table des attributs

Nom de l'attribut	Type	Cardinalité (Min/Max)	Valeur par défaut	Domaine des valeurs
Nom	Littéral	1-1	-	-
Prénom	Littéral	1-N	-	-
e-mail	Littéral	0-N	-	-
Site	Littéral	0-N	-	-
Téléphone	Littéral	1-N	-	-
Fax	Littéral	1-N	-	-
Titre	Littéral	1-1	-	-
Adresse	Littéral	1-1	-	-
Duré	Temps	1-1	-	-

Table 4.6 : table des attributs pour le système enseignant

<i>Nom de l'attribut</i>	<i>Type</i>	<i>Cardinalité (Min/Max)</i>	<i>Valeur par défaut</i>	<i>Domaine des valeurs</i>
Nom	Littéral	1-1	-	-
Prénom	Littéral	1-N	-	-
Numéro	Littéral	1-1	-	-
e-mail	Littéral	0-N	-	-
Site	Littéral	0-N	-	-
Téléphone	Littéral	1-N	-	-
Fax	Littéral	1-N	-	-
Titre	Littéral	1-1	-	-
Adresse	Littéral	1-1	-	-
Duré	Temps	1-1	-	-

Table 4.7 : table des attributs pour le système étudiant

IV.7 La table d'axiomes logiques

Nom du concept	Description	Expression logique
Personne	Chaque personne est soit une femme, soit un homme.	$\forall (X), \text{Personne}(X) \Rightarrow \text{Femme}(X) \vee \text{Homme}(X)$
Homme	Les concepts Homme et Femme sont incompatibles	$\forall (X), \text{Homme}(X) \Rightarrow \text{not Femme}(X)$
Femme	Les concepts Femme et Homme sont incompatibles	$\forall (X), \text{Femme}(X) \Rightarrow \text{not Homme}(X)$
Enseignant	Un enseignant travail avec au moins avec un enseignant et il doit avoir un grade et il doit enseigne au moins une spécialité et donne un cours et il doit travail au moins dans une classe et collabore dans un institut et il dirige un projet.	$\forall (X) \text{enseignant}(X) \Rightarrow (\exists Y \text{enseignant}(Y) \text{ AND travail_avec}(X,Y))$ $\wedge (\exists S \text{grade}(Y) \text{ AND avoir}(X,S))$ $\wedge (\exists Z \text{spécialité}(Z) \text{ AND enseigne}(X,Z))$ $\wedge (\exists U (\text{cours}(U) \text{ AND donne}(X,U))$ $\wedge (\exists W (\text{classe}(W) \text{ AND travail_dans}(X,W))$ $\wedge (\exists L (\text{institut}(L) \text{ AND collabore_dans}(X,L))$ $\wedge (\exists M (\text{projet}(M) \text{ AND dirige}(X,M))$

Grade	Peut être magister, docteur, maitre de conférences, professeur.	$\forall (X) \text{Grade}(X) \Rightarrow \text{magister}(X) \vee \text{docteur}(X) \vee \text{maitre_de_conférences}(X) \vee \text{professeur}(X).$
Encadreur	Un encadreur est un enseignant oriente une thèse.	$\forall (X) \text{encadreur}(X) \Rightarrow (\exists \text{enseignant}(Y) \text{ AND } \text{est_un}(X,Y)) \wedge (\exists Z (\text{Thèse}(Z) \text{ AND } \text{Oriente}(X,M)))$
Thèse	Elle orientée par au moins un encadreur et elle Peut être thèse de recherche, thèse de développement.	$\forall (X) \text{Thèse}(X) \Rightarrow (\exists \text{Encadreur}(Y) \text{ AND } \text{Orientée_par}(X,Y))$ $\wedge (\exists \text{Projet}(Z) \text{ AND } \text{Inclut_dans}(X,Z))$ $\wedge (\text{Thèse_de_recherche}(X) \vee \text{Thèse_de_développement}(X))$
Projet	Il est dirige par un enseignant.	$\forall (X) \text{Projet}(X) \Rightarrow (\exists \text{Enseignant}(Y) \text{ AND } \text{dirige_par}(X,Y))$
Cours	Un cours doit être donne par au moins un enseignant, et doit être concerne un module et donne dans une classe.	$\forall (X), \text{cours}(X) \Rightarrow (\exists Y \text{enseignant}(Y) \text{ AND } \text{donne_par}(X,Y))$ $\wedge (\exists Z \text{module}(Z) \text{ AND } \text{concerne}(X,Z))$ $\wedge (\exists W \text{classe}(W) \text{ AND } \text{donne_dans}(X,W))$
Module	Un module est constitue de au moins un cours et il partie d'une spécialité.	$\forall (X) \text{module}(X) \Rightarrow (\exists Y \text{cours}(Y) \text{ AND } \text{Constitue_de}(X,Y))$ $\wedge (\exists Z \text{spécialité}(Z) \text{ AND } \text{partie_de}(X,Z))$
Spécialité	Une spécialité est enseigne par au moins un enseignant et elle est composition de modules.	$\forall (X) \text{spécialité}(X) \Rightarrow (\exists Y \text{enseignant}(Y) \text{ AND } \text{enseigne_par}(X,Y))$ $\wedge (\exists Z \text{module}(Z) \text{ AND } \text{ensemble_de}(X,Z))$
Classe	Une classe doit être liée à un département	$\forall (X) \text{classe}(X) \Rightarrow (\exists Y \text{classe}(Y) \text{ AND } \text{liée_à}(X,Y))$
Département	Un département est un Collectivement des classes et doit être relié à un institut.	$\forall (X) \text{Département}(X) \Rightarrow (\exists Y \text{classe}(Y) \text{ AND } \text{Collectivement_de}(X,Y))$ $\wedge \exists Z \text{institut}(Z) \text{ AND } \text{relié_à}(Z,X)$
Institut	Est comporte de départements et il doit attacher à une université.	$\forall (X) \text{Institut}(X) \Rightarrow (\exists Y \text{Départements}(Y) \text{ AND } \text{ensemble_de}(X,Y))$ $\wedge \exists Z \text{université}(Z) \text{ AND } \text{attacher_à}(Z,X)$
Université	Une université compose des instituts	$\forall (X) \text{université}(X) \Rightarrow (\exists Y \text{instituts}(Y) \text{ AND } \text{compose_de}(X,Y))$

Table 4.8 : table d'axiomes logiques pour le système enseignant

Nom du concept	Description	Expression logique
Personne	Chaque personne est soit une femme, soit un homme.	$\forall (X), \text{Personne}(X) \Rightarrow \text{Femme}(X) \vee \text{Homme}(X)$
Homme	Les concepts Homme et Femme sont incompatibles	$\forall (X), \text{Homme}(X) \Rightarrow \text{not Femme}(X)$
Femme	Les concepts Femme et Homme sont incompatibles	$\forall (X), \text{Femme}(X) \Rightarrow \text{not Homme}(X)$
Etudiant	Un étudiant doit étudier au moins une spécialité et il doit Possède un numéro d'inscription et a apprendre un cours et former dans une année universitaire et il doit étudier dans une classe	$\forall (X) \text{ Etudiant}(X) \Rightarrow (\exists Y \text{ spécialité}(Y) \text{ AND Etudier}(X,Y))$ $\wedge (\exists U \text{ numéro_d'inscription}(U) \text{ AND Possède}(X,U))$ $\wedge (\exists V \text{ cours}(V) \text{ AND a_apprendre}(X,V))$ $\wedge (\exists W \text{ années_universitaire}(W) \text{ AND Former_dans}(X,W))$ $\wedge (\exists Z \text{ classe}(Z) \text{ AND étudier_dans}(X,Z))$
Cours	Un cours doit être concerne un module et a apprendre par au moins un étudiant et étudier dans une classe	$\forall (X), \text{cours}(X) \Rightarrow (\exists Y \text{ module}(Y) \text{ AND concerne}(X,Y)) \wedge (\exists Z \text{ étudiant}(Z) \text{ AND a_apprendre}(X,Z)) \wedge (\exists W \text{ classe}(W) \text{ AND étudier_dans}(X,W))$
Module	Un module est constitue de au moins un cours et il partie d'une spécialité.	$\forall (X) \text{ module}(X) \Rightarrow (\exists Y \text{ cours}(Y) \text{ AND Constitue_de}(X,Y)) \wedge (\exists Z \text{ spécialité}(Z) \text{ AND partie_de}(X,Z))$
Spécialité	Une spécialité est étudier par des étudiants et elle est composition des modules.	$\forall (X) \text{ spécialité}(X) \Rightarrow (\exists Y \text{ Etudiant}(Y) \text{ AND étudier_par}(X,Y))$ $\wedge (\exists Z \text{ module}(Z) \text{ AND ensemble_de}(X,Z))$
Année universitaire	Année universitaire est le constituant de spécialité	$\forall (X) \text{ Année universitaire}(X) \Rightarrow (\exists Y \text{ spécialité}(Y) \text{ AND constituant_de}(X,Y))$
Classe	Une classe doit être liée à un département	$\forall (X) \text{ classe}(X) \Rightarrow (\exists Y \text{ classe}(Y) \text{ AND liée_à}(X,Y))$
Département	Un département est un Collectivement des classes et doit être relié à un institut.	$\forall (X) \text{ Département}(X) \Rightarrow (\exists Y \text{ classe}(Y) \text{ AND Collectivement_de}(X,Y))$ $\wedge \exists Z \text{ institut}(Z) \text{ AND relié_à}(Z,X)$
Institut	Est comporte de départements et il doit attacher à une université.	$\forall (X) \text{ Institut}(X) \Rightarrow (\exists Y \text{ Départements}(Y) \text{ AND ensemble_de}(X,Y))$ $\wedge \exists Z \text{ université}(Z) \text{ AND attacher_à}(Z,X)$
Université	Une université compose des instituts	$\forall (X) \text{ université}(X) \Rightarrow (\exists Y \text{ instituts}(Y) \text{ AND compose_de}(X,Y))$

Table 4.9 : table d'axiomes logiques pour le système étudiant

IV.8. La table des instances

La table des instances décrit toutes les instances (incluses dans le champ instances de dictionnaire de concepts) avec leurs attributs et valeurs.

Kazar Okba	Personne	Nom	Kazar
		Prénom	Okba
		Adresse	Biskra
		Téléphone	07 77 30 30 30
		e-mail	kazarokba@yahoo.fr
	Homme		
	Enseignant	Grade	Maitre de conférences
	Encadreur		
Mohamed Tayeb Laskri	Personne	Nom	Laskri
		Prénom	Mohamed Tayeb
		Adresse	Annaba
		Téléphone	07 77 50 55 50
		e-mail	laskri@univ-annaba.org
	Homme		
	Enseignant	Grade	Professeur
Sahnoun_Zaidi	Personne	Nom	Sahnoun
		Prénom	Zaidi
		Adresse	Constantine
		Téléphone	07 70 80 80 50
		e-mail	Sahnoun_Zaidi @yahoo.fr
	Homme		
	Enseignant	Grade	Professeur
Soltani Mohamed	Personne	Nom	Soltani
		Prénom	Mohamed
		Adresse	Souk Ahres
		Téléphone	06 62 12 12 12

		e-mail	soltani_mohamed@yahoo.fr
	Homme		
	Enseignant	Grade	Magister
Ben Idir Kamel	Personne	Nom	Soltani
		Prénom	Mohamed
		Adresse	Ain Beida
		Téléphone	07 77 10 20 30
		e-mail	ben_idir_kamel@yahoo.fr
	Homme		
Boudiaf Noura	Personne	Nom	Boudiaf
		Prénom	Noura
		Adresse	Ain Mlila
		Téléphone	07 71 05 05 05
		e-mail	boudiafn@yahoo.com
	Femme		
	Enseignant	Grade	Docteur
Intelligence artificielle	Spécialité	Duré	2 ans
Génie logiciel			2 ans
Système multi agent	Module		
Réseaux de neurones			
Web-sémantiques			
cycles de génie logiciel			
Type d'agent	Cours	Duré	3 H
SMA			4.5 H
Les ontologies			6 H
Les neurones multi couches			4 H
Introduction en génie logiciel			1.5 H
INF 1	Classe		
ELEC 5			

AR 6			
GES 10			
DR 8			
Informatique	Département		
Electronique			
Gestion			
Langue arabe			
Droit			
Séances de l'ingénieur	Institut		
Economie Gestion			
Lettres et Sciences Sociales			
Droit et Séances politiques			
Université de Biskra	Université		
C.U de Souk-Ahras			
Université de Constantine			
Université d'Oum el Bouaghi			
Uiversité Badji Mokhtar Annaba			
Web	Domaine		
Réseaux			
Atelier de génie logiciel			
Web sémantique	Projet		
Réseaux des capteurs			
Approches par composant			
Multi ontologies	Thèse de recherche	Thèse	
Apprentissage dans les Réseaux des capteurs			
La création d'une ontologie	Thèse de développement		
Une nouvelle approche par composant			

Table 4.10 : Table des instances pour le système Enseignant

Goudjil Lakhdar	Personne	Nom	Goudjil
		Prénom	lakhdar
		Adresse	Ain Beida
		Téléphone	07 77 10 10 10
		e-mail	goudjillak@yahoo.fr
	Homme		
	Etudiant	Année universitaire	Deuxième
Spécialité		Intelligence artificielle	
Nour adine Gasmaallah	Personne	Nom	Gasmaallah
		Prénom	Nour adine
		Adresse	Souk ahras
		Téléphone	07 70 77 77 70
		e-mail	Nouradine@yahoo.fr
	Homme		
	Etudiant	Année universitaire	Deuxième
Spécialité		Intelligence artificielle	
Tebassi Kamel	Personne	Nom	Tebassi
		Prénom	Kamel
		Adresse	Constantine
		Téléphone	06 66 20 20 20
		e-mail	Kml12@yahoo.fr
	Homme		
	Etudiant	Année universitaire	Troisième
Spécialité		Génie logiciel	
Sahrawi Brahim	Personne	Nom	Sahrawi
		Prénom	Brahim
		Adresse	Oum el Boughi
		Téléphone	05 55 50 50 50
		e-mail	SahrawiBrahim @yahoo.fr
	Homme		
	Etudiant	Année universitaire	Première
Spécialité		Génie logiciel	
Zaidi Sana	Personne	Nom	Zaidi
		Prénom	Sana
		Adresse	Ain Beida
		Téléphone	07 77 15 15 15
	Femme		

Première	année universitaire		
Deuxième			
2006/52	Numéro d'inscription		
2006/20			
2007/12			
2007/05			
Intelligence artificielle	Spécialité	Duré	2 ans
Génie logiciel			2 ans
Système multi agent	Module		
Réseaux de neurones			
Web-sémantiques			
cycles de génie logiciel			
Type d'agent	Cours	Duré	3 H
SMA			4.5 H
Les ontologies			6 H
Les neurones multi couches			4 H
Introduction en génie logiciel			1.5 H
INF 1	Classe		
ELEC 5			
AR 6			
Informatique	Département		
Electronique			
Langue arabe			
Séances de l'ingénieur	Institut		
Economie Gestion			
Lettres et Sciences Sociales			
Droit et Séances politiques			
Université de Biskra	Université		
C.U de Souk-Ahras			
Université de Constantine			
Université Oum el Bouaghi			
Université Badji Mokhtar Annaba			

Table 4.11 : Table des instances pour le système Etudiant

IV.9. La table des assertions

Les assertions affirment l'existence de relations entre des instances. La table des assertions définie pour chaque concept source 'Cs' représenté dans la table des relations binaires et pour chaque instance 'I' de 'Cs' définie dans la table des instances les instances du concept cible 'Cc' qui sont en relation par 'R' avec l'instance 'I'.

a) représentation de la partie terminologique T-Box:

Les concepts et les rôles sont exprimés de manière déclarative à travers une partie terminologique. Cette dernière comprend des définitions de concepts et de rôles ainsi que des inclusions de concepts.

$\text{Personne} := \text{Homme} \cup \text{Femme} \cap (\exists \text{ Nom. String}) \cap (\exists \text{ Prénom. String}) \cap (\exists \text{ Adresse. String})$ $\cap (\exists \text{ Téléphone. String}) \cap (\exists \text{ E-mail. String})$
$\text{Homme} := \text{Personne} \cap \neg \text{Femme}$
$\text{Femme} := \text{Personne} \cap \neg \text{Homme}$
$\text{Enseignant} := \text{Personne} \cap (\exists \text{ grade. String}) \cap (\geq 1 \text{ enseigne. Spécialité}) \cap (\exists \text{ donne. Cours})$ $\cap (\exists \text{ dirige. projet}) \cap (\geq 1 \text{ travail_dans. Classe}) \cap (\geq 1 \text{ travail_avec. Classe})$ $\cap (\exists \text{ collabore_dans. Enseignant})$
$\text{Encadreur} := \text{Enseignant} \cap (\geq 1 \text{ Oriente. Thèse})$
$\text{Thèse} := (\exists \text{ Titre. String}) \cap (\exists \text{ Orientée_par. Encadreur}) \cap (\exists \text{ Includ_dans. Projet})$ $\cap (\text{Thèse_de_recherche} \cup \text{Thèse_de_développement})$
$\text{Grade} := (\exists \text{ Nom. String})$
$\text{Cours} := (\exists \text{ Titre. String}) \cap (\geq 1 \text{ donne_par. Enseignant}) \cap (\exists \text{ concerne. Module}) \cap (\exists \text{ donne_dans. Classe})$
$\text{Module} := (\exists \text{ Titre. String}) \cap (\geq 1 \text{ constitue_de. Cours}) \cap (\exists \text{ partie_de. spécialité})$
$\text{Spécialité} := (\exists \text{ Nom. String}) \cap (\geq 1 \text{ enseigne_par. Enseignant}) \cap (> 4 \text{ Ensemble_de. Module})$
$\text{Classe} := (\exists \text{ Nom. String}) \cap (\exists \text{ lier_à. Département})$
$\text{Département} := (\exists \text{ Nom. string}) \cap (\exists \text{ adresse. string}) \cap (\exists \text{ telephone. string}) \cap (\exists \text{ fax. string})$ $\cap (\exists \text{ Collectivement_de. Classe}) \cap (\exists \text{ lier_à. Institut})$

$\text{Institut} := (\exists \text{Nom.string}) \cap (\exists \text{adresse.string}) \cap (\exists \text{telephone.string}) \cap (\exists \text{fax.string})$ $\cap (\exists \text{Comporte_de.Département}) \cap (\exists \text{Attacher_à.Université})$
$\text{Université} := (\exists \text{Nom.string}) \cap (\exists \text{adresse.string}) \cap (\exists \text{telephone.string}) \cap (\exists \text{fax.string}) \cap (\exists \text{Compose_de.Institut})$

Table 4.12 : T-Box pour le système Enseignant

$\text{Personne} := \text{Homme} \cup \text{Femme} (\exists \text{Nom. String}) \cap (\exists \text{Prénom. String}) \cap (\exists \text{Adresse. String})$ $\cap (\exists \text{Téléphone. String}) \cap (\exists \text{E-mail. String})$
$\text{Homme} := \text{Personne} \cap \neg \text{Femme}$
$\text{Femme} := \text{Personne} \cap \neg \text{Homme}$
$\text{Etudiant} := \text{Personne} \cap (\exists \text{Etudier.spécialité}) \cap (\exists \text{Possède .Numéro_d'inscription})$ $\cap (\exists \text{a_apprendre.cours}) \cap (\exists \text{étudier_dans.classe})$
$\text{Cours} := (\exists \text{Titre .String}) \cap (\exists \text{concerne.Module}) \cap (\exists \text{donne_dans.Classe}) \cap (\geq 1 \text{ a_apprendre.Etudiant})$
$\text{Module} := (\exists \text{Nom. String}) \cap (\geq 1 \text{ constitue_de.Cours}) \cap (\exists \text{partie_de.spécialité})$
$\text{Spécialité} := (\exists \text{Nom. String}) \cap (> 1 \text{ Etudier_par.Etudiant}) \cap (> 4 \text{ ensemble_de.Module})$ $\cap (> 2 \text{ Etudier sur. Année_universitaire})$
$\text{Classe} := (\exists \text{Nom. String}) \cap (\exists \text{lier_à.Département})$
$\text{Département} := (\exists \text{Nom.string}) \cap (\exists \text{adresse.string}) \cap (\exists \text{telephone.string}) \cap (\exists \text{fax.string})$ $\cap (\exists \text{Collectivement_de.Classe}) \cap (\exists \text{lier_à.Institut})$
$\text{Institut} := (\exists \text{Nom.string}) \cap (\exists \text{adresse.string}) \cap (\exists \text{telephone.string}) \cap (\exists \text{fax.string})$ $\cap (\exists \text{Comporte_de.Département}) \cap (\exists \text{Attacher_à.Université})$
$\text{Université} := (\exists \text{Nom.string}) \cap (\exists \text{adresse.string}) \cap (\exists \text{telephone.string}) \cap (\exists \text{fax.string})$ $\cap (\exists \text{Compose_de.Institut})$
$\text{Année universitaire} := (\exists \text{Numéro.string}) \cap (\exists \text{constituant_de.Spécialité})$

Table 4.13 : T-Box pour le système Etudiant

a).1 Définition de concepts

Ensemble_de (Spécialité, Module)	Partie_de(Module, Spécialité)
collabore_dans (Enseignant, Institut)	
Travaille_avec (Enseignant, Enseignant)	Travaille_avec (Enseignant, Enseignant)
Donne (Enseignant, Cours)	Donne_par (Cours, Enseignant)
Concerne (Cours, Module)	
Avoir (Enseignant, Grade)	
Travaille_dans (Enseignant, Classe)	
Enseigne (Enseignant, Spécialité)	Enseignée_par (Spécialité, Enseignant)
Participe_à (Enseignant, Institut)	
Orienté (Encadreur, Thèse)	Orientée_par (Thèse, Encadreur)
Donne _dans (Cours, Classe)	
Dirige (Enseignant, Projet)	Dirige_par (Projet, Enseignant)
Est_un (Encadreur, Enseignant)	
Sur_un(Projet, Domaine)	
Inclut_dans(Thèse, Projet)	
Un_raisonnement_de (Thèse_de_développement, Thèse)	
Un_raisonnement_de (Thèse_de_recherche, Thèse)	
Collectivement_de(Département, Classe)	Lier_à (Classe, Département)
Comporte_de(Institut, Département)	Relier_à (Département, Institut)
Compose_de(Université, Institut)	Compose_de(Institut, Université)

Table 4.14 : Définition de concepts pour le système Enseignant

Ensemble_de (Spécialité, Module)	Partie_de(Module, Spécialité)
Former dans(Etudiant,Année_universitaire)	
Ensemble_de (Classe, Etudiant)	
Etudier_dans (Etudiant, Classe)	
Etudier_avec (Etudiant, Etudiant)	Etudier_avec (Etudiant, Etudiant)
A_apprendre (Etudiant, Cours)	
Concerne (Cours, Module)	Constitue de(Module, Cours)
Possède(Etudiant, Numéro_d'inscription)	

Etudier_dans (Etudiant, Classe)	
Etudier (Etudiant, Spécialité)	
Collectivement_de(Département, Classe)	Lier_à (Classe, Département)
Comporte_de(Institut, Département)	Relier_à (Département, Institut)
Compose_de(Université, Institut)	Compose_de(Institut, Université)
Etudier_dans (Etudiant,Classe)	
Constituant_de (Année_universitaire, Spécialité)	

Table 4.15 : Définition de concepts pour le système Etudiant

a).2 Définition de rôles

Personne \subseteq T	Enseignant \subseteq Personne	Thèse de développement \subseteq Thèse
Domaine \subseteq T	Département \subseteq Institut	Thèse de recherche \subseteq Thèse
Spécialité \subseteq T	Module \subseteq Spécialité	Magister \subseteq Grade
Grade \subseteq T	Encadreur \subseteq Enseignant	Docteur \subseteq Grade
Université \subseteq T	Institut \subseteq Université	Maitre de conférences \subseteq Grade
Institut \subseteq T	Département \subseteq Institut	
Homme \subseteq Personne	Classe \subseteq Département	Professeur \subseteq Grade
Projet \subseteq Domaine	Thèse \subseteq Projet	

Table 4.16 : Définition de rôles pour le système Enseignant

Personne \subseteq T	Année universitaire \subseteq T	Institut \subseteq Université
Numéro d'inscription \subseteq T	Etudiant \subseteq Personne	Département \subseteq Institut
Spécialité \subseteq T	Homme \subseteq Personne	
Université \subseteq T	Femme \subseteq Personne	Classe \subseteq Département
Institut \subseteq T	Module \subseteq Spécialité	

Table 4.15 : Définition de rôles pour le système Etudiant

b) Représentation de la partie assertionnelle A-BOX :

La partie assertionnelle (A-box), est un ensemble d'axiomes décrivant des situations concrètes, par rapport au domaine que l'on investit. Pour chaque individu (instance définie dans la table des instances) et pour chaque rôle (relation définie dans la table des assertions), nous décrivons les deux formes suivantes :

- **a : C** où *C* est un concept défini et *a* est un individu.
- **(a1, a2) : R** où *R* est un rôle défini et *a1, a2* sont deux individus définis.

Kazar Okba : Enseignant	(Intelligence artificielle, Système multi agents) : Ensemble_de
Mohamed Tayeb Laskri : Enseignant	(Intelligence artificielle, Réseaux de neurones) : Ensemble_de
Soltani Mohamed : Enseignant	(Génie logiciel, cycles de génie logiciel) : Ensemble_de
Boudiaf Noura : Enseignant	(Intelligence artificielle, cycles de génie logiciel) : Ensemble_de
Intelligence artificielle : Spécialité	(Intelligence artificielle, Type d'agent) : Ensemble_de
Génie logiciel : Spécialité	(Intelligence artificielle, Type d'agent) : Ensemble_de
Système multi agent : Module	(Kazar Okba, Séances de l'ingénieur) : collabore_dans
Réseaux de neurones : Module	(Kazar Okba, Mohamed Tayeb Laskri) : Travaille_avec
Web-sémantiques : Module	(Kazar Okba, Système multi agents) : Donne
cycles de génie logiciel : Module	(Système multi agents, Kazar Okba) : Donne_par
Type d'agent : Cours	(Type d'agent, SMA) : Concerne
SMA : Cours	(Kazar Okba, Docteur d'état) : Avoir
Les ontologies : Cours	(Kazar Okba, INF 1) : Travaille_dans
Les neurones multi couches : Cours	(Kazar Okba, Intelligence artificielle) : Enseigne
Introduction en génie logiciel : Cours	(Intelligence artificielle, Kazar Okba) : Enseignée_par
INF 1 : Classe	(Kazar Okba, Séances de l'ingénieur) : Participe_à
ELEC 5 : Classe	(Kazar Okba, Ontologies) : Oriente
AR 6 : Classe	(Cours, Salle 3) : Donne_dans
GES 10 : Classe	(Kazar Okba, Web sémantique) : Dirige
DR 8 : Classe	(Web sémantique, Kazar Okba) : Dirige par
Informatique : Département	(Kazar Okba, Encadreur) : Devient
Electronique : Département	(Web sémantique, Web) : Sur_un
Gestion : Département	(Ontologies, Web sémantique) : Inclut_dans
Langue arabe : Département	(La création d'une ontologie, Ontologies) : Un_raisonnement_de
Droit : Département	(Multi_ontologies, Ontologies) : Un_raisonnement_de
Séances de l'ingénieur : Institut	(Informatique, Salle 3) : Collectivement_de
Economie Gestion : Institut	(INF 3, Informatique) : Lier_à
Lettres et Sciences Sociales : Institut	
Droit et Séances politiques : Institut	
Université de Biskra : Université	

<p>C.U de Souk-Ahras : Université Université de Constantine : Université Université d'Oum el Bouaghi : Université Université Badji Mokhtar Annaba : Université Web : Domaine Réseaux : Domaine Atelier de génie logiciel : Domaine Web sémantique : Projet Réseaux des capteurs : Projet Approches par composant : Projet Multi ontologies : Thèse de recherche Apprentissage dans les Réseaux des capteurs : Thèse de recherche La création d'une ontologie : Thèse de développement Une nouvelle approche par composant : Thèse de développement</p>	<p>(Séances de l'ingénieur , Informatique) : Comporte_de (Informatique, Séances de l'ingénieur) : Relier_à (Université de Biskra, Séances de l'ingénieur) : Compose_de (Séances de l'ingénieur, Université de Biskra) : Attacher_à </p>
--	---

Table 4.16: la table **A-BOX** pour le système Enseignant

<p>Goudjil Lakhdar : Etudiant Nour adine Gasmaallah : Etudiant Tebassi Kamel : Etudiant Première : année universitaire Deuxième : année universitaire 2006/52 : année universitaire 2006/20 : année universitaire 2007/12 : année universitaire 2007/05 : année universitaire Intelligence artificielle : Spécialité Génie logiciel : Spécialité Système multi agent : Module Réseaux de neurones : Module Web-sémantiques : Module cycles de génie logiciel : Module Type d'agent : Cours SMA : Cours Les ontologies : Cours Les neurones multi couches : Cours Introduction en génie logiciel : Cours</p>	<p>(Intelligence artificielle, Système multi agents): Composition_de (Système multi agents, Intelligence artificielle) : Partie_de (Etudiant,Année_universitaire) : Former_dans (Goudjil Lakhdar, INF 1) : Etudier_dans (Goudjil Lakhdar, Nour adine Gasmaallah) : Etudier_avec (Goudjil Lakhdar, Type d'agent) : A_apprendre (Goudjil Lakhdar, Type d'agent) : A_apprendre_par (Type d'agent, Système multi agents) : Concerne (Système multi agents, Type d'agent) : Constitue_de (Goudjil Lakhdar, 2007/52) : Contient (Goudjil Lakhdar, INF 1) : Etudier_dans (Goudjil Lakhdar, Intelligence artificielle) : Etudier (Informatique, INF 1) : Collectivement_de (INF 1, Informatique) : Lier_à (Séances de l'ingénieur , Informatique) :</p>
--	--

INF 1 : Classe	Comporte_de
ELEC 5 : Classe	(Informatique, Séances de l'ingénieur) : Relier_à
AR 6 : Classe	(Université, Séances de l'ingénieur) :
Informatique : Département	Compose_de
Electronique : Département	(Séances de l'ingénieur, Université de Biskra) :
Langue arabe : Département	Attacher_à
Séances de l'ingénieur : Institut	(Informatique , INF 1) : Collectivement _de
Economie Gestion : Institut	(Salle 3, Informatique) : Reliée à
Lettres et Sciences Sociales : Institut	(Goudjil Lakhdar, INF 1) : Etudier_dans
Droit et Séances politiques : Institut
Université de Biskra : Université	
C.U de Souk-Ahras : Université	
Université de Constantine : Université	
Université d'Oum el Bouaghi : Université	
Université Badji Mokhtar Annaba : Université	

Table 4.16: la table A-BOX pour le système Etudiant

V. CONCLUSION

Dans ce chapitre, nous avons présenté une étude de cas qui consiste à construire deux ontologies locales de l'encadrement universitaire. Nous avons montré ainsi comment chaque ontologie locale a été identifier et à organiser, à partir des sources d'informations, pour décrire la sémantique dans un langage semi-formel, et à travers un ensemble de représentations semi formelles puis formalisée en utilisant le formalisme de la logique de descriptions.

Dans le chapitre suivant nous utilisons l'éditeur Protégé 3.3.1 pour généré un code OWL pour créer les ontologies, et la plate-forme JADE pour créer les agents d'ontologies qui communique entre eux pour gérer le mapping ontologique.

Le chapitre suivant, ces l'implémentation, L'objectif de cette étape est de généré un code OWL pour créer les ontologies, nous utiliserons pour ce la protégé 3.3.1.en suite pour les agents d'ontologies, Nous avons, intégré dans la plate-forme JADE l'éditeur d'ontologies Protégé. En suite, on génère et sauve les ontologies en fichiers « Web Ontology Language»(OWL). Pour le mapping d'ontologie les agents communiqués entre eux s'échangent des messages en respectant un formalisme basé sur la théorie des actes de langage. Le langage de communication FIPA-ACL.

CHAPITRE 05:

IMPLEMENTATION

I. INTRODUCTION

De nombreux outils permettent aujourd'hui d'éditer des ontologies. Parmi ceux-ci, quelques uns essaient de guider leur utilisateur dans l'élaboration de l'ontologie en suivant une méthodologie de conception plus ou moins complète, que ce soit en respectant des principes de cycle de vie et validation logiciels, d'un côté, ou, de l'autre côté, en outillant une réflexion épistémologique. Dans tous les cas, force est de constater qu'aucun de ces outils n'a réussi à s'imposer et la réflexion sur l'outillage de la construction des ontologies reste donc ouverte. Les outils proposés aujourd'hui proposent à leur utilisateur de créer l'ontologie de manière relativement indépendante de tout langage implémenté et prennent ensuite automatiquement en charge l'opérationnalisation de l'ontologie, en la transposant dans divers langages. Cette évolution tend à rapprocher les ontologies de leur but original.

II. L'IMPLEMENTATION DES ONTOLOGIES

L'objectif de cette étape est de généré un code OWL pour les deux Ontologies formaliser précédemment, nous utiliserons pour ce la protégé 3.3.1. Protégé-3.3.1 est un éditeur qui permet de construire une ontologie pour un domaine donné, de définir des formulaires d'entrée de données, et d'acquérir des données à l'aide de ces formulaires sous forme d'instances de cette ontologie. Protégé est également une librairie Java qui peut être étendue pour créer de véritables applications à bases de connaissances en utilisant un moteur d'inférence pour raisonner et déduire de nouveaux faits par application de règles d'inférence aux instances de l'ontologie et à l'ontologie elle-même (méta-raisonnement).

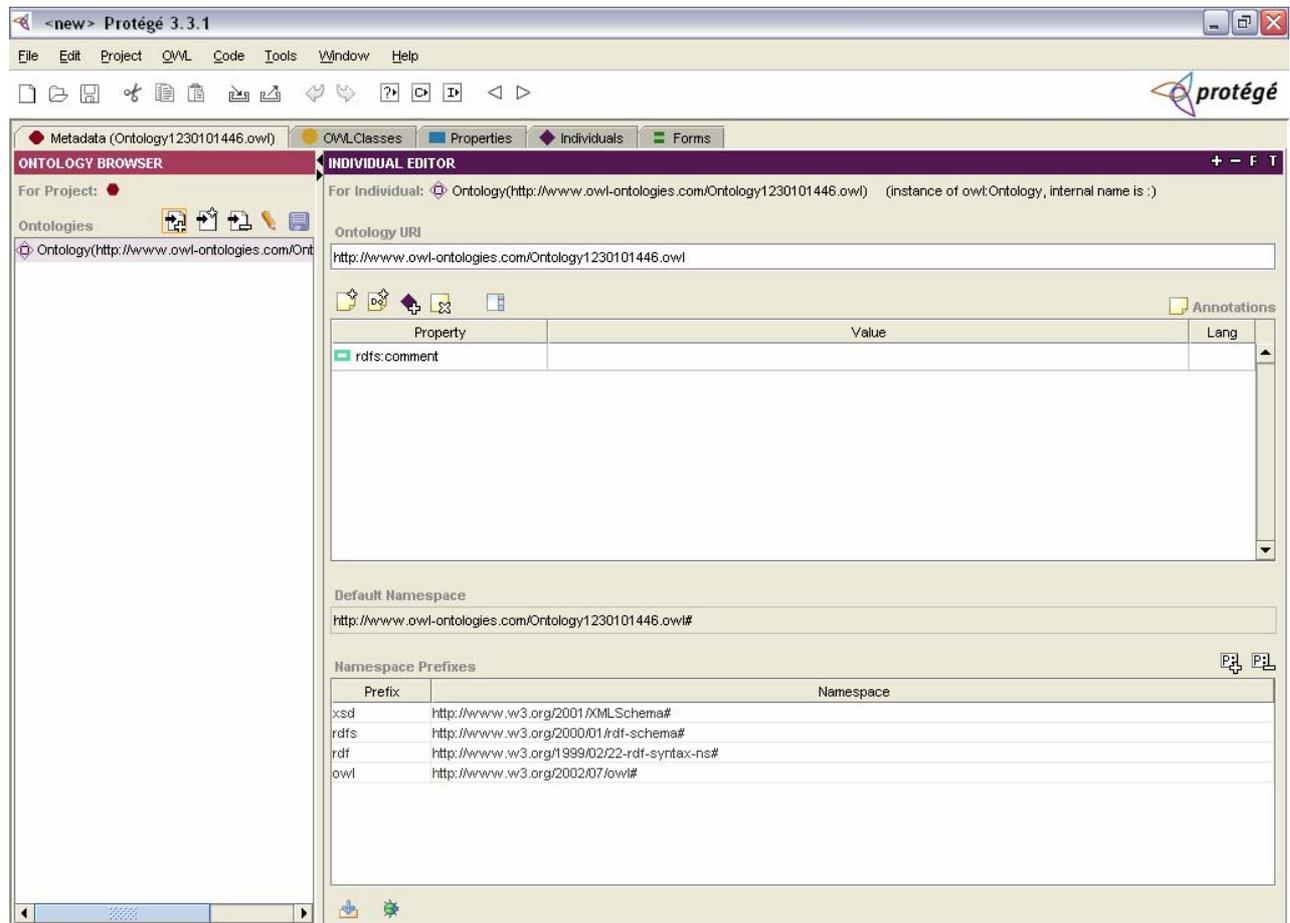


Figure 5.1: L'interface du protégé 3.3.1

Pour chaque ontologie nous suivons les étapes suivantes.

1. *Création de la hiérarchie des classes*
2. *Création des propriétés d'une classe*
3. *Création des rôles entre les classes*
4. *descriptions des classes*
5. *Création des instances*

II.1. Définir les classes et la hiérarchie des classes

Il existe un certain nombre d'approches possibles pour développer une hiérarchie de classes (Uschold and Gruninger 1996) :

- Un procédé de développement de haut en bas commence par une définition des concepts les plus généraux du domaine et se poursuit par la spécialisation des concepts.
 - Un procédé de développement de bas en haut commence par la définition des classes les plus spécifiques, les feuilles d'une hiérarchie, et se poursuit avec le regroupement de ces classes en concepts plus généraux.
 - Un procédé combiné de développement est une combinaison des deux approches, de haut en bas et de bas en haut. Au tout début, les concepts les plus saillants sont définis, ensuite ils sont généralisés ou spécialisés, suivant le cas.

▪ Aucune de ces trois méthodes n'est fondamentalement meilleure que les autres. L'approche à adopter dépend fortement du point de vue personnel sur le domaine. Si un développeur à un point de vue systématique de-haut-en-bas du domaine, il peut lui être plus commode d'utiliser l'approche de-haut-en-bas. L'approche combinée est souvent, la plus facile à utiliser pour la plupart des développeurs d'ontologies, étant donné que les concepts « du milieu » ont tendance à être les concepts les plus descriptifs du domaine (Rosch 1978).

▪ **ONTOLOGIE ENCADREUR**

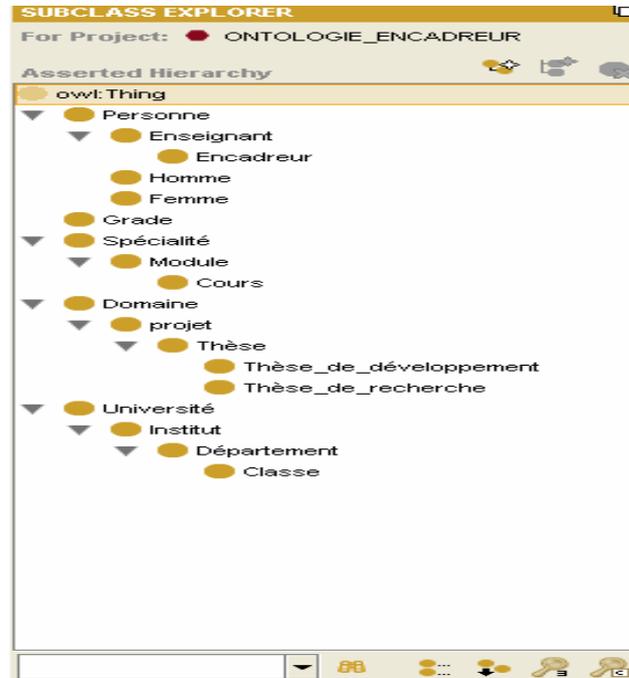


Figure 5.2 : la hiérarchie des classes d'ontologie Encadreur

▪ **ONTOLOGIE ETUDIANT**

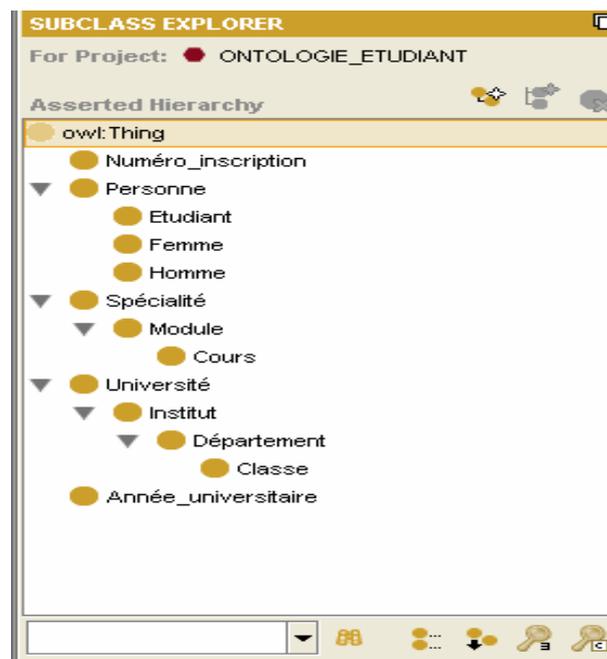


Figure 5.3 : la hiérarchie des classes d'ontologie Etudiant

II.2. Définir les propriétés des classes – attributs

Les classes seules ne fourniront pas assez d'information pour répondre à différentes questions de compétence. Après avoir défini les classes de chaque ontologie, nous devons décrire la structure interne des concepts. Nous avons déjà sélectionné des classes à partir de la liste des termes que nous avons créée pendant la première étape, il convient de déterminer les propriétés (les attributs) qui leur sont associées et de spécifier les valeurs permises pour ces propriétés.

Si en prend par exemple pour créer la propriété *Titre* (La figure ****) il faut suivi les étapes suivantes :

- Dans l'onglet property insérer une nouvelle propriété Data Property.
- Dans l'onglet For Property écrire le nom de la propriété : *Titre*
- Définir le type de la propriété dans la zone Range : *String*
- Définir le domaine de la propriété qui représente les classes de cette propriété : *Cours* , *Domaine* , *Projet* , *Thèse* , *Thèse de développement* , et *Thèse de recherche*.

▪ ONTOLOGIE ENCADREUR

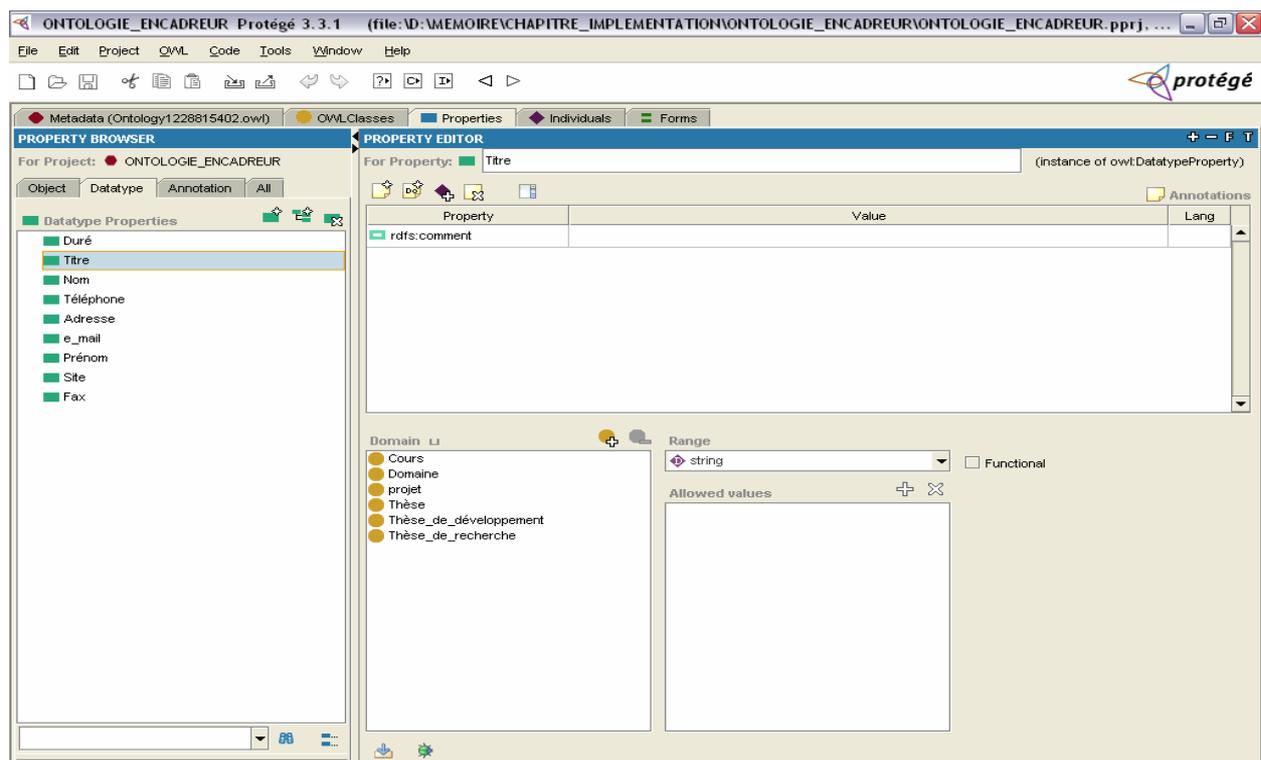


Figure 5.4 : les propriétés des classes pour l'ontologie Encadreur

▪ ONTOLOGIE ETUDIANT

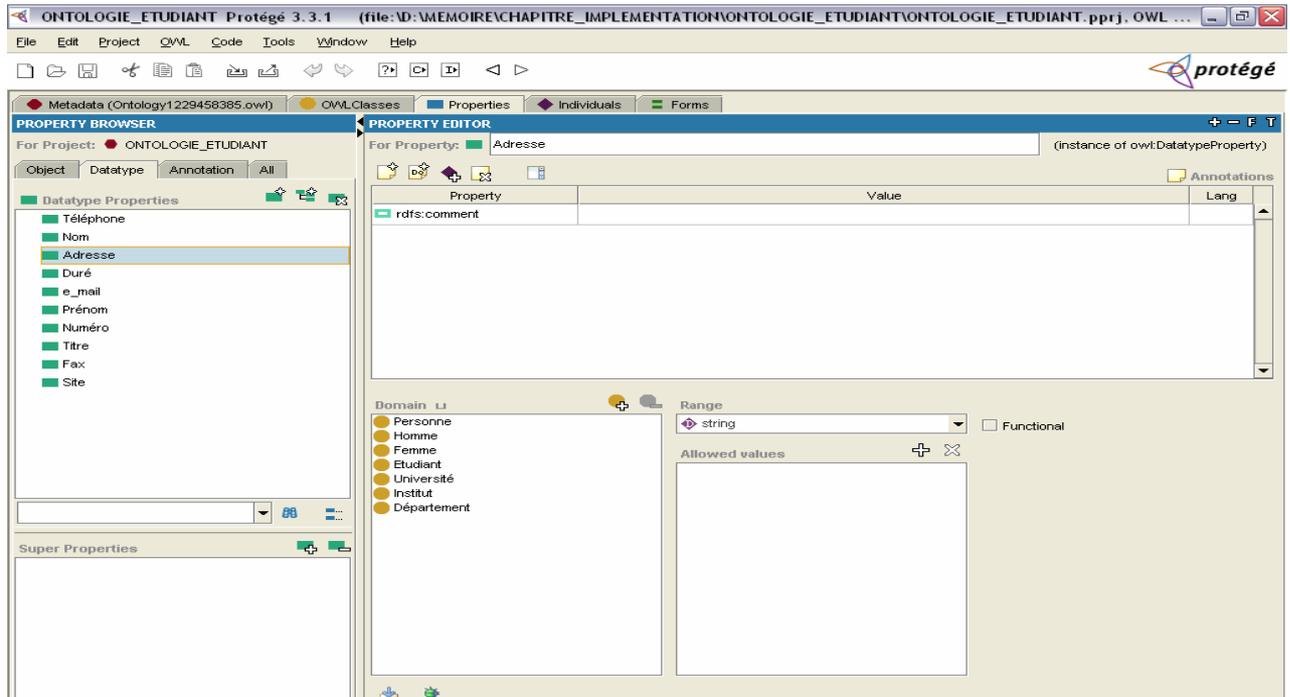


Figure 5.5 : les propriétés des classes pour l'ontologie Etudiant

II.3. Création des rôles entre les classes

La définition d'un rôle suit un schéma presque identique: la seule différence étant cette fois-ci la spécification d'Instance comme *Value type*, jointe au nom de la classe autorisée comme valeur d'objet. Pour définir une relation entre deux concepts (classe), il faut ajouter dans la zone des propriétés et donnée le nom de celle-ci, puis définir le classe source dans la zone **Domaine** et le classe cible dans la zone **Range** et aussi définir une relation inverse à cette propriété dans la zone **inverse** (si elle existe).

▪ ONTOLOGIE ENCADREUR

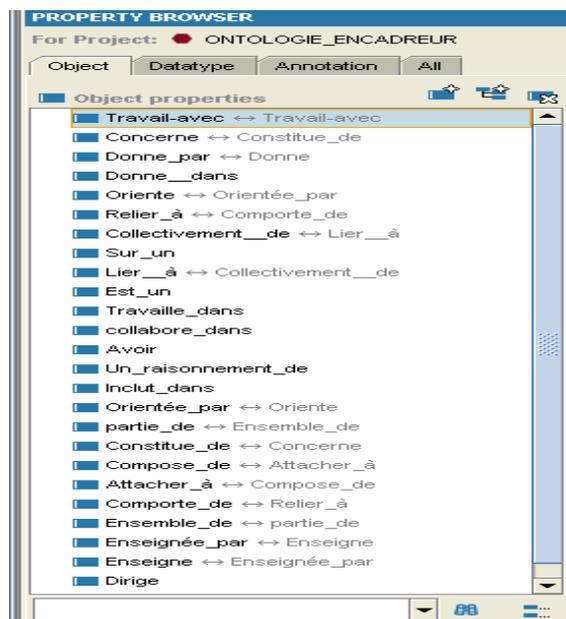


Figure 5.6 : des rôles entre les classes d'ontologie Encadreur

La figure suivante rend compte de la définition du rôle *travail_avec* entre la classe source *Enseignant* (dans la zone **Domaine**) et la classe cible *Enseignant* (dans la zone **Range**).

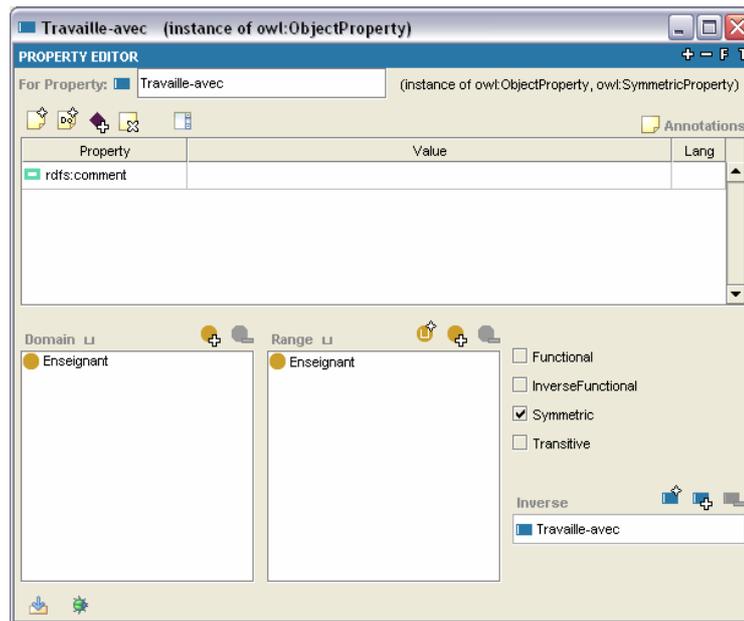


Figure 5.7: la création du rôle Travail_avec dans d'ontologie Encadreur

- **ONTOLOGIE ETUDIANT**

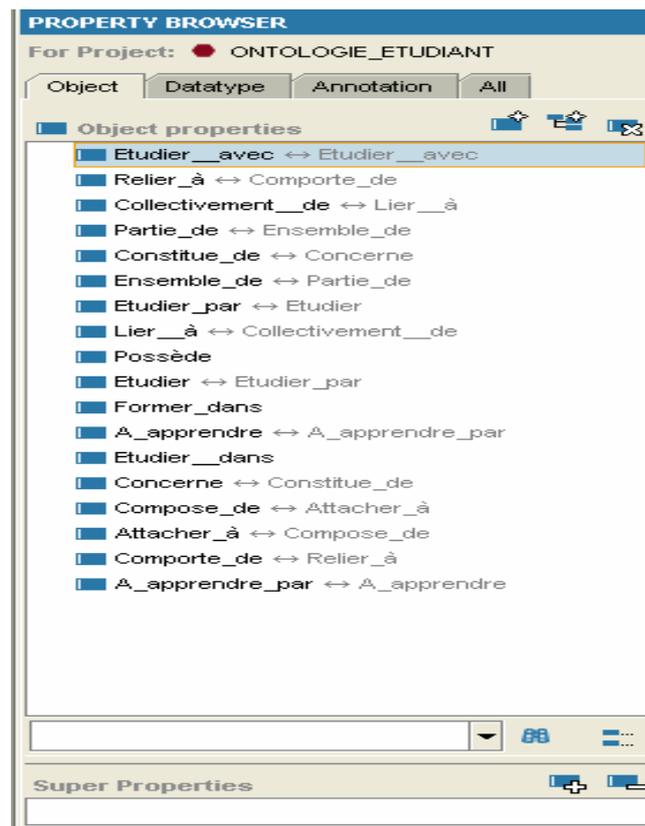


Figure 5.8: des rôles entre les classes d'ontologie Etudiant

La figure suivante rend compte de la définition du rôle *Etudier_avec* entre la classe source *Etudiant* (dans la zone **Domaine**) et la classe cible *Etudiant* (dans la zone **Range**).

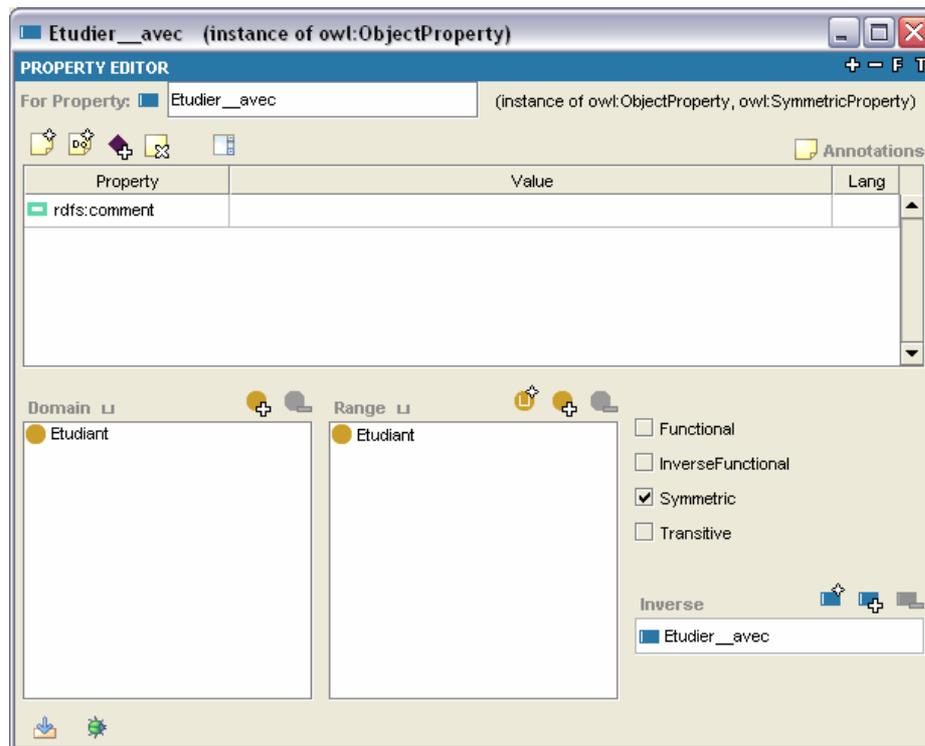


Figure 5.9: la création du rôle *Etudier_avec* dans l'ontologie *Etudiant*

II.4. descriptions des classes

Dans cette partie nous allons implémenter les règles de la logique descriptive définie dans la partie formalisation, certaines règles sont plus descriptives que la formalisation telle que la règle **Nom= 1** qui représente une et une seule instance ne peut pas être représentée dans la logique descriptive.

- **ONTOLOGIE ENCADREUR**

La figure suivante donne la description de la classe *Encadreur*. *Encadreur* est un enseignant qui oriente une thèse. Un enseignant et une personne qui travaillent au moins avec un enseignant et il doit avoir un grade et il doit enseigner au moins une spécialité et donner un cours et il doit travailler au moins dans une classe et collaborer dans un institut et il dirige un projet.

II.5. Créer les instances

La dernière étape consiste à créer les instances des classes dans la hiérarchie. Définir une instance individuelle d'une classe exige (1) choisir une classe, (2) créer une instance individuelle de cette classe, et (3) la renseigner avec les valeurs des attributs.

▪ **ONTOLOGIE ENCADREUR**

A titre d'exemple, pour instancier la classe *Encadreur*, Nous retrouvons comme il se doit des propriétés héritées de la classe *Personne* (Nom , Prénom ,E_mail Téléphone et Adresse) et des propriétés héritées de la classe *Enseignant* (spécialité, module, cours,...) dont la classe *Encadreur* dérive.

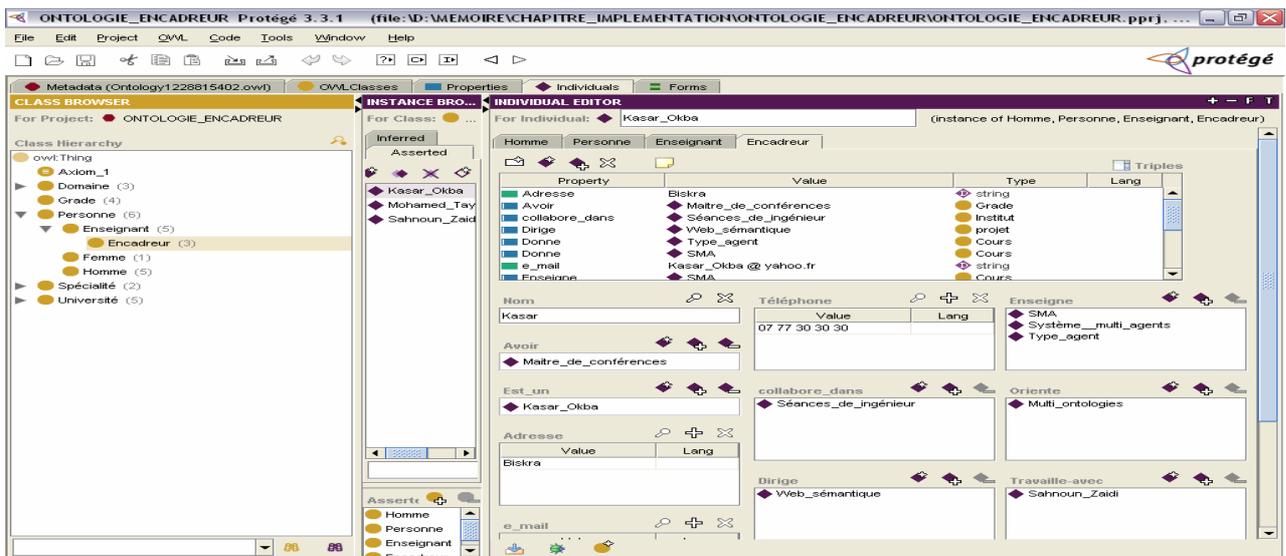


Figure 5.12 : la création des instances pour l'ontologie Encadreur

▪ **ONTOLOGIE ETUDIANT**

A titre d'exemple, pour instancier la classe *Etudiant*, Nous retrouvons comme il se doit les propriétés héritées de la classe *Personne* (Nom, Prénom, E_mail Téléphone et Adresse) dont la classe *Etudiant* dérive.

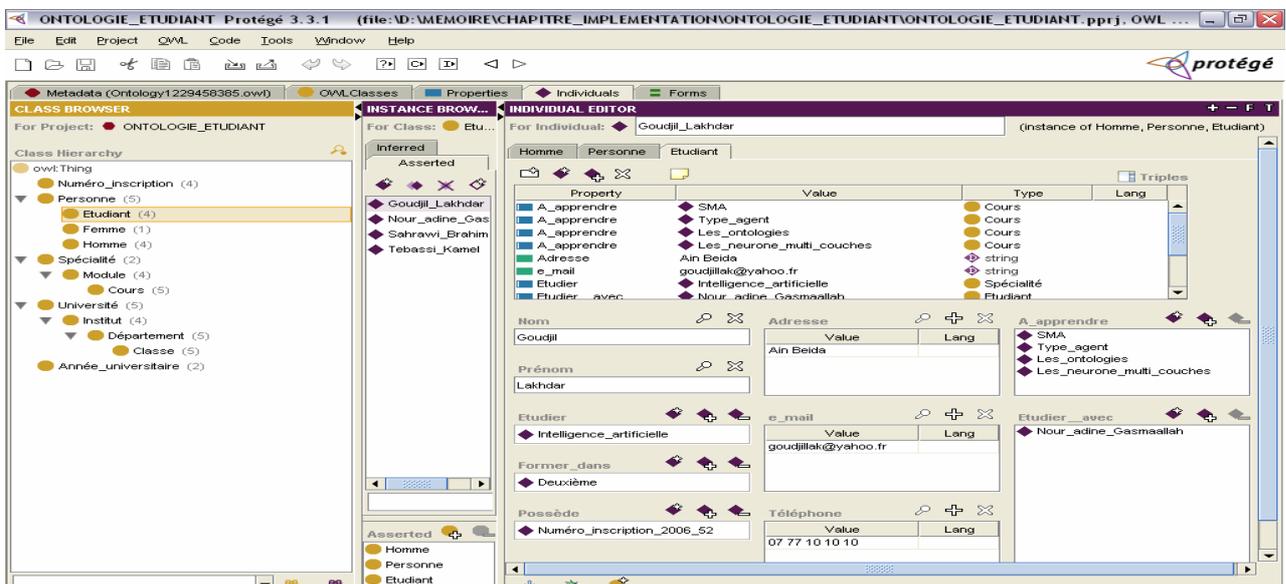


Figure 5.13 : la création des instances pour l'ontologie Etudiant

II.6. Code OWL généré par Protégé 3.3.1

Une fois notre ontologie construite et enregistré, nous obtiendrons un document au format OWL qui peut être utilisé par d'autre application (JADE).

Dans ce qui suit, nous présentons quelques déclarations selon la syntaxe OWL pour la classe Encadreur :

```
<owl:Class rdf:ID="Encadreur">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Oriente"/>
      <owl:minCardinality rdf:datatype="&xsd:int">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Enseignant"/>
</owl:Class>
<owl:Class rdf:ID="Enseignant">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Avoir"/>
      <owl:cardinality rdf:datatype="&xsd:int">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Personne"/>
  <rdfs:subClassOf>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:onProperty rdf:resource="#collabore_dans"/>
          <owl:someValuesFrom rdf:resource="#Institut"/>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
```

III. L'IMPLEMENTATION DES AGENTS

III.1 Les agents d'ontologies

Le système multi-agent a été développé en utilisant la plate-forme intégrée de développements d'agents Java Agent Développment Framework (JADE). Pour parvenir à implémenter notre système multi-agent il nous a fallu répondre à trois questions:

Comment est-il possible de créer des agents aux ontologies déférentes de façon systématique, automatique et intégrée?

Comment peut-on assurer une compréhension nette des intentions commerciales (de négociation) entre agents ayant des vocabulaires distincts?

☑ Quels protocoles d'interaction devra-t-on utiliser an de négocier et de résoudre les problèmes d'interopérabilité?

Pour répondre à la première question, nous utilisons une nouvelle méthodologie. Cette méthodologie est intégrée dans la plate forme JADE et permet la création et l'usage, dans un même système multi-agent, d'ontologies déférentes. Pour y parvenir, nous avons, d'abord, intégré dans la plate-forme JADE l'éditeur d'ontologies Protégé. En suite, on génère et sauve les ontologies en fichiers « Web Ontology Language»(OWL). Développer précédemment Ainsi, au moment de la création du système multi-agent, l'utilisateur peut choisir le fichier d'ontologie OWL qui correspond à l'ontologie désirée.

III.2 Description et Architecture logiciel de la plate-forme JADE

JADE (Java Agent DEvelopment framework) est une plate-forme multi-agent créé par le laboratoire TILAB et décrite par Bellifemine et al. JADE reprend l'architecture de l'Agent Management Reference Model proposé par FIPA. Les différents modules présentés dans la figure suivante sont présentés sous forme de services. Les services de base proposés sont le Directory Facilitator (DF) et l'Agent Management System (AMS). Il est possible de lui demander de tenir en plus le service de Message Transport Service (MTS) pour communiquer entre plusieurs plates-formes. Mais ce service sera chargé à la demande pour ne conserver par défaut que les fonctionnalités utiles à tout type d'utilisation.

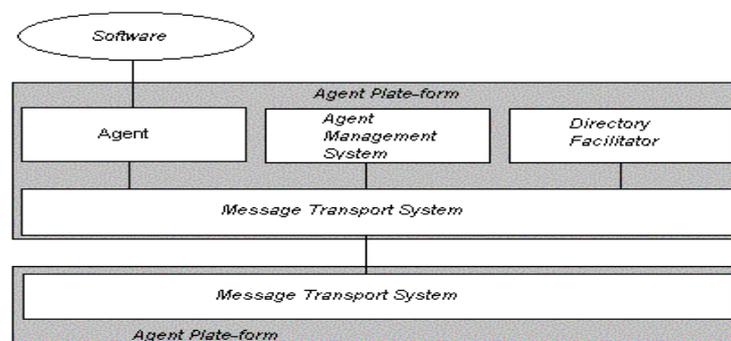


Figure 5.14 : Architecture logiciel de La plate-forme JADE

L'objectif de cette étape est de transférer ici le code OWL d'ontologie Encadreur et d'ontologie Etudiant formalisé précédemment, à deux agents cognitifs : Agent d'Ontologies Encadreur **AOEn**, et Agent d'Ontologie Etudiant **AOEt**. et nous créons aussi un agent utilisateur (agent réactifs), pour chaque système interopérable (système Encadreur et système Etudiant). chaque agent d'ontologie utilise le modèle utilisateur proposé par l'agent utilisateur, pour proposer une solution pour une requête étendue.

IV. SERVICES DE MAPPINGS D'ONTOLOGIES

Pour Chaque ontologie encadreur et ontologie étudiant leur propre agent ontologique (AO), les systèmes multi-agents gagnent à s'appuyer sur des ontologies ici pour marcher le mapping entre les deux ontologies, et, chaque agent d'ontologie en collaboration avec un agent utilisateur dans chaque système, et utilise pour communiquera avec l'utilisateur. Pour communiquer entre eux les agents s'échangent des messages, en utilisant le « Agent Communication Langage » .Ce langage d'agent que nous avons présenté ACL-FIPA, il a été développé de façon à permettre l'interaction entre agents artificiels.

V. SCENARIO D'UTILISATION

Dans cette partie nous considérons notre cas d'utilisation *l'encadrement universitaire* relativement, sans toutefois être implantable directement. Nous partons en effet d'hypothèses relatives à l'environnement technique dans lequel se déroule le scénario, ainsi que d'hypothèses par rapport au niveau applicatif se trouvant en amont de l'exécution de notre protocole. Signalons que nous ne considérons ici que des agents d'ontologies encadreur et étudiant par rapport aux contraintes d'utilisation du protocole.

VI.1 Les acteurs du système

Nous présentons ici les différentes entités intervenant au sein du scénario, ainsi que leurs objectifs.

- √ Une ontologie Encadreur **OEEnc**
- √ Une ontologie Etudiant **OEtu**
- √ Un agent utilisateur **AU**
- √ Un agent d'ontologie Encadreur **AOEnc**,
- √ Un agent d'ontologie Etudiant **AOEtu**,

Nous considérons que l'agent d'ontologie Encadreur AOEnc, et l'agent d'ontologie Etudiant AOEtu, évolue dans le même domaine de connaissances (l'encadrement universitaire), mais pour échanger des informations liées au ce domaine. Nous déjà créons précédemment les ontologies locale : l'ontologie Encadreur OEEnc et l'ontologie Etudiant OEtu pour gérer le mapping entre eux par ces deux agents.

V.2 Déroulement du scénario

Nous étudions dans ce paragraphe une communication entre l'agent AOEnc et l'agent AOEtu, au cours de laquelle notre protocole PMAO sera amené à être utilisé pour accélérer le mapping entre les deux ontologies OEEnc et OEtu.

▪ Etape 1

Un étudiant de fin des études besoin un thème à réalisera et donc d'un encadreur, et aussi un encadreur peut demandera à un étudiant réaliser un thème.

Pour effectuer cette opération il est possible de définir le mapping entre les deux d'ontologies *OE_{Enc}* et *OE_{Tu}* comme un ensemble de correspondances (liens et nommées aussi mappings). qui établit une relation sémantique entre les deux ontologies *OE_{Enc}* et *OE_{Tu}*.

▪ Etape2

Pour effectuera des lien entre les deux ontologies *OE_{Tu}* et *OE_{Enc}* , l'agent d'ontologie *AOE_{Tu}*, communique avec l'agent d'ontologie Etudiant *AOE_{En}*.

▪ Etape3

Pour réalisera des liens entre les deux ontologies : *OE_{Tu}* et *OE_{Enc}* il faut premièrement tester la communication entre les deux agents *AOE_{Tu}* et *AOE_{Enc}* (exprimé en FIPA ACL, que l'on suppose connu des 2 agents).

✓ Etape 3a

Si l'agent *AOE_{En}* ne supporte pas notre protocole et ne peut pas interpréter le message, deux options s'offrent à lui: soit il ne répond pas et l'agent *AOE_{Tu}* aura le choix entre réessayer d'envoyer un message ou abandonner, soit il a par défaut la faculté de répondre par un message du type NOT-UNDERSTOOD ou FAILURE.

✓ Etape 3b

Dans ce cas l'agent *AOE_{En}* va tenter de "comprendre" le message en exploitant le protocole. Il s'utilise alors le module de service mapping pour comprendre le message transmit.

▪ Etape 4

L'agent d'ontologie encadreur *AOE_{Enc}* reprendre envoyer une repense positif à l'agent d'ontologie étudiant *AOE_{Tu}*.

▪ Etape 5

L'agent *AOE_{Tu}* demande donc une mapping leur ontologie locale *OE_{Tu}* et l'ontologie locale *OE_{Enc}* d'agent encadreur *AOE_{Enc}* (règle map-xxx du protocole).

▪ Etape 6

L'agent d'ontologie *AOE_{Tu}* et L'agent d'ontologie *AOE_{Enc}* mapping les deux ontologies *OE_{Tu}* et *OE_{Enc}* pour reprendre à la demande de la requête (la repense) de l'*AOE_{Tu}*. dans le but d'envoyer les informations du l'étudiant et d'obtenir plus d'informations sur les encadreurs, les projets, et les thèses proposant. Il envoie le message suivant à l'agent *AOE_{En}*.

▪ Etape 7

L'agent AOEnc initie le dialogue aussi dans le but d'obtenir les informations manquantes.

▪ Etape 8

Les noms de concepts communs aux deux ontologies OEtU et OEnc sont similaires. Un algorithme simple de mapping obtenir un mapping exploitable immédiatement., et prenant éventuellement en compte les attributs des concepts.

▪ Etape 9

Le mapping obtenu par l'agent AOEnc est satisfaisant, selon lui, pour interpréter le message envoyé par AOEtU. Il estime cependant qu'il sera amené à interagir de nouveau avec l'agent AOEtU. Il demande donc au module de service mapping d'utiliser le programme de traduction pour traduire le message basé sur le mapping qu'il vient d'obtenir.

▪ Etape 10

L'agent AOEtU calcul un mapping en fonction des paramètres qui lui ont été transmis. Nous considérons que le mapping est produit et renvoyé à AOEtU sans problème (cas de la règle map-success du protocole);

▪ Etape 11

Le mapping est mémorisé dans le module de service mapping dans les agents d'ontologies AOEtU et AOEnc.

▪ Etape 12

L'agent AOEtU peut désormais convertir les messages exprimés par l'agent AOEnc, en fonction de leur ontologie OEtU, en des messages qu'il peut directement interpréter, c'est-à-dire exprimés en fonction de son ontologie OEtU. Signalons que la traduction inverse est possible si le programme de traduction dans le module de service mapping a été conçu pour la permettre.

▪ Etape 13

Nous considérons ici qu'AOEtU reconvertit la réponse qu'il obtient suite à l'interprétation du message. Il pourra donc répondre à AOEnc en lui envoyant un message exprimé en fonction de l'ontologie OEnc.

▪ Etape 14

L'agent AOEnc répond donc à l'agent AOEtU par l'envoi du message suivant : L'agent AOEtU peut alors mettre à jour son ontologie OEtU. L'interaction s'est terminée avec succès.

VI. CONCLUSION

Dans ce chapitre, nous avons utilisé la protégé 3.3.1 pour généré un code OWL du ontologies de notre cas utilisation (*Encadrement Universitaire*):l'ontologie Encadreur et l'ontologie Etudiant ,en suite nous avons, intégré dans la plate-forme JADE l'éditeur d'ontologies Protégé pour créer les agents d'ontologies Encadreur et Etudiant. la communication entre agents est réalisée par des échanges de messages, éventuellement englobés dans un protocole de communication qui définit une session de dialogue entre deux agents. l'utilisations du modèle FIPA ACL, dont le but est de promouvoir des standards dans le monde des systèmes à agents hétérogènes et coopérants, il est largement reconnu et supporté par la communauté d'individus et d'organismes manipulant des agents. Il définit, entre autre, le format des messages échangés entre agents. FIPA-ACL supporte un certain nombre de protocoles qui sont décrits en utilisant la notation AUML.

Bien que le mapping qui nous avons présenté ici flexible dans notre cas, rappelons qu'il revient à l'algorithme d'mapping d'identifier des couples cohérents de concepts entre ontologies, en fonction de leurs noms, de leurs attributs ou encore des relations qui lient ces concepts. Le protocole que nous avons introduit et présenté tout au long de ce mémoire, et mis en situation dans ce chapitre, permet de guider le partage de connaissances entre agents d'ontologies et donc la communication entre les ontologies locales. Notons enfin que la communication présentée ici étant asynchrone, et les messages identifiés, tout échec d'une opération de la part d'un acteur du système ne met en aucun cas le système global lui-même en péril.

CONCLUSION ET PERSPECTIVES

Le Web sémantique est la vision du Web de demain, où nous pouvons effectuer plusieurs tâches de manière informatisée et automatisée grâce à l'explicitation de la sémantique des ressources et des connaissances, explicitation les rendant « compréhensibles » et manipulables par les machines et les logiciels. Dans le noyau du partager sémantiquement des connaissances et des informations. En tant qu'entités du Web, un environnement très large, très complexe et divers, les entités du Web sémantique possèdent aussi des caractéristiques de cet environnement si hétérogène. Les ontologies du Web sémantique peuvent donc être variées et hétérogènes même si elles sont créées dans un même domaine. Une des clés importantes pour le succès du Web sémantique est de maîtriser cette hétérogénéité et de cohabiter avec elle.

Pour atteindre l'objectif de la thèse qui est de permettre la construction et l'exploitation d'une approche multi ontologies dans une organisation hétérogène, nous envisageons la problématique de l'hétérogénéité de l'ontologie, qui est une composante de base importante du Web sémantique. Notre approche pour répondre à cette problématique a été la suivante :

Étudier les différentes approches d'ontologies et proposer pour l'approche multi ontologie de mettre des correspondances entre les ontologies. Bien que l'ontologie soit utilisée pour capitaliser des connaissances d'un domaine, du fait de l'ouverture du Web et donc du Web sémantique. L'objectif de nos approches est de permettre donc à des ontologies locales pour des systèmes interopérables d'échanger, de partager des données, des informations entre elles.

Étudier l'utilisation des systèmes multi-agents sur des ontologies pour fonctionner et créer des liens entre ontologies est une des dispositions de leur bonne communication. Et proposer pour chaque ontologie locale un agent d'ontologie qui prenant en charge des calculs sur cette ontologie afin de l'aligner avec les autres ontologies.

Étudier et proposer une méthode pour transférer le code OWL des ontologies locales à des agents d'ontologies cognitifs.

Étudier les architectures des agents et proposer pour les agents d'ontologies une architecture générique. Les agents sont homogènes dans leur structure et se différencient par leurs connaissances (ontologie locale) de domaine. Ainsi proposer une architecture pour l'agent utilisateur qui dialogue avec l'utilisateur.

Etudier la communication entre agents et proposer un protocole pour gérer la communication entre agents d'ontologies de notre approche.

En revanche, il est un peu plus rare de trouver, dans la littérature, des exemples où les systèmes multi-agents sont une aide pour la construction de ressources ontologiques. Dans ce cas, ils sont le plus souvent utilisés comme support pour des ateliers de construction d'ontologies collaboratifs et n'exploitent pas de résultat en provenance de textes [Bao et Honavar, 2004]. Ils sont essentiellement là pour assurer la cohérence du produit modélisé en suivant les modifications concurrentes des utilisateurs.

Dans l'état actuel de notre approche, nous distinguons plusieurs améliorations se situant à plusieurs niveaux :

1. nous avons donné une démarche à suivre, pour réaliser chaque étape de notre approche ; en déterminant quel est le comportement attendu de chaque sous système interopérable, une méthode pour l'ingénierie des besoins, La construction d'ontologies basée sur *la méthodologie d'Uschold et King.*, Le cycle de vie c'est celui proposé par l'équipe de Fernandez.

2. la création pour chaque ontologie un agent dit agent d'ontologie, cognitif basé sur une architecture contient un ensemble de modules comme le Module de communication inter-agents *qui utilise* pour faciliter, la communication entre les agents d'ontologies et avec l'agent utilisateur par envois de messages.

3. pour améliorer le mapping entre les différentes ontologies locales , nous avons proposé agents un module de Service mapping pour chaque agent d'ontologie ; et qui utilise d'un module de traduction. un programme de traduction doit permettre à un agent d'ontologie de pouvoir transformer localement un message exprimé en fonction d'une autre ontologie.

4. pour gérer les mappings sémantiques entre les différentes ontologies locales. Et donc créer des correspondances sémantiques pour concept, et les *correspondances sémantiques pour propriété*, nous avons proposé un nouveau protocole. ce protocole que nous avons présenté est à la fois modulaire et flexible : n'importe quel agent d'ontologie, peut demander à un autre agent d'ontologie d'effectuer une opération (mapping bidirectionnel). Les opérations sont atomiques et le protocole laisse donc toute liberté aux agents d'ontologies de choisir leur façon de gérer les mappings. De plus il n'y a aucune contrainte pour un agent d'ontologie ; tout arrêt de suivi du protocole est transparent et celui-ci peut être repris ultérieurement. Enfin, l'intégration dans notre approche du module de service mapping dans chaque agent d'ontologie qui contient un programme

de traduction permet d'accélérer et d'automatiser les traductions et donc géré rapidement le mapping d'ontologies.

Limites et Perspectives

1. Les ontologies du notre approche sont des ontologies locales sur des systèmes interopérables du même domaine.

2. Un certain nombre de techniques ont été proposées dans pour réaliser l'interopérabilité. Elles sont généralement utilisées pour permettre le partage des connaissances hétérogènes entre les différentes ontologies. dans notre approche système multi ontologie basé agent, nous avons basé seulement sur le mapping d'ontologie.

3. Une autre approche intéressante est d'étudier les relations entre des ontologies « classiques » et des ontologies multi-points de vue selon le modèle multi-points de vue proposé. Il est probablement intéressant de pouvoir détecter des correspondances entre des entités d'une ontologie traditionnelle et d'une ontologie multi-points de vue.

4. Une perspective est la réalisation d'un système complet permettant d'une part de gérer des multi ontologies de vue par la définition, la modification, la suppression des points de vue, des entités de l'ontologie selon des points de vue ; et d'autre part de représenter des connaissances, de filtrer des connaissance selon des points de vue des utilisateurs, de naviguer contextuellement dans la base des connaissances ou la mémoire des ontologies selon les points de vue des utilisateurs, d'exploiter des points de vue des utilisateurs pour une présentation des résultats adaptée à l'utilisateur.

REFERENCES

- [MELLAL 2007] Naçima MELLAL Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information *19 Décembre 2007*.
- [F. VERNADAT 2007]François VERNADAT. Interoperable enterprise systems: Architectures, methods and metrics. Rapport technique, LGIPM, Université de Metz, France, 2007.
- [MELLAL 2007]Naçima MELLAL Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information thèse pour obtenir le grade de docteur de l'université de Savoie *2007*.
- [Wache et al. 2001] Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., and Hübner S., "Ontology-Based Integration of Information – A Survey of Existing Approaches". In Stuckenschmidt, H., editor, *IJCAI-2001 Workshop on Ontologies and Information Sharing*, pp., 2001.
- [Mena et al. 2000] Mena E., Illarramendi A., Kashyap, V., and 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.
- [Goh 1997] Goh C. H., "Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources". Phd Thesis, MIT, 1997.
- [Wache et al. 1999] Wache H., Scholz Th., Stieghahn H., and König-Ries B., "An integration method for the specification of rule-oriented mediators". Proceedings of *DANTE'99*, pp. 109-112, Kyoto, Japan, 1999.
- [Visser et Stuckenschmidt 2002] Visser U., and Stuckenschmidt H., "Interoperability in GIS – Enabling Technologies". In Proceeding of *5th AGILE Conference on GIS*, Spain, 2002.
- [Klein 2001] Klein M., "Combining and relating ontologies: an analysis of problems and solutions". International Joint Conferences on Artificial Intelligence (IJCAI), Workshop on Ontologies and Information , USA, 2001.
- [Corcho et al. 2003-b] Corcho O., Gomez-Perez A., Leger A., Rey C., and Toumani F., "An ontology-based mediation architecture for e-commerce applications". In Proceedings of *Intelligent Information Systems (IIS2003)*, [On Line] <http://citeseer.ist.psu.edu/corcho03ontologybased.html>, 2003.
- [KnowledgeWeb 2004] KnowledgeWeb, "Deliverables of KWEB Project". EU-IST- 2004-507482, [On Line] <http://knowledgeweb.semanticweb.org/>, 2004.
- [Kavouras 2003] Kavouras M., "A unified ontological framework for semantic integration". *International Workshop on Next Generation Geospatial Information*, Cambridge, UK, 2003.
- [Euzenat et Valtchev 2004] Euzenat J., and Valtchev P., "Similarity-based ontology alignment in OWL-Lite". In the *16th European Conference on Artificial Intelligence (ECAI'04)*, 2004.
- [Maedche et Staab 2002] Maedche A., and Staab S., "Measuring Similarity between Ontologies". In Proc. of the *13th European Conf. on Knowledge Engineering and Knowledge Management EKAW-02*, 2002.
- [Sheth et Kashyap 1993] Sheth A., and Kashyap V., "So far schematically yet so near semantically". In Proceeding IFIP TC2/WG2.6 [On Line] <http://citeseer.ist.psu.edu/sheth93so.html>, 1993.

- [Saïd IZZA 2006] Saïd IZZA « *INTEGRATION DES SYSTEMES D'INFORMATION INDUSTRIELS Une approche flexible basée sur les services sémantiques* » Pour obtenir le grade de Docteur de l'Ecole Nationale Supérieure des Mines de Saint-Etienne le 20 novembre 2006
- [Bruijn et Polleres 2004] Bruijn J. and Polleres A., "Towards an Ontology Mapping Specification Language for the Semantic Web". DERI Technical report 2004-06-30, Digital Enterprise Research Institute (DERI), 2004.
- [Kalfoglou et Schorlemmer 2003] Kalfoglou Y., and Schorlemmer M., "IF-Map: an ontology mapping method based on information flow". *Journal on Data semantics*, Vol. 1 (No. 1) pp. 98-127, 2003.
- [McGuinness et al. 2000] McGuinness D. L., Fikes R., Rice J., and Wilder S., "The Chimaera Ontology Environment". In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI), 2000.
- [Bruijn et al. 2005] Bruijn J., Lausen H., Krummenacher R., Polleres A., Kifer M. and Fensel D., "Deliverable D16.1v0.2, The Web Service Modeling Language WSML". Final Draft, *WSMO project*, [On Line] <http://www.wsmo.org/TR/d16/d16.1/v0.2/20050320/>, 2005.
- [Kalfoglou et Schorlemmer 2003-b] Kalfoglou Y., and Schorlemmer M., "Ontology Mapping: The State of The Art". *The Knowledge Engineering Review*, Vol. 18 (No. 1), pp. 1-31, Cambridge University Press, 2003.
- [Noy 2004] 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.
- [Sekt 2004] Sekt, "Deliverables of SEKT Project". [On Line] <http://www.sektproject.com/>, 2004.
- [Stumme et Madche 2001] Stumme G., and Madche A., "FCA-Merge: Bottom-up merging of ontologies". In 7th *International. Conference on Artificial Intelligence (IJCAI'01)*, pp. 225-230, Seattle, WA, 2001.
- [Berlin et Motro 2002] Berlin J., and Motro A., "Database schema matching using machine learning with feature selection". In Proc. *Conference on Advanced Information System Engineering (CAiSE)*, 2002.
- [Li et Clifton 1994] Li W. S., and Clifton C., "Semantic integration in heterogeneous databases using neural networks". Jorge B. Bocca, Matthias Jarke, Carlo Zaniolo (Eds.): In Proceedings of the *20th International Conference on Very Large Data Bases (VLDB'94)*, Santiago de Chile, Chile. Morgan Kaufmann, pp. 1-12, 1994.
- [Noy et Musen 2003] Noy N. F., and Musen M. A., "The Prompt suite: Interactive tools for ontology merging and mapping". *International Journal of Human Computer Studies*, 59(6): 983-1024, 2003.
- [A. GÓMEZ-PÉREZ,al 1995] Asunción GÓMEZ-PÉREZ, Natalia JURISTO et Juan PAZOS. Evaluation and assessment of the knowledge sharing technology. In *Towards very large knowledge bases*, pages 289–296. IOS Press, 1995.
- [Alexander M; al 2002] Alexander MAEDCHE, Boris MOTIK, Nuno SILVA et Raphael VOLZ. MAFRA - A Mapping Framework for Distributed Ontologies. In *Proc. of 13th European Conference on Knowledge Engineering and Knowledge Management (EKAW)*, Siquenca, Spain, 2002.
- [Yannis K ;al 2003] Yannis KALFOGLOU et Marco SCHORLEMMER. If-map: an ontology mapping method based on information flow theory. *Journal on Data Semantics*, 1(1):98–127, october 2003.
- [B OMELAYENKO;al 2003] Borys OMELAYENKO. Rdf: A mapping meta-ontology for web service integration.
- [P.BOUQUET,al 2003] Paolo BOUQUET, Fausto GIUNCHIGLIA, Frank van HARMELEN, Luciano SERAFINI et Heiner STUCKENSCHMIDT. Contextualizing ontologiesstar, open. *Web Semantic: Science, Services and Agents on the World Wide Web*, 1(4):325–343, 2004. 137–153. 2003.

- [J. ANGELE ;al 2005] Jürgen ANGELE et Hans-Peter SCHNURR. Do not use this gear with a switching lever! automotive industry experience with semantic guides. In *GI Jahrestagung (1)*, pages 48–52, 2005.
- [NAMYOON et YEOL et HYOIL 2006] Choi NAMYOON, Song IL-YEOL et Han HYOIL. A survey on ontology mapping. *SIGMOD Rec.*, 35(3):34–41, September 2006.
- [NOY et MUSEN 2000] Natalya Fridman NOY et Mark MUSEN. Prompt: Algorithm and tool for automated ontology merging and alignment. pages 450–455, 2000.
- [MCGUINNESS] Deborah MCGUINNESS, Richard FIKES, James RICE et Steve WILDER. The chimaera ontology environment. pages 1123–1124. AAAI Press / The MIT Press.
- [STUMME et MAEDCHE 2001] Gerd STUMME et Alexander MAEDCHE. Fca-merge: Bottom-up merging of ontologies. In *IJCAI*, pages 225–234, 2001.
- [DOU, MCDERMOTT et Peishen QI 2002] Dejing DOU, Drew MCDERMOTT et Peishen QI. Ontology translation by ontology merging and automated reasoning, 2002.
- [Erhard RAHM et Philip BERNSTEIN 2001] Erhard RAHM et Philip BERNSTEIN. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.
- [GIUNCHIGLIA, SHVAIKO et YATSKEVICH 2005] Fausto GIUNCHIGLIA, Pavel SHVAIKO et Mikalai YATSKEVICH. S-match: an algorithm and an implementation of semantic matching. In Y, Schloss Dagstuhl, Germany, 2005.
- [EHRIG et STAAB 2004] Marc EHRIG et Steffen STAAB. Qom quick ontology mapping. In *International Semantic Web Conference (ISWC2004)*, Japan, November 2004.
- [DOAN, MADHAVAN, DOMINGOS et HALEVY 2003] AnHai DOAN, Jayant MADHAVAN, Pedro DOMINGOS et Alon HALEVY. *Ontology matching: A machine learning approach*. 2003.
- [LE, Rose DIENG-KUNTZ et GANDON 2004] Bach Thanh LE, Rose DIENG-KUNTZ et Fabien GANDON. On ontology matching problems - for building a corporate semantic web in a multi-communities organization. In *ICEIS (4)*, pages 236–243, 2004.
- [Antoine 2002] Antoine Billet. "Xyleme veut faciliter la recherche de documents XML". *01net*, www.01net.com/rdn?oid=177270&rub=3369, mars 2002.
- [Tim Berners,al 2001]: Tim Berners-Lee, James Hendler, and Ora Lassila. "The Semantic Web". www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2, May 2001.
- [Regis 2002] Regis Naefaire. "Xyleme se lance sur le marché de la recherche et de l'intégration de contenus XML". *Digital Business Globe*, www.xyleme.com/fr/presse/doc1.pdf, mars 2002.
- [Bach, 2006] :Bach T. L., Construction d'un web sémantique multi-points de vue, PhD thesis, École des Mines de Nice à Sophia Antipolis, 2006.
- [Franz,al 2003]: Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
- [Minsky 1980]:M. Minsky.A framework for representing knowledge. In D. Metzger, editor, *Frame Conceptions and Text Understanding*, pages 99–125. de Gruyter, Berlin, 1980.

- [Fournier 2005] Philippe Fournier-Viger. Un modèle de représentation des connaissances à trois niveaux de sémantique pour les systèmes tutoriels intelligents 2005.
- [Horrocks 1998] Ian Horrocks. The FaCT system. Pages 307–312, 1998.
- [Haarslev ,al 2001] V. Haarslev and R. Møller. Racer user's guide and reference manual version 1.6. Technical report, University of Hamburg, Computer Science Department, 2001.
- [Evren,al 2006] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet : A practical owl-dl reasoner. Submitted for publication to Journal of Web Semantics, 2006.
- [Tsarkov,al 2004] Dmitry Tsarkov and Ian Horrocks. Efficient reasoning with range and domain constraints. In Proc. of the 2004 Description Logic Workshop (DL 2004), pages 41–50, 2004.
- [f-o 2003] F-owl : An owl inference engine in flora-2. <http://fowl.sourceforge.net/>, 2003.
- [Surnia 2000] Surnia web page. <http://dev.w3.org/cvsweb/2000/10/swap/surnia/>, 2000.
- [Hoolet 2004] Hoolet web page. <http://owl.man.ac.uk/hoolet/>, 2004.
- [NOY 2002] NOY N & MUSEN M. A., Evaluating ontology-mapping tools : requirements and experience, in *Proceedings of the Workshop on Evaluation of Ontology Tools (EON'2002) at EKAW'2002*, 2002
- [Bechhofer et al 2001]: BECHHOFER S., HORROCKS I., GOBLE C. & STEVENS R. (2001). OilEd: a Reason-able Ontology Editor for the Semantic Web. In *Joint German/Austrian conference on Artificial Intelligence (KI'01)*, volume (2174) of *Lecture Notes in Artificial Intelligence*, p. 396–408, Autriche: Springer Verlag.
- [Sure et al 2002] : SURE Y., ERDMANN M., ANGELE J., STAAB S., STUDER R. & WENKE D. (2002). OntoEdit: Collaborative Ontology Engineering for the Semantic Web. In I. HORROCKS & J. HENDLER, Eds., *First International Semantic Web Conference (ISWC'02)*, volume (2342) of *Lecture Notes in Computer Science*, p. 221–235, Chia, Sardaigne, Italie: Springer Verlag.
- [Arpirez et al 2001] : ARPIREZ J., CORCHO O., FERNÁNDEZ-LÓPEZ M. & GÓMEZ-PÉREZ A. (2001). WebODE : a Workbench for Ontological Engineering. In *First international Conference on Knowledge Capture (K-CAP'01)*, p. 6–13, Victoria, Canada: ACM.
- [Bachimont et al 2002; Troncy & Isaac, 2002] : BACHIMONT B., ISAAC A. & TRONCY R. (2002). Semantic Commitment for Designing Ontologies: A Proposal. In A. GOMEZ-PÉREZ & V. BENJAMINS, Eds., *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02)*, volume (2473) of *Lecture Notes in Artificial Intelligence*, p. 114–121, Sigüenza, Espagne: Springer Verlag.
- [Angele & Sure 2002] J. ANGELE & Y. SURE , Eds. (2002). *First International Workshop Evaluation of Ontology-based Tools (EON'02)* , volume (62) of *CEUR-WS*, Sigüenza, Espagne. <http://CEURWS.org/Vol-62/>.
- [W3C RDQL, 2004] : Soumission W3C RDQL, 2004. <http://www.w3.org/Submission/RDQL/>
- [Sirin , Parsia et Grau 2006] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet : A practical owl-dl reasoner. Submitted for publication to Journal of Web Semantics, 2006.
- [W3C 2006]Candidate recommandation W3C SPARQL, 2006. <http://www.w3.org/TR/ rdf-sparql-query/>.
- [Haarslev et Moller 2001] V. Haarslev and R. Moller. Racer user's guide and reference manual version 1.6. Technical report, University of Hamburg, Computer Science Department, 2001.
- [Gandon, 2002] Fabien GANDON : Ontology Engineering : a Survey and a Return on Experience. INRIA, 2002.

- [Greenwood et al., 2007] Dominic GREENWOOD, Margaret LYELL, Ashok MALLYA et Hiroki SUGURI : The IEEE FIPA Approach to Integrating Software Agents and Web Services. AAMAS'07, May 2007.
- [Elmore et al., 2003] Mark T. ELMORE, Thomas E. POTOK et Frederick T. SHELDON : Dynamic data fusion using an ontology-based software agent system. Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics, 2003.
- [Bergenti et al., 2002] Federico BERGENTI, Agostino POGGI, Giovanni RIMASSA et Paola TURCI : Comma : a multiagent system for corporate memory management. AAMAS '02 : Proceedings of the first international joint conference on Autonomous agents and multiagent systems, pages 1039–1040, 2002.
- [Siebes et van Harmelen, 2002] Ronny SIEBES et Frank van HARMELEN : Ranking agent statements for building evolving ontologies. Workshop on Meaning Negotiation, in conjunction with the Eighteenth National Conference on Artificial Intelligence, July 2002.
- [Laera et al., 2007] Loredana LAERA, Ian BLACOE, Valentina TAMMA, Terry PAYNE, Jérôme EUZENAT et Trevor BENCH-CAPON : Argumentation over Ontology Correspondences in MAS. AAMAS'07, May 2007.
- [Lister et al., 2006] Kendall LISTER, Leon STERLING et Kuldar TAVETER : Reconciling Ontological Differences by Assistant Agents. AAMAS'06, May 2006.
- [Hernandez, 2005] Nathalie HERNANDEZ : Ontologies de domaine pour la modélisation du contexte en recherche d'information. Thèse de doctorat, Université Toulouse III, 2005.
- [Bao et Honavar, 2004]. Jie BAO et Vasant HONAVAR : Collaborative ontology building with wiki@nt. Proceedings of the Workshop on Evaluation of Ontology-Based Tools (EON2004), 2004.
- [Jiang et Conrath, 1997] Jiang, J. J., Conrath, D. W. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy, In Proceedings of International Conference Research on Computational Linguistics, Taiwan, 1997.
- [Klein et al., 2002] Klein, M., Kiryakov, A., Ogniano, D., Fensel, D. Finding and specifying relations between ontology versions. In Proceedings of the ECAI-02 Workshop on Ontologies and Semantic Interoperability. Lyon, July, 2002.
- [Rodriguez et Egenhofer, 2003] Rodriguez, M. A., Egenhofer, M. J. Determining Semantic Similarity Among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering, Vol.15(2):442-456, 2003.
- [Maedche and Staab, 2001] Maedche, A., Staab, S. Ontology Learning for the Semantic Web, IEEE Intelligent Systems, Vol.16(2), 2001.
- [Mitra et Wiederhold, 2002] Mitra, P., Wiederhold, G. Resolving Terminological Heterogeneity in Ontologies, In Proceedings of the ECAI-02, Workshop on Ontologies and Semantic Interoperability, Lyon, 2002.
- [Guarino et Welty, 2001] Guarino, N. Welty, C., Supporting ontological analysis of taxonomic relationships, Data & Knowledge Engineering, Vol.39, pp. 51-74, 2001.
- [Suguri et al., 2001] Suguri, H., Kodama, E., Miyazaki, M., Nunokawa, H., Noguchi, S. Implementation of FIPA Ontology Service. In Proceedings of the Workshop on Ontologies in Agent Systems, AAMAS, Montreal, Canada, 2001.

- [Steels, 1998] Steels, L. The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, Vol.1(2):169-194, 1998.
- [Bailin et Truszkowski, 2002] Bailin, S. C., Truszkowski, W. Ontology negotiation between intelligent information agents. *The Knowledge Engineering Review*, Vol.17(1), pp. 7-19. 2002.
- [van Eijk et al., 2001] van Eijk, R. M., Boer, F. S., van der Hoek, W., Meyer, J-J Ch. On Dynamically Generated Ontology Translators in Agent Communication, *International Journal of Intelligent Systems*, Vol.16, pp.587-607, 2001.
- [Tzitzikas et Meghini, 2003] Tzitzikas, Y., Meghini, C. Ostensive automatic schema mapping for taxonomybased peer-to-peer systems. In *Proceedings of the 7th International Workshop on Cooperative Information Agents*, Helsinki, Finland, 2003.
- [Burnstein et al., 2003] Burnstein, M., McDermott, D., Smith, D.R., Westfold, S.J., Derivation of glue code for agent interoperation. *Autonomous Agents and Multi-Agent Systems*, Vol.6(3):265-286, 2003.
- [Doherty et al., 2004] Doherty, P., Szalas, A. Lukaszewicz, W. Approximative Query Techniques for Agents with Heterogeneous Ontologies and Perceptive Capabilities, In *Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning*, 2004.
- [Williams et al., 2003] Williams, A., Padmanabhan, A., Blake, M. B. Local Consensus Ontologies for B2B-Oriented Service Composition, In: Rosenschein, J., Sandholm, T., Wooldridge, M., Yokoo, M. (eds.), *AAMAS*, pp. 647-654. ACM Press, Melborne, 2003.
- [Wiesman et Roos, 2004] Wiesman, F., Roos, N. Domain independent learning of ontology mappings, In: Jennings, N., Sierra, C., Sonenberg, L., Tambe, M. (eds.), *AAMAS*, ACM Press, New York, USA, pp.846-853, 2004.
- [van Diggelen et al., 2005] van Diggelen, J., Jan Beun, R., Dignum, F., van Eijk, R. M., Meyer, J-J. Optimal Communication Vocabularies and Heterogeneous Ontologies, In: van Eijk, R. M., Huget, M. P., Dignum, F. (eds.), *Developments in Agent Communication*, LNAI 3396, Springer Verlag, Berlin Heidelberg, pp. 76-90, 2005.
- [Malucelli et Oliveira, 2006] Andreia Malucelli :Supervisor: Professor Eugénio da Costa Oliveira :Co-supervisor: Professor Marcos Augusto Hochuli Schmeil "*Ontology-based Services for Agents Interoperability*" Porto, 2006.
- [Finkelstein 90] :Finkelstein A., Gabbay, D., Hunter, A., Kramer J., Nuseibeh, B.: Inconsistency handling in multi-perspective specifications; *Proc. 4th European Software Eng. Conf.*, Garmisch, Germany, 84-99.
- [Gruber, 1993] : Gruber T., "A Translation Approach to Portable Ontology Specifications". *Knowledge Acquisition*, Vol. 5 (No. 2) pp. 199-220, 1993.
- [Fernandez et al. 1997] : Fernandez M., Gomez-Perez A., and Juristo N., "METHONTOLOGY". *Proceedings of the AAAI97, Spring Symposium Series on Ontological Engineering*, Stanford, USA, pp. 33-40, 1997.
- [Gandon 2002] : Gandon F., "Distributed Artificial Intelligence and Knowledge Management: ontologies and multi-agent systems for a corporate semantic web". PhD Thesis, INRIA and University of Nice - Sophia Antipolis, 2002.
- [Bachimont 2001] : Bachimont B., "Modélisation linguistique et modélisation logique des ontologies : l'apport de l'ontologie formelle". In *12th Journées Francophones d'Ingénierie des Connaissances (IC'01)*, pages 349-268, Grenoble, 2001.

- [OntoWeb 2002] : OntoWeb Consortium, "Deliverable 1.3: A survey on ontology tools - OntoWeb Ontology-based information exchange for knowledge management and electronic commerce". IST Project IST-2000-29243, 2000.
- [Demazeau, 1995] : Yves Demazeau. From interactions to collective behaviour in agent based systems. European Conference on Cognitive Science, Saint-Malo – France, Avril 1995.
- [Franc et Sanders, 1998] : Modèles et systèmes multi-agents en écologie et en géographie : état de l'art et comparaison avec des approches classiques. In Modèles et systèmes multi-agents pour la gestion de l'environnement et des territoires, Ferrand N. (Ed), Editions Cemagref, Paris : 17-34.
- [Bousquet et al., 2002] : BOUSQUET F., BARRETEAU O., D'AQUINO P., ETIENNE, M., BOISSAU S., AUBERT S., LE PAGE C., BABIN D., CASTELLA J.C., et al., 2002. Multi-agent systems and role games : an approach for ecosystem co-management. In Complexity and ecosystem management : the theory and practice of multi-agent approaches, Janssen M. (Ed), Elgar Publishers, Northampton.
- [Drogoul, 1993] : Drogoul A., De la simulation multi-agent à la résolution collective de problèmes, Thèse de l'Université P et M Curie, 1993.
- [Ferber, 1995] : J. Ferber. Les systèmes multi-agents. Vers une intelligence collective, Editions InterEditions, 1995.
- [Brooks, 1989] : Brooks R., A robot that walks : emergent behaviour from a carefully evolved network, Neural computation, 1:253-62, 1989.
- [Steels, 1994] : Steels, L. Building agents with autonomous behavior systems, in L. Steels & R. Brooks, eds, 'The artificial life' route to 'artificial intelligence'. Building situated embodied agents.', Lawrence Erlbaum Associates, New Haven, 1994.
- [Gallone & al., 1994] : Gallone J-M., Charpillat F. et Chevrier V., Un modèle d'agent pour un raisonnement contraint par le temps, - In: Actes Journée PRC-IA sur les Systèmes Multi-Agents, Paris, Décembre 1994.
- [Deneubourg & al., 1990b] : Deneubourg J-L., Goss S., Francks N.R., Sendova-Franks A., Detrain C. et Chretien L., The dynamics of collective sorting : robot-like ants and ant-like robots, in Meyer J-A et Wilson S.
- [Erceau, 1991] : Erceau J., Intelligence Artificielle Distribuée et Systèmes Multi-Agents de la théorie aux applications. 23ème Ecole Internationale d'Informatique de l'AFCEP, Neuchâtel, 1993.
- [Tanenbaum, 92] : Andrew Tanenbaum. Réseaux — Architectures, protocoles, applications InterEditions 1992.
- [Austin, 1962; Austin, 1970] : J.L. Austin. How to do things with words. Clarendon Press, Oxford, 1962.
- [W3C, 2004a] : Web Ontology Language. <http://www.w3.org/TR/2004/REC-owl-features-20040210>
- [W3C, 2004b] : RDF Schema. <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>
- [Zou et al., 2003] : Youyong Zou, Tim Finin, Li Ding, Harry Chen, Rong Pan. Using semantic web technology in multi-agent systems: a case study in the TAGA trading agent environment. Proceedings of the 5th international conference on Electronic commerce. Pages: 95 – 101, 2003.